

The Open University

Faculty of Science, Technology, Engineering & Mathematics
Knowledge Media Institute (KMi)

Designing a Personal Awareness Agent to Ameliorate Information Overload

David Goddard

A thesis submitted for the degree of
Doctor of Philosophy

January 2025



orcid.org/0000-0002-5565-9692

Supervised by Dr. Paul Mulholland, Dr. Lara Piccolo & Dr. Enrico Daga

Copyright & Licensing

© 2025 **David Goddard**. All rights reserved.

This thesis is licensed under CC BY-NC-ND 4.0.

 CC Creative Commons

 BY Attribution Required

 NC Commercial Use Not Allowed

 ND No Derivatives

To view a copy of this license, visit: <https://creativecommons.org/licenses/by-nc-nd/4.0/>.

Document Object Identifiers

The DOI for this thesis is: <https://doi.org/10.21954/ou.ro.00104313>

Supplemental material is indexed at: <https://doi.org/10.21954/ou.rd.28045100>

Thesis Organisation

This thesis is organised into several chapters and appendices, with supplemental material accessible via the DOI listed above. References to supplemental material in the thesis content are prefixed with the letter 'S'.

A roadmap outlining the structure and content of each chapter is provided in Section 1.3.2, offering a guide for navigating the document.

*I dedicate this work to my family:
to Helen, Naomi and Kara – thank you for everything, you're amazing;
to Mum – Dorothy – who has always had such faith and been so supportive;
to Dad – John – who didn't get to see it through to the end.*

Abstract

Ubiquitous devices provide users with content and notifications that often blur the distinction between work and personal activities and can lead to information overload (IO). This research aimed to support users of multiple collaborative and social systems who may experience this, by designing and evaluating a modular personal software agent – the Awareness Agent – to support the user and ameliorate the effects of overload.

Looking at the issue of information overload through a lens of the CSCW concept of Awareness, a survey of users showed that there may be no one size fits all solution, with people perceiving and reacting to IO in different ways – but with common themes emerging showing a need for more control over the algorithms that determine what users see. A set of personas was devised based on this input, which were used to inform and evaluate the design for the agent. With respect for user control and ownership of Artificial Intelligence (AI) interaction being at the heart of the work, a modular framework was designed, able to pull in messages from different sources and make use of varied commodity cognitive computing services to manage that content for the user. A system called User-Directed Machine Learning (UD-ML) was introduced as part of this framework, to enable the user to control the lifecycle of the AI that makes decisions about the content that they see.

A novel system for generating and using synthetic messages and conversations was created in order to address practical issues relating to access to confidential and walled-off data in academic studies. A technique for using Generative AI-driven synthetic ‘virtual study participants’ was also introduced with twin purposes of easing the workload on human participants in high data volume studies, and to more broadly investigate how AI-driven evaluation techniques can form part of the solution to IO. The process of evaluating the Awareness Agent combined human and synthetic study participants to draw conclusions on the efficacy of both UD-ML and the synthetic evaluation techniques themselves.

Key findings highlight the importance of balancing AI systems with user control to foster trust and usability. The prototype Awareness Agent showed strong potential as a basis for a practical solution to mitigate information overload, demonstrating effective capabilities in filtering and consolidating information across multiple domains. The techniques for synthetic content and evaluation were found to be an effective part of the study process and are an asset that can be re-used in other research contexts.

Acknowledgements

I would like to thank my supervisors – Dr. Paul Mulholland, Dr. Lara Piccolo and Dr. Enrico Daga – for their support, guidance and persistence over these years; I know that supervising a part-time remote PhD student can present its own challenges. If I had just one takeaway from this process, it would be how important good supervisors are; every difficult question that they have asked me has been worth the answer.

On the topic of people asking difficult but also helpful questions, I would like to thank Richard Stockley for all of the help and support. His insightful feedback, willingness to take the time, and unique perspective on the process have been invaluable.

I would also like to express my gratitude to the examination committee for taking the time to read my somewhat unwieldy thesis and conduct a stimulating and rewarding viva voce.

I am grateful to my five study participants for their time and effort. The lot of a PhD study participant is maybe not the most glamorous and exciting, so I thank you very much for your time, diligence and input. I'd not have been able to complete the work without you.

Many others have helped me during this time – be it with advice, proof-reading, bouncing ideas off, or just plain support. So thank you to my family, friends and former colleagues at IBM who have also been an essential part of this process.

Working as a remote student has not always been easy – particularly when spending most of that time overseas. I'm grateful to the staff of the various parts of the Open University who have helped make it work, from the library and IT support staff to the administrators in KMi and the Graduate School; you are the essential glue that holds it together.

Somewhat grandiosely I'd like to thank the founders of the Open University – from Michael Young and Harold Wilson who conceived and promoted the idea through to everyone whose efforts made and make it a reality. I spent more time than I'd like to admit in my formative years watching OU content on the BBC (another priceless British institution), and it had a significant effect on my outlook and development. It's been as a privilege to study with the OU, and I hope it thrives in its mission for many more years to come.

Finally, my thanks also go to the countless contributors to Free and Open Source Software that has made possible not only my work but also pretty much everything else in the modern world – and most of them have done it for little more than the love of creating. You guys are why we *can* have nice things ☺

Declaration of Authorship

I, David Goddard, confirm that the research in this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated.

I attest that I have exercised reasonable care to ensure that the work is original, and to the best of my knowledge does not breach any UK law, infringe any third party's copyright or other intellectual property rights, or contain any confidential material.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Notices

Slack is a trademark and service mark of Slack Technologies, LLC (formerly Slack Technologies, Inc.), registered in the U.S. and in other countries.

OpenAI is a trademark and service mark of OpenAI, Inc., registered in the U.S. and in other countries.

This research is independent and is not affiliated with, endorsed by, or sponsored by Slack Technologies or OpenAI.

All trademarks and logos used are the property of their respective owners and are used here for informational and academic purposes only.

Contents

Title Page	1
Abstract	v
1 Introduction	1
1.1 Introduction and Motivation	1
1.2 Problem Discussion	4
1.3 About This Document	10
2 Literature Review	15
2.1 Introduction	15
2.2 Awareness Problem	16
2.3 The Social Machine	25
2.4 Software Agents	31
2.5 Cognitive Computing and AI	39
2.6 Conclusions	60
2.7 Chapter Summary	60
3 Problem Analysis and Research Questions	61
3.1 Problem Analysis	61
3.2 Research Questions	63
3.3 Inferences From Literature Review	68
3.4 Chapter Summary	71
4 Overall Methodology	73
4.1 Overall Approach	73
4.2 Research Elements	74

4.3 Chapter Summary	78
5 User Survey and Personas Development	79
5.1 Survey	79
5.2 Personas Development	89
5.3 Personas	101
5.4 Chapter Summary	105
6 Awareness Agent Design and Implementation	107
6.1 An Awareness Agent	107
6.2 Personas Perspective	108
6.3 Requirements Analysis	111
6.4 Awareness Agent Design Concept	113
6.5 Process	120
6.6 System Model	121
6.7 Architecture	141
6.8 Implementation	172
6.9 Technical Status	179
6.10 Reflections	191
6.11 Chapter Summary	196
7 Synthetic Content	197
7.1 Motivation	198
7.2 Research Through Design	200
7.3 Approach to Synthetic Content	200
7.4 Content Generation	202
7.5 Consumption of Synthetic Content	212
7.6 Reflections	217
7.7 Chapter Summary	223
8 Synthetic Evaluation	225
8.1 Approach	225
8.2 Concepts	228

8.3	Implementation	241
8.4	Reflections	252
8.5	Chapter Summary	257
9	Awareness Agent Prototype Study	259
9.1	Overview	259
9.2	Design	261
9.3	Implementation	270
9.4	User Interfaces	274
9.5	Results	278
9.6	Reflections	292
9.7	Conclusions	297
9.8	Chapter Summary	299
10	Conclusions and Further Work	301
10.1	Scientific Contributions	302
10.2	Addressing the Research Questions	305
10.3	Limitations	316
10.4	Further Work	319
10.5	Final Thoughts	322
Bibliography		323
Appendices		353
A Survey Appendix		355
A.1	Survey Questions	355
A.2	Survey Results	355
B Personas Appendix		357
B.1	Cluster Analysis	357
B.2	Final Personas	360
C Awareness Agent Application Appendix		367
C.1	Additional Figures	367

C.2	Slack Application Manifests	370
D	Design and Development Log Appendix	371
D.1	Overview	371
D.2	Abridged Design and Development Log	372
E	Synthetic Content Appendix	395
E.1	Simulation Schemas	395
E.2	Simulated Content Prompting	396
E.3	Dramatis Personae Document	396
E.4	Entities Document	396
E.5	Simulated Content Topic Examples	396
E.6	Simulated Content Output Examples	401
F	Synthetic Evaluation Appendix	403
F.1	Static Prompt Elements	403
F.2	Evaluation Schemas	408
F.3	Evaluation Schemas - Data Service	410
F.4	Evaluation JSON Examples	411
F.5	Evaluation Processor	411
F.6	Fine-Tuning	411
G	Study Appendix	413
G.1	Study Protocol	413
G.2	Persona to Agent ID Mappings	414
G.3	Persona Configurations Spreadsheets	414
G.4	OpenAI Models Used	415
G.5	Evaluation Output Tags	416
G.6	Instance Preparation	417
G.7	Information Provided to Participants	418
G.8	Evaluation Token Usage	419
H	Supplementary Materials Appendix	421

List of Tables

1.1 Descriptions of the four research domains	4
1.2 Thesis Roadmap – Part 1	11
1.3 Thesis Roadmap – Part 2	12
1.4 Thesis Roadmap – Appendices	13
5.1 Survey questions 9-14	81
5.2 Survey Notable Results 1	86
5.3 Survey Notable Results 2	87
5.4 Kruskal-Wallis Test Results for QQ_9_14_5CL	94
5.5 Sentiment assignment by statement/cluster based on mean response value . .	95
5.6 Pearson’s chi-square correlation between each question and the QQ_9_14_5CL cluster and inclusion in persona creation	96
5.7 Mapping of QQ_9_14_5CL Cluster Number and Persona Name	101
6.1 Requirement Areas	111
6.2 Agent Requirements	112
6.3 Data Service Routes – Core Awareness Agent	177
6.4 Data Service Routes – Simulate, Evaluate, Study	178
6.5 Agent Requirement Compliance	182
7.1 Synthetic Data Request Elements	204
8.1 Evaluation Processing Inputs for a Single Item	242
8.2 Prompting Messages for Mode 1 (Compound Ordinal Evaluation)	245
8.3 Prompting Messages for Mode 2 (Simplified Ordinal Evaluation)	246
8.4 Prompting Messages for Mode 3 (Simplified Binary Evaluation)	246
8.5 Messages for Base Training Item (Mode 2 Only)	247

8.6	Messages for Base Training Item (Mode 3) and Extended Training Item	248
9.1	Classification Agreement vs Evaluation Score for Prime Evaluation	287
D.1	List of Design & Development Log Topics	372
G.1	Mapping of Persona Name to Agent ID	414
G.2	Evaluation Output Tags	416

List of Figures

1.1	Research Journey	3
1.2	Venn diagram showing intersection of theoretical domains	7
5.1	Derived cluster-statement mapping	98
5.2	Blank PATHY template	99
5.3	Final PATHY document for persona “Susan”	102
6.1	Awareness Agent System Model	122
6.2	Content Item Structure	128
6.3	Content Item Data Mapping Illustration	130
6.4	Simple Classification Augmentation Item	131
6.5	Content Item Data Mapping – LD Mapping	143
6.6	Content Item Data Mapping – Standard Fields	144
6.7	Content Item Data Mapping – Extended Fields	145
6.8	Content Item Data Mapping – Native Fields	146
6.9	Content Item Formatters	147
6.10	Content Item Creation Flow	148
6.11	Content Item Data Generation	149
6.12	Awareness Agent Queues & Services	150
6.13	Awareness Agent Service Launchers	152
6.14	Awareness Agent Acquire Service for RSS	153
6.15	Awareness Agent Acquire Service for Slack	154
6.16	Awareness Agent Augment Service	155
6.17	Awareness Agent Allocate Service	157
6.18	Awareness Agent Interact Service for Slack	161
6.19	Slack Content Item Formatter within Interact Service	162

6.20 Example of Slack Interact messages	162
6.21 Example of Slack Interact messages (annotated)	163
6.22 Key of User-Directed ML Request & Response Objects	165
6.23 Generation of User-Directed ML Classification Request from Content Item . .	166
6.24 User-Directed ML stage of Augmentation flow	167
6.25 Population of Augmentation Item from UD-ML Classification Response	167
6.26 User-Directed ML Content publication in Interact Service for Slack	169
6.27 User-Directed ML Training process example in Interact Service for Slack . .	170
6.28 Generation of UD-ML Training Request from CI & Interact data	171
6.29 ML Service Model Directory Structure	176
6.30 Awareness Agent UI showing UD-ML channel #interested-personal	188
6.31 Awareness Agent UI showing UD-ML channel #cycling-cycling	189
6.32 Awareness Agent UI showing UD-ML item reclassification	190
6.33 Awareness Agent UI showing summarisation feedback	190
7.1 Synthetic Content Generation & Storage	203
7.2 Synthetic Content Publication Using Slack	209
7.3 Slack web app for persona Adam showing channel #bis-team-manager	213
7.4 Slack web app for persona Adam showing channel #bis-team-client	214
7.5 Slack web app for persona Adam showing channel #bis-announce	215
7.6 Slack web app for persona Adam showing channel #family-group-chat	215
7.7 Slack web app for persona Adam showing channel #friends-chat	216
7.8 Slack web app for persona Adam showing channel #bwcc-general-chat	217
8.1 Classification Evaluation Request – Mode 1 (Compound)	232
8.2 Classification Evaluation Request – Mode 2 & 3 (Simplified)	234
8.3 Classification Evaluation Response – Mode 1 (Compound)	235
8.4 Classification Evaluation Response – Modes 2 & 3 (Simplified)	237
8.5 Synthetic Evaluation Process for Single Content Item	243
8.6 Flow of Content Items from Allocate to Evaluation Services	249
8.7 Storage of Content Items for Later Processing	249
8.8 Synthetic Evaluation Batch Processing by Tag and Subset	251

9.1	Awareness Agent Training UI – Full Browser Page	273
9.2	Awareness Agent Training UI – Single Item	274
9.3	Awareness Agent Evaluation UI – Full Browser Page	276
9.4	Awareness Agent Evaluation UI – Partial Item	277
9.5	Mean Synthetic vs Participant Evaluation r by Evaluation Tag	279
9.6	Mean Synthetic Evaluation vs Participant Classification r_{pb} by Evaluation Tag	280
9.7	Mean Synthetic Evaluation vs Participant Classification r_ϕ by Evaluation Tag	281
9.8	Selection of Evaluation Performances by UD-ML Model	282
9.9	Synthetic Evaluation Tag Ratings	283
9.10	Cohen’s Kappa for Manual Classification Agreement with UD-ML	285
9.11	Percent of Manual Classification Agreement with UD-ML Time Series	285
9.12	Classification Agreement vs Evaluation Score for Prime Evaluation	286
B.1	Demographics for cluster 1	357
B.2	Demographics for cluster 2	358
B.3	Demographics for cluster 3	358
B.4	Demographics for cluster 4	359
B.5	Demographics for cluster 5	359
B.6	Final PATHY persona “Susan”	361
B.7	Final PATHY persona “Adam”	362
B.8	Final PATHY persona “Phoebe”	363
B.9	Final PATHY persona “Kenton”	364
B.10	Final PATHY persona “Usha”	365
C.1	CI Queue Flow in Awareness Agent	368
C.2	Awareness Agent Allocate Mappings	369

Chapter 1

Introduction

1.1 Introduction and Motivation

Most users of multi-user collaborative software and social applications will attest to experiencing some degree of information overload. Studies on social media platforms have shown such information overload to have a measurable effect on information processing performance [Rodriguez, Gummadi, and Schoelkopf, 2014].

The term, ‘information overload’ was coined by Toffler [1970] and can be said to be “a condition in which an agent has – or is exposed to, or is provided with – too much information, and suffers negative consequences as a result (experiences distress, finds itself in a ‘problematic situation’, is unable to make a decision or to stay informed on a topic, etc.)” [Himma and Tavani, 2009].

The first aspect of this research considers an approach to ameliorate such difficulties associated with large volumes of content coming from disparate sources in either Computer Supported Collaborative Work (CSCW) or social media environments. It is motivated by the notion that we can make better use of computing systems to help users with information overload problems than we currently do, in particular by using the paradigm of the autonomous software agent, supported by Artificial Intelligence (AI) and Machine Learning (ML) techniques. We hope that by shifting more of the work of dealing with high information load to computing systems, we can improve the productivity and well-being

of individual users. To be effective, we must do this in ways that do not compromise the quality of handling of information by erroneously discarding important information while favouring items that are less important. The “gold standard” that we might aim for is for a support system that acts as the user themselves would, if they had an unlimited amount of time and attention to process information, and we might measure such a system based on its deviation from this standard.

We adopt the concept of *awareness* [Metaxas and Markopoulos, 2008] to describe the user’s relationship with data arising from the the activities of others. In broad terms, we can say this is an individual’s understanding of the activities of others that they use to provide context to and inform their own activities. To maintain awareness, the user must be exposed to information arising from others’ activities, but over exposure to this information risks information overload.

The second aspect of this research examines how we can test and evaluate such systems by making use of Generative AI and Large Language Models (LLMs) to perform tasks that might otherwise take a considerable amount of human contributors’ time and effort. We will develop LLM-based techniques for performing evaluations of solutions to the awareness problem, keeping humans in the loop to validate the work of the LLMs and allow us to validate the techniques themselves.

We consider that this research exists at the intersection of four domains: *Awareness*, *the Social Machine*, *Software Agents*, and *Cognitive Computing*, as described in Table 1.1. In this thesis we will consider each of these domains and how our work relates to them.

We begin by looking at the problem of information overload through a lens of awareness, and develop a more grounded understanding by collecting data on perceptions and strategies to cope with information overload through a survey. Based on the survey results, we model a set of personas to represent different perspectives on the problem and appropriate solutions. We use this work as the basis for developing an awareness agent concept, which we then prototype and test. In parallel, we develop a technique for using synthetic data and evaluation in the experimental process, as illustrated in Figure 1.1.

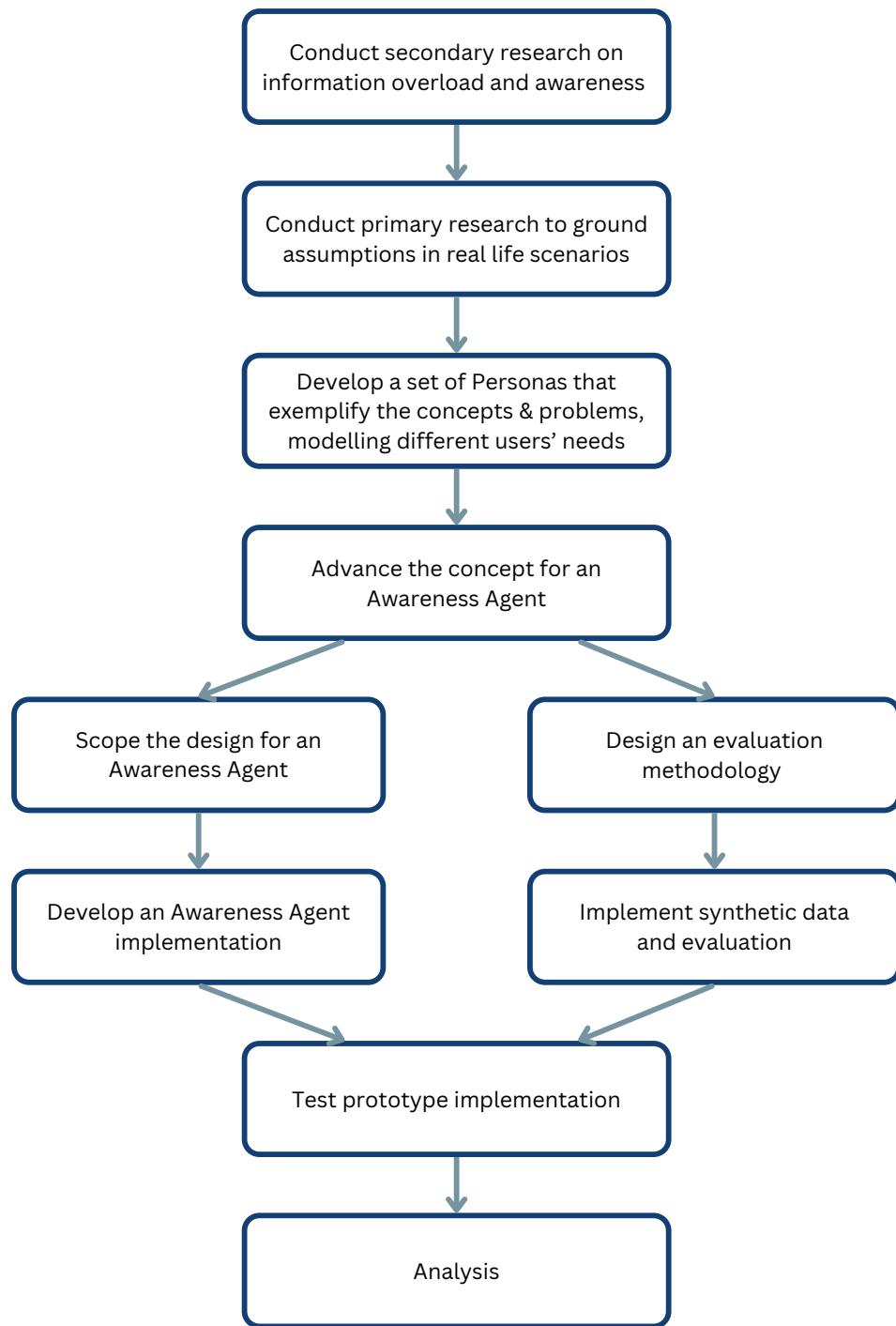


Figure 1.1: Research Journey

Table 1.1: Descriptions of the four research domains

Domain	Description
Awareness	Awareness refers to the capacity of an individual or system to be cognisant of its environment, context, and the entities within it. In the context of computing and artificial intelligence, awareness involves understanding and interpreting the surrounding environment to make informed decisions. This can include situational awareness in software systems, where the system perceives and reacts to changes in its environment.
The Social Machine	The Social Machine concept comprises a synergistic integration of humans and machines to accomplish tasks that neither could achieve independently, using the collective intelligence of human networks and the computational power of machines to solve complex problems. This domain explores the interaction patterns, collaborative behaviours, and the co-evolution of social and technical systems, with an emphasis on the roles of social networking, participatory design, and crowdsourcing in modern computing environments.
Software Agents	Software agents are autonomous, pseudo-intelligent applications capable of performing tasks on behalf of other entities with some degree of independence and adaptability. They are commonly designed to perceive their environment, make decisions, and take actions to achieve specific goals, and can range in functionality from simple bots performing repetitive tasks to complex systems capable of learning, negotiation, and interaction with other agents and humans.
Cognitive Computing	We are using Cognitive Computing as an umbrella term to refer to computer systems that simulate human thought processes to some degree. Such systems can utilise artificial intelligence, machine learning, and data mining techniques to understand, reason, and learn from interactions and data.

1.2 Problem Discussion

1.2.1 The Awareness Problem

We can surmise that a key problem for a user of a system where there is a high information load, is identifying and accessing content that is relevant to them in a format that is useful. Information load is a concept of more than one dimension: it relates to the quantity, variety and complexity of items as well as the number of individual sources in the system.

In this research we take, ‘awareness’ to mean the ability of actors¹ to perceive the activities and output of other actors in the system. This process is influenced by the actor’s own activities and goals, as well as the actions of other actors (such as explicit sharing of activities and information). We can frame consideration of this problem in terms of the *awareness* concept, as defined by Benford and Fahlén [1993] and extended by Rodden [1996] and later Metaxas and Markopoulos [2008]. The term ‘awareness’ is used in quite a broad sense in CSCW [Harper, 2016], and as K. Schmidt [2002] commented, researchers often qualify it with a variety of adjectives to confer more precise meaning. It is also applied to a variety of contexts ranging from telepresence or virtual presence in a simulated space to entirely non-spatial collaborative processes. We can phrase this as the “awareness problem” – how does a user maintain awareness of what is happening without being so aware of noise that it impairs their function? While the concept of awareness is rooted in CSCW, we believe that it can be applied to other systems that do not fall into the traditional ‘work’ sphere.

The multi-dimensional nature of information load also means that there is more than one aspect to the mechanics of awareness: an actor needs to be aware not only of the existence of other activities, but also in many cases of the *content* so that it may better evaluate the priority it should assign to them. Computational support is particularly useful in this area where the volume of data associated with activities is very large. While many techniques exist for aggregating large volumes of numerical information, such as displaying data in chart form, aggregation of large volumes of *textual* data is less well explored. As Vilaplana [2015] noted, the visual representation and exploration of bodies of texts has not evolved as far or fast as other visual techniques. This poses a challenge for maintaining awareness where the target information is mostly textual in nature, but it is also an area where technological advances have opened new opportunities, with the advent of a new generation of Generative AI [Leslie, 2023].

¹An actor in this system is not necessarily a person, it can also be a software agent or an automated emitter of data

1.2.2 An Awareness Agent

Simple computational approaches to ameliorating information overload, such as applying filters to incoming data can be quite effective, but we aim to investigate how a more sophisticated software agent might act on a human user's behalf to enhance their interaction with a multi-user collaborative system.

A software agent might simplistically be defined as a computer program that acts independently on behalf of a user. The range of function of different software agents is very wide, encompassing task automation, system monitoring and interactive chatbots. In this work, we look at an agent that acts on behalf of a user to monitor one or more online systems, bringing the right information to their attention at the right time. We can call this an *Awareness Agent*, taking inspiration from Luff *et al.*'s Awareness Machine [Luff, Heath, and Svensson, 2008], Ye *et al.*'s agent-supported adaptive group awareness [Ye *et al.*, 2001], and the Awareness Server of Ahmad and Wegner [1999]².

Such an agent will extend the *focus* of a user of a collaborative system, while conversely you can say that such agents will occupy the *nimbus* of users of these systems (as Benford, Bowers, *et al.* [1994] summarised it: "The more an object is within your focus, the more aware you are of it. The more an object is within your nimbus, the more aware it is of you"). We can further adapt this terminology for the social media age, by relating the focus to the people you are *following* and the nimbus to your *followers*. More succinctly, you might say that the agent will follow people and things on your behalf while others' agents may follow you.

A user's agent may also manage their own presence as seen by followers. In this context, the agent would monitor the output created by its 'owning' user and perform actions based on that content. These actions could include generating push notifications that other actors may receive and act on. The distinction of this mode of operation is that the agent is processing content using rules defined by the *creator* of the content rather than by a consumer, allowing them to deliberately promote content that they themselves judge to be noteworthy. An actor that receives such notifications may of course choose to

²The term "awareness agent" has also been used by Oh and Look [2003], although in the different and more specific context of instant messenger availability

disregard them, or simply to consider them as just one factor in an overall evaluation of the content.

The type of system where software agents and humans co-exist and cooperate within the same ecosystem can be seen as an example of a Social Machine as envisioned by Berners-Lee and Fischetti [1999]. In some aspects, the agents are avatars or representatives of the user, presenting an outward face on their behalf; in others, they act as gatekeepers or concierges for the users, filtering, managing and assisting with inbound information; in many aspects, software agents as peers of the human elements of the system.

We consider that our awareness agent sits at the intersection of four theoretical domains: awareness, software agents, cognitive computing, and the social machine – see Figure 1.2 and Table 1.1.

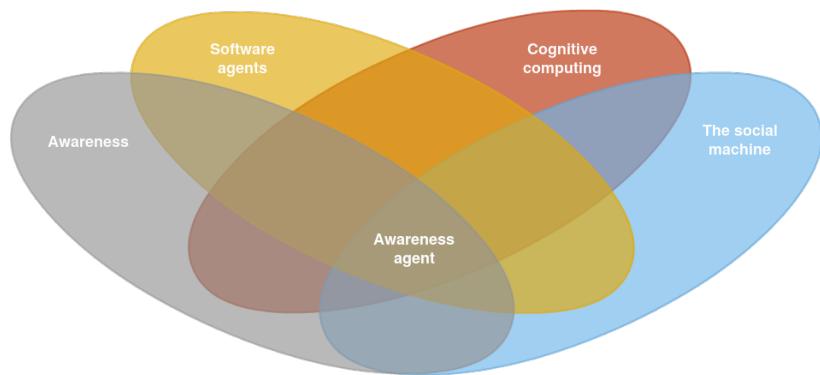


Figure 1.2: Venn diagram showing intersection of theoretical domains

1.2.3 Research Objectives

The goal of this work is to explore how software agents can best support users of collaborative and social systems, such that they are able to receive, process and act on the maximum amount of relevant information without experiencing information overload. We can view this as a process of ‘information targeting’, where the information is filtered, refined and presented to the user in such a way that balances their need to *know* with their desire to avoid overload.

We approach this by proposing a design for a conceptual intelligent software agent and implementing some of the functionality that such an agent might possess. However, the

process of evaluating such an agent is also a challenge, and can itself be a fruitful source of research. The nature of the problem – large volumes of information, some of which may be confidential – means that efforts to evaluate a solution need to address issues of data volume and privacy. This presents a parallel opportunity to develop and study techniques to address this.

We seek to answer the following research questions:

- RQ1 [Problem Understanding]: What problems with information overload are experienced by users of information systems and what attitudes do they have towards providers and solutions?
- RQ2 [Solution Development]: What design direction and system features can address information overload for diverse users managing multiple online information sources?
- RQ3 [Evaluation Techniques]: Can synthetic techniques be used effectively to study and evaluate potential solutions to Information Overload?

These questions are discussed in greater detail in chapter 3 of this thesis.

1.2.4 Expected Contributions

We hope to contribute in the following ways:

1. Create a design for an agent-based solution that uses AI techniques to address information overload and improve awareness; we aim for this design to provide the basis for developing a real-world application to assist users.
2. Gain and share insight into the design and practical challenges involved in developing such a solution, and ways to address those challenges.
3. Produce detailed analysis of the operation of such a solution that can form the basis for further development and improvement.
4. Develop reproducible techniques and materials for generating and using synthetic content to make it easier to study information systems where real data might be limited for technical or confidentiality reasons.
5. Develop reproducible techniques for using synthetic evaluation in academic study of information overload and similar topics so that the performance of classifiers and other systems can be evaluated and analysed at scale.

1.3 About This Document

1.3.1 Links

This document contains links to a number of web resources that are current at time of publication. Due to the phenomenon of link rot – whereby URLs and the content that they point to change or are lost over time – these links might not still be valid at time of reading. We have chosen to use the service <https://perma.cc/> to preserve what we hope is a permanent record of those links most likely to change [Zittrain, Albert, and Lessig, 2014] [Dulin and A. Ziegler, 2017] and have created Perma Links with archived content for many of the URLs that we reference here.

We have not generated Perma Links for every URL referenced in this document – priority has been given to those that we judge most at risk of link rot, such as commercial websites and personal blogs. We've assumed that URLs generated by standards bodies and similar organisations have greater longevity; where some form of permalink already exists we have given that preference, and we use DOIs³ to address scholarly articles where available.

1.3.2 Roadmap

This section provides an overview of the thesis structure, detailing the progression of the research and its components. Tables 1.2 and 1.3 outline the main chapters, which encompass the background research, formulation of research questions, development of the methodology, and the execution of the methodology. This includes the creation of personas, the design and implementation of an Awareness Agent, the study techniques employed, and the study itself.

Additionally, Table 1.4 provides a list of the Appendices included in the thesis, offering supplementary information and supporting materials relevant to the main content.

³<https://www.doi.org/>

Table 1.2: Thesis Roadmap – Part 1

Chapter	Description
1 - Introduction	This chapter
2 - Literature Review	A review of the existing literature covering the domains: <ul style="list-style-type: none"> • Awareness • The Social Machine • Software Agents • Cognitive Computing & AI
3 - Research Questions	An analysis of the research problem and the formation of an umbrella research topic with three specific sections: <ul style="list-style-type: none"> • Problem Analysis • Research Questions • Inferences from Literature Review
4 - Methodology	An overview of our research methodology, including a summary of the Overall Approach and more detailed methodologies for the elements: <ul style="list-style-type: none"> • Survey • Personas • Awareness Agent Application • Study
5 - Personas	The development of survey data-driven personas: <ul style="list-style-type: none"> • Survey • Personas Development • Final Personas
6 - Awareness Agent	Design and implementation of a prototype Awareness Agent: <ul style="list-style-type: none"> • Personas Perspective • Requirements Analysis • Design Concept • System Model • Architecture • Prototype Implementation • Technical Status • Reflections

Table 1.3: Thesis Roadmap – Part 2

Chapter	Description
7 - Synthetic Content	The first of two chapters bridging between the Awareness Agent and the Study, covering the rationale for using synthetic content and the creation of a system to support the study. Includes: <ul style="list-style-type: none"> • Motivation • Approach • Content Generation • Content Consumption • Reflections
8 - Synthetic Evaluation	The second bridging chapter, covering the design and implementation of a system for synthetically evaluating elements of the Awareness Agent that will go on to be a core part of the study. Includes: <ul style="list-style-type: none"> • Approach • Concepts • Implementation • Reflections
9 - Study	The study itself, using synthetic content and evaluation techniques to investigate the performance of aspects of the Awareness Agent, including: <ul style="list-style-type: none"> • Overview • Design • Implementation • User Interfaces • Results • Conclusions
10 - Conclusions	Draws together the overall work making a number of conclusions, observations and highlighting areas for further work.

Table 1.4: Thesis Roadmap – Appendices

Appendix	Description
A - Survey	Contains detailed data supporting the user survey part of the Personas chapter: <ul style="list-style-type: none">• Survey Questions• Survey Results
B - Personas	Includes details of the cluster analysis used to develop the personas, the final personas themselves and a set of persona-based scenarios: <ul style="list-style-type: none">• Cluster Analysis• Final Personas
C - Awareness Agent	Supporting information for the Awareness Agent: <ul style="list-style-type: none">• Additional Figures• Slack Application Manifests
D - Design & Development Log	The abridged Design and Development Log.
E - Synthetic Content	Supporting information for Synthetic Content: <ul style="list-style-type: none">• Simulation Schemas• Prompting Code & Template• Topics for Personas
F - Synthetic Evaluation	Supporting information for Synthetic Evaluation: <ul style="list-style-type: none">• Static Prompt Elements• Evaluation Schemas• Evaluation JSON Examples• Evaluation Processor Code• Fine Tuning
G - Study	Supporting information for the Study: <ul style="list-style-type: none">• Study Protocol• Persona to Agent ID Mappings• Configurations Spreadsheets• Evaluation Output Tags• Instance Preparation• Information for Participants
H - Supplementary Material	Index of supplementary material referenced in this document.

Chapter 2

Literature Review

2.1 Introduction

Information overload is a big problem which continues to worsen. Technical developments are often implicated in contributing to the problem, but may also offer solutions if placed in the hands of the user.

We have divided the literature review into four sections:

1. Awareness
2. The Social Machine
3. Software Agents
4. Cognitive Computing

Each of these sections represents a field that has a direct bearing on our research, which sits at the intersection of these domains. We are driven by the need to address problems with *awareness* (including information overload); our proposal is to address this with a digital solution that we can describe as a *social machine*; it is implemented as a *software agent* using *cognitive computing*, which also forms part of our evaluation methodology.

2.2 Awareness Problem

In this section, we explore the literature relating to the awareness problem that we defined in the previous chapter. We will begin by looking more closely at the concept of ‘awareness’, then covering the literature relating to information overload and finally reviewing work on alerting and notification strategies, which are a key practical consideration for the problem as posed.

2.2.1 Awareness

The understanding of ‘awareness’ that we base this work on derives from the concepts developed by Rodden [1996], who postulated a model for awareness that diverged from the spatial concepts previously advanced by Benford and Fahlén [1993], who based their work on a metaphor of interaction within virtual worlds [Benford, Bowers, et al., 1994]. This metaphor treated a computer system as a set of spaces through which people move, interacting with other people and objects within those spaces. Rodden considered a specialisation of a model of users’ interactions with shared objects that was not dependent on some analogous shared spatial geometry but instead applied to shared graph structures found in computer applications. While Rodden maintained the concepts of focus and nimbus, he mapped users and objects into a set of presence positions in a logical space, rather than any virtual representation of physical space.

Rodden’s work was later extended by Metaxas and Markopoulos [2008], who added the concepts of aspects, attributes and resources, relating to communication aspects of awareness systems. They noted the lack of clarity and consistency in awareness research that had been highlighted by K. Schmidt [2002] and accepted his argument for describing awareness in reference to “activities, practices or phenomena or object that a person is made aware of”. They also considered the work of Boyle and Saul Greenberg [2005] on privacy, specifically the concepts of solitude, confidentiality and autonomy, providing a basis for modelling how actors may choose to share aspects of their work with other users.

A recurrent theme in the literature is that nobody can entirely agree on a definition of

what awareness actually *is*, or at least not without qualifying the term. Continuing this, Papadopoulos [2006] itemised several different definitions of awareness in CSCW, including:

- *Presence awareness*: knowledge of who is around [Milewski and T. M. Smith, 2000]
- *Workspace awareness*: knowledge of peers' activities within a shared work context [Carl Gutwin and Saul Greenberg, 2002]
- *Peripheral awareness*: ability to be to some extent aware of others' activities while focusing on your own different task [Grudin, 2001]
- *Contextual awareness*: knowledge of surrounding environment/location and broader context [Izadi et al., 2002]
- *Passive awareness*: knowledge of others' actions gained by passive rather than active means [Dourish and Bellotti, 1992]
- *Situation awareness*: the timely cognizance of product and process needed to operate or maintain a system [Carl Gutwin and Saul Greenberg, 2002]

This is not an exhaustive list, and some of the definitions are open to interpretation. For example in the case of presence awareness, "who is around" may be taken to be who is *physically* located nearby, but may also mean who (or what) is currently active in the same online environment.

Also of interest are the following distinctions for modes of awareness made by Bürger [1999] in his PhD dissertation:

- *Synchronous vs. asynchronous awareness*: ability to act before somebody else has completed a task *vs.* notification after the fact, e.g. during absence.
- *Symmetrical vs. asymmetrical awareness*: if I can see you then you can see me *vs.* quiet observations and actions, but controlled by the observed.
- *Implicit vs. explicit awareness*: let the system guess what is relevant to you *vs.* tell the system about what you are interested in.
- *Coupled vs. detached awareness*: receive notifications for objects in your focus *vs.*

notifications for objects outside your view.

These are interesting definitions in the sense that they seek to differentiate awareness based on the context of the application; awareness means a different thing when users are acting asymmetrically for example. Additionally, the distinction between explicit and implicit awareness is one that we will return to when considering implicit vs. explicit learning behaviour for an awareness agent.

As we stated in the introduction, we are taking ‘awareness’ here to mean the ability of actors to perceive the activities and output of other actors in the system. While this broad phrase encompasses most of the various definitions of awareness in CSCW and other fields, we need to be mindful of the distinctions that apply to the term in a variety of modes.

Luff, Heath, and Svensson [2008] considered how awareness technologies would fit into a workplace, postulating an *awareness machine*. Their study was based on an operational surveillance system for London Underground, but was able to draw general inferences. An important consideration in this context was how and when to notify operators about specific events. They also considered that a machine that acts as some kind of awareness surrogate must not only take information from explicitly defined sources (such as CCTV in this case), but also from other “off the radar” resources that may be available to a human operator, such as a human supervisor’s knowledge of actual physical layout of the stations and the routine behavioural characteristics of passengers. The operation of such a machine would also need to be highly tailor able to individual circumstance. While Luff *et al.* were studying a safety-critical control room environment, these considerations are likely to be relevant to any computer system that attempts to handle delegated awareness for users.

On a philosophical level, Tenenberg, Roth, and Socha [2016] explored the idea of *shared intentionality* among cooperating actors in a CSCW system, taking the view that the first-person perspective taken by traditional views of awareness did not incorporate newer philosophical understandings of *shared* intentionality. They argued that such an approach exposes the weakness of the existing individual intentionality models, which are insuf-

ficient for what they term the *socially recursive inference* underwriting cooperative behaviour [Tomasello, 2014]. The meaning of this term is similar to the terminology of *grounding*, or achieving common ground [C. Gutwin and S. Greenberg, 2000] [J.-s. Lee and Tatar, 2014] [Dourish and Bellotti, 1992]. Tenenberg *et al.* considered this a recursive process as the development of a common ground is often done as a recursive conversation: “‘I know X and you know X’ and ‘I know that you know X’ and ‘You know that I know X’ recursively all the way down”.

2.2.2 Discerning Value in High Volume Information

The comprehensive study of information overload (IO) undertaken by Arnold, Goldschmitt, and Rigotti [2023] notes that there is no single, universally accepted definition, due to the topic’s appearance in disciplines as diverse as psychology, education, medicine, social sciences, marketing, and computer science¹. They found that a relatively large number of measures to reduce information overload relate to the personal rather than organisational level – for example tools and filtering strategies to manage emails, individual training, and coping strategies. They reviewed a number of technical and algorithm based approaches to the problem, which may be individual or organisational solutions. Their review also found that the information itself in IO cannot be considered in isolation; it is received and processed in a wider social context of tasks, team processes, and other factors such as organisational rules.

While Arnold *et al.*’s work references an enormous number of papers, the following stood out for us in terms of relevance to our work:

- Landale [2007] – description of a 3 step process for dealing with assimilation of new documents at the personal level
- Wu et al. [2016] – how the presentation of information at the human-machine interface should be of low complexity to reduce the cognitive load on the user
- Kluge, Antoni, and Ellwart [2020] – delegation of tasks from people to “digital agents”
- Norri-Sederholm et al. [2015] – information categories for situation awareness in an

¹Perhaps related to this, most of us have an intuitive view on what IO means in practice

emergency medical setting

- Graf and Antoni [2021] – appropriate technology choice is relevant for the transmission of information; more attention should be paid to quality rather than quantity in order to reduce information overload
- Sappelli et al. [2016] – system to categorise email according to the intent of the sender and extract task information

Discussing the progress of the Cochrane Collaboration for medical research information² in the decades since it was launched, R. Smith [2010] noted that progress addressing the issue of information overload pertaining to medical practitioners has been poor. Considering a number of strategies used by doctors to stay up to date on medical research, Smith described the ‘inhuman strategy’ (in other words doctors relying on machines to help them practise medicine); he noted that teams of people contributing to shared knowledge systems can keep up with information load far better than individuals can.

Bettis-Outland [2012] examined the relationship between information overload, decision making and organisational learning in a business setting. She considered the possibly non-intuitive question of whether decision making itself can influence the nature of organisational learning and risk of information overload, as certain decision making processes generate and use different amounts of information. She also noted that a tendency to simply ignore or discard excess information was one way that people adapted to information overload. We can regard this situation as one of the ways in which effective awareness can decrease in the face of overly high information load.

Shang, T. Wang, and Lv [2011] investigated machine learning and text classification techniques as a way of addressing information overload, noting that other computer-based solutions such as search engines and RSS technology solved some problems but were not adequate to fully address information overload. They took an approach of first aggregating data using RSS feeds, then using a Naïve Bayes classifier to classify this data. They argued that by classifying in this way the effect of information overload could be reduced (because classification helps the user process information). However, this technique does

²<http://www.cochrane.org/>

not explicitly focus or filter the incoming information (although the user could do so based on classifier outputs, in a relatively course-grained way).

Sorower, Slater, and Dietterich [2015] took a similar approach with email as the medium, by investigating tagging systems for incoming email. Tagging has an advantage over many classifier systems in that multiple tags can be attached to one entity, improving searchability and handling. Investigating the relative merits of implicit vs. explicit training of the classifier, they noted a balance in user effort relating to providing explicit and implicit feedback with consequent mixed results. In their example, explicit feedback would occur when a user manually corrects/confirms tagging information; conversely, implicit feedback is obtained when the user takes a number of normal email processing actions: setting/removing a flag or tag, moving the message to a folder, reading/replying/forwarding a message and so on. Each of these can be used to provide feedback to the system. In their user study, they found that explicit correction of tags was much poorer than expected (20% rather than 50% of incorrect tags corrected), but that implicit feedback was more successful. They also noted that any tagging system must be able to cope with incomplete feedback as explicit user feedback rates are low and generally only provide negative feedback (highlighting errors but not confirming when tags are correct).

The concept of information having a ‘half-life’ akin to that of the radioactive decay of atoms is also relevant. Davis [2013] explored this in relation to academic research papers³ where two obvious metrics can be used to measure academic paper half-lives: download count and citation count. While Davis used download data supplied by publishers, Abt [1998] used citations over time. The two metrics have characteristics that make them useful in different ways: citations show the judged relevance of a paper to ongoing research, while downloads provide more plentiful data but are less of an indicator of assessed relevance to current research, as many downloads are speculative. Researchers including Abt have observed that some academic papers gain a status where they have continued long-term relevance; these are often foundational papers that describe some research or concept that many others build on. This is not to say that short-lived papers do not have value: they can have a strong impact in the time that they are relevant for, but it may be the case

³<https://scholarlykitchen.sspnet.org/2013/12/18/what-is-the-lifespan-of-a-research-article/>
[<https://perma.cc/UHV6-VD5W>]

that as research moves on they themselves are much less frequently cited or viewed.

2.2.3 Notification Strategies

An important axis to consider for a system that notifies users is the receptivity of those users to interruption (a ‘push’ notification) compared to the possibly lesser utility of passive notification (‘pull’ notification). Research has shown [Cutrell, Czerwinski, and Horvitz, 2001] [Brian P. Bailey and Konstan, 2006] both that interruptions can affect users’ concentration and productivity when concentrating on a task, but also that users often find notifications helpful. Iqbal and Horvitz [2010] noted that users generally prefer to have notifications enabled, but that their reaction to individual notifications was influenced by content and context. They also found that in cases where notifications displayed sufficient information for the user to understand the nature of the notification, users were able to more effectively discriminate between them and achieve better balance between attentiveness to notifications and task concentration.

Xia and Sudharshan studied the effects of interruptions on users of e-commerce applications [Xia and Sudharshan, 2002], and drew some conclusions that can also be applied to collaborative and social fields. While they found that frequency of interruptions increased the time necessary to accomplish a task, they also found that giving the user some degree of control over interruptions led to improved user attitudes to them.

J. E. Fischer et al. [2010] observed that while the timing of pushed notifications was important to users, the *content* was generally considered more so. They conjectured that this is partly due to users’ personal methods for mitigating unimportant interruptions, and that a distinction is made when the content of the interruption demands attention. A side result from their research was also that many users have an aversion to *push* type notifications, preferring instead a *pull* strategy.

Okoshi, Tsubouchi, and Tokuda [2019] emphasised the growing importance of personalized notification strategies. They showed that machine learning models trained on individual preferences and behavioural data could significantly improve the relevance and timing of notifications by performing interruptibility estimations, addressing both content

and timing concerns.

Pejovic and Musolesi [2014] identified that in an environment where pervasive technology makes it hard to escape unwelcome interruptions, the application of context to a notification strategy can play a valuable role. They found that use of context such as user location, activity and collocation with other users to improve timing of interruptions, caused those interruptions to be viewed more favourably and acted on more quickly. We can learn from this research that as well as *whether* and *how* to notify users, *when* is also an important consideration for any notification that is likely to interrupt the user.

Iqbal and Brian P. Bailey [2008], among others, have explored the effectiveness of scheduling notifications around natural *breakpoints* in a user's work pattern, which has been shown to reduce the effect of interruption on productivity [Monk, Boehm-Davis, and Trafton, 2002] [Adamczyk and B P Bailey, 2004]. Iqbal and Bailey found that by deferring notifications until breakpoints, they achieved faster reaction times and lower levels of frustration for users, with study participants reporting that this approach was consistent with their own notification preferences.

2.2.4 Application to Research Questions

2.2.4.1 Awareness

The conceptual anchor for this work is *awareness*; we can see that this is a term with many interpretations. We should bear in mind that this research aims to address cases where users are probably not physically collocated (or at least where it is unimportant whether or not they are), and that we are not trying to solve some of the classical workspace problems of awareness.

The discourse between Tenenberg *et al.*, Harper and Schmidt poses some interesting questions about the nature of collective endeavour. We can say that a notional awareness agent, acting as a proxy for a user in a constellation of independent actors and systems, represents the opposite of the shared intentionality envisaged by Tenenberg. This research is predicated on an assumption that each agent is an independent actor; while we may say

that many of the actors might share goals, the presumed role of the awareness agent is a selfish one on behalf of its owner. While a level of enlightened self-interest might prove useful in the design of an effective awareness agent, this is far from a group mind type of shared intentionality among multiple actors in the system.

The work of Sorower *et al.* gives some useful context to how an agent might be trained, as it covers implicit vs. explicit training. The relatively low participation rate for providing explicit feedback should be noted in particular.

The concept of an information half life is important because it drives the time value of information, influencing any notification strategy that we might adopt for a given item; for example, if we have a way of identifying items that have a high information value over a short time period but a reduced value over a longer period, we might choose to prioritise those for intrusive notification.

2.2.4.2 Notification Strategies

Notification and alerting is an important element of an awareness agent to get right from a user's perspective. The reviewed literature shows a clear relationship between interruptive notifications and loss of productivity for users, yet it also shows that users value notifications in the right context and quantity, provided the information in them is relevant.

We can concentrate on the following aspects of alerting:

- *Timing*: not all moments are equal when it comes to receiving notifications and our strategies must take this into account.
- *Content*: the content of a notification is important to users and significantly affects its utility.
- *If*: should an alert or notification be issued at all? A solution should be able to discriminate between incoming items of information well enough to safely discard some.

2.3 The Social Machine

This section discusses several linked concepts – looking at the concept of the Social Machine (and the closely related concept of the Social Web), and Semantic Web and Linked Data standards that underpin much research in this area. Many of the terms used in these fields are often employed to some degree interchangeably, or there is ambiguity in specific meaning, as well as some overlap in the function of some of the relevant standards; we will attempt to clarify this to some degree.

2.3.1 The Social Machine and the Semantic Web

Berners-Lee introduced the concept of the social machine in his 1999 book, *Weaving the Web* [Berners-Lee and Fischetti, 1999]:

Real life is and must be full of all kinds of social constraint – the very processes from which society arises. Computers can help if we use them to create abstract *social machines* on the Web: processes in which the people do the creative work and the machine does the administration. . . The stage is set for an evolutionary growth of new social engines. The ability to create new forms of social process would be given to the world at large, and development would be rapid. (pp. 172–175)

Jim Hendler and Berners-Lee [2010] noted that the concept of the social machine had been realised in practice, citing as examples wikis, blogging platforms and social media applications in particular. However, they also expressed a view that these were merely early implementations of the social machine, limited by the fact that they functioned largely in isolation from one another. Their approach to advancing the state of the art rested in large part on the Semantic Web [Berners-Lee, James Hendler, and Lassila, 2001], “an enhancement that gives the Web far greater utility” [Feigenbaum et al., 2007]. They noted that an important property of the Semantic Web is that it raises the level of abstraction offered to programmers above that offered by the Internet and then the Web, “allowing programmers and users to make reference to real-world objects – whether people,

chemicals, agreements, stars or whatever else – without concerning themselves with the underlying documents in which these things, abstract and concrete, are described”.

The importance of contextual mechanisms is also considered: the type of social machine they proposed “must have an ability to be able to appropriately apply different policies in different situations, based on their use contexts”. Interestingly, Hendler & Berners-Lee noted that many datamining and knowledge discovery techniques of the time relied on the use of domain-specific knowledge to function properly, and cautioned that this knowledge can bias the results of the system, yet there has been a subsequent movement to formalise the process of domain-specific knowledge usage [Vaithyanathan, 2016] [Sinha, 2016] [Dinh and Tamine, 2012]. They identified the problem as being that such knowledge is often embedded in the procedural code or algorithms of systems, where any introduced bias cannot easily be isolated, identified and corrected. They posited that the solution to this is to design systems where the context applied to the data is logically separated from the mechanisms for processing, and that multiple domain contexts could equally be applied to the same data set (possibly producing different results). In this way, bias can be explicitly accounted for or even controlled for.

The third pillar identified by Hendler & Berners-Lee is that of information access and control, with particular emphasis on provenance of data and *information accountability* [Weitzner et al., 2008]. They cited e-Science as one area where this was of importance [Simmhan, Plale, and Gannon, 2005] [Deelman et al., 2009]. By way of example, they give an example of the ‘polite’ vs. ‘impolite’ web crawler, being held to account by way of the instructions given to them and the adherence (or otherwise) to those instructions revealed in servers’ web logs. This is a case where the receiver of information is being held accountable by the source.

N. Shadbolt [2013] described the growth of the Semantic Web in his paper, *Knowledge acquisition and the rise of social machines* discussing Gaines’ work [Gaines, 2012], noting that between 1996 and 2002 the proportion of papers referencing this topic at EKAW⁴ had gone from zero to 22 out of 34. Shadbolt augmented Berners-Lee’s concept of social machines with two examples: the creation of a temporal and geospatial archive of events

⁴<http://ekaw.org/>

in the 2007 Kenyan general election, built by people using blogs, mobile phone messaging and web applications⁵ [Okolloh, 2009]; and the spontaneous crowdsourced creation of detailed maps of Port au Prince using technologies such as WikiProject and OpenStreetMap, after the devastating 2010 Haiti earthquake⁶.

Shadbolt cited these as examples of a “new kind of emergent and collective problem solving that is anticipated in Gaines’ analysis. It is social computing, that generates social machines” [N. Shadbolt, 2013] [Jim Hendler and Berners-Lee, 2010]. He described these systems as being relatively simple in computation terms but being highly robust and providing comprehensive data, discussing Wikipedia⁷ and Galaxy Zoo⁸ as examples playing different roles. He employed a two dimensional chart, having compute/data complexity and social complexity as axes, to differentiate social computation from conventional computation. In this representation, conventional computation (such as air traffic control systems or climate modelling) occupies a band close to the Y-axis – that is, having low social complexity but varying compute complexity. Conversely, social computation is arranged along the X-axis – that is, having greater (but varying) social complexity than conventional computing. However Shadbolt also saw social computing extending up in the Y-axis of compute complexity, even beyond the current level of conventional computing. This was a *futures* analysis – Shadbolt was postulating that social machines could in future have both high social and computation complexity (the top right quadrant of the chart), and noting that this was an area for development.

2.3.2 Web or Machine?

The social machine has also been described as the ‘Social Web’ by Smart and N. R. Shadbolt [2014], with the Web serving as the platform on which various social processes are implemented. They note that, while the processes are a mixture of the familiar and the novel, all of them are shaped by the nature and properties of the Web. Smart & Shadbolt made the point that the actual meaning of the term ‘social machine’ remained the subject

⁵<https://www.ushahidi.com/> [<https://perma.cc/Z6J7-EQHK>]

⁶<https://www.ushahidi.com/blog/2012/01/12/haiti-and-the-power-of-crowdsourcing> [<https://perma.cc/EX5H-PZZK>]

⁷<https://www.wikipedia.org/>

⁸<https://www.zooniverse.org/projects/zookeeper/galaxy-zoo/> [<https://perma.cc/89QW-NGB9>]

of some confusion (and arguably still does). Active human participation is identified as a critical element of a true social machine (as opposed to passive content consumption for example), but beyond this there is limited consensus on what exactly constitutes a social machine except by identifying examples such as Facebook, Twitter, Ushahidi, Galaxy Zoo, and reCAPTCHA.

They went on to examine the statement by Berners-Lee and Fischetti [1999] defining social machines as “processes in which the people do the creative work and the machine does the administration”. In particular, they queried the definition of ‘creative work’, questioning the extent to which this activity was exclusive to the human elements in the social machine, with the computer elements being confined to administrative duties. While some kinds of content were indeed considered to be something that only a human can create (with Wikipedia content cited as an example of this), they argued that this was not exclusively the case. Their preferred characterisation is that “social machines are best understood as systems in which human and machine components make complementary contributions with respect to the performance of some larger joint process”. Given the substantial expansion of synthetic creativity offered by cognitive systems since then [2.5.1], this viewpoint has aged well.

One area in which we might think to challenge this definition is the notion that a social machine is necessarily web-based. While Smart & Shadbolt rightly note that the existing examples of social machines are web-based, and that the nature of the web has shaped the development of these machines, in our view there is no reason why the web must be the *only* platform on which a social machine can function. For example, while software agents forming part of a social machine may well operate using the Web, this is not necessarily the only mechanism that they may function on; and non-Web protocols may be used to interact and to access information.

We should consider the social web or social machines in the context of the W3C Social Web Working Group (SocialWG) definition of Social Web Protocols W3C 2017. The group’s output, defined in the Social Web Working Group Charter⁹, defined deliverables in the contexts of Social Data Syntax, Social API, and Federation Protocol; we can consider these

⁹<https://www.w3.org/2013/socialweb/social-wg-charter>

to form a suite of standards that can be used to design and evaluate a social machine. The SocialWG acknowledged the existence of multiple specifications in this space and detailed the relationship between them and how they covered the spread of standards of interest to the group¹⁰ ¹¹. We note that the proposed ActivityPub protocol¹² covers a wide range of the group's requirements, although in many cases taking a complementary role to other protocols such as Linked Data Notifications (LDN) [Capadisli, Guy, Lange, et al., 2016]. New research or development in the area of social machines should be conducted as much as possible within the framework defined by the SocialWG, building on these specifications (although this leaves the matter of which of the overlapping tools is most suitable for any given job). Considering ActivityPub in particular, several projects are already implementing some subset of the specification¹³, including for example dokiel [Capadisli, Guy, Verborgh, et al., 2017], Mastodon¹⁴ and Bridgy Fed¹⁵.

2.3.3 Linked Data and the Semantic Web

Echoing how lighter weight [Mousavi, 2011] RESTful web services [Severance, 2015]¹⁶¹⁷ were fast to gain traction in the commercial world over 'heavyweight' XML Web Services¹⁸ [Peinl, 2016], we can compare the lighter weight implementation of Linked Data [Auer et al., 2013] with the full vision of the Semantic Web. In this sense we can think of both RESTful web services and Linked Data as pragmatic alternatives, with a lower barrier to entry, than the heavyweights that they either complement or supplant (depending on your point of view¹⁹²⁰). It's notable that Berners-Lee himself has described Linked (Open) Data as "Semantic Web done right"²¹. A significant point to note about Peinl's analysis is that it is a study of *corporate* adoption; we can infer two things about commercial users: firstly,

¹⁰https://www.w3.org/wiki/Socialwg/Social_API/Requirements

¹¹<https://www.w3.org/TR/social-web-protocols/#requirements>

¹²<https://www.w3.org/TR/activitypub/>

¹³<http://activitypub.rocks/implementation-report/> [<https://perma.cc/N3TN-GM4N>]

¹⁴<https://joinmastodon.org/> [<https://perma.cc/92VP-VXAN>]

¹⁵<https://fed.brid.gy/> [<https://perma.cc/9QEK-4QXQ>]

¹⁶<https://www.w3.org/2005/Talks/1115-hh-k-ecows/>

¹⁷<https://www.w3.org/TR/ws-arch/#relwwwrest>

¹⁸i.e. using WSDL/SOAP/RDF – <https://www.w3.org/TR/ws-arch/>

¹⁹<http://tomheath.com/blog/2009/03/linked-data-web-of-data-semantic-web-wtf/> [<https://perma.cc/J76A-5XYJ>]

²⁰<http://vadimeisenberg.blogspot.co.at/2011/10/on-difference-between-linked-data-and.html>

[<https://perma.cc/35ZC-R5HK>]

²¹[https://www.w3.org/2008/Talks/0617-lod-tbl/#\(3\)](https://www.w3.org/2008/Talks/0617-lod-tbl/#(3))

that they do see a need for linked data; and secondly that they tend to favour a more pragmatic “get it done” approach.

We can argue that Linked Data as an expression of the Semantic Web vision presents a more easily adoptable concept in practice – especially for commercial rather than purely academic applications.

2.3.4 Application to Research Questions

If we think of awareness as the conceptual basis for the research, we can say that the social machine describes its embodiment (and we might describe the semantic web & linked data as the plumbing). The literature on awareness – such as that of Luff, Heath, and Svensson [2008] – supports the concept of an awareness agent as part of a social machine, consisting of many actors and systems.

We note that the social machines described by N. Shadbolt [2013] tend to both high social and computational complexity, and this is consistent with the designs we wish to investigate. The computational effort required to discriminate between information sources when determining a data presentation strategy is not trivial, but we can probably say that we now have some resources at our disposal that were conjectural to Shadbolt in 2013.

Smart and Shadbolt’s views on the differentiation between administrative and creative work within a social machine are also important. While we consider that an awareness agent might primarily be doing the mundane work of sifting through information on the user’s behalf, the elements of implicit learning and advanced content selection blur the lines – we don’t wish to evaluate the awareness agent as a dumb factotum but instead ideally a fuller partner in the user’s process of comprehending information and interacting with others. We need to evaluate the success of a solution based on how well it is able to rise to this challenge.

It is implicit in the research questions that the awareness agent is operating in a networked web based environment, where it can both consume and expose web services. The research points to linked data as the most apposite channel for inter-agent and inter-actor communication.

2.4 Software Agents

We are interested in a software agent intended to enhance the process of human cognition of information in a collaborative or social environment, but wish to stay on the right side of the line between understanding and using agents as a tool, and on the other hand the study of agents themselves. Our work is *not* a study of agent design theory in and of itself, but instead we study the use of an agent – the awareness agent – as a means to an end, specifically supporting users with IO issues in a connected environment. Nevertheless, we should establish a grounded understanding of the theory of software agents in order to frame this research. This section reviews some of the significant literature in the area of software agents with two aims: 1) to be able to describe, based on reference to established theory, the agent-like properties of our proposed solution; 2) to inform the design process of the awareness agent.

2.4.1 What is an Agent?

In their 1995 paper, *Intelligent agents: theory and practice*, Michael Wooldridge and Nicholas R. Jennings [1995] documented a number of features of a software agent and postulated some hypothetical examples of agents that demonstrate those principles. Of particular relevance here, some of the examples they used were based on the notion of a “personal digital assistant (PDA)” that performed tasks on behalf of the user including information retrieval and differentiated handling of notifications. They noted that the technological basis for agents of such sophistication did not yet exist, but progress in Artificial Intelligence (AI) and general computing systems since then has made these far more realisable now than they were then.

Wooldridge and Jennings also reflected on the uncomfortable nature of the question, “what is an agent?”, and addressed this by defining two notions of agency: weak and strong. Under the weak notion, an agent is considered to enjoy the following properties:

- Autonomy: agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state [Castelfranchi, 1995];

- Social ability: agents interact with other agents (and possibly humans) via some kind of agent-communication language [Genesereth and Ketchpel, 1994];
- Reactivity: agents perceive their environment, and respond in a timely fashion to changes that occur in it;
- Pro-activeness: agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking the initiative.

They also discussed a stronger – and in their view potentially more contentious – notion of agency that was more popular in the field of AI in particular, noting that “these researchers generally mean an agent to be a computer system that, in addition to having the properties identified above, is either conceptualised or implemented using concepts that are more usually applied to humans”. Among other examples, they referenced Shoham’s view [Shoham, 1993] that it was common in AI to characterise an agent using mentalistic notions, such as knowledge, belief, intention, and obligation. Of these, *intention* is probably the hardest to rationalise as a property of an agent in the absence of true intelligence; how can we truly say that a computer system intends to do something when it is merely reacting to a combination of instructions and events? The authors approach this via technical and philosophical viewpoints, including those of Dennett [1987] and McCarthy [1979]. They contend that such mentalistic properties become most useful as a way of comprehending the behaviour of more complex systems: “The intentional notions are thus abstraction tools, which provide us with a convenient and familiar way of describing, explaining, and predicting the behaviour of complex systems”. We expand on this topic later in the chapter when discussing cognitive computing [2.5.2].

A comment by Petrie [1997] – that is admittedly taken somewhat out of context here²² – sums up one of the aspects where an agent can deliver most value: “One common intuition about agents is that they should surprise one by offering unexpected, but helpful, information”.

The value of an agent in the information sphere might be measured in its ability to high-

²²Petrie was actually making the point that this in itself no more defines an autonomous agent than it might a printer daemon, but we think that this is actually a behaviour that is valuable in itself for an agent designed to assist with awareness

light relevant and useful information of unexpected source and provenance. In this context, the term “unexpected” is used advisedly; as well as information coming entirely ‘out of the blue’, this term could also encompass information from familiar sources that the user wasn’t seeking out at any given moment. This would potentially be an example of computer support widening the scope of information that a user can comprehend.

When considering an architecture for a real-time agent, J. H. M. Lee and Zhao [2002] applied the cognitive psychology theories of Neisser [1976] on the perceptual process of human cognition to a design for a real-time agent. They transposed Neisser’s model elements to subsystems for Perception, Cognition and Action. While the paper’s focus on the demands of a true real-time system are probably not relevant for this research (although the notion that an agent itself may suffer information overload is an interesting counterpoint to this research theme), their overall architecture is potentially applicable.

Mathieu, Routier, and Secq [2002] also applied a human-centric understanding of agents, arguing that multi-agent systems can be seen as “societies of interacting agents”. As we will cover later, this is a compatible viewpoint with the concept of the social machine [Jim Hendler and Berners-Lee, 2010].

We are establishing this grounding of the definition of an agent in order to avoid devaluation of the term, which is particularly important if we later want to use it to describe an awareness agent. As Michael Wooldridge [1996] wrote in response Franklin and Graesser [1996], “if the term ‘agent’ is not to become an empty, meaningless term attached to everything [...], then we must try to label as agents only those systems that really deserve it.”

There is also a small matter of evolving terminology. In 2024, users are far more likely to think of interacting with *bots* [Ferrara et al., 2016] than agents. It could be argued that this is a matter of fashion [Canter, 2017], or that there is a distinction between bots occupying the front-end role and agents providing the back-end substance. Krafft, Macy, and Pentland [2017] provide a more constrained definition of a bot as “digital agents who act according to tailored algorithms, often as peers, in online spaces such as online social networks or chatrooms”. From this we can infer that the term ‘bot’ primarily refers to

the interactive facet of a software agent; a bot is a software agent whose chief role is to interact with peers (human and bot) on a given medium.

2.4.2 Multi-agent Systems

An agent does not usually exist in isolation, and in many cases is considered to form part of a multi-agent system (MAS). Michael Wooldridge [2002] stated that agents in a MAS have these important characteristics:

- Autonomy: the agents are at least partially independent, self-aware, autonomous.
- Local views: no agent has a full global view of the system, or the system is too complex for an agent to make practical use of such knowledge.
- Decentralisation: there is no designated controlling agent.

There is an assumption that agents in a multi-agent system are cooperating towards a common goal or set of goals [Kolp, Giorgini, and Mylopoulos, 2001] or at least work within a shared social environment. Many MAS implementations carry an implicit assumption that the agents within the MAS have a common design or progenitor, or at least are in some way orchestrated. Castelfranchi [1995] used the term ‘Commitment’ to describe the obligations an individual agent may have to a common goal, which may be commitments to an overall organisational role or to other agents individually. However, as Poslad [2007] noted while many MAS designers choose to assume that agents act sincerely in their interactions, this is not always the case (Poslad’s example of ecommerce was apt: an unquestioning assumption of the sincerity of those you interact with in business is a good way to lose money).

This has a particular relevance when we come to consider the common distinction between closed and open MAS: those where the agents are developed by parties with competing or differing interests, and where there is no external access to the internal state of a given agent, are often classified as ‘open’ [Artikis and Sergot, 2009]. Additionally, García-Camino [2009] introduced the concept of *regulation* of MAS in his thesis, to address the problem posed by N. R. Jennings, Sycara, and M. Wooldridge [1998] of how to mitigate

harmful system behaviour. In an open MAS, a given agent might be *expected* to undertake and fulfil certain commitments, but in the case of an unregulated system, there might be no central mechanism to hold the agent to these commitments other than what actions other actors in the system might permit it to take (we could call this concept *peer regulation*, which would necessarily be carried out by individual actors, although they may make use of centralised resources such as peer tracking and rating systems [S. Schmidt et al., 2007]).

Poslad [2007] described the environment in which agents function as an “Information Communication Technology (ICT) environment, often called the MAS platform, in which the agents are embedded”. It is in this context that the FIPA (Foundation for Intelligent Physical Agents)²³ standard specifications for MAS interoperability are designed.

Bogg, Beydoun, and Low [2008] considered a number of features that should be taken into account when making a decision about using a MAS or agent-oriented approach in general. These include the system’s distributiveness, robustness, flexibility and interactions. We highlight in particular their definition of ‘interactions – deliberation’: “Agents may be *proactive* or *reactive* in cooperating with other agents in order to achieve their goals. As a *social* entity, an agent may communicate with other agents. As an *autonomous* entity an agent may have control over if, when and how to cooperate and what information is relevant to the cooperation”.

Garcia, Giret, and Botti [2008] described a detailed framework for evaluating the design methodology and tools for a MAS based on what they considered to be a number of differentiating characteristics of an agent based on the architecture, design, functionality and implementation of the MAS. As we noted, the modelling requirements for an awareness agent are likely to differ from many typical MAS designs, but the evaluation criteria defined by Garcia *et al.* provide potentially useful support for the design process.

²³<http://www.fipa.org/>

2.4.3 Agents Applied to Information Management

The 1997 work by Derbyshire et al. [1997] on agent-based digital libraries is an early discussion of information curation using agents. They used a definition of an agent as a system capable of flexible, autonomous action; by ‘flexible’ they meant: reactive, proactive & social. With their case study on the Zuno digital library (ZunoDL), they defined three types of agent in the system:

- Producers: Producers correspond to owners of information – organisations or individuals that have content they wish to make available.
- Consumers: Consumers correspond to end users of the library, who obtain access to library services via a web interface.
- Facilitators: Facilitators take the role of brokers, mapping between producers and consumers.

Derbyshire *et al.* defined a specific case of Consumer agent – *User Interface Agents (UIAs)*: “UIAs represent the interests of the user within the ZunoDL system. Each user that is logged on to the system will be associated with exactly one UIA”. We will later examine how this relates to the concept of an awareness agent.

Tran and Hoang [2008] also described a system with agents that act on a user’s behalf – in this case, an agent system used to obtain information from heterogeneous web sites and inform its owner about content and events. An interesting feature of their research is the use of Web Ontology Language (OWL) [McGuinness and Harmelen, 2004] – approaching the challenge of giving meaning to disparate web sources by converting to OWL.

The concept of an agent based ‘personal assistant’ with the goal of enabling a user to find relevant information quickly in the Internet without suffering information overload has been explored before. For example by Mianowska and Nguyen [2010], who used the concept of personalisation as defined by Adomavicius and Tuzhilin: “the ability to provide content and services tailored to individuals based on knowledge about their preferences and behavior” [Adomavicius and Tuzhilin, 2005]. They noted that it’s not simply enough that the user should explicitly tell the agent what they are interested in, but that this

should be an autonomous function [Pan et al., 2007].

The common features of a personalisation/recommender system [Resnick and Varian, 1997] are discussed by Montaner, López, and De La Rosa [2003], who identified common features of various examples of these on the web in 2003. This work discusses several key concepts, including: initial profile generation, evolving the profile over time (in response to changing user interests for example), training the model via explicit and implicit feedback from the user, and profile exploitation (using the profile to offer items that are relevant to the user). Mianowska & Nguyen describe a JADE-based prototype of a system based on these principles; while it is limited to web search, its design is informative. Another example of a personalised internet assistant is from Kaboré et al. [2013], although this example employs a more limited architecture and is not agent based, instead providing a reactive service responding to user queries based on a profile of the user.

2.4.4 Agents' Place in the Social Machine

Yee-King, D'Inverno, and Noriega [2014] applied MAS theory to social machine design, developing work on MAS-based socio-cognitive systems (including crowd-based socio-cognitive systems (CBSCS)) [Pablo Noriega and Mark D'Inverno, 2014] [Castelfranchi, 2014]. They postulated a MAS description of social machines as: "systems which have large numbers of rational agents, each with the ability to model the other agents in the system, and that interact in order to achieve shared or individual goals." They illustrated this by looking at Wikipedia, which has a combination of human contributors and software agents that perform tasks such as basic formatting checks, with an overlap between the tasks performed by human and computer actors in the system.

Their application of MAS terminology provides a useful tool to help us describe a social machine. By considering all actors – human and computer – as 'agents' we have a model for describing the diverse elements of the machine in a uniform way, using a set of general characteristics that apply agent behaviours and concepts to complex social systems.

Other research looks at cases where agents designed to act as personal assistants take a role interfacing with other humans and agents. Maes [1994] looked at agents that act to

reduce information overload as well as making interaction easier for untrained users. She covered four use cases that still apply today: email handling, meeting scheduling, news filtering and entertainment media recommendation; while the technology has changed significantly, we can see that Maes' agents are compatible with the concepts of agents working in a social machine. More recently, the major technology companies have all developed virtual assistant software that understands natural language interaction and attempts to assist their users in multiple ways – such as Apple's Siri and Amazon's Alexa [Tuohy, 2024].

2.4.5 Application to Research Questions

If we wish to situate our solution as an agent according to the literature, we can apply the weak definition of agency – saying that we seek to address the research questions by designing an autonomous, social agent that is able to both react and act proactively in partnership with its user.

We can consider that the research questions imply an awareness agent to be an actor in an *unregulated, open* MAS; the questions ask how can an agent meet the user's needs rather than how can a system of agents meet them. This distinction is important because the design approach and considerations are very different. In our case, the heterogeneity of actors in the system is a crucial point – a notional awareness agent would act on its user's behalf in an environment where common design, goals and interests among the various actors is very much *not* a given. This viewpoint is supported by the definitions put forward by Yee-King *et al.*, which allows us to clearly define how a social machine may be considered as a MAS. The literature also throws up several frameworks that might serve as the practical basis for answering the research questions, most notably FIPA.

2.5 Cognitive Computing and AI

As with software agents, the theories of machine learning and artificial intelligence (AI) underpin this research but are not themselves the main topic. We aim here to give a view of some of the aspects of this field that have a bearing on our intended research, without straying into detail about the theoretical basis of AI.

2.5.1 Cognitive Computing and the Social Machine

The technology and associated terminology in the field of artificial intelligence has evolved rapidly. We take AI to be the overall term encompassing a number of sub-fields such as *Machine Learning* (ML) and more recently *cognitive computing* [Modha et al., 2011]. While ML was a set of specific techniques for detecting patterns and surfacing information, Earley [2015] wrote that the emerging field of cognitive computing is “about making computers more user friendly, with an interface that understands more of what the user wants”. There are obvious parallels with Human-Computer Interaction (HCI) in this description, but the distinction with cognitive computing is the emphasis on understanding the user, replicating some of their cognitive processes in order to better meet the interaction requirements. We can say that ML is a subset of AI, while cognitive computing is a specific application of AI. This depiction of cognitive computing is one that resonates well with the aims of this research: the application of AI to make the interface between user and online systems more user-friendly by easing the cognitive burden.

We see from this that AI may play a role in the social machine. As Hearst [1999] commented: “I suggest that to make progress we do not need fully artificial intelligent text analysis; rather, a mixture of computationally-driven and user-guided analysis may open the door to exciting new results.” This viewpoint suggests that cognitive computing be used to *supplement* human cognition rather than *supplant* it. This is an important assumption on which we base our concept of an awareness agent, and is distinct from the ‘cognitivist’ approach to AI where the aim has been to simulate the full range of biological intelligence [Langley, 2012] [Lieto and Radicioni, 2016].

We see a distinction between two interesting classes of task: those that are easy for humans but difficult for computers, and conversely those that are easy for computers but difficult for humans (ignoring for now the other two polar classes of tasks that are easy or difficult for both humans and computers). While much research understandably focuses on how to make AI better at tasks that are considered to be difficult for computers, we are more interested in playing to the strengths of either side: letting computers handle the tasks that are easy for them, in order to support humans to do likewise. This is a link between cognitive systems and the social machine: by dividing tasks between the human and computer elements of the machine according to natural ability to process those tasks, we can improve the overall efficiency of the machine. As Chi [2017] notes, as well as continuing to explore the boundaries of what is possible for AI to achieve, we should also be exploring the boundaries of human interaction with cognitive systems.

2.5.2 Theory of Mind and World Models

Theory of Mind (ToM) [Premack and Woodruff, 1978] in psychology and neuroscience refers to the ability to attribute mental states – such as beliefs, desires, and intentions – to oneself and others, and to understand that others may hold perspectives different from our own. This concept has been instrumental in exploring human social cognition and communication. More recently, ToM has been applied to AI, particularly in evaluating the capabilities of large language models (LLMs). Researchers have sought to determine whether advanced LLMs can mimic human-like ToM by performing benchmark tasks for social intelligence, such as predicting intentions.

Strachan et al. [2024] investigated the performance of LLMs on ToM tasks, comparing them to human participants in a diverse range of tests ranging from understanding false beliefs to interpreting indirect requests and recognising irony & faux pas. They observed that while LLMs performed at or above human levels in some tasks, their failures often stemmed from shallow heuristics rather than genuine inferential reasoning. While these findings demonstrated that LLMs exhibited behaviour that is consistent with the outputs of mentalistic inference in humans, they also highlighted the importance of systematic testing to ensure that comparison between human and artificial intelligences was performed

at more than a superficial level. Such systematic testing would help to differentiate between surface-level successes and robust ToM-like behaviours in LLMs.

To understand and predict others' beliefs and actions through Theory of Mind, an internalised representation of how the world works is needed. These comprehensive representations, often called 'world models' [Ha and Schmidhuber, 2018], form the foundation for reasoning about both the environment and the minds of others within it.

Vafa et al. [2024] examined the possibility that LLMs may implicitly learn world models, such as those underlying logical reasoning, game-playing or geographic navigation. Inspired by the classic Myhill-Nerode theorem from language theory, they used deterministic finite automata (DFA) as a theoretical framework to recover and then evaluate world model coherence. They found that despite LLMs performing well on standard diagnostics and common tasks, there was significant incoherence in world representation, leading to fragility in novel or subtly altered tasks. We can interpret this as exposing a lack of true understanding of the world on the part of LLMs – while they may produce good results in practice, this is driven by factors other than a coherent internal world model and true understanding.

Richards and Wessel [2024] looked at ToM in the context of an LLM-based conversational assistant named ToMMY, intended to support developers with understanding code. An interesting feature of this application was the personalisation of the user's experience based on their mental state as inferred by the assistant. In this, they were building on previous work showing that predicting and reflecting on mental states improved LLM performance [Zhou et al., 2023] [Leer, Trost, and Voruganti, 2023]. This work is effectively simulating a world model of the user's knowledge and goals to enhance interaction and understanding.

Street [2024] explored ToM and AI with a perspective on Alignment [Gabriel, 2020], examining potential applications and risks of LLMs' ToM-like capabilities in individual and group contexts – such as goal specification, empathy, and moral judgment. When considering ToM on an individual level, she noted a central challenge in personal computing: accurately characterising user goals and preferences without requiring explicit and ex-

haustive advance definition of these from the user. She considered that LLM ToM might be able to avoid the limitations of such inflexible rules-based systems based on insight into users' underlying thoughts, feelings, and desires.

Street et al. [2024] also looked more broadly at LLM performance in higher order ToM tasks, finding that the best-performing LLMs have developed a generalised capacity for ToM, with adult-level performance in a number of ToM benchmarks. They noted in conclusion that we "may have to recognise LLM behaviours that are functionally-equivalent to those of humans as evidence of a new kind of understanding that cannot be reduced to 'spurious' correlation" – suggesting to us that we can't so easily dismiss the idea that LLMs do form a useful world model of sorts, and that their performance in real tasks reflects something that we could consider to be a form of understanding (at some level).

Some research also related inter-agent cooperation to ToM. For example H. Li et al. [2023] evaluated LLMs in collaborative multi-agent tasks requiring ToM, finding that LLM-based agents demonstrated emergent collaborative behaviours but faced challenges in maintaining long-horizon contexts and avoiding hallucination. They proposed using explicit belief state representations to address some of the limitations of collaborative LLM-based agent. They suggested that incorporating these dynamic ToM mechanisms could improve the role of LLMs in complex teamwork scenarios, offering a model for simulating understanding in interactive systems. They also commented that their observations hinted at the potential emergence of advanced cognitive skills such as world knowledge understanding.

S. Zhang et al. [2024] considered ToM in the context of human-AI cooperation. Focusing on real-time shared workspace tasks, they focussed on Mutual Theory of Mind (MToM) in human-AI teams, designing a shared cooperative workspace and tasks based on the Overcooked game²⁴. They found that in real-time tasks within workspaces shared with agents, the ToM abilities of those agents could both enhance humans' understanding of the agent and make people feel understood. However, they also found that this did not significantly impact team performance. They also found that the humans in the system relied more on the non-verbal *actions* of the agents rather than verbal communication to understand those agents. These findings stress the need for balancing explicit and implicit

²⁴<https://www.team17.com/games/overcooked> [<https://perma.cc/28FN-2PEL>]

communication in designing AI systems for collaborative environments.

2.5.3 Cognition as a Service

A number of commercial organisations have continued to advance the field of machine learning and other AI capabilities delivered as a service, often referred to as Cognition as a Service (CaaS). The global market for cognitive services has expanded significantly, with a valuation of approximately USD 15.27 billion in 2024. Within this, it has been estimated that the commercial ML as a Service (MLaaS) market will be worth over US\$300 Billion by 2029 [Intelligence, 2024].

Emerging trends include the integration of advanced AI models, such as OpenAI's GPT-3 [Yenduri et al., 2024], the development of AI-specific hardware by companies like Nvidia, and the expansion of AI cloud services, democratising access to high-performance AI tools [Lins et al., 2021] [Ghassemi et al., 2023]. On a related note, Talia [2011] has studied the use of software agents as a cloud-based intelligent service, with similar benefits found in terms of convenience and flexibility.

2.5.4 Classifying of Content and Understanding Sentiment

Sentiment analysis and content classification [Witten et al., 2017] are two broad and intersecting fields; while a full review of each is beyond our scope, we focus on areas directly relevant to our work. We are looking at them in conjunction here because we are interested in the intersection of the two – where the sentiment and meaning of textual content significantly influence classification outcomes. We have a particular interest in research comparing LLMs to other AI tools, examining evaluation techniques and user-centric systems. Much of the research covered in this section is very recent, reflecting the dynamic state of current AI research.

Accuracy was a traditional issue with sentiment analysis: even with constantly improving techniques [Liu, 2012] [Prabowo and Thelwall, 2009] [Clavel and Callejas, 2016], human language does not lend itself to neat machine classification. The use of sarcasm and irony

are common offenders for example [Maynard and Greenwood, 2014] [Mukherjee and Bala, 2017] [Liebrecht, Kunneman, and Bosch, 2013]. Clearly, sentiment analysis is also not the only way to analyse text, and it does not provide all the answers. In many case, users want to know the broad thrust of what is being said on a topic, perhaps summarising common themes or understanding how topics of conversation evolve over time [Cui et al., 2011].

Recent advancements in LLMs, such as GPT-type architectures, have demonstrated strong capabilities in both sentiment understanding and content classification tasks, often outperforming traditional approaches. For example, the application of multilingual text categorization techniques to social media data, as examined in Manias et al. [2023], highlights how such models handle the complexities of classifying diverse linguistic inputs. Similarly, Jin et al. [2024] introduced MM-Soc, a benchmark for evaluating multimodal LLMs' capacity to comprehend and classify social media content, addressing challenges specific to modern platforms like sentiment nuance and misinformation.

We have also seen LLMs integrated into content moderation tasks, for example as explored by OpenAI themselves: <https://openai.com/index/using-gpt-4-for-content-moderation/>²⁵.

Thejaswee et al. [2020] covered a number of techniques and ML algorithms for text classification, comparing metrics including accuracy and performance across different classifiers.

Q. Li et al. [2022] conducted an extensive survey of text classification between 1961 and 2021, tracing the evolution of methodologies from traditional machine learning models, such as Naïve Bayes and Support Vector Machines, to deep learning approaches such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformers. The survey highlighted the transition from manual feature engineering in traditional models to automated feature extraction and representation learning in deep learning models, and provided a taxonomy of text classification techniques. They noted that the introduction of pre-trained Transformer-based models such as BERT and its successors, marked a significant milestone in text classification, enabling more robust semantic understanding and improved task generalisation across diverse datasets. They also noted that while some of the new models make continual improvement on the accuracy index

²⁵<https://perma.cc/6KUU-7642>

for classification tasks, these results can't indicate whether the model actually "understands" the text from the semantic level like human beings can (reinforcing the points on this that we covered in Section 2.5.2). They also noted that performance of classification models is significantly negatively affected by zero-shot or few-shot learning, due to a high dependency on numerous labelled data.

A study by W. Zhang et al. [2024] evaluated the performance of LLMs across various sentiment analysis tasks, including aspect-based sentiment analysis and multifaceted subjective text analysis. They compared LLMs to smaller language models (SLM) trained on domain-specific datasets, and found that while LLMs performed satisfactorily in simpler tasks, they lagged behind in more complex tasks that required deeper understanding. However, they did find that LLMs significantly outperformed SLMs in few-shot learning settings, which suggested their potential in cases where resources for annotating SLMs is limited. Zhang *et al.* also noted the limitations of current evaluation practices for assessing the sentiment analysis ability of LLMs, proposing a novel benchmark, *SentiEval*. One area that they highlighted was that existing evaluations tended to focus narrowly on specific tasks or datasets, noting that they wanted to provide a more holistic evaluation against more diverse content. They also noted that LLMs such as ChatGPT were very sensitive to variations in prompt design, and that such nuanced sensitivity introduced challenges for testing the sentiment analysis capabilities of these LLMs, commenting that a single prompt may not be universally appropriate for all the capabilities of a model. Their intent with *SentiEval* was to mitigate these the limitations, by breaking the boundary between individual sentiment analysis tasks to establish a unified testing benchmark, testing the model using natural language instructions presented in various styles, and equipping the benchmark with diverse but fixed instructions to improve stability and reliability across different LLMs and studies.

Sun et al. [2023] introduced an extensible, annotation-free and efficient framework for text classification via LLM that they named CARP (for Clue And Reasoning Prompting), to address two challenges that they had identified with models such as GPT-3: the lack of reasoning ability in addressing complex linguistic phenomena such as irony, and the

limited number of tokens allowed in in-context learning²⁶. These were significant reasons why LLMs using in-context learning had under-performed trained models. They designed CARP to enhance LLMs' reasoning abilities in text classification tasks by mimicking human decision-making, using three steps: firstly, to collect clues such as keywords & semantic relationships from the input text; secondly, to employ these clues for diagnostic reasoning, addressing linguistic complexities like negation and irony; and finally, to integrate this information to produce a final classification decision. CARP selected task-specific examples for in-context learning by utilising fine-tuned models, bridging the gap between LLM generalisation and training data specificity. The authors found that CARP outperformed benchmarks, particularly excelling in low-resource and domain-adaptive scenarios.

Krugmann and Hartmann [2024] explored the proficiency of LLMs, such as GPT-3.5, GPT-4, and Llama 2, in sentiment analysis within marketing research, benchmarking the performance of these against established transfer learning models including SiBERT and RoBERTa. They found that LLMs – particularly GPT-4 – often matched or exceeded traditional methods in zero-shot settings, in particular with binary sentiment tasks. However, they also found that the performance of LLMs tended to diminish with the complexity of the task or when analysing data with informal language, such as tweets. They highlighted the impact of textual features, such as word length and document structure, on classification accuracy and explored the influence of prompting techniques like few-shot and contextual prompting.

They found that of the LLMs that they tested, GPT-4 showed the strongest performance in binary sentiment analysis (the first of their three experiments), which surpassed all other models except for SiBERT. Overall however, none of the three LLMs tested consistently outperformed traditional transfer learning models in sentiment classification of Tweets, with a transfer learning model achieving superior accuracy over the LLMs in three out of the five Twitter datasets examined. The results indicate that the general capabilities of LLMs may not always be able to match those of specialised models as the complexity of classification task increases.

They also found it noteworthy that none of the LLM models adhered to the prompted

²⁶They specifically noted the context window for OpenAI's GPT-3 of 4,096 tokens

instruction of exclusively using positive or negative sentiment classifications in their responses, with neutral sentiments sometimes being used counter to this. This is a point that is relevant to the question of trust [2.5.8].

Their second experiment, an impact analysis of prompting method and data characteristics, highlighted the central role of the prompting technique. They found that variability in results with prompt selection posed an important reproducibility challenge, but also suggested that automated prompt optimisation may bring significant advantages. They also noted that an observed decrease in accuracy for more complex classification tasks indicated an increased need for task-specific fine-tuning for LLMs.

Their third experiment looked at explainability of sentiment classifications, traditionally perceived as a “black box”, noting that the achievement of the highest explainability rating by open-source model Llama 2 stood out as a significant finding. This suggested that open-source models could lead advancements in transparent and explainable AI (although they cautioned that while explanations generated by LLMs are helpful, they do not decode the full inner working mechanisms of those highly complex models).

They also commented in conclusion that it is essential to manage the use of LLMs in research – taking into consideration their inherent biases, tendencies, and reproducibility challenges – to ensure that results sourced from these are both accurate and ethical.

Z. Wang et al. [2024] presented a novel framework using LLMs for text classification. Their intent was to make text classification accessible to non-expert users, simplifying workflows by eliminating the need for extensive preprocessing, domain-specific expertise and feature engineering. Their framework supported zero-shot, few-shot, and fine-tuning strategies for tasks including sentiment analysis, spam detection, and multi-label classification. Incorporation of domain knowledge into the framework was via customised prompts. Implementing an evaluation subsystem to continuously monitor performance and reliability, the authors also introduced a new evaluation metric – the Uncertainty/Error (U/E) rate – to assess model reliability under unknown or uncertain conditions. This metric quantifies the frequency at which an LLM either refuses to classify content, or provides an output deemed unrelated to it or hallucinated.

Their experimental results showed fine-tuned LLMs outperforming traditional ML models and neural network architectures, particularly on domain-specific tasks. However, the study identified limitations, including inconsistent output formats, content classification restrictions, and high hardware demands for LLMs. Despite these challenges, the authors believe that their proposed system democratises access to advanced AI technology.

2.5.5 Content Assessment and Recommendation

Content assessment and recommendation is interesting to us because much of the work looks at the understanding that LLM-based systems have of the content and meaning of texts (or at least their ability to generate outputs apparently consistent with those founded in understanding).

Minaee et al. [2021] conducted a comprehensive review into deep learning-based text classification, reviewing more than 150 models and 40 popular datasets. They noted that new transformer-based LLMs have surpassed traditional models in many Natural Language Processing (NLP) tasks – a trend that has arguably accelerated in the few years since this paper was published in 2021. This has largely been due to parallelisation, deeper contextual understanding, and pre-training on large corpora.

Han [2020] looked at personalised news recommendation and collaborative filtering, taking information overload as a starting point: “the massive amount of information presented at the same time makes it difficult for users to discover what they are interested in.” Han’s approach starts with building a user-interest model for each user, in their case from browsing data rather than explicit input, and then uses an algorithm to recommend content that is similar to the user’s model which does not already feature in their browsing history. They extend this to collaborative filtering/recommendation by comparing the models of multiple users to establish similarity between users and make recommendations based on that. The paper discusses how Han’s improved algorithm addresses specific issues with the existing approach: *data sparsity* – in many large recommendation systems, the user-item rating matrix is often sparse because users typically rate only a small fraction of the available items, a point of interest to us; *cold start* – dealing with new users; *scalability*

- computational complexity rising with user count; *recommendation accuracy* – failure of traditional algorithms to capture nuance; *dynamic user preferences* – change of individual preferences over time.

Franco, Gaggi, and Palazzi [2023] investigated how LLMs can improve content moderation processes. Their rationale stemmed from the perceived need to address the shortcomings of existing moderation approaches, which often failed to consider the diverse and context-specific needs of users. Traditional moderation systems have been criticized for being unfair to minorities and fragile users, lacking transparency, and failing to provide adequate explanations for their decisions. The authors argued that LLMs can help overcome these challenges by offering a more flexible, transparent, and user-centric moderation process. By providing detailed explanations and allowing for rule customisation, LLMs could in theory make moderation systems more inclusive and effective in managing harmful content, while respecting user preferences. They found that while LLMs can address these issues, the implementation must be carefully managed to address ethical, privacy, and operational concerns. The proposed hybrid approach – combining AI capabilities with human oversight – aims to create safer and more inclusive online environments. We will discuss some of these considerations a little more in section 2.5.8.

Chan et al. [2023] also looked at the content moderation space, discussing using culturally attuned LLM models to recognise cultural and societal variations in what is considered offensive content when performing content moderation on social media.

2.5.6 Addressing Information Overload with Cognitive Systems

The problem of information overload is a long-standing challenge in domains like email management, where the volume of irrelevant content hampers human cognition. Spam disrupts human cognition of email by flooding them with unwanted content; a term originally applied to unsolicited commercial email (UCE), the volume of email of dubious relevance within many corporate environments has also come to be known as ‘spam’. The task of filtering spam exemplifies how AI can assist by triaging and classifying content to alleviate this cognitive burden [Klimt and Yang, 2004]. AI algorithms for spam filtering

have achieved varying degrees of success [Seewald, 2004] [Metzger, Schillo, and K. Fischer, 2003]. Moreover, studies such as Stumpf et al. [2009] have explored how human feedback can enhance AI-based triage processes, underlining the collaborative potential between humans and AI.

Recommender systems represent another avenue for AI to address information overload, aiding user cognition through scoring and ranking mechanisms [Aljukhadar, Senecal, and Daoust, 2010] [Martin et al., 2011] [Segaran, 2007]. Unlike spam filtering, recommender systems focus on tailoring content to user preferences, often using explicit ranking methodologies.

Algorithmic approaches to addressing information overload have been explored in various applications that have been described by Pan as “personalized information service agents” [Pan et al., 2007] or cognitive personal assistants. The agent described by Pan is interesting in that it attempts to learn a profile for the user running it; we can say that this is in some way the inverse of what many AI systems do – rather than analysing content and extracting scores, classifications etc., it analyses the user behaviour and identifies content for them.

A personal assistant application project that is interesting for us is RADAR (Reflective Agents with Distributed Adaptive Reasoning) [Steinfeld et al., 2007] [Faulring et al., 2010], a sub-project within the DARPA PAL (Personalized Assistant that Learns) programme; this agent system uses AI to classify incoming email into a set of tasks, partly building on the work of Shen et al. [2006]. The RADAR research is distinctive in that it invested significant resources (a team of undergraduate English majors) to create an email text corpus with a consistent and detailed ‘backstory’ for an analysis scenario, with a combination of signal and noise messages related to a plausible business example. This provided strong material to base the research on and the researchers were able to demonstrate that the use of ML has a measurable positive impact in this scenario over and above that provided by commercial off-the-shelf tools. This ‘old school’ approach to generating synthetic content is in contrast with the techniques that have become available since.

2.5.7 Evaluation of Generative AI and LLMs

A large survey by Chang et al. [2024] covers topics of evaluation of LLMs in significant breadth. In their rationale for the survey, Chang et al. noted “Evaluation is of paramount prominence to the success of LLMs due to several reasons. First, evaluating LLMs helps us better understand the strengths and weakness of LLMs. Second, better evaluations can provide better guidance for human-LLMs interaction, [...] Finally, as LLMs are becoming larger with more emergent abilities, existing evaluation protocols may not be enough to evaluate their capabilities”

Chang et al. categorised literature by according to *what, where* and *how* to evaluate, as well as summarising evaluation metrics, datasets and benchmarks, and considering future ‘grand’ challenges. Under the ‘what’ category, of most interest to us is natural language processing tasks, which they have subdivided into sentiment analysis, text classification, natural language inference, and semantic understanding. Of relevance to us is semantic understanding, which they describe as involving “the interpretation and comprehension of words, phrases, sentences, and the relationships between them. Semantic processing goes beyond the surface level and focuses on understanding the underlying meaning and intent.” Literature referenced for this particular topic include preprints by Tao et al. [2023] and Riccardi and Desai [2023], discussing techniques for evaluating event semantics and simple combinatorial phrases respectively. The latter is notable from our perspective as it discusses a study that compares LLM and human ratings for phrases in order to evaluate the effectiveness of different models – something not dissimilar to our own work in general terms. However, none of the work referenced by Chang et al. looks at our own area, namely applying the semantic understanding of a LLM to assess the appropriateness of classification of some text.

The category of text classification in Chang et al.’s study is of course also relevant to us. Overall, they noted, “LLMs perform well on text classification and can even handle text classification tasks in unconventional problem settings as well.” Of particular note, Peña et al. [2023] discussed a system of leveraging LLMs in conjunction with Support Vector Machine (SVM) classifiers for topic classification in the domain of public affairs.

This has a lot of overlap with our own work, in that we also use SVM classifiers alongside LLMs, but the application is different: while we look to use LLMs to evaluate the output of SVMs, Peña *et al.* utilised an LLM backbone in a processing sequence that then passed to SVM for classification. Nevertheless, their research highlights a number of points that are also relevant to our own work. One conclusion of Peña *et al.* that we should consider when it comes to our own experimental results is the strong performance of SVMs in the topic classification process: “The results show how text understanding models with SVM classifiers supposes an effective strategy for the topic classification task in this domain, even in situations where the number of data samples is limited”. While we note that this conclusion relates to the combination of SVM and LLM in a single classification process, the strength of SVM classification with limited data volumes is pertinent.

2.5.8 Human-AI Factors

The increasing integration of AI into various spheres of human life spurred the development of Human-Centred AI (HAI) [Xu, 2019], which emphasises the alignment of AI systems with human needs, ethics, and usability. Bond *et al.* [2019] proposed a multidisciplinary framework for HAI centred on the concept of Explainable AI (XAI), which aims to make AI decision-making processes transparent and interpretable in order to foster trust among end-users. They highlighted challenges that could be mitigated by HAI approaches, such as automation bias – the tendency of users to over-rely on AI recommendations – and algorithmic bias, which arises from socio-demographic disparities in training data. These challenges underscore the importance of involving end users throughout the AI development lifecycle, to design systems that are not only technically sound but also ethically aligned. Additionally, they explored ethical and practical implications of conversational user interfaces and the democratisation of AI tools, and advocated for greater AI literacy to mitigate risks associated with the widespread adoption of these technologies.

In their editorial in the IJHCI Special Issue on AI in HCI, Antona *et al.* [2023] presented papers examining the relationship between AI and HCI, highlighting potential of such collaboration to address challenges in technology design and deployment. These argued that integrating AI technologies such as natural language processing, machine learning, and

computer vision into HCI practices gave opportunities for creating systems that are more ethical, explainable, and responsive to human needs. Themes such as fostering user trust, enhancing transparency, and designing systems that prioritise human values and usability also recurred in this material. Authors also explored how HCI can contribute to responsible AI development by offering methodologies for explainability and fairness, and their paper demonstrates the growing importance of interdisciplinary approaches in addressing emerging issues such as trust calibration, ethical considerations, and the practical deployment of AI in daily life.

A number of researchers have explored issues of collaboration and trust in relation to LLMs. D. Wang et al. [2020] noted that this is not actually a new concept, pointing to Licklider's early discussion of Man-Computer Symbiosis [Licklider, 1960]. Technology advances since then have expanded the scope of the topic, and added to the core concepts.

Bansal et al. [2023] conducted a workshop at ACM CHI 2023²⁷ on Trust and Reliance in AI-Human Teams (TRAIT) with a focus on establishing appropriate levels of trust and reliance in human interaction and collaboration with AI systems. While the results of the workshop have not yet been published as of writing, the framing of it is useful to note: the authors noted the interdisciplinary nature of this topic – addressing trust in AI requires insights from the fields of HCI, AI, psychology, and social sciences. The intent of the workshop was to explore three broad aspects: 1) How to *clarify definitions and frameworks* relevant to human-AI trust and reliance (what does trust mean in different contexts?); 2) How to *measure trust and reliance*; and 3) how to *shape trust and reliance*?

Cabrero-Daniel et al. [2024] explored human-AI collaboration in the context of Agile development, looking at customised LLM meeting assistants. They worked with Agile practitioner Austrian Post Group, to attempt to answer the following questions using action research: 1) How can AI assist in identifying potential problems and risks in Agile meetings, and provide actionable and useful recommendations? 2) To what extent do the AI meeting-assistants generate sensible recommendations in the context of real meetings in real time? and 3) How do users perceive the AI meeting-assistants in terms of user experience, and what impact does it have on overall performance? This use of AI is not

²⁷<https://chi2023.acm.org/>

strongly related to our work, but some of the user perceptions exposed during the study are interesting. Participants experienced initial curiosity, with mixed reactions as the study progressed. While some appreciated the AI-generated insights for identifying issues such as over-commitments, others found the recommendations too prescriptive and feared enforced workflow changes. Many saw potential benefits with the AI collaboration, but concerns arose about the accuracy and relevance of AI's recommendations. Human oversight was considered to be very important. Other issues arose relating to communication style and customisation needs for individual teams. Overall, the study highlighted the potential of AI in Agile environments, while stressing the importance of careful design and respect for human expertise.

Correia et al. [2023] designed and evaluated SciCrowd, a hybrid human–AI tool for scientometric analysis, the measurement and analysis of scholarly literature. This is a large study, so we focus here on a few points arising from it. The authors noted that LLMs failed “to capture contextual insights into the structure, dynamics, and implications of scientific activity since they have limited ability to reason, interpret, and contextualize research outputs” in addition to issues arising from the quality, quantity and relevant of the training data. To address issues, they chose to integrate crowdsourcing using a Reinforcement Learning from Human Feedback (RLHF)-based model. They suggested that “the integration between AI-driven data discovery and human-driven crowdsourced interpretations enables a more differentiated assessment of scientific production”. By involving crowdsourcing, they sought to harness diverse perspectives and expertise in SciCrowd, with the intent of ensuring that its output was not only accurate but also relevant to the specific needs of researchers (users). Some of the issues experienced/reported are familiar: data quality, accuracy and relevance of output, transparency and replicability, and contextualisation of results. We can argue that this study – as with others – highlights common concerns among end users about the provenance and reliability of data from such systems, while also being positive about the value they can bring. In the end, the combination of AI and crowdsourcing was seen by the authors as a key success factor, allowing for more accurate and comprehensive scientometric analyses through human verification.

Returning to the Franco, Gaggi, and Palazzi [2023] paper introduced previously, we see how they have also proposed a human-AI hybrid system, combining AI capabilities with human oversight to address identified issues with LLM usage. Their proposed content moderation pipeline utilises LLMs for their ability to analyse data at high volume and velocity, but places human moderators in the loop. Immediate actions are taken where appropriate, but in cases where the confidence is not high, the content is flagged for human review. This provides human moderators with context in the form of the AI's reasoning for its decision, alongside the content itself, and they have the opportunity to override the AI's decision – which itself provides an iterative feedback loop for continual training and improvement. There is of course also a mechanism in their design for the human user to appeal, obviating the need for a low confidence score to cause human involvement. Many aspects of this process are similar to our own designs for evaluation with human oversight (or could be used as the basis for enhancements of our own process).

Siemon [2022] took the approach of considering AI systems as coequal teammates in collaborative work, noting the increasing trend for people to perceive computer systems as human (or at least with human-like characteristics). They noted that in order for AI systems perceived as an equal partner, they must "fulfill a compelling and consistent team role, but one that should not reflect an omnipotent and omniscient partner, rather a teammate who is also limited in its skills and abilities". This limitation is important, lest humans would instead become overly reliant on their AI-based teammate and eventually exert less effort [Karau and Williams, 1993]. Siemon recognised the importance of diversity within a team – including diversity of abilities, strengths and weaknesses – and proposed four team roles that could apply to AI members: Coordinator, Creator, Perfectionist, and Doer. Of these roles, the characteristics that most closely match a notional awareness agent are probably those of Doer, which are much more associated with concrete actions and prioritisation. More interesting maybe is the paper's discussion on how humans relate to an AI team member, and inferences we can draw from that which might apply to an awareness agent.

Kolbjørnsrud [2024] defined six principles for human-AI collaboration from an organisational management perspective:

1. **Addition Principle:** Adding more actors with higher levels of intelligence, whether human or digital, increases organisational intelligence.
2. **Relevance Principle:** The type of intelligence must match the nature of the problems to be solved. AI solutions often match or exceed human intelligence in highly specialised domains, but humans excel at solving ambiguous problems.
3. **Substitution Principle:** Replacing intelligent humans with intelligent machines does not make an organisation more intelligent but more efficient.
4. **Diversity Principle:** Increasing the diversity of intelligent actors (with different knowledge, skills, and mindsets, or different forms of AI), improves an organisation's ability to solve complex problems and adapt.
5. **Collaboration Principle:** Organisational intelligence requires collaborative skills from both human and digital actors.
6. **Explanation Principle:** Intelligent organisations seek explanations and act responsibly. Explainable AI is crucial for understanding AI models, detecting biases, ensuring accountability, and fostering human learning and motivation.

There are several relevant points in these principles – in particular the appropriate match between problem and type of intelligence tasked with addressing it; diversity of both skills and types of AI, and the importance of explainability. These are themes that apply to our own work also.

A Pew Research report by Rainie and Anderson [2017] examined concerns among professionals and the public about the role of algorithms across public life and the hazards of bias, filter bubbles, inequality and the stifling of choice & creativity. The report presented views on both the positive and negative effects of algorithms: on the positive side, algorithms can enhance decision-making efficiency, automate mundane tasks, and create personalised user experiences. Conversely, the article highlights several risks, such as the potential for biased or opaque decision-making, lack of transparency, and the challenge of algorithmic accountability. These systems – which are often invisible to users – shape

critical aspects of daily life. These range from content recommendations on social media to financial, legal, and medical decisions, raising concerns about fairness, privacy, and the ethical implications of handing over control to machine-driven processes. The article underscores the need for better oversight, transparency, and public awareness as algorithms become more ubiquitous in society.

Tsamados et al. [2022] examined the ethics of algorithms, building on the work of Mittelstadt et al. [2016] and noting both the potential for these to improve individual and social welfare and the significant ethical risks that accompany them. They noted that of the ethical concerns from Mittelstadt *et al.* refer to epistemic factors: *inconclusive*, *inscrutable*, and *misguided evidence*. Two are normative: *unfair outcomes* and *transformative effects*; while the final, *traceability*, is relevant both for epistemic and normative purposes. They discussed how algorithms often perpetuate social inequalities due to biased datasets, lack of transparency, and limited accountability structures – this highlights the risks of relying on algorithms for critical decisions, such as in criminal justice, healthcare, and social media, where they can reinforce unfair outcomes. The authors proposed potential solutions, such as implementing fairness checks, enhancing algorithmic explainability, and increasing public oversight. They also argued for interdisciplinary collaboration to develop more ethical, accountable algorithms that prioritise societal well-being over purely technological advancements.

Petrescu and Krishen [2020] explored the ethical and societal challenges posed by the increasing use of algorithms on social media platforms. The authors examined discussions around the 2020 Netflix documentary The Social Dilemma²⁸, conducting a semantic analysis of 8,812 Twitter messages to gauge public opinion on the issues raised by the documentary. Key themes included the manipulation of user data by major tech companies such as Google, Facebook, and Twitter, as well as concerns about freedom of speech and censorship. The paper also covered the issue of algorithmic bias and how such systems can perpetuate social inequality. The authors argued for more research into improving social media platform designs, calling for stakeholder collaboration to address ethical issues, privacy concerns, and user awareness. The study also referenced the Persuasion Knowl-

²⁸<https://www.imdb.com/title/tt11464826/> [<https://perma.cc/WL9Y-Q57E>]

edge Model [Friestad and Wright, 1995], which argues that consumers must be given the necessary opportunities and tools to learn about manipulation and persuasion from social interactions. It concluded with a call for the development of ethical frameworks and legal reforms to ensure that social media platforms are transparent and accountable in their use of data analytics.

People are aware of the importance of AI algorithms to their lives and social media usage; Oeldorf-Hirsch and Neubaum [2023] examined users' algorithmic literacy or awareness. Their survey of social media users in the USA and Germany highlighted differences in patterns across demographics within the study participants and also across the two geographies. They found a positive correlation between the youth and education level of participants with their level of both algorithmic awareness and positive attitudes toward engaging with algorithms. A higher level of positivity towards algorithms among the US users also highlighted the relevance of cultural factors.

2.5.9 Application to Research Questions

We can see that AI has a significant role to play in addressing the research questions. Each of the areas covered can be used to support the sifting and narrowing of information in a system that addresses information overload: scoring, classification, topic extraction, semantic analysis. The output from a scoring or recommender system may only be used as one input to the overall system; for example, while a Bayesian spam filter might assign a score to each email, the end user can determine which score thresholds relate to specific actions, or can override scores with their own logic-based algorithms. Pan's work, describing an agent that learns the profile of its user, can additionally be applied to the question of how an awareness agent might represent its owner to other actors, as this understanding of the user's preferences can be applied to both outgoing and incoming information.

These types of classification and meaning extraction can ameliorate information overload by allowing users to view items on aggregate ("show me statistics on negative sentiments") or to narrow down the content ("show me all the messages asking for my help"). These

support analysis of the relative effectiveness of these approaches.

The increasing commoditisation and distribution of artificial intelligence is important. By interacting with the cognitive component as a commodity service, we can to some degree avoid consideration of how this part actually works and focus on how it is used and the comparative quality of the outputs. It also makes it easier to exchange one commoditised service for another, which supports additional comparative study.

Many of the studies that we have reviewed show similar results about how people will engage with AI systems: a generally positive approach, tempered with awareness of the risk of trust issues that can be mitigated with transparency and engagement. This suggests a ready audience for an AI based agent to help individuals with information overload and similar problems.

The work of Vafa et al. [2024] and others into LLMs internal world models and Theory of Mind as applied to AI show us that we should be cautious about expecting behaviours from AI systems that could be attributable to those systems having a coherent internal understanding of the world. We have to take outputs from AI systems at face value, without ascribing deeper qualities – in our research this means that we should be cautious of those situations where we ask an AI to fulfil the role of a human in a system, and should seek to evaluate its effectiveness when we do this. While an AI system may be very effective at tasks where it substitutes for a human, we should acknowledge that this might not apply for all inputs and contexts, as the AI may adapt poorly to situations that are out of the ordinary.

The work of Strachan et al. [2024] supports the need to perform systematic testing of the performance of LLMs in tasks that emulate human reasoning and understanding. As with the findings of Vafa et al, this work highlights that the strength of LLMs in performing such work may be built on superficial foundations that need to be tested.

Although it was not the focus of her paper, Street [2024] illustrated for us how LLMs might be employed to address one of the main pain points in addressing information overload – the burden of setting up and maintaining explicit rules-based systems in communication or social-based applications.

2.6 Conclusions

When considering the gap in scientific knowledge that we aim to fill with this research, it is important to bear in mind that we are not aiming to advance any individual one of the domains studied in isolation, but to advance the way that those domains are used in conjunction.

The CSCW concept of awareness is well supported by existing research and is sufficient to provide a theoretical basis for our work; we can say the same about MAS theory, and in particular the link between MAS and the concept of the social machine defined by Yee-King, D'Inverno, and Noriega [2014]. However, the body of work examining the intersection between awareness and the social machine (in the context of supporting awareness for the human actors) is thin. While there is research on using software agents as assistants to address information overload, we have not found studies on this combination where a multi-agent system that forms a social machine.

On the one hand, the commoditisation of Large Language Models in particular allows us to bypass in-depth study on the internals of AI implementations work, but on the other we find that commoditised black box solutions carry trust risks that must be mitigated.

2.7 Chapter Summary

In this chapter we presented an overview of the literature relevant to our work, covering the four domains:

- Awareness [2.2]
- The Social Machine [2.3]
- Software Agents [2.4]
- Cognitive Computing & AI [2.5]

For each of these we considered how the topics that we had discussed applied to our research questions [3.3] and then went on to draw some overall conclusions.

Chapter 3

Problem Analysis and Research Questions

3.1 Problem Analysis

In the course of our literature review, we established a number of pain points experienced by users of information systems [2.2.2], in particular those systems that generate volumes of information and notifications for users to consume such as social media apps.

While techniques and tools that attempt to deal with personal information overload abound, we identified two gaps in the ecosystem that presented opportunities for further study:

- Solutions where the emphasis is on personal control or ownership by the user [2.2.4]
- Agent-based systems, where an agent acts on behalf of the user [2.4.5]

Many past solutions to information overload or content prioritisation did not adequately serve the user, often due to being ineffective or outside their control. The opaqueness of algorithms run by the large tech companies has long been an issue, and even when the code they use has been open-sourced¹, there is still a perceived lack of control [Chavez, 2023].

While some researchers have looked at the role of agents in solving IO problems [Kluge,

¹<https://github.com/twitter/the-algorithm>

Antoni, and Ellwart, 2020], there is not a strong body of work addressing this aspect.

As a corollary to the main problem of information overload and awareness, we also encounter practical questions about how to research and evaluate a solution due to the nature of the problem [Arnold, Goldschmitt, and Rigotti, 2023]. A user experiencing information overload may be receiving a large amount of information from multiple sources; asking them to evaluate a solution to this may mean asking them to process an even larger amount of data than they would otherwise contemplate, so that we might build a full picture. There is also the potential for privacy and data confidentiality concerns, with certain types of information such as private messages and confidential corporate material being unsuitable to use.

The topic of information overload is large in both the nature of the problem, and the scale of some of the solutions that might be implemented to address it. For example, a person experiencing this problem may be receiving content and/or notifications from multiple different sources (both work and personal) throughout the day, and the problem is experienced only when these are of a volume sufficient for it to become problematic. The volumes of data involved may also be high. Additionally, many areas where the users suffer the problem occur in applications that are walled (or partially walled) gardens [Kamdar, 2015].

These factors present some practical barriers to researching any comprehensive solution:

- We cannot make unlimited demands on the time of study participants
- Some data sources could be unavailable to use due to technical or policy barriers
- Testing a solution that consumes data indiscriminately may have problems with privacy and data security

These considerations have affected how we could approach this research, placing constraints on the scope of what can be implemented and tested, as well as the evaluation processes that work in practice. This in turn affects the scope and phrasing of the research questions themselves.

We decided to approach the research problem in part using a Research Through Design

(RtD) approach [Gaver, 2012] [Godin and Zahedi, 2014] in concert with simulated content production and evaluation, to enable us to design a solution to the problems and to test aspects of the solution in various ways. The resulting overall methodology is discussed in detail in Chapter 4.

3.2 Research Questions

The overall aim of our research is to investigate how we can design and build a software agent that will address issues of information overload and (lack of) awareness in an information-dense environment. We can approach this by considering:

- How to gain a better understanding of the nature and extent of the problem for real users of such systems
- What potential solutions or systems that could address some of these issues
- How a prototype system could be used to evaluate some aspects of the notional solution
- What methods we can use for evaluating such systems

These topics will broadly inform the research methodology, covered in chapter 4, and we have formulated the following Research Questions to drive our work.

3.2.1 Topic: Understanding Information Overload

Research Question RQ1: What problems with information overload are experienced by users of information systems and what attitudes do they have towards providers and solutions?

Section 2.2 of the Literature Review discusses issues relating to information overload in everyday professional and personal life. However, in order to develop a design to meet the needs of actual users, we believe that we need a more precise understanding.

We seek to improve our understanding of the following topics that affect real-world users:

- The extent of the effects of information overload
- Distinction between information overload in personal and professional spheres
- Overlap or conflict between personal and professional information load
- Attitudes to and experience with existing solutions to information overload
- Opinions on privacy and control in relation to information overload solutions

We propose the following hypotheses in relation to this, in order to frame the research and guide the methodology:

Hypothesis H1: Users desire better ways to manage information overload than are currently available.

Hypothesis H2: The distinction between work and personal communications is important to most users.

Hypothesis H3: Users are prepared to put in some effort to help a system improve their information overload situation.

3.2.2 Topic: Solution Development

Research Question RQ2: What design direction and system features can address information overload for diverse users managing multiple online information sources?

This primary question can be expanded into three sub-questions:

Sub-Question RQ2a: How can agent-based systems be designed to effectively manage and mitigate information overload across diverse online information sources?

Sub-Question RQ2b: How effective is a prototype of a designed system in reducing information overload and improving user awareness across multiple communication and information platforms?

Sub-Question RQ2c: What insights can be gained through the design and development of an agent-based system addressing information overload, particularly regarding user needs, design trade-offs, and implementation challenges?

We framed this question with the expectation of addressing it using a Research Through Design approach: in the context that will be provided by the answers we find to our understanding of the problem of information overload [3.2.1], what design can we develop for an agent that can address those issues?

We propose the following hypotheses in relation to the development of a solution design:

Hypothesis H4: A system with explicit user self-training can be effectively used to manage content from multiple sources.

Hypothesis H5: Users will see rapid value from efforts to train personalised content prioritisation models.

Hypothesis H6: Self-trained AI models are a viable alternative for personal IO management to other techniques such as filtering or third party systems.

3.2.3 Topic: Evaluation Techniques

Research Question RQ3: Can synthetic techniques be used effectively to study and evaluate potential solutions to Information Overload?

This primary question can be expanded into two sub-questions:

Sub-Question RQ3a: Is synthetic content an effective substitute for real content to support analysis of Information Overload problems?

This sub-question has the following aspects:

1. What limitations do we find with this?
2. Do users find synthetic content sufficiently realistic?

Sub-Question RQ3b: Can we use a synthetic evaluation approach to evaluate a potential solution or service to address Information Overload?

This sub-question has the following aspects:

1. How effective is synthetic evaluation compared to human evaluation for similar content?
2. What benefits do we get from synthetic evaluation over human evaluation?
3. What limitations do we find with this?
4. What factors affect synthetic evaluation quality? (i.e. topic, context, technologies used)

We are interested not only in the end ‘product’ – a solution design that could address issues with information overload and awareness – but also the methods for simulating and evaluating the functionality of the design.

We are specifically interested in:

- How can we use synthetic data in the process of prototyping and evaluating our design?
- What techniques can we develop to introduce synthetic evaluation to our research process?
- Can synthetic data be used successfully to avoid privacy and data confidentiality issues when conducting this type of research?
- How can we judge the effectiveness and reliability of these synthetic experimental components?

We propose the following hypotheses in relation to evaluation techniques and synthetic data:

Hypothesis H7: By using synthetic as well as real data for studies, we can avoid some data confidentiality and access issues that would otherwise limit scope without significant negative consequences.

Hypothesis H8: We can use AI-driven evaluation techniques to partially replace humans in studies of information overload solutions, minimising the problem of overloading the study participants.

Hypothesis H9: Human feedback on synthetic content and evaluation processes is an effective strategy to ensure overall experimental integrity.

3.3 Inferences From Literature Review

In this section, we will draw together some of the key points from the referenced literature and consider how they inform this research. We will address in particular how the current state of the art influences and directs the concept and design of the notional awareness agent. This analysis should form the basis for the design and testing of the agent.

3.3.1 An Awareness Agent Within a Multi-agent System

In the context of this research, we tend towards a more prosaic view of an agent, as something which exhibits certain behaviours and properties rather than ascribing higher mentalistic characteristics to them. As such, we could take a view that the weak notion of agency defined by Michael Wooldridge and Nicholas R. Jennings [1995] is a suitable model for our research, and the four properties they describe map closely to our notional design of how an awareness agent should behave.

We should also consider how – or if – an awareness agent forms part of a multi-agent system (MAS). Wooldridge's characteristics of autonomy, local views, and decentralisation appear to be consistent with an awareness agent. However, we should consider the compatibility of the notional awareness agent with some of the common norms for interaction within a MAS – namely the degree to which agents in a MAS are *cooperating*, and whether we can expect an awareness agent to be able to make a meaningful (and enforceable) *commitment* as defined by Castelfranchi and extended by others [Fasli, 2003].

If we consider the conceptual guide that an awareness agent is an agent that acts *on behalf of* an individual user and can interact with heterogeneous systems (many of which may themselves be unaware of the existence of an awareness agent), we have to conclude that an awareness agent would exist within an *unregulated* rather than a *regulated, open* MAS. The MAS has to be open, because there is no restriction on an awareness agent or other similar agent joining it, and similarly it must be unregulated because there is no central regulation system or commonly agreed protocol for regulation². Each agent

²For example, such as the use of a robots.txt file might be considered as form of commonly agreed regulation for web spiders

instance would be subject to no regulation other than its own parameters and such peer regulation that exists. This would lead us towards not formally considering an awareness agent as part of a MAS, but rather a standlone entity operating in a complex environment that may or may not contain other entities similar to itself.

3.3.2 Designing a Social Machine

We can utilise Yee-King, D'Inverno, and Noriega [2014]'s application of a MAS descriptive model to socio-cognitive systems. Having already discussed how actors in the system can be either human or computer, on this understanding that the human actors will have personal models of the social environment, we need to consider how to implement analogous models in the software awareness agent. This can have relevance both in the case of the agent representing its owner to the other agents in the system in an outward-facing sense, as well as for how the agent acts for inbound information.

An idealised agent's social model could account for what we might term as the social position of the other actors: are they known to us (and do we 'like' them)? Do they have bad habits such as promoting poor quality or irrelevant information? Are they newcomers or old hands? How do they get on with other actors that I already have a relationship with? And so on. Some of this might be formalised in any protocol that we develop for inter-agent communication – for example by using content signing and trust scores – but it may also need to be more generally implemented using flexible models to build understanding of people and topics so that they can be rated and handled appropriately.

3.3.3 Cognitive Computing and AI

As indicated, we seek to take a *commodity* approach to Cognitive Computing resources. To use the agent *belief* paradigm, the awareness agent should be able to use commodity ML services to help it inform its internal model and beliefs (for example the belief that a particular piece of work should be shared, or that a given actor is worth listening to). The nature of the services used in part depends on the functionality that we need them to facilitate.

On a pragmatic level, the work that we have reviewed on content classification and the relative strength of systems such as SVM classifiers [Peña et al., 2023] for quality and initialisation times, leads us in the direction of using content classification as a significant part of our design for an awareness agent. While classification is not the only way that we can make use of AI in our system, it is an established technique for alleviating IO [Faulring et al., 2010] [Klimt and Yang, 2004] that has options for flexible application within an agent.

On the other hand, we also see the potential of LLMs for processing and evaluating content [Franco, Gaggi, and Palazzi, 2023] [Minaee et al., 2021] [Riccardi and Desai, 2023], and this leads us towards a synthetic evaluation design that utilises commodity LLM resources.

LLMs also have potential for introducing higher order capabilities to an awareness agent, such as autonomy, a more sophisticated system of belief and intention and natural language communication with its owner. Some of this is outside the scope of the immediate research questions, but should be considered in our overall agent design.

3.4 Chapter Summary

In this chapter we conducted a problem analysis [3.1], leading us to formulate a set of research questions [3.2]:

- RQ1 [Problem Understanding]: What problems with information overload are experienced by users of information systems and what attitudes do they have towards providers and solutions?
- RQ2 [Solution Development]: What design direction and system features can address information overload for diverse users managing multiple online information sources?
- RQ3 [Evaluation Techniques]: Can synthetic techniques be used effectively to study and evaluate potential solutions to Information Overload?

We then went on to discuss a number of Inferences [3.3] from the Literature Review that would help frame our approach to these questions in subsequent chapters.

Chapter 4

Overall Methodology

4.1 Overall Approach

We have introduced the concept of an Awareness Agent [1.2.2] as a way to ameliorate information overload and address what we have termed as the awareness problem [1.2.1]. When considering our approach to the research problem [3.1], we opted in part for the Research Through Design approach described by Gaver [2012] and further documented by Godin and Zahedi [2014] – taking a generative approach of producing a theoretical design for our awareness agent concept, and then testing aspects of this out in practice with a partially functional prototype implementation.

Gaver emphasised that the knowledge generated through RtD is often provisional and context-specific rather than universal. Similarly, by implementing a subset of the full functionality of an Awareness Agent, we aimed to create a prototype that provides specific insights into how users might interact with the system and what challenges or opportunities might arise. We expected the findings from this prototype to be contextually rich, offering useful but provisional insights that can guide further development and refinement.

While Gaver suggested using annotated portfolios to document and communicate the outcomes of RtD, the evaluation of our prototype – including the insights gained from testing its functionality – can serve as a form of documentation that reflects the design decisions, challenges, and user interactions. This documentation should fulfil a similar role to anno-

tated portfolios in RtD.

The concept of Personas [Friis Dam and Teo, 2024] is also central to our work, both from a design and evaluation perspective, and we opted to ground our RtD design with a data-driven persona development process.

A final important consideration is the use of generative AI and other aspects of cognitive computing. This technology gave us the ability to expand our experimental scope by providing some capabilities that would otherwise be unavailable, and to introduce simulation and synthetic evaluation to our research method. Several elements of the methodology are thus influenced by or entirely dependent on this technology.

Our overall process was as follows:

- Conduct a survey to elicit users' experiences with information overload and multiple information sources, with emphasis on the distinction between different categories of content (work, personal etc.) and timing
- Model a set of personas to guide the design and support evaluation
- Design a platform that can be used to investigate how an autonomous agent can be used to manage social communications and alleviate the pain points that have been identified.
- Implement a version of the platform to facilitate testing of aspects of the general proposed solution
- Conduct a study using synthetic techniques with human supervision/input.

The following sections provide more detail on each element of the overall methodology.

4.2 Research Elements

In this section, we discuss a number of elements of the research, which form parts of the overall methodology.

4.2.1 Survey

We included a survey of users in our methodology for two reasons:

1. To substantiate our understanding of the problems experienced by actual users
2. To support the development of resources to inform our research and design

Based on this, we considered that the survey should have the following properties:

- It should be able to inform our understanding of IO as experienced by actual users
- Questions should gather opinions on specific areas of interest including:
 - Attitudes to notifications and interruptions
 - Prioritisation of different types of content
 - Importance of context such as time/place/mode to attitudes
 - Views on existing solutions and services
- It must elicit sufficient demographic information to allow us to potentially derive related persona types

We chose a limited distribution for the survey, with a focus on audiences that we had identified as likely to experience issues with information overload and possible potential candidates for a solution such as an awareness agent. The survey is detailed in Section 5.1.

4.2.2 Personas

Personas play an important role in our work: not only have we used them to help understand the problem and inform the design of our notional Awareness Agent, they also play a significant role in the evaluation process.

In his book *The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity*, Cooper [1998] argues that many software products fail because they are designed based on the technical preferences of engineers rather than the actual needs of users. By creating detailed personas – fictional characters that

represent different user types – designers can focus on real user behaviours, preferences, and goals. Cooper argues that this approach helps ensure that the final product is intuitive, accessible, and aligned with the users' needs, ultimately leading to more successful and satisfying user experiences.

Goodwin and Cooper [2009] also discuss personas as part of a design process. While Cooper introduced personas as a way to ensure user-centred design, Goodwin takes this further by integrating personas into the entire design process, using them alongside scenarios to envision how different user types will interact with a product in various contexts. This approach not only guides design decisions but also helps validate them by ensuring that the end product aligns with user needs and behaviours.

Our approach similarly integrates personas into a data-driven methodology, combining quantitative clustering with qualitative insights with the intention to create personas that are both evidence-based and reflective of real user experiences. This approach, emphasises the importance of grounding design decisions in a deep understanding of user needs, using personas as a central tool in this process.

Persona development is discussed in detail in Section 5.2.

4.2.3 Awareness Agent Application

Our approach to using an Awareness Agent prototype as part of our RtD methodology leans heavily on the “project-grounded research” approach defined by Findeli et al. [2008], integrating both design practice and research in an iterative process. The term project-grounded refers to the idea that the design project itself becomes the foundation for generating new knowledge. By following this methodology we have been able to focus both on creating functional design artefacts (in this case, the Awareness Agent prototype) and contributing to the theoretical understanding of how we can design and use such a system to manage personal information flow and information overload.

Findeli emphasised two crucial criteria for design research: rigour and relevance, arguing that rigorous research must meet the usual scientific standards, ensuring it stands up to critical scrutiny, while relevance demands that the research contributes to the improve-

ment of design practice. Our methodology embodies both, with the development of the Awareness Agent following a systematic process of prototyping, testing, and reflection.

We followed a path of Reflection-in-Action/Reflection-on-Action: as described by Schon [2008], RtD often involves designers reflecting on their actions during the process (*reflection in action*) and afterward (*reflection on action*). Each iteration of our work required us to reflect on challenges and feedback, taking time to reflect and growing our work based on problems and feedback. For example, a limitation in a third party API may require us to adapt our approach, or input from a tester may lead to a better way to present information. The process of reflecting on challenges and changes could also lead to serendipitous changes that allow us to create new knowledge or identify a better way to do something.

Our method for implementing this methodology is expanded on in Section 6.5, describing the development process for the Awareness Agent.

4.2.4 Study

As described above, our aim with the study was to gain insight into our concept for an Awareness Agent by constructing and evaluating a partial prototype, with the intent of better understanding particular elements of the design. The design decisions for the study are discussed in detail in Chapter 9 and we will not pre-empt them here, but we wish to highlight two aspects at this point.

Firstly, a significant part of our method relies on synthetic rather than real data. Much of the data associated with IO in the real world exists behind barriers, including technical and privacy related ones, with the content driving IO existing in private chats on mobile platforms without easy API access, or within corporate networks. This raises many issues which would impede research, such as difficulty of gaining consent from all parties involved in private discussions, handling confidential data and accessing closed systems. We discuss this in more detail in Section 7.1. Using only synthetic and public data makes the study process more feasible, and also presents a study opportunity in itself.

Secondly, while recognising the need to keep humans in the loop for the study, we also saw the opportunity to bring synthetic components into the process – allowing us to study

larger quantities of data using novel techniques. We discuss the rationale for this in detail in Section 8.1. This led to a modified role for the humans involved in the process; we sought to use a smaller number of more sophisticated study participants who could take an active role in supervising and assessing the study process, in addition to taking on the role of a persona.

4.3 Chapter Summary

In this chapter we gave an overview of our research methodology, discussing our overall approach [4.1] and documenting more detailed methodologies for the elements:

- Survey [4.2.1]
- Personas [4.2.2]
- Awareness Agent application [4.2.3]
- Study [4.2.4]

We will then go on to apply this methodology, starting with running a study and developing personas [5], then designing and implementing a prototype Awareness Agent [6], before working on techniques for synthetic content [7] and evaluation [8], finally applying these in our study [9].

Chapter 5

User Survey and Personas Development

In order to better understand how users may be affected by information overload and tailor solutions and testing plans appropriately, we developed a number of data-led personas, as discussed in the Methodology chapter [4.2.2]. To facilitate this, user opinions were solicited in a survey, which was advertised with an intention to reach people who may experience information overload. We then used the output of the survey to inform persona development.

5.1 Survey

5.1.1 Survey Design

The survey was grouped around five themes:

- Attitudes to interruptions from notifications**

The literature review [2.2.3] had identified a number of ways that notifications and their timing affect users, with varying attitudes being shown by users to different types of notification. We selected questions within this theme to gain further understanding into this and look at angles of particular relevance to our study. In partic-

ular we wanted to further explore attitudes to timing of notifications, prioritisation of content, whether notifications caused problems and how well these aspects were currently being served by online applications.

- **How well online services understand respondents' preferences and interests**

The concept of user control over their information environment is a core element of our concept for an awareness agent [1.2.2]. We included this theme in the survey to help us establish what degree of control is desired by different users, how much control they feel that they have at present, and their appetite for proactively taking steps to exert better control.

- **Degree of trust and confidence in online services**

The question of trust in online services and AI systems in particular [2.5.8] is also important to our work, and we grouped a number of questions designed to explore that in this theme. Questions took two broad forms: trust in competence of these services to perform well, and trust related to privacy and control of information. These questions would help us establish the levels of trust that different groups of users had in online services and how a personal agent under their own control might be useful.

- **General views on online services, connected applications and smartphones**

We sought to differentiate between types of user by asking a set of questions with a general theme of views on technology. The questions within this theme sought to establish respondents propensity to explore and tinker with technology, and their general approach to computers – utilitarian or more exploratory in nature.

- **Differentiation between work and personal use of apps and services**

Personal experience and informal discussion with people employed in multiple fields had led us to form a view that there was significant variation in how people differentiated between interruptions and use of devices in work and personal contexts. We included questions in this theme to substantiate this and distinguish users based on how they regarded overlap and potential conflict for work and personal-related communications.

Table 5.1: Survey questions 9-14

9. Please think about your attitude to being interrupted by notifications from your smartphone or computer originating from online services such as Slack, news apps, Facebook, Twitter, Instagram and messaging applications	
9.1	I receive so much information online that I often miss things that are important or time critical
9.2	I don't mind being interrupted when it's about something important
9.3	I often receive notifications about things that could have waited for later
9.4	Getting interrupted by notifications/alerts when I'm trying to get things done is a problem for me
9.5	My online services always get it right about what I want to be interrupted with
9.6	Online services always get it right when they judge what I'm interested in
10. We would like to ask you about how well online services such as Slack, Facebook, news apps, Twitter and Instagram understand your preferences.	
10.1	Online services always get it right when they judge what I'm interested in
10.2	Not all of the things that I follow (hashtags, people) are equally important to me
10.3	I want to be able to tell online services what matters to me most
10.4	I don't have enough control over what online services choose for me
10.5	I'm happy to have a computer make decisions about what content I should see
10.6	The idea of being able to rank or prioritise hashtags and other content appeals to me
10.7	I am happy to put in effort to 'train' the online services in order to see better results
11. We would like you to tell us about your trust in online services.	
11.1	I trust online services to make the best decisions about what to show me
11.2	I'm happy to share information about my interests and activities with online services if it will improve the service
11.3	I am uncomfortable about sharing personal information with online services because I don't know what they do with it
11.4	I prefer to keep information about my interests & activities under my control
11.5	I'm more willing to let a computer program have personal information if I know that I can control it
13. Please tell us about some of your general views on smartphone and connected applications.	
13.1	I always take time to customise the applications and devices that I use
13.2	I regularly update and ensure I have the latest version of applications I use
13.3	I'm always looking to try new applications and services
13.4	I consider myself very tech-savvy
13.5	Computers are just tools rather than interesting in themselves
13.6	I use all of the features on my phone and connected applications
14. Please can you tell us what you think about the relationship between personal and work use of computers and notifications on connected/online device.	
14.1	I keep my work and personal applications entirely separate
14.2	I find it easy to switch off from work
14.3	I'm happy to receive some work-related notifications during personal time
14.4	I prefer to keep certain applications (such as instant messaging on my phone) for personal things and not for work
14.5	I'm happy to see personal notifications while I'm at work
14.6	My employer is happy for me to receive personal notifications while I'm at work
14.7	I don't mind both work and personal information coming from the same device or application as long as it gets the timing and content right

The survey questions were split into different types:

- Questions 1-8 looked at users' use of applications and their general attitude to interruptions and notifications in various circumstances.

- Questions 9, 10, 11, 13 and 14 were designed to elicit opinions on the respondent's views on information overload, notification, interruptions, work/life split and so on. In each of these questions, respondents were asked to rate their agreement with a statement on a Likert scale of 1 (strongly disagree) to 5 (strongly agree). These are listed in Table 5.1. We will refer to these questions in this document as 'Q9_14'.
- Other questions were designed to elicit other types of information from the respondents, such as demographic data and free text inputs.

The full list of survey questions is referenced in Appendix A.

5.1.2 Survey Participants

We did not attempt to conduct a fully demographically balanced survey across all potential user types; this would have been beyond the scope and practical limitations of the research. Instead, we focussed recruitment efforts on participants who were likely to have relevant experience with the themes under investigation, using channels that were accessible and appropriate to the context.

Our primary goal was to obtain responses from a sufficiently diverse pool of participants to enable the derivation of distinct personas, while also providing quantitative input on IO-related assumptions. The intention was not to conduct a representative or generalisable study of user attitudes, but to gather meaningful insights from a sufficient cross-section of users to support design and evaluation tasks.

We acknowledge that this approach introduced inherent sampling bias, as the resulting survey population would be unlikely to capture a complete demographic spectrum of the potential users, which we also did not attempt to validate or control for. However, we consider this acceptable for the goals of this research, which focussed on persona development and exploratory validation rather than statistical generalisability. The effect of bias on eventual persona development is discussed in Section 5.2.5.

Channels used to advertise the survey included LinkedIn, Facebook, a university notice-board, internal channels in a large technology company, and direct outreach to personal

contacts. To encourage responses, optional entry into a prize draw of two gift vouchers was offered. Anonymous responses were permitted, although contact details were required if the respondent wished to enter the prize draw.

5.1.3 Survey Results

The survey was open for a total of 40 days in 2018, with a total of 135 responses received. The detailed survey results are referenced in Appendix A.2. The nature of the survey distribution resulted in the predominant groups being IT and Education professionals as shown in Chart S3.5 [doi:10.21954/ou.rd.28045442]. The respondents were highly educated, with over 75% reporting having an undergraduate degree or higher [S3.7]. Over 70% of the respondents were based in the UK [S3.8]. The gender balance of respondents was roughly equal [S3.3]. The age range of respondents was spread mainly across working-age, with 78% being in the range 25 to 54 years [S3.2].

5.1.3.1 Notification strategies by information type

Questions 2-4 of the survey had asked users about how they prefer to receive information that had different levels of urgency, with the users asked to rate preference level for each of multiple channels for communicating this information. Some notable results are shown in Table 5.2. We found for example that where information was something that might need to be acted on quickly (Question 2), Text/SMS was a preferred channel for 78% of respondents, while Email was also preferred but less strongly (59%). Interestingly 61% of respondents did *not* prefer to receive such information via computer desktop notifications.

In contrast, Email was strongly preferred (87%) for information that might be important but does not need immediate action (Question 3), with the surveyed channels being either actively not preferred (desktop notification, voicemail) or no strong preference.

For information that is interesting but not particularly important or urgent (Question 4), preference either way was less marked. Email was the most preferred at 66%, while Text/SMS, desktop notification and voicemail were all generally not preferred.

5.1.3.2 General views on information overload and related topics

Some notable results of questions 9-14 are shown in Table 5.3. These questions all have the same structure: grouped under an umbrella topic, the respondents indicated agreement on a Likert scale with a series of related statements. For the purposes of summarising the results, we have treated Likert responses 1 & 2 ('strongly disagree' & 'disagree') as 'disagree' and 4 & 5 ('strongly agree' & 'agree') as 'agree'.

5.1.3.2.1 Question 9: Attitudes to information overload and interruptions Respondents' attitude to being interrupted by online service notifications from smartphones or computers showed evidence of some problems being experienced. While 85% agreed that they didn't mind being interrupted about something important (Q9.2), 79% reported often receiving notifications about things that could have waited for later (Q9.3) and 59% agreed that being interrupted by notifications/alerts when I'm trying to get things done was a problem for them (Q9.4). On the other hand, respondents were split on whether they receive so much information online that they often miss things that are important or time critical, with 49% agreeing and 36% disagreeing¹. Respondents were more unified on the topic of online services always getting it right about what they want to be interrupted with (Q9.5) with only 9% agreeing.

5.1.3.2.2 Question 10: Online services and user preferences Under the topic of how well online services such as Slack, Facebook, news apps, Twitter and Instagram understand their preferences, 61% of respondents disagreed that 'online services always get it right when they judge what I'm interested in' (Q10.1), showing a level of dissatisfaction with these services (only 16% said they agreed to any degree with the statement). 69% of respondents did not feel they had enough control over what online services chose for them (Q10.4), and there was a desire/willingness to influence this: 66% wanted to be able to tell online services what matters to them most (Q10.3) and 56% were happy to put in effort to 'train' services in order to see better results (Q10.7). However, 64% also responded that they were not happy with a computer making decisions about what content

¹This is one of the areas where we later found some distinction between different clusters of users

they should see (Q10.5) – suggesting a general level of reticence towards any automated decision making.

5.1.3.2.3 Question 11: Trust in online services Respondents showed a low level of general trust in online services. Only 12% agreed to any degree that they trusted these services to make the best decisions about what to show them (Q11.1). 79% agreed that they were uncomfortable about sharing personal information with online services because they don't know what they do with it (Q11.3) and 83% prefer to keep information about their interests & activities under their control (Q11.4).

5.1.3.2.4 Question 13: General views on smartphone and connected applications The responses under topic 13 showed a possibly surprising willingness for proactive effort with technology. 73% agreed that they take time to customise the applications and devices that they use (Q13.1), and 73% also claimed to regularly update and ensure I have the latest version of applications that they used (Q13.2). 63% considered themselves to be tech savvy (Q13.4)

5.1.3.2.5 Question 14: Relationship between personal and work use of devices Question 14 asked respondents if they used devices or applications for both work and personal purposes. The majority (n=126) responded that they did, with a small minority not doing so (n=9). The positive respondents were asked to complete the questions on this topic (Q14.a.X). 60% agreed that they keep their work and personal applications entirely separate (Q14.a.1). Perhaps unsurprisingly, while 66% agreed that they were happy to see personal notifications while at work (Q14.a.5), only 31% agreed that they were happy to see work notifications in personal time (Q14.a.3). There was no consensus on ease of switching off from work (Q14.a.2), with 49% agreeing and 38% disagreeing that they found this easy. A plurality of 50% agreed that they did not mind both work and personal information coming from the same device or application as long as it got the timing and content right (Q14.a.7).

5.1.4 Survey Conclusions

The survey supported our view that users in the real world experience information overload and issues with interruptions (with the caveat that the pool of respondents to this survey are skewed towards information workers who may experience this problem more). Responses to questions on trust and control showed a lack of confidence that online service providers would act in the interests of the user and that there is a desire to have more control over decision-making about content and notifications. Again with the caveat of the respondents' demographic, the responses also indicated a willingness to invest time in solutions that work. We believe that the survey provides evidence to support the use of a personal agent that acts on behalf of users who might experience information overload.

Table 5.2: Survey Notable Results 1

Question	Channel	Response	N	%
2. How do you prefer to receive information when it is something you might need to act on quickly?				
2.1	Email	Not prefer	28	20.7%
		Prefer	80	59.3%
2.2	Text/SMS	Not prefer	20	14.8%
		Prefer	106	78.5%
2.4	Smartphone notification	Not prefer	35	25.9%
		Prefer	68	50.4%
2.6	Desktop notification	Not prefer	82	60.7%
		Prefer	24	17.8%
3. How do you prefer to receive information when it is something that might be important but does not need immediate action?				
3.1	Email	Not prefer	118	87.4%
		Prefer	46	34.1%
3.2	Text/SMS	Not prefer	42	31.1%
		Prefer	46	43.1%
3.6	Desktop notification	Not prefer	77	57.0%
3.8	Voicemail	Not prefer	81	60.0%
4. How do you prefer to receive information when it is interesting but not particularly important or urgent?				
4.1	Email	Prefer	89	65.9%
4.2	Text/SMS	Not prefer	82	60.7%
4.7	Phone call	Not prefer	100	74.1%
4.8	Voicemail	Not prefer	102	75.6%

Table 5.3: Survey Notable Results 2

Response	N	%
9.1 - I receive so much information online that I often miss things that are important or time critical		
Disagree	48	35.6%
Agree	66	48.9%
9.2 - I don't mind being interrupted when it's about something important		
Disagree	10	7.4%
Agree	115	85.2%
9.3 - I often receive notifications about things that could have waited for later		
Disagree	8	5.9%
Agree	107	79.3%
9.4 - Getting interrupted by notifications/alerts when I'm trying to get things done is a problem for me		
Disagree	33	24.4%
Agree	79	58.5%
9.5 - My online services always get it right about what I want to be interrupted with		
Disagree	82	60.7%
Agree	12	8.9%
10.1 - Online services always get it right when they judge what I'm interested in		
Disagree	82	60.7%
Agree	21	15.6%
10.3 - I want to be able to tell online services what matters to me most		
Disagree	26	19.3%
Agree	89	65.9%
10.4 - I don't have enough control over what online services choose for me		
Disagree	15	11.1%
Agree	93	68.9%
10.5 - I'm happy to have a computer make decisions about what content I should see		
Disagree	87	64.4%
Agree	20	14.8%
10.7 - I am happy to put in effort to 'train' the online services in order to see better results		
Disagree	27	20.0%
Agree	75	55.6%

Continued on next page

Table 5.3 – continued from previous page

Response	N	%
11.1 - I trust online services to make the best decisions about what to show me		
Disagree	95	70.4%
Agree	16	11.9%
11.3 - I am uncomfortable about sharing personal information with online services because I don't know what they do with it		
Disagree	19	14.1%
Agree	106	78.5%
11.4 - I prefer to keep information about my interests & activities under my control		
Disagree	9	6.7%
Agree	112	83.0%
13.1 - I always take time to customise the applications and devices that I use		
Disagree	24	17.8%
Agree	99	73.3%
13.2 - I regularly update and ensure I have the latest version of applications I use		
Disagree	27	20.0%
Agree	98	72.6%
13.4 - I consider myself very tech-savvy		
Disagree	26	19.3%
Agree	85	63.0%
14.a.1 - I keep my work and personal applications entirely separate		
Disagree	36	26.7%
Agree	81	60.0%
14.a.2 - I find it easy to switch off from work		
Disagree	51	37.8%
Agree	66	48.9%
14.a.3 - I'm happy to receive some work-related notifications during personal time		
Disagree	65	48.1%
Agree	42	31.1%
14.a.5 - I'm happy to see personal notifications while I'm at work		
Disagree	12	8.9%
Agree	89	65.9%

Continued on next page

Table 5.3 – continued from previous page

Response	N	%
14.a.7 - I don't mind both work and personal information coming from the same device or application as long as it gets the timing and content right		
Disagree	39	28.9%
Agree	67	49.6%

5.2 Personas Development

5.2.1 Methodology

5.2.1.1 Overview

We applied a cluster analysis process to map respondent groups from within the 135 responses to personas [Tu et al., 2010], using IBM SPSS software². Persona construction used a hybrid of quantitative and qualitative inputs. We used the quantitative output of the clustering process to evidence the personas, but also used qualitative analysis to supply some more subjective criteria to enable the creation of a balanced and representative set.

We followed the following steps:

- Group the survey respondents into a set of clusters having traits in common that were distinct from each other
- Extract traits from the clusters by examining average survey question response values for each cluster; an extracted trait would be a survey question that is on average highly or lowly scored for each cluster and which also distinguished it from other clusters
- Define a Persona for each cluster, assigning demographic information that is consistent with the demographics of the cluster respondents
- Complete a PATHY template for each persona, based on the related traits and demographics

²<https://www.ibm.com/spss>

5.2.1.2 Data-Driven PATHY Technique

We took an approach to persona development integrating elements from both the PATHY technique [B. M. Ferreira, Diniz Junqueira Barbosa, and Conte, 2016] [B. Ferreira et al., 2018] and the data-driven methodology outlined by McGinn and Kotamraju [2008]. PATHY is the more qualitative technique, having a focus on empathy [Gray, Brown, and Macanufo, 2010] and understanding the user's emotional journey; qualitative insights, such as those derived from stakeholder workshops or interviews, are applied to structured empathy mapping to create personas. In our case, these qualitative elements came directly or indirectly from our user survey, as either direct comments or general inferences from data. McGinn & Kotamraju took a more quantitative approach, applying a statistical analysis to survey data to inform persona development, following up with interviews as a refinement process.

We were keen to make use of a PATHY style output for personas to support our further work, while wishing also to give this a statistical basis – resulting in a hybrid of the two approaches with some variations of our own. While McGinn & Kotamraju used factor analysis, we instead took an approach of using a clustering technique; we felt that the size of the survey may not be sufficient for the former approach, and initial tests with clustering also yielded strong results. McGinn & Kotamraju also used follow-up interviews to refine the personas; we instead took the approach of applying a second step of statistical validation and using general survey data (demographics, free-text, etc.) to validate and inform the personas.

5.2.1.3 Clustering Approach

Our survey design had grouped questions into five overall themes relating to different aspects of our research [5.1.1]. We used these themes as a basis for our approach to creating clusters of respondents via a two-stage k-means clustering approach.

Stage 1 – clustering within themes During initial informal testing of different clustering techniques, we had found distinct clusters at theme-level (i.e. ‘Question 9.x: attitudes

to interruption'), where groups of correspondents could be identified that responded similarly across a theme's questions. We wanted to capture this thematic association between respondents and individual research-related topics and chose to do this by clustering responses within each theme as the first stage. This allowed us to identify groups of respondents whose views were consistent within a theme in a way that we could not do by clustering questions from across all themes at once.

Stage 2 – clustering across themes The second stage of clustering was then based on the output of Stage 1, by calculating clusters of respondents based on their similarity across themes. For example, there might be a relationship between respondents who fell into *Attitudes to Interruption* Cluster A, *Trust in Online Services* Cluster C, *Personal/work Use* Cluster B and so on. This second stage allowed us to identify groups of respondents that shared similar sets of views across the thematic level.

This two stage process allowed us to develop groups of respondents that had commonalities at the higher theme level while retaining distinction at the sub-question level.

Cluster profiling We took the output from Stage 2, and mapped these clusters back against the granular question responses to perform cluster profiling with the aim of extracting distinctive information to drive the development of individual personas. Each cluster was assessed for how members had responded to each of the individual questions/statements of the survey (for example, how did Cluster 1 members respond to the question: *I receive so much information online that I often miss things that are important or time critical?*). We filtered the statements by applying some criteria in order to select only those that would contribute to the persona in a way that would be differentiating and statistically significant, and applied the resulting statements to individual clusters in order to frame archetypes.

These attributes combined with demographic information that also emerged from the clusters fed the PATHY technique to derive individual personas. As well as data-driven development, some subjective input was also used to generate realistic personas and achieve a reasonable balance of types and demographics.

The two-stage k -means clustering approach that we adopted is detailed in Section 5.2.2, and the use of cluster profiling inform persona development is described in Section 5.2.3.

5.2.1.4 Limitations

This process necessarily involved some degree of subjectivity and the fabrication of some personal details for each persona to represent a blend of characteristics that we expect to embody typical use cases. However, while the name, biography and other such features of the persona have been created in this way, the key properties of each persona are connected to the survey evidence for its associated cluster.

For example, let's assume that Cluster N had survey results that had high average scores for questions indicating that they were overloaded and miss items, that they are happy to put in effort to train systems and that they like to keep work and personal information separate. Furthermore, assume that these characteristics are a distinguishing factor from other clusters, which do not share that combination. In this case, we assign these features from the survey to be characteristics or traits of the related persona. On the other hand if members for every cluster scored a low agreement for the question "online services always get it right about interruptions", then that would not be considered a defining trait for that cluster/persona and should not necessarily form part of the persona definition.

For the purposes of persona development in this study, we decided to limit the dataset to those who had answered the question 14 statements ($n=124$) – i.e. those who answered Yes to the question "Do you use connected/online devices or applications for both work and personal purposes?" and provided Likert ratings for all the corresponding statements.

5.2.2 Clustering

To implement our clustering design [5.2.1.3], we selected the question set Q9_14 as the basis for our clustering implementation, as they describe respondents' opinions, and can be used to differentiate different groups within the respondents. As seen in Table 5.1, each top level question represents a topic or theme, such as "We would like you to tell us

about your trust in online services”, under which several related statements are grouped for the user to respond to. We tested several clustering approaches, and eventually we adopted the following two stage meta-clustering process to reflect this structure³.

5.2.2.1 Stage 1 – Within Themes

Firstly, we used SPSS to generate a k -means cluster of 3 clusters ($k=3$), separately for each of the top-level questions in Q9_14. We initially ran the first-level clustering process using unmodified data, with the Likert values in the 1-5 range of the survey data. We then produced an additional *merged* data set, where Likert value responses were merged or mapped to a range of 1-3⁴ and repeated the clustering process with this data. We found significantly better cluster output with the merged data compared to the original, with all responses being placed in a cluster (with the exception of questions 13 & 14 where 11 respondents did not answer the questions). So for example, for question 9, we ran a job to generate 3 clusters relating to Q9_1, Q9_2, Q9_3, Q9_4, and Q9_5, outputting a new data item called ‘Q9M_3CL’ containing the cluster number for each response record.

We also tested different permutations of k in this step, to generate varying numbers of clusters. We applied Kruskal-Wallis tests to evaluate the differences across clusters in these cases and found that $k=3$ gave us a number of clusters that would be useful for persona development while still having significant differences between clusters.

5.2.2.2 Stage 2 – Across Themes

The second stage was to use each of these outputs⁵ as inputs for the second stage clustering process, also using k -means. We also tested a number of different cluster counts for this process, generating QQ_9_14_3CL ($k=3$), QQ_9_14_4CL ($k=4$) and so on, up to $k=7$. We again applied Kruskal-Wallis tests to evaluate the differences across second stage clusters, and additionally took into account the range of maximum persona counts that would be practical to study balanced with the need to create sufficient personas to reflect the

³We selected k -means as the clustering algorithm based on early testing with different clustering methods. We had also tested hierarchical cluster generation, but this did not yield well-defined clusters

⁴So survey data Likert values of 1 & 2 mapped to 1, 2 remained 2, and 3 & 4 mapped to 3

⁵Q9M_3CL, Q10M_3CL, Q11M_3CL, Q13M_3CL and Q14M_3CL

diversity of the data. We settled on k=5 for this stage, generating the cluster named “QQ_9_14_5CL”⁶. Table 5.4 shows the Kruskal-Wallis test results for this cluster.

Table 5.4: Kruskal-Wallis Test Results for QQ_9_14_5CL

Test Statistics^{a,b}	Q9M_3CL	Q10M_3CL	Q11M_3CL	Q13M_3CL	Q14M_3CL
Kruskal-Wallis H	59.180	68.968	107.470	7.053	48.997
df	4	4	4	4	4
Asymp. Sig.	.000	.000	.000	.133	.000

^a Kruskal-Wallis Test
^b Grouping Variable: Cluster of clusters of individual Q9-14 clusters (5CL)

5.2.3 Cluster-Persona Mapping

We performed a cluster profiling process based on the derived set of clustered respondents, seeking to associate individual clusters with positive or negative statements from the Q9_14 questions. For each question we calculated the mean Likert value of response for each cluster. We then assigned Positive, Neutral and Negative sentiments to the cluster based on this mean value. For example, if the mean Likert value for a statement was 1.5, we would assign a Negative sentiment, while a mean Likert of 4.7 would give a Positive sentiment. Likert values around 3 would be assigned Neutral sentiment. This was a manual process and hard boundaries were not defined, although it was found that most values fell to a reasonably unambiguous sentiment assignment. The results of this can be seen in Table 5.5

Not all statements were included in mappings. To favour only those where there was a statistically significant relationship between question/statement and cluster, we applied Pearson’s chi-square tests on the correlation between each question and the QQ_9_14_5CL cluster⁷. The results are listed in Table 5.6. By default we included only those statements where there was a significant relationship ($p \leq 0.05$) – however we did also make some specific subjective exceptions to this in order to make certain statements available for persona creation; these are also listed in Table 5.6.

⁶We chose ‘QQ’ as the prefix for Stage 2 cluster names to differentiate them from Stage 1 clusters

⁷See also:

		Positive	Neutral	Negative	1	2	3	4	5
Q9_1	Overload, miss things	1,2,4	3	5	TRUE TRUE TRUE	TRUE TRUE TRUE	TRUE TRUE TRUE	TRUE TRUE TRUE	TRUE TRUE TRUE
Q9_2	Don't mind interruptions when important	1,2,3,4,5							
Q9_3	Receive not about things that could have waited	1,2,3,4,5							
Q9_4	Interruptions are a problem	4,5	1,2,3						
Q9_5	Services get it right about interruptions			1,2,3,4,5					
Q10_1	Services get it right about my interests				2,1,3,4,5	TRUE TRUE TRUE	TRUE TRUE TRUE	TRUE TRUE TRUE	TRUE TRUE TRUE
Q10_2	Not all things equally important	1,2,3,4,5	1,5						
Q10_3	Want to be able to say what matters most	2,3,4							
Q10_4	Don't have enough control	1,3,4,5	2						
Q10_5	Happy for computer to make content decisions	1,2,3,4							
Q10_6	Ranking/prioritising important	2,4	1,3,5						
Q10_7	Happy to put in effort to train								
Q11_1	Trust online services to make best content decisions				2,1,3,4,5	TRUE TRUE TRUE	TRUE TRUE TRUE	TRUE TRUE TRUE	TRUE TRUE TRUE
Q11_2	Happy to share to improve service	2,4							
Q11_3	Not comfortable sharing	1,3,5	2,4						
Q11_4	Prefer to keep under my control	1,2,3,5	4						
Q11_5	Happy to share if I can control it	2,4	1,3,5						
Q13_1	Always take time to customise	1,2,3,4							
Q13_2	Always update	1,2,3,4	5						
Q13_3	Always looking to try new apps	1,2,3,4,5							
Q13_4	Tech savvy	1,2,3,4							
Q13_5	Computers just tools	1,2,3,4,5							
Q13_6	Use all features	1,2,4,5	3						
Q14_1	Like to keep work and personal separate	1,5	2,3,4		TRUE TRUE	TRUE TRUE	TRUE TRUE	TRUE TRUE	TRUE TRUE
Q14_2	Easy to switch off from work		1,2,4	3					
Q14_3	Happy to receive work notifications in pers time	1,2,3,4,5	2,3,4	1,5					
Q14_4	Prefer to keep certain apps for personal	1,2,3,4,5							
Q14_5	Happy to see personal notifications while at work	1,2,3,4,5							
Q14_6	My employer is happy for personal notifications while at work	1,2,3,4,5	2,3	1,4,5					
Q14_7	Pers & work same device								

Table 5.5: Sentiment assignment by statement/cluster based on mean response value

Table 5.6: Pearson's chi-square correlation between each question and the QQ_9_14_5CL cluster and inclusion in persona creation

Question	P Value	Significant?	Included?	Exception?
Q9_1	0.000	Y	Y	
Q9_2	0.523			
Q9_3	0.647			
Q9_4	0.035	Y	Y	
Q9_5	0.586			
Q10_1	0.000	Y	Y	
Q10_2	0.500			
Q10_3	0.060		Y	Y
Q10_4	0.306			
Q10_5	0.000	Y	Y	
Q10_6	0.020	Y	Y	
Q10_7	0.000	Y	Y	
Q11_1	0.000	Y	Y	
Q11_2	0.000	Y		
Q11_3	0.000	Y	Y	
Q11_4	0.000	Y	Y	
Q11_5	0.000	Y	Y	
Q13_1	0.541		Y	Y
Q13_2	0.074			
Q13_3	0.337		Y	Y
Q13_4	0.725			
Q13_5	0.342			
Q13_6	0.129			
Q14_1	0.083		Y	Y
Q14_2	0.147		Y	Y
Q14_3	0.001	Y	Y	
Q14_4	0.556			
Q14_5	0.215			
Q14_6	0.068	Y	N	
Q14_7	0.001	Y	Y	

We also applied other criteria on for judging the prominence and inclusion of statements in mappings:

- Statements where only the one cluster matched were marked in bold to indicate that they are distinguishing features for the cluster
- Statements where two or three clusters in total matched were marked in normal text to signify that they are relevant but not unique
- Statements where more than three clusters have a match were discarded as not being sufficiently distinctive

Looking by default only at Positive and Negative mappings, this process gave us a True/False value for each combination of cluster and statement. We used this to build a list of statements applicable to each cluster. For cases where a cluster/statement mapping was True,

we applied the statement to the cluster. Where the mapping was False, we applied the inverse of the statement to the cluster.

In cases where there were only a single pair of mappings (i.e. Positive & Neutral, or Negative & Neutral), we took the Neutral result to be the opposite of the other mapping. For example in Q10_1 (“Online services always get it right when they judge what I’m interested in”), four clusters had a Negative sentiment, while one cluster had Neutral. In this case we treated the single Neutral as a Positive sentiment for the purposes of building a statement list for the clusters because it was a differentiating factor from the other clusters (in other words it was positive *relative* to the other clusters if not positive in absolute terms).

In this way we were able to create a list of weighted statements for each cluster that could then be used to turn the cluster into an evidenced persona. This is shown in Figure 5.1.

5.2.4 Persona Derivation

Although we had a set of statements that could be applied to each cluster, the process of converting the clusters to personas was necessarily partly subjective. We took our own design of PATHY template [Figure 5.2] for each cluster/persona and populated the relevant fields with statements applicable to that persona, and to use demographic information that was consistent with what we found for the cluster but still allowing us some latitude to create convincing personas. We loaded the survey results and cluster data into an interactive reporting application⁸ and used this to explore the demographic properties and other survey data by cluster. Some of this output is included in Appendix B.1.

We treated demographic and other information as only a *hint* rather than a strict determinant of the persona content, on the basis that: 1) an individual is not completely defined by their demographics, and the same persona type can exist in multiple demographic variations; 2) we only had a limited range of demographics in the survey respondents, and forcing a strict relation would limit our choices. Instead we created personas that we believed were generally but loosely consistent with the cluster demographics.

⁸<https://www.ibm.com/products/cognos-analytics>

Cluster	Mapped statements
	<p><i>Bold: Statement is unique to this cluster</i></p> <p><i>Normal: This cluster is one of only 1 or 2 matching this statement</i></p> <p><i>Grey: This cluster is one of 3 matching this statement</i></p>
1	<p>Keep work and personal separate</p> <p>Easy to switch off from work</p> <p>Overload, miss things</p> <p>Not comfortable sharing</p> <p>Not happy to share to improve service</p> <p>Not happy to receive work notifications in personal time</p>
2	<p>Happy to share personal & work notifications on same device</p> <p>Overload, miss things</p> <p>Happy to receive work notifications in personal time</p> <p>Experiences information overload</p> <p>Services get it right about my interests</p> <p>Want to be able to differentiate</p> <p>Happy for computer to make decisions</p> <p>Happy to put in effort to train</p> <p>Trust online services to make content decisions</p> <p>Happy to share to improve service</p> <p>Happy to share if I can control it</p>
3	<p>Not comfortable sharing</p> <p>Happy to share personal & work notifications on same device</p> <p>Always takes time to customise apps and devices</p> <p>Always looking to try new apps & services</p> <p>Don't use all features</p> <p>Not happy to share to improve service</p> <p>Not easy to switch off from work</p>
4	<p>Trust online services to make content decisions</p> <p>Experiences information overload</p> <p>Interruptions are a problem</p> <p>Want to be able to differentiate</p> <p>Ranking/prioritisation are important</p> <p>Happy to put in effort to train</p> <p>Happy to share to improve service</p> <p>Happy to share if I can control it</p>
5	<p>Keep work and personal separate</p> <p>Interruptions are a problem</p> <p>Not comfortable sharing</p> <p>Not happy to share to improve service</p> <p>Not happy to receive work notifications in personal time</p> <p>Does not experience information overload</p>

Figure 5.1: Derived cluster-statement mapping

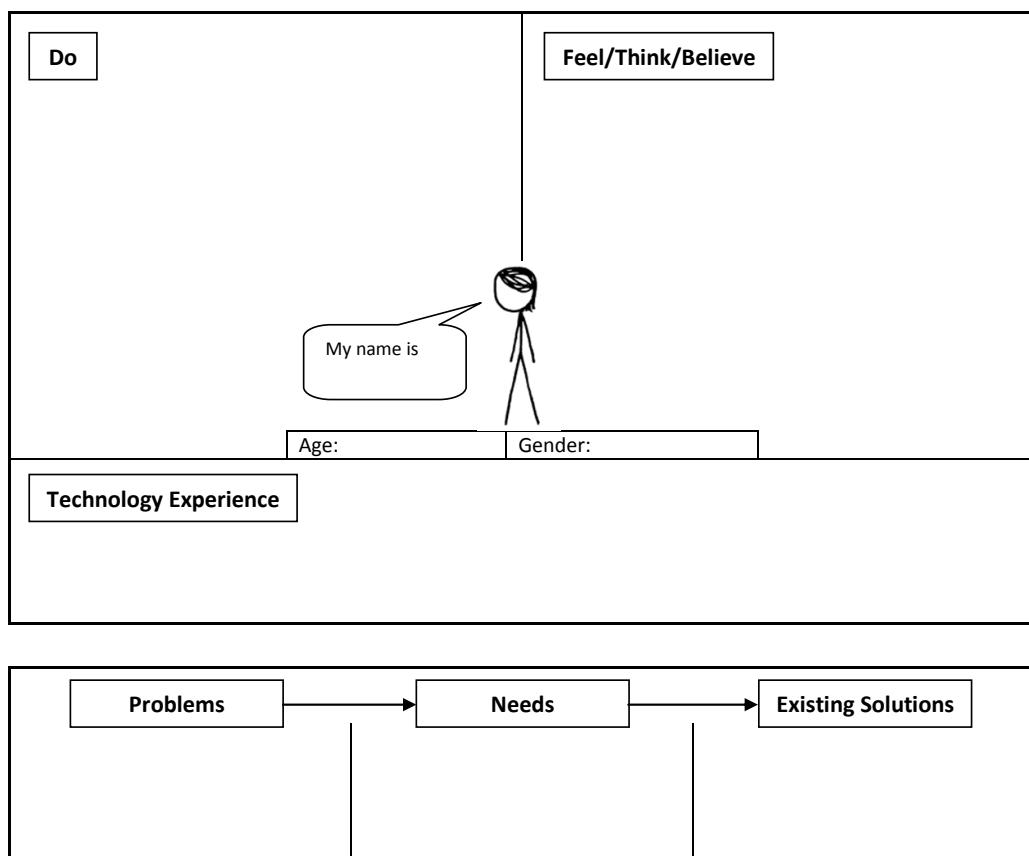


Figure 5.2: Blank PATHY template
Credit to [xkcd](#) for the stick figures

To add a theme and some consistency to our fictional personas, we borrowed names and some character information from the long-running BBC Radio series *The Archers*⁹. However, this was done purely to add colour and provide some inspiration – the completed personas bear little resemblance to their *Archers* namesakes, as any aficionado would quickly identify.

5.2.5 Effect of Survey Bias on Persona Development

As noted in Sections 5.1.2 and 5.1.3, the selected approach to survey participant recruitment led to a demographically unbalanced set of responses, with a preponderance of IT and Education professionals. This led to a set of final personas that is not fully representative of the overall population. However, while we acknowledge this as a limitation, we do not consider it to significantly undermine the value of the personas for the following reasons:

- A fully representative set of personas would likely require a substantially larger and more demographically diverse sample, resulting in more personas than could be feasibly studied within the scope of this research.
- Our intent was to develop personas relevant to individuals likely to experience IO, which implicitly targets more intensive users of connected applications. This subgroup is not itself representative of the general population, and that limitation is both acknowledged and intentional.
- The survey data and clustering process yielded a sufficiently diverse set of five personas, which was consistent with the expected number of final study participants.
- The final personas are evidenced by data, internally coherent, and appropriate for the study's design and evaluation objectives.

While a broader survey incorporating demographic controls might yield a wider array of personas, we argue that the sample used here was appropriate for our specific research requirements.

⁹https://en.wikipedia.org/wiki/The_Archers

5.3 Personas

Table 5.7: Mapping of QQ_9_14_5CL Cluster Number and Persona Name

Cluster Number	Persona Name
1	Susan
2	Adam
3	Phoebe
4	Kenton
5	Usha

We created personas for identities with names Susan, Adam, Phoebe, Kenton and Usha. The relationship between QQ_9_14_5CL cluster and persona name is shown in Table 5.7. The Susan persona is included here as an example, Figure 5.3. The full set of personas is included in Appendix B.2 and can also be found in Supplement S4 [doi:10.21954/ou.rd.28045454].

5.3.1 Persona 1 – Susan

The Susan persona [B.6] had data-led traits of:

- Keep work and personal separate
- Easy to switch off from work
- Overload, miss things
- Not comfortable sharing

The demographics for the cluster showed slightly lower higher level of education on average, with a fairly wide spread of age range; they were slightly more skewed in favour of administrative/clerical work and civil servant.

Susan

Age: 58 Gender: Female

<p>Do</p> <p>I live in the Midlands in the UK and work as an administrator at the nearby university.</p> <p>I'm married and have two children – unlike me they both went to university. Since graduating one settled down quite close but the other lives and works in London.</p> <p>I have a few hobbies outside of work – baking, tennis and getting out into the local countryside.</p> <p>My name is Susan</p>	<p>Feel/Think/Believe</p> <p>While I enjoy my job, it's not the most important thing in my life; I have no trouble switching off at the end of the day, even when it's been very busy. When I socialise with work colleagues we rarely talk about work (otherwise I probably wouldn't socialise with them).</p> <p>I know my children lead busy lives now, but I miss seeing as much of them as I used to, particularly the youngest who is in London now. They do try and keep in touch but I don't always know what they're up to or how they are doing.</p> <p>I really enjoy my tennis and spend a lot of time helping organise club events.</p> <p>I'm not always entirely on top of things in my personal life because I don't check my email often enough.</p>						
<p>Technology Experience</p> <p>I use a desktop computer at work for admin, email and maybe a little web browsing. We have one at home too, but it's mostly my husband on that.</p> <p>I've had IT training at work and get along fine with computers – although I prefer to stay in my comfort zone.</p> <p>My son made me get one of those smartphones. I didn't really see the point at first, but it is actually quite useful for staying in touch and organising things. I think I mostly use Facebook and WhatsApp as well as things like the weather app.</p> <p>I admit I do use Facebook quite a lot, but there are a lot of stories about how much they know about you and what they do with that. If it wasn't so handy, I'd use it a lot less.</p>							
<p>Problems → Needs → Existing Solutions</p> <table border="1"> <thead> <tr> <th>Problems</th> <th>Needs</th> <th>Existing Solutions</th> </tr> </thead> <tbody> <tr> <td>Not as aware of her adult childrens' activities and day to day lives as she would like to be. Because of relatively low level of engagement with computers at home, Susan sometimes misses items of news or things to act on – particularly when they come in via email or get lost in Facebook feeds.</td> <td>Tools to help her track what is going on with friends, family and hobbies in social media without needing to log in all the time. A way to ensure that she does not miss important emails.</td> <td>Existing algorithms in social services that select content for users. Email filtering. Notification functionality in social media smartphone apps.</td> </tr> </tbody> </table>		Problems	Needs	Existing Solutions	Not as aware of her adult childrens' activities and day to day lives as she would like to be. Because of relatively low level of engagement with computers at home, Susan sometimes misses items of news or things to act on – particularly when they come in via email or get lost in Facebook feeds.	Tools to help her track what is going on with friends, family and hobbies in social media without needing to log in all the time. A way to ensure that she does not miss important emails.	Existing algorithms in social services that select content for users. Email filtering. Notification functionality in social media smartphone apps.
Problems	Needs	Existing Solutions					
Not as aware of her adult childrens' activities and day to day lives as she would like to be. Because of relatively low level of engagement with computers at home, Susan sometimes misses items of news or things to act on – particularly when they come in via email or get lost in Facebook feeds.	Tools to help her track what is going on with friends, family and hobbies in social media without needing to log in all the time. A way to ensure that she does not miss important emails.	Existing algorithms in social services that select content for users. Email filtering. Notification functionality in social media smartphone apps.					

Figure 5.3: Final PATHY document for persona "Susan"

5.3.2 Persona 2 – Adam

The Adam persona [B.7] had data-led traits of:

- Happy to share personal & work notifications on same device
- Overloaded, misses things
- Happy to receive work notifications in personal time
- Experiences information overload
- Services get it right about my interests
- Want to be able to differentiate
- Happy for computer to make decisions
- Happy to put in effort to train
- Trust online services to make content decisions
- Happy to share to improve service

The demographics for the cluster showed a higher propensity to be an IT professional and lower for civil servant. The average education level was slightly higher than average.

5.3.3 Persona 3 – Phoebe

The Phoebe persona [B.8] had data-led traits of:

- Not comfortable sharing
- Happy to share personal & work notifications on same device
- Always takes time to customise apps and devices
- Always looking to try new apps & services
- Don't use all features
- Not happy to share to improve service
- Not easy to switch off from work

Demographically, members of this cluster were younger than average, more likely to be a knowledge worker and less likely to be a civil servant.

5.3.4 Persona 4 – Kenton

The Kenton persona [B.9] had data-led traits of:

- Trust online services to make content decisions
- Experiences information overload
- Interruptions are a problem
- Want to be able to differentiate
- Ranking/prioritisation are important
- Happy to put in effort to train
- Happy to share to improve service
- Happy to share if I can control it

Members of this cluster had some quite distinctive demographics, falling mostly in the 49-54 age range, and with school/further education being the most common highest education level.

5.3.5 Persona 5 – Usha

The Usha persona [B.10] had data-led traits of:

- Keep work and personal separate
- Interruptions are a problem
- Not comfortable sharing
- Not happy to share to improve service
- Not happy to receive work notifications in personal time
- Does not experience information overload

The demographics for this cluster showed a higher average age, more propensity to have higher education and greater likelihood of being a knowledge worker.

5.3.6 Persona Scenarios

We took these personas and used them to develop a number of Persona Scenarios, which are listed in Supplement S5 [[doi:10.21954/ou.rd.28045460](https://doi.org/10.21954/ou.rd.28045460)]. These scenarios are fictionalised expansions on the defined persona texts, introducing scenarios for information usage and issues that are consistent with the persona. We have then used these to better understand how an Awareness Agent might address the differing needs of a diverse user base.

5.4 Chapter Summary

This chapter is formed of two major sections: documenting our survey into information overload, attitudes to interruption and trust [5.1], and developing a set of personas based on the results of this survey [5.2].

We also included details of the personas themselves [5.3] and a set of notional scenarios that could apply to these personas [5.3.6].

These personas form a foundation for the work described in the remainder of the thesis: we will go on to document the design and implementing a prototype Awareness Agent [6], before working on techniques for synthetic content [7] and evaluation [8] and applying these in our final study [9].

Chapter 6

Awareness Agent Design and Implementation

6.1 An Awareness Agent

We conceived the Awareness Agent as a possible solution for the problem of information overload. Our concept for an awareness agent is for an autonomous, intelligent software entity capable of assisting users in managing high volumes of incoming data from disparate sources. Its primary goal is to enhance the user's awareness and consumption of this information without overwhelming them with unnecessary data. It should do this by ensuring that relevant and actionable items are surfaced to the user at appropriate times, while extraneous content is filtered out.

The solution needs to balance the flow of information in such a way that the user is always aware of the right level of important or interesting content, while being protected from overload. The agent addresses this by acting as a mediator, leveraging cognitive computing techniques to evaluate, prioritise, and deliver content that aligns with the user's preferences, goals and context. Drawing from the concept of the Social Machine [Berners-Lee and Fischetti, 1999], the Awareness Agent can be seen as part of a broader ecosystem where humans and machines collaborate.

This chapter will explore the design of the notional awareness agent, examining the re-

quirements, establishing design principles, developing a system model for its core concepts, and leading to an architecture. We will then go on to discuss how we have formed this into a partial prototype that we will later go on to evaluate.

6.2 Personas Perspective

In this section we look at the design of an awareness agent in light of several Persona Scenarios that we developed for the personas¹. We examine several scenarios and consider how the design of an agent might best facilitate success for the user in those cases. The personas and scenarios should not be considered a firm design directive, but rather something to provide general design scope and direction.

Susan [B.6 & S5.1]

“As Susan, I want to be able to focus on family updates so that I can stay up to date on what the children are up to without getting overwhelmed by other content on social media” [Scenario 1]

- It may help Susan to see everything from the family in one place, so smart filtering could help by including only social content from her children in the *Family - Children* channel of the agent

“As Susan, I don’t want non-work distractions while trying to get on with work tasks” [Scenario 2]

- Not all interruptions are bad – some are important, but context is key, and Susan wants to demarcate work from personal content
- A context-aware solution that suppresses interruptions of certain types when they are not wanted may be useful

¹Contained in Supplement S5 [doi:10.21954/ou.rd.28045460]

Adam [B.7 & S5.2]

"As Adam, I need to keep track of multiple different projects at the same time – I don't want to miss anything important" [Scenario 1]

- We could help Adam by identifying project-related messages that could be important or urgent and put these in one place

"As Adam, I'd like to be able to take a moment in the morning to catch up on everything relevant to me in my personal and work life before I get on with my day" [Scenario 3]

- Adam doesn't necessarily want to see content split on work/personal lines, but would rather discriminate between important to catch up on and not important
- A solution that analyses content across multiple sources and prioritises content would help with this

Phoebe [B.8 & S5.3]

"As Phoebe, I want my social media content feed personalised to my own preferences" [Scenario 3]

- If Phoebe is prepared to invest time in providing training data, she could see benefits from content better personalised to her needs

Kenton [B.9 & S5.4]

"As Kenton, I sometimes want to be able to turn off all interruptions unless they are truly urgent" [Scenario 2]

- A combination of smart filtering and context aware notifications settings would help Kenton by identifying only the more urgent items and surfacing only those while he is concentrating on something

- Kenton needs to be able to tell the app when he doesn't want extraneous interruptions

"As Kenton, I love the social chat of my golf club, but sometimes I just want to see the stuff about actually meeting up to play" [Scenario 3]

- Kenton's golf messages all go into one pot at the moment, so the messages about golf logistics are mixed in with all the general chat
- It would help Kenton if the content from that single source that relates to actually playing golf could end up in a different place in his app so that he can find it more easily

Usha [B.10 & S5.5]

"As Usha, I only want to see high priority work content during office hours" [Scenario 2]

- Usha would likely benefit from an app that is aware of her office hours and adjusts its behaviour accordingly
- The app would need to be able to distinguish likely high priority work content from everything else.

6.3 Requirements Analysis

We can say that the driving requirement is for users to be aware of the work and output of others in their sphere without being *too* aware. We break these down into a more detailed list of requirements for an awareness agent, based on inferences drawn from the literature. Table 6.1 shows general areas of requirement and their relevant references in the literature review; Table 6.2 shows a list of agent requirements mapped to these areas.

Table 6.1: Requirement Areas

Area	Area Description	Reference
Autonomy	Independent operation of the agent	2.4.1
Collaboration & Trust	Collaboration between user & agent; trust in competence & benign nature of agent	2.5.8
Social Machine	The Social Machine & Semantic Web	2.5.1
Awareness	Awareness	2.2.1
Notification Strategies	Notification strategies within the greater context of awareness	2.2.3
Linked Data	Linked Data for information storage and exchange	2.3.3

These requirements collectively establish a framework for an agent that not only manages and filters information at scale, but also ensures that its behaviour aligns closely with the user's preferences and goals. By balancing independent decision-making with user oversight and control, the agent is designed to act as an intelligent intermediary, handling information flow without overwhelming the user. The agent's ability to communicate effectively, adjust to multiple resource types, and evolve through both explicit and implicit feedback ensures that it remains adaptable and user-centric. Ultimately, this requirements analysis provides a robust foundation for the design and implementation of an Awareness Agent that promotes efficiency, minimises cognitive load, and maintains trust in its interactions.

Table 6.2: Agent Requirements

	Requirement	Area
1	Should act on behalf of the owner and access information resources that are of interest to them	Autonomy Collaboration & Trust
2	Must not discard items of content that are important for the user to see	Collaboration & Trust
3	Must not delay items of content that are important for the user to see promptly.	Collaboration & Trust
4	Must make a decision for each resource, on behalf of its owner, about how to handle that resource (for example: act now, defer or ignore).	Collaboration & Trust Autonomy
5	Must be able to query target resources using appropriate methods such as RSS or a proprietary API.	Awareness
6	Must be able to receive and process incoming information, for example in the form of Linked Data Notifications.	Linked Data
7	Must be capable of managing its own scheduling according to various patterns (such as polling a resource on pre-set or flexible intervals, or communicating with its owner at defined intervals).	Autonomy
8	Multiple target resource types should be supported, while presenting a consistent user interface.	Awareness Social Machine
9	Multiple channels of communication with the owner must be supported (i.e., mobile notifications, emails).	Notification Strategies Awareness
10	Information presentation methods should be chosen based on the agent's understanding of its owner's preferences.	Autonomy Collaboration & Trust
11	Mechanisms must exist to allow the agent to be trained by the user.	Collaboration & Trust
12	Where the agent self-trains based on implicit feedback from the user, it should provide the user with the ability to view and correct this training.	Collaboration & Trust Autonomy
13	Must present an <i>outward</i> representation on behalf of its owner and be able to send its own notifications to other actors in the system about its owner's activities.	Social Machine Linked Data
14	Should be <i>conservative</i> in its outgoing actions; must not cause its owner difficulty due to antisocial message volume or inappropriate content.	Collaboration & Trust

6.4 Awareness Agent Design Concept

This section covers the design principles behind the awareness agent.

6.4.1 Owner Autonomy & Control

The primary principle underlying our concept of an awareness agent is that of owner control, where the agent is under control and acts on behalf of its owner, the user. As established in Section 2.5.8, human-AI collaboration must be built on trust, transparency, and accountability [Van Noorden and Perkel, 2023]. The awareness agent must provide mechanisms for its owner to not only interact with but also control the underlying machine learning models and algorithms.

6.4.1.1 Trust – Confidence Through Transparency and Accountability

The Awareness Agent can only be effective in its task of managing its owner's awareness and information flow if the user has confidence in its ability to perform this task. If the user is not confident in the ability of the agent to meet two critical requirements – not to discard important content or delay time-sensitive content – it is likely that the agent will be unable to relieve the user of IO as the user may feel compelled to double-check the agent's decisions.

To achieve this, the agent requires transparency in its decision-making process so that the user may understand decisions made by it and modify criteria. This design choice aligns with the growing body of literature advocating for user control over AI systems [2.5.8].

Accountability is also necessary: the agent must have built-in mechanisms for ensuring that its actions can be traced and evaluated. If an agent makes an inappropriate decision (*i.e.* missing a critical notification), there should be systems in place to identify how and why the error occurred and how this can be corrected.

6.4.1.2 Tailoring

Further, user control requires that the Awareness Agent can be tailored to individual preferences, allowing adjustments to reflect the user's evolving goals and contexts. Feedback mechanisms should allow the agent to learn and refine its behaviour based on explicit user input and implicit behavioural patterns. This approach is consistent with increasing user awareness of AI algorithms and their role [Oeldorf-Hirsch and Neubaum, 2023], and provides for a system which explicitly empowers the user.

6.4.2 Data Abstraction & Standardisation

A central function of the Awareness Agent is to take content from multiple sources, process it and present it to the user. Because the information landscape is highly heterogeneous, we face challenges in how a single agent can process resources from multiple such sources in a consistent manner. We face two distinct issues – practical and conceptual.

The *practical* issue is simply one of data structures and formats. There is no one standard for how data is stored in the various different email, messaging and other systems that contain information that we want to process. So that the agent can process these efficiently, we need to represent them using a common format internally to the agent.

The *conceptual* issue relates to the meaning of the data, and in particular the metadata. For example, an email message will have a set of headers including `From`, `To` and `Subject`. A corporate messenger app on the other hand may have a different set of metadata, such as `Sender`, and `Channel`. In some cases these mean different things, while in others they are conceptually the same or similar – in the above example `From` and `To` from email could be considered analogous to `Sender`, and `Channel` from the messaging application.

Conceptually similar items such as these should be represented within the agent in such a way that it can handle them equivalently. One area where this is important is how the agent represents items to the user where the items have heterogeneous sources – in order to be able to do so consistently, the agent must be able to identify logically similar components that it can then display in a consistent manner.

Similarly, the agent may perform some processing operations that are related to specific logical properties, such as who sent a message or when it was sent. To do this, the agent must be able to understand each item in a consistent way.

Therefore it is an essential design principle that content items processed by the agent are converted to a common form having standardised but extensible metadata.

6.4.3 Modularity & Commoditisation

For an Awareness Agent to be practical in the real world, it must have innate *flexibility*. This means it must be able to adapt to a diverse and constantly changing information landscape while also coping with deployment constraints, such as data access policies or restrictions. We define two key design principles to this end: the agent must be modular, and it must be able to leverage commodity services.

6.4.3.1 Modularity

Modular design is essential for the Awareness Agent in three primary ways:

Acquisition A diverse and constantly changing information landscape implies diverse and constantly changing data sources from which to acquire content, each with their own characteristics such as different APIs, access requirements, and data formats. A modular design allows for the integration of individually tailored components to handle these sources within a cohesive framework. This ensures that the system can easily adapt as new data sources emerge or existing sources evolve.

Deployment In some cases, it may not be feasible to deploy all components of an Awareness Agent to the same physical or logical system. For example, certain data acquisition modules may only function on a private network due to security or policy constraints. Modularity enables the separate deployment and independent updating of these components, ensuring the agent can function effectively across varied environments.

Interaction As with acquisition, the available or desired data egress and interaction implementations may change with time and context. A particular owner or use case may prefer one type of user interaction mechanism, while another user (or the same user in a different context) may need something else. Modularity supports this flexibility.

6.4.3.2 Commoditisation

The diversity and change principle also applies to the tools available to the agent to perform its task. For example, commercially available cognitive computing resources are a dynamic and extensive landscape. The agent must be designed to take advantage of whichever resource is most appropriate for a given task, whether it be a public cloud service or a specialised bespoke solution.

By treating these external tools and services as interchangeable commodities, the Awareness Agent reduces implementation complexity and gains the flexibility to evolve over time. The agent can also seamlessly incorporate new or improved cognitive services over time. This approach not only reduces workload during development but also allows the system to scale and adapt with minimal disruption. Modularity is a prerequisite for such commoditisation, as it enables the seamless integration and replacement of external services within the agent's architecture.

6.4.4 User Interaction

The Awareness Agent's paradigm for user interaction is guided by the literature discussed in Section 2.2.4, but also builds on the aforementioned principles of control, modularity and commoditisation.

User engagement with systems featuring notifications is maximised when these systems are intuitive, transparent, responsive to user needs, and contextually appropriate [Iqbal and Horvitz, 2010] [Pejovic and Musolesi, 2014]. The awareness agent, therefore, must have a focus on minimising friction in user interactions, so that users can easily configure and receive feedback from the system.

A two-fold interaction model is needed: proactive and reactive. In the proactive mode, relevant information should be pushed to the user based on predefined rules and learned preferences; in the reactive mode, the user can query the agent for specific information or ask for pre-built content reports.

6.4.4.1 Proactive Interaction

Users must be made aware of important information while not being overwhelmed by other information. Important, time-sensitive, information should be delivered or announced via a notification on a device that the user is using or carrying – but these must be the only items announced triggering such interruptions. This type of interaction should be context-sensitive, with the determination of which items pass the threshold for intrusive interaction being determined by the implicit and explicit context set by the user. For example, work-related notifications should have a much higher threshold for action when the user is in personal time than when they are at work.

6.4.4.2 Reactive Interaction

Reactive or on-demand interaction should represent the bulk of the user's interaction with the agent. It has less need to be context sensitive, but rather is task driven. We consider that reactive interaction occurs for three purposes: content consumption, direction/training, control/administration.

Content Consumption In this case, users interact with the agent to consume information, typically viewing content items that have been gathered and filtered/organised by the agent. They may view singular or aggregated items, depending on the user's context and intent. Information should be organised according to the purpose of the interaction and the nature of the content. For example the user may wish to view a catch-up report containing all items that were considered important but not requiring a time-sensitive notification, or alternative they may wish to view all new items relating to a particular topic.

This information should be hierarchical and linked – that is, available optionally in aggregate form with the ability to progress to view individual detail items or where appropriate to link to other related items.

Direction/Training Direction and training should be related closely to content consumption; we could even consider it to be a sub-topic, although it is also a standalone activity. The aim of this mode of interaction should be to allow the user to influence the content operation of the agent, for example by telling the agent that they are or are not interested in a given item. We should recognise that there are different modes to this type of interaction: in some cases a user may want to invest time into training the agent, in which case they should have a dedicated interface available for this. In other cases training may be given in the process of consumption – for example when the user is presented with something that has been handled incorrectly by the agent, they should have the ability to correct this and give future guidance *in situ*. In this latter case, the training and consumption UIs should be integrated.

Control/Administration This type of interaction enables the user to control or administer the agent. Tasks could include altering settings, setting up or modifying data sources, or changing operating mode. The design goal is that these operations should be as intuitive and friction-free as possible – with the most frequently carried out tasks presenting the fewest barriers to the user. For example, switching operating modes (a *control* task) is something that the user might do frequently, so should be quick and easy to do, whereas editing the functionality of a given operating mode (an *administration* task) is a less frequent task that can tolerate a less immediate UI.

6.4.5 Modal Operation

The functionality of the agent should be modal, such that it functions differently in different modes or contexts. There are multiple types of mode, including but not limited to:

- Work context – whether the user ‘at work’ or in personal time (being ‘at’ work may

not mean being in an actual workplace)

- *Time of day* – the time of day for the user
- *Date/day* – what day it is, whether it is a weekend etc.
- *Location* – where the user is physically located at the time
- *Current task* – what the user is currently doing
- *Mood* – current user mood or interests

Different operating modes should determine agent behaviours such as content filtering, notification decisions and methods of interaction. Ideally, the agent should be able to implicitly choose its current operating mode without user input, but the user must always have the ability to manually override.

6.4.6 User-Directed Machine Learning Models

Machine learning models are widely used in the processing and categorisation of information. However, end-users generally have limited control over the behaviour and use of these models. Acknowledging this, we have devised an overall design concept of multiple user-directed ML models. To ensure flexibility, transparency, and user autonomy in content processing and classification, the following design principles guide the development of user-directed machine learning models within the awareness agent:

User Control Over Model Lifecycle Users must have control over the lifecycle of their ML models, with the ability to deploy, manage, and retire models as their needs evolve. This allows the system to remain dynamic and adaptable to the user's changing priorities.

Customisable and Trainable by Users The ML models should be designed to be user-trainable, allowing users to customise them based on their personal content preferences. By providing examples and feedback, users can direct how the model learns and refines its classification over time.

Modular Design for Independent and Sequential Use Each ML model should operate independently, yet be capable of working in conjunction with others. Users should have the option to combine models in sequence, allowing for layered classification processes capable of more sophisticated processing than any one model.

Minimal Technical Expertise Required The system should abstract technical complexity, ensuring that users without advanced machine learning knowledge can still effectively train and manage models. The interface should be intuitive, offering guided model creation and feedback mechanisms to improve performance.

Transparency in Model Behaviour The system should provide transparency into how each model is making decisions. This includes clear explanations of classification criteria and performance, enabling users to trust the system and make informed adjustments as needed.

6.5 Process

As noted in Section 4.2.3, we have taken a Research Through Design (RtD) approach to our study of the Awareness Agent, with our primary design artefact being a prototype implementation of our notional Awareness Agent.

Our general approach to agent development was to start with a conceptual idea of what we wanted to achieve – a design for an agent that could meet the information needs and address the issues experienced by the user research and the personas that we had developed – and then refine that via a process of iterative prototyping, reflection and further development.

At the outset of this process, much of what would become our System Model [6.6] was not clearly defined; it was only by following a process of iteration and reflection that we were able to formulate this. The most important drivers for this process were *problem-solving* and *feedback*. It may seem counter-intuitive to consider problem-solving as a driver for new design, but we found that many new conceptual features emerged from

reflection on problems encountered during the prototyping process. For example, many of the data abstraction techniques that we developed were born of a need to address a practical difficulty in implementing a prototype. Feedback is more obvious: by considering ongoing feedback from testers we were able to refine what did and did not work, and make design decisions based on this.

In order to capture our reflections and significant design decisions, we kept a contemporaneous log of the design and development process while we were working on the Awareness Agent prototype and testing. The log itself is a cornerstone of our research through design, capturing constraints and decisions made during the process, documenting the basis for the eventual design.

The Design and Development Log forms an important part of our RtD methodology, capturing our design reflections and actions. These elements have been used to form the System Model [6.6] and Architecture [6.7] that we have discussed in this chapter. Appendix D contains an abridged version of this log [D.2] and the full log is located in Supplement S1 [[doi:10.21954/ou.rd.28045163](https://doi.org/10.21954/ou.rd.28045163)].

Our prototyping process culminated in a set of studies with volunteer participants that we will cover in Chapter 9 – although it should be noted that this does not represent a finished design, as we discuss in Section 6.10.7.

6.6 System Model

This section describes the system model that we have developed for the Awareness Agent, based on the requirements and design principles previously established. It serves as an abstract conceptual model of the system, able to inform the basis of software development.

Central to our design is the concept of a Content Item [6.6.2], which in part is our answer to the design requirement of abstraction & standardisation [6.4.2]. Most aspects of our overall system model flow from the concept of the Content Item (CI) and how this is used.

6.6.1 Overview

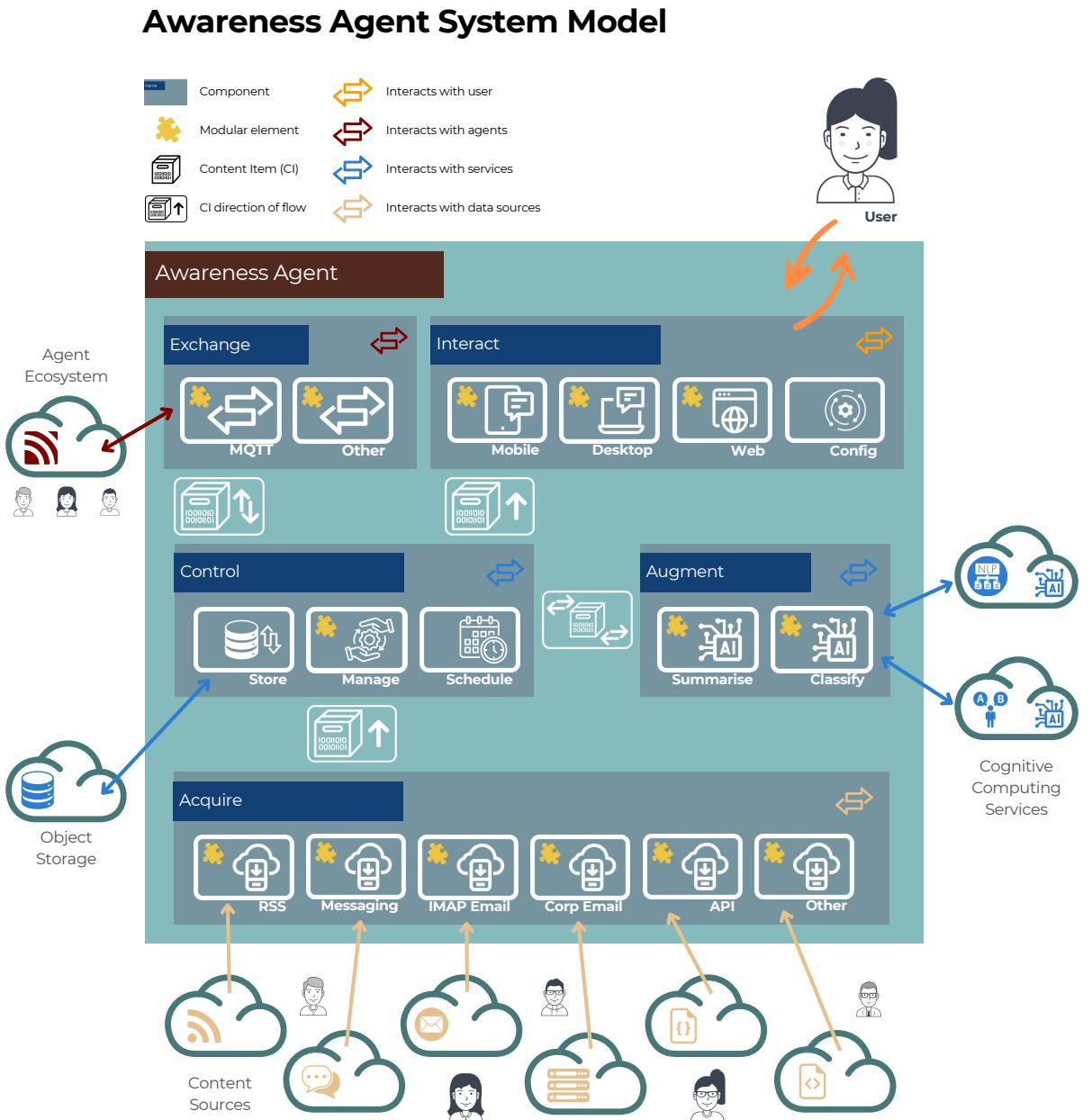


Figure 6.1: Awareness Agent System Model

The overall system model for the Awareness Agent and its place in the ecosystem is shown in Figure 6.1. The agent model comprises 5 high-level design components that we introduce in the following subsections.

6.6.1.1 Acquire

The *Acquire* component is the agent's interface with all incoming content, consisting of *Modules* (such as RSS, or IMAP Email), and *Instances* (such as *The Times RSS Feed*, or *Personal Email*). There are two fundamental types of module: *listener*, and *client*; these represent different ways of interacting with external systems – some systems may push content to external systems via an API (requiring a listener type module), while others may require an external system to actively query them (necessitating a client type module). The former requires the module to be exposed externally (for example as a callback URL) and is reactive, while the latter does not expose a URL but needs to be run on a scheduled or triggered basis.

The mechanism by which a given Acquire module works is transparent to the rest of the agent: the end product of operation is that one or more Content Items are periodically emitted from the component for processing by the agent.

6.6.1.2 Control

The *Control* component is a grouping of a number of functionality elements that carry out necessary tasks. Content Items are received by Control from the Acquire component and processed accordingly. This component is responsible for:

- Initial triage & allocation of incoming CIs
- Orchestrating augmentation of CIs via the Augment component
- Storing and retrieving persisted data (CI, schedules, configuration)
- Passing items to Exchange component for inter-agent communication
- Passing items to Interact component for user interaction via the Allocate component

6.6.1.3 Augment

The *Augment* component adds augmentations [6.6.2.6] to Content Items. This is also a modular component, containing *Modules* and *Instances*. However, rather than source-

based as with Acquire modules, Augment modules are functionality based. For example a Simple Classification module may use an AI classifier service to add a classification augmentation to a CI, with each instance of that module having its own settings. Unlike Acquire, where the number of instances rises according to the number of data sources, it is possible that an agent implementation has only one Augment module of each type.

Augment is passed CIs and emits the same CIs with augmentations added.

6.6.1.4 Exchange

The *Exchange* component passes content items to other agents. Again, the component is modular, with *Modules* (such as *MQTT*), and *Instances* (such as *MQTT Broker X* and *MQTT Broker Y*). The component also receives CIs from other agents and injects these into the agent via the Control component, in much the same way as an Acquire module.

6.6.1.5 Interact

The *Interact* component is the user's interface with the agent. Each type of user interaction is a module type, again using the module instance paradigm for individual configurations. The nature of specific user interaction types is intentionally not strongly defined in the system model; an Interact module encompasses any type of component that can take content items and present them to the user, while also taking user input. For example one type of Interact module may push filtered and formatted content item summaries out to a messenger service and listen for responses from it; another may expose a web interface allowing the user to explore and interact with items. It is a requirement that an instance of an Interact module be able to extract all the data it needs from fully augmented CIs passed to it; this is something achieved with the help of the standardised CI structure.

6.6.2 Content Item

In order to fulfil the design requirements of Abstraction & Standardisation [6.4.2], as well as Modularity & Commoditisation [6.4.3] we developed the concept of the Content Item

(CI), as the core data object used by the Awareness Agent. These requirements translate into the following aspects of the CI model: *semantic & formatting commonality*, and *self-containment*.

6.6.2.1 Semantic & Formatting Commonality

Practical Challenge

Most existing potential content sources such as email, messaging platforms or RSS have their own native data structures and format². To enable consistent processing, the Awareness Agent converts each source's native format into a unified structure and medium – in this case a JSON document extending *JSON-LD*³. By converting all content to a single format having common structure rules, we can use consistent techniques to process and manipulate the content. Any format could potentially be suitable, but we chose JSON – and specifically JSON-LD – due to its widespread use, ubiquitous software tooling, and the other properties that help us address the conceptual challenges in addition to the practical ones.

Conceptual Challenge

Beyond structural differences, the metadata itself must be conceptually aligned across different source types so that the Awareness Agent can recognise, process, and represent conceptually similar metadata from different platforms. The problem of forming a common semantic understanding of different elements is not a new one and has for example been addressed by Dublin Core⁴ and FOAF⁵ for their specific fields. Just as these standards can easily be expressed in JSON-LD, so can our own approach to semantic harmonisation.

For the Awareness Agent, we apply a similar approach to the aforementioned standards.

For example, an email's `From` field and a Slack message's `Sender` field serve the same con-

²For example <https://www.rfc-editor.org/info/rfc5322> describes email, RSS has its own specification defined at <https://www.rssboard.org/rss-specification>, while Slack has its own proprietary API and message format: <https://api.slack.com/>

³<https://json-ld.org/> [<https://perma.cc/UQK6-2QSL>]

⁴<https://www.dublincore.org/>

⁵<http://xmlns.com/foaf/spec/>

ceptual purpose but are stored differently. To address this, the Awareness Agent standardises metadata through *Standard Fields* and *Extended Fields*, enabling uniform processing across systems, much like these systems align diverse data.

Consideration of Existing Ontologies: Dublin Core and FOAF

Dublin Core and *FOAF* are widely used ontologies that could potentially provide some standardisation for certain types of metadata. However, they are not fully suited to the communication-based content that the Awareness Agent processes, such as social media messages and instant messaging content.

Dublin Core: This ontology provides a standard set of concepts aimed primarily at publishing and library sciences, such as `dc:title`, `dc:creator`, and `dc:subject`. While useful for describing static resources like documents or web pages, *Dublin Core* does not capture the specificities of communication metadata, such as message content, sender, recipient, or the channel through which a message is sent. These are essential fields for the Awareness Agent, where the focus is on processing communications and social interactions, which are not the primary focus of *Dublin Core*.

FOAF (Friend of a Friend): *FOAF* is more aligned with identifying entities and relationships between people or organisations, with properties like `foaf:name`, `foaf:knows`, and `foaf:Person`. While *FOAF* is particularly suited to describing these structural elements in a social network, it does not include fields that represent communication events or content, such as where a message was sent (`T0`) or the body of a message (`BODY`). Thus, while *FOAF* might be helpful for identifying the actors in a communication, it is less relevant for the content or the context of those communications.

While both *Dublin Core* and *FOAF* are valuable ontologies, neither fully addresses our requirements of processing and standardising communication content, where the metadata needs to include fields such as message body, sender, recipient, timestamp, and communication channel. For this reason, we have opted not to extend these standards but rather

to define a standalone schema for handling such content. This also gave us the ability to define a mechanism for storing additional data in the CI [6.6.2.3].

This decision is not meant to create redundancy but rather to establish a *specialised framework* tailored to communication metadata. By designing this as a standalone but compatible framework, the Awareness Agent provides a flexible and extensible system that can incorporate elements from existing ontologies where appropriate, while ensuring the metadata structure is comprehensive enough to meet the specific needs of our design.

6.6.2.2 Self-Containment

The Awareness Agent concept is highly modular throughout, and the overall system model is based on the concept of Content Items flowing between modular components. To support this, the CI must be self-contained: a module must be able to examine a CI in isolation and access all the information it needs to perform its tasks. Similarly, it must be able to add any information that it needs to pass on to other modules within the CI structure itself.

Our model approaches this by using a CI structure that contains three top level elements, as illustrated in Figure 6.2:

- **Data** – the JSON-LD containing the item content itself [6.6.2.3]
- **Metadata** – other metadata relating to the item that can be used to identify the type, source and history of the item [6.6.2.5]
- **Augmentations** – augmentations of the CI that have been applied by individual modules [6.6.2.6]

The final top-level component of a Content Item is a unique identifier (UUID) that is assigned at creation time and is used to uniquely identify a CI within the agent ecosystem.

The CI structure is illustrated in Figure 6.2.

The relevant Architecture section for the process of CI generation is 6.7.3.

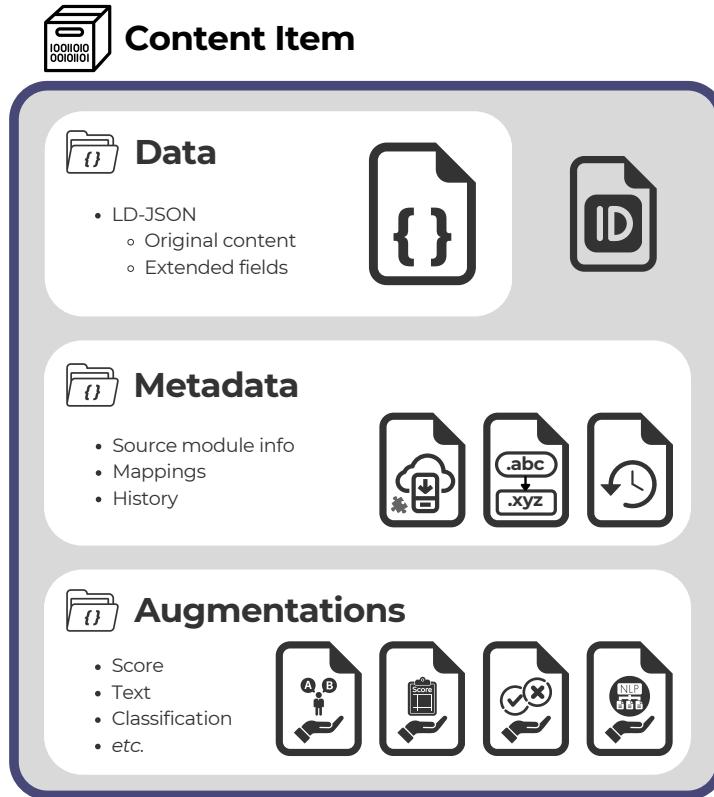


Figure 6.2: Content Item Structure

6.6.2.3 Data

Our model for Content Item Data (CI Data) is based on a JSON-LD structure that is directly or indirectly derived from the native format of the content that the CI relates to. The intent of the CI Data structure is to:

- Maintain the integrity of the source content
- Store additional information that is available at creation time
- Allow systematic access to data items

The corresponding Architecture section is 6.7.1.

The JSON-LD structure includes core @ fields to provide context, while other native fields are retained for source-specific details.

To realise the intents listed above and to ensure consistency across content types, Awareness Agent understands three key categories of metadata:

Standard Fields: These fields represent common conceptual properties, mapped from native fields. Examples include TYPE, ID, FROM, T0, SUBJECT, DATE_SENT, and BODY. These fields are not stored as tangible entries in the CI Data structure, but are instead always mapped representations of Native or Extended fields. The mapping concept is outlined in Section 6.6.2.4.

Extended Fields: These fields provide additional data for each content item, which is standardised at creation time. This data may be derived from native fields or fetched via external queries (for example by retrieving additional information about a message's author from an API). By storing these fields during creation, the Awareness Agent makes each content item *self-contained*, ensuring consistency even as external systems evolve. It also provides a way to capture information that may only be reliably available at creation time via a contemporaneous API query. An example of an extended field would be the name of a Slack channel that a message is posted to: the native message object that is passed to external apps contains the ID of the channel but not the name, so this value can be queried and stored as an extended field.

Native Fields: These fields exist in the source content and are carried across directly to the CI Data; the Awareness Agent may read values from these either directly or indirectly via Standard Fields, but will not change or rename them. This satisfies the requirement to maintain the integrity of the source data.

6.6.2.4 Mapping

As introduced above, mapping of data items is an important feature of the Content Item design and use as it allows us to address the conceptual challenges outlined in Section 6.6.2.1. To do this we have introduced several conceptual standard fields (T0, FROM, BODY etc.) and established a mechanism for associating these with tangible data fields using mapping.

In accordance with our approach of using linked data, we can model the mapping as a JSON-LD document. Figure 6.3 shows a conceptual example of this, showing how concep-

tual terms are mapped to specific native or extended fields in the data JSON.

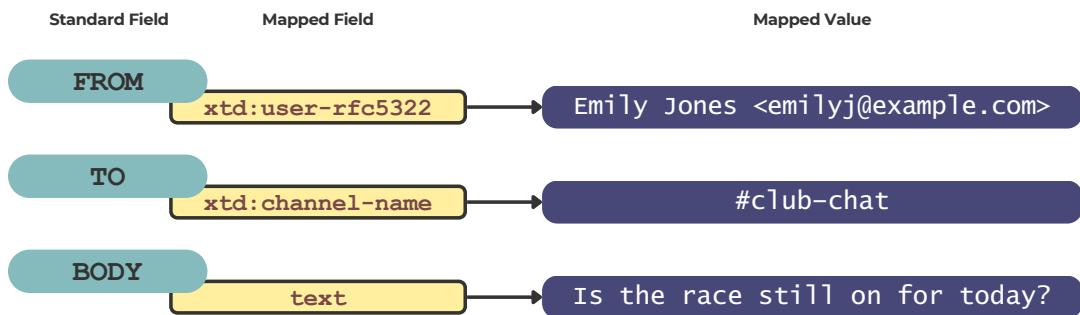


Figure 6.3: Content Item Data Mapping Illustration

Mapping is detailed in Section 6.7.1.4.

6.6.2.5 Metadata

Each Content Item has a history metadata where all operations on the CI (ingress, transfer, augmentation and so on) are recorded. The intent of the metadata is to allow any agent (or human) inspecting the CI to understand where it has come from and what has been done to it. This can be useful for items that are shared between agents, for example identifying items that have already been seen, or understanding what augmentations have been applied by other agents. It is also generally useful for explainability and tracing of content, allowing use to understand the full provenance of each CI that we see and act accordingly.

6.6.2.6 Augmentations

The concept of CI Augmentation is designed in accordance with the requirement of modularity and model of Self Containment [6.6.2.2]. An augmentation is something that we add to a Content Item to increase its value or utility to the agent or user. For example, assigning a classification to a message (“work” or “personal”) is one type of augmentation, while another may be to add a short AI-generated summary of a text item.

The *Augment* component is modular, consisting of *Modules* (such as *Classification*, or *Summarise*), and *Instances* (such as *Simple AI Classifier X*, or *LLM Text Summariser Y*).

The augmentation process is self-contained; the agent will pass a CI to each configured Augment instance in turn, and the output of these instances will be added to the CI Augmentations data. The Augment Awareness Agent component understands only how to add individual augmentations by passing to a configured module instance, and does not need to know the how the augmentation was created or how it should be used.

Similarly, components that make use of CI Augmentations do not need to know how a specific augmentation was produced, only that it conforms to a type that they are expecting to handle. All required information for processing the augmentation is contained in the collection of Augmentations.

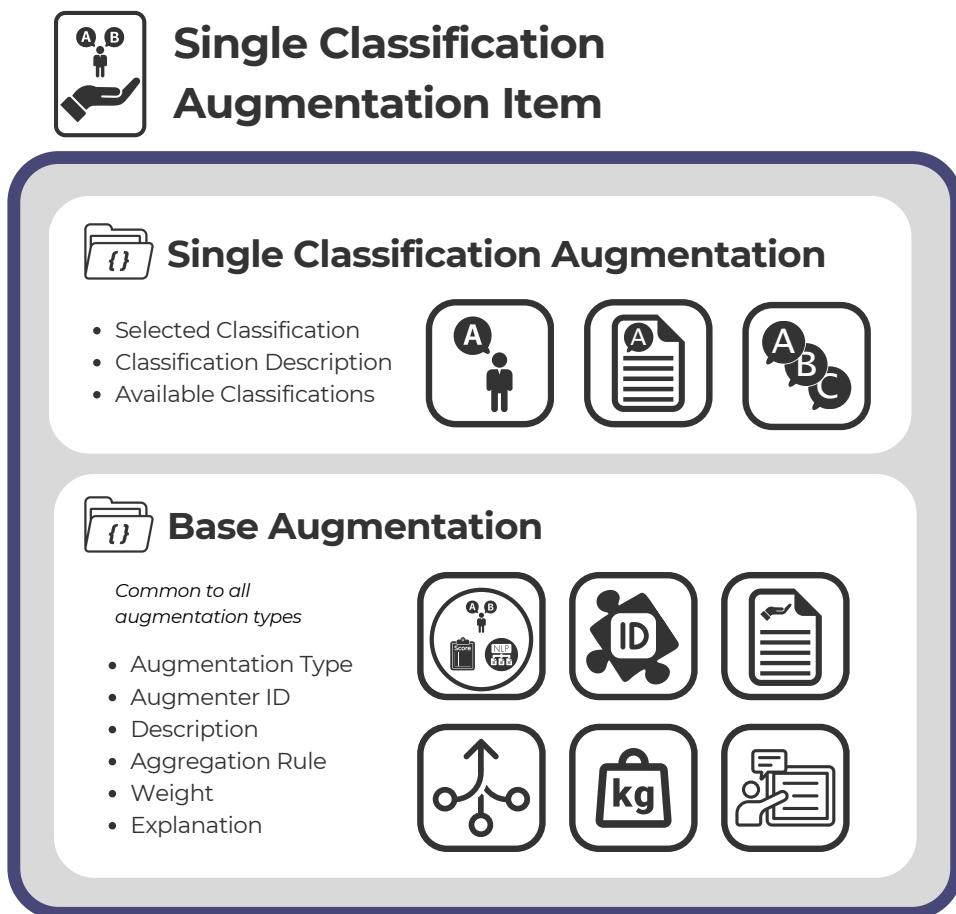


Figure 6.4: Simple Classification Augmentation Item

The relevant Architecture section covering CI generation and augmentation is 6.7.3.

All Augmentations have the following common properties, as shown in Figure 6.4, which illustrates the structure of a Simple Classification Augmentation Item:

- **Augmenter ID** – Text string identifying the Augment module instance that performed the augmentation; this may be useful to receiving components that are looking to use a specific augmentation
- **Type** – A categorical value describing the type of augmentation (see below)
- **Description** – Simple textual description of the augmentation supplied by the augmenting module
- **Weight** – Numerical relative weight that can be applied when comparing augmentations (see below)
- **Aggregation Rule** – A categorical value showing what aggregation rule should be applied to this aggregation, such as taking a maximum or average value, or using the most recent value. Other modules can use this to decide how to handle situations where augmentation values need to be aggregated or combined
- **Explanation** – A collection of named data items added to the augmentation that provide explanations of the basis for creating the augmentation; this could be a simple text description or algorithmic details

Augmentation types include FLAG, SIMPLE_SCORE, SIMPLE_TEXT, SINGLE_CLASSIFICATION, and MULTI_CLASSIFICATION.

Aggregation rules include AVERAGE, MAX, MIN, FIRST, and LAST.

Weighting tells the system about the relative importance of each augmentation when it makes judgements about how to process or present items to the user. This may be part of an aggregation process or standalone. For example, two augmentations each having the default weight of 1.0 would be considered equal, whereas if one carried higher numerical weight value it would be considered to have higher significance. This could lead to the augmentation with the higher weight being chosen if two separate augmentations make conflicting classification or flagging choices for example.

Additional properties contained in an augmentation item will depend on the nature of the

Augmentation. For example, the Single Classification Augmentation Item shown in Figure 6.4 – which is the result of an Augment process applying a single text classification to a CI – will contain the following type-specific properties:

- **Classification Text** – The text of the classification chosen (i.e. “work”)
- **Classification Description** – Descriptive text for the selected classification (i.e. “Work-related message”)
- **Available Classifications** – List of the classification options that were available for the classifier to select (i.e. [“work”, “personal”])

Similarly, a Simple Text Augmentation Item – for example one containing a short summary of the item – has only one type-specific property, **Text**.

6.6.3 Context Mapping

The Awareness Agent approach to the modal operation requirement [6.4.5] is to implement Context Mapping, performed within the Allocate part of the Control component [6.6.1.2]. This component uses the notion of a currently selected *Context* to determine where and when Content Items should be sent to the Interact layer, with a map being maintained of conceptual Contexts (i.e. Home, Work, Travelling, “Sorting out club fixtures”) to outputs. If a given Content Item does not qualify for immediate distribution to the user – for example because it is deemed low priority and not in line with the current context, it is held up until the context is compatible with sending the item.

So for example, if the current context is set to “Personal” then mappings can be configured to hold up any work items, unless they have an augmentation that overrides this (such as “very-urgent”).

The implementation for this is described in architecture Section 6.7.4.3.

6.6.4 User Interaction

6.6.4.1 Modular Concept and User Interaction

The variety of different ways that users interact with technology has shown us that one size does not generally fit all when it comes to user interfaces, which was borne out by the survey and the persona scenarios that we developed.

As discussed in Section 6.4.4, our concept for an interaction model is two-fold: proactive and reactive. As different paradigms these may be suited to entirely different implementations, so we take the approach of a modular structure: for example proactive interaction could be handled by one module implementation (for example something based on a messenger app) while reactive interaction could use a different module (such as a web UI).

We will later go on to document one possible modular implementation of the UI component – the Slack Interact service – in Section [6.7.4.5].

6.6.4.2 Existing Platforms

We take the view that while it is often useful to create something from scratch, particularly something with unique characteristics, this is not always warranted or efficient. In particular, many application behaviours rely on significant infrastructure or application development investment. Take for an example a platform such as Slack⁶, which the user can interact with via multiple mechanisms (mobile app, desktop app, web) and in different ways (granular notification settings, mechanisms for formatting and managing content etc.). We could use these existing assets to build a system that interacts with the user in multiple ways utilising natively supported app platforms. At the expense of the customisation that bespoke development makes possible, this would confer some advantages: existing mature UI elements, an application infrastructure, user familiarity and so on.

The modular approach to UI design makes this possible, as we can choose to implement one UI aspect on (say) Slack, while maybe choosing a bespoke solution for another aspect.

⁶<https://api.slack.com/docs>

6.6.5 Platform & Modularity

As discussed in Section 6.4.3, modularity and flexibility is an important part of the design. Each of the agent components [6.6.1] is conceived as modular in part or in whole. That is, the agent has a common code base that supports functionality and information flow, with modular implementations of those components conforming to the common interface that implement for specific cases. For example, the Acquire component has a set of defined methods to acquire new Content Items and pass them on to the rest of the agent; a particular implementation may implement this for a given type of data source such as RSS data.

The guiding principle is for the system to be able to support diverse data sources and use cases, and to be able to adapt over time as they change, possibly in unexpected directions. The intent is also for the system treat the resources that it uses – such as Machine Learning and cognitive systems – as commodities that can be swapped out when something more useful is available.

Central to the modular approach is a design based around Queues and Services. Each component of the system model is considered to be a service internally, performing a specific function for the overall agent. The communication between these services uses a queue-based mechanism, with the item being exchanged being the Content Item. As we briefly touched on in Section 6.6.2.2, the Awareness Agent design is based on the concept of Content Items flowing between components (or services). As a CI leaves one service (such as Acquire) it is placed on the queue for the next (in the case of Acquire, the CI is placed on the Triage queue, for the next service – Triage – to pick up). This is elaborated on in the Architecture section 6.7.4. Awareness Agent queues are First In, First Out (FIFO)

The queue system is a key feature of our approach to modularity; a service picking up a Content Item from a queue does not need to know anything about how the item found its way onto the queue or what was involved in this process, as long as the item conforms to the expectations of the service that receives it. This allows each service to be a distinct modular implementation – in theory there is no requirement for the services to even be co-located on the same host, as long as the queue mechanism is defined.

In the face of changing user requirements or the availability of new types of service, we can enhance the agent to accommodate this by changing or adding services as needed.

6.6.6 User-Directed Machine Learning Models

A central design principle of the Awareness Agent is user autonomy and control over AI and algorithms that determine what they see [6.4.1]. We have introduced the *User-Directed ML* (UD-ML) concept to allow users to deploy, train, and control their own lightweight ML models to categorize content based on their unique preferences and contexts. This empowers users to tailor the agent's classification capabilities to align with their specific needs, such as sorting information into categories like work, personal, or urgent.

The underlying technology of the ML models used by UD-ML is intentionally not specified at this level, but the initial design is for the service to utilise a classifier model such as that provided by a Support-Vector Network (SVN) [Cortes and Vapnik, 1995]. Our work focuses on the process around using such models, with the expectation that the end result would permit the transparent swapping out of the underlying ML implementation.

Within the overall Awareness Agent system model, we consider User-Directed ML to be an *implementation* approach at the Augment and Interact level. That is, the overall Awareness Agent design does not depend on having UD-ML but it is a feature of our approach. The design of User-Directed ML is discussed in detail in Section 6.7.6.

User-Controlled Model Management The User-Directed ML system gives users full control over the lifecycle of their models, allowing them to spin up and take down models as needed. Users can easily create and configure new models that focus on particular content dimensions, such as work-related versus personal content, or classify information as urgent versus non-urgent. Each model's existence and function are determined entirely by the user, ensuring that they retain authority over how their data is processed and categorised.

Customisable and Trainable Models User-Directed ML models are designed to be trainable by the user, without requiring advanced technical knowledge. Using a process that is integrated with the consumption of content, users can provide data to teach the model how to classify content. Over time, the models can refine their accuracy based on feedback and usage patterns, allowing the system to evolve as the user's preferences change over the lifespan of the model. This makes the system adaptable and responsive to highly personalised content filtering.

Modular and Sequential Integration The modular nature of the User-Directed ML system allows for sequential model integration, where multiple models can work together to achieve complex classification tasks. For example, a user might deploy a first model to filter content by work-related versus personal categories, and a second model to further classify work-related content by urgency. This sequential combination provides a layered approach to classification, offering sophisticated content filtering based on user-defined priorities.

Flexible Lifespan The lifespan of each model is entirely under the user's control. Users can retire models when they are no longer relevant or create new ones as their classification needs change. This ensures that the system is dynamic, reflecting the evolving contexts and requirements of the user without becoming overly rigid or cumbersome. This approach also addresses issues of drift over time on model requirements or models becoming stale – if necessary it is easy for the user to tear down an existing model and replace with a fresh one that is more appropriate to their current needs. For this approach to work, the classifier must be rapidly, incrementally trainable.

User Empowerment Through Control By decentralising the control of ML models, the User-Directed ML concept enhances the transparency and accountability of the system, two critical factors identified in the literature review and survey. Users are not only aware of how content is classified, but they also have direct oversight of the models' responsible for these classifications. This approach is intended to foster trust, ensuring that the user feels empowered to make adjustments and fine-tune the system without reliance on pre-

built, opaque algorithms.

6.6.7 Data Storage

The Awareness Agent requires data storage at multiple points in its operation process for:

- Configuration data
- Content Item persistence
- State information

We adopted an entirely object-based storage mechanism, where objects are stored as JSON documents, identified by key-value pairs. The storage interface identifies each stored item by a parent namespace, and an individual object ID. So for example the agent configuration is stored in a Configuration namespace, using the Agent ID as the identifier. It is expected that the object store offers object persistence, although most state information does not require this.

6.6.8 AXP

We have defined Awareness Exchange Protocol (AXP) as the mechanism by which the Exchange process [6.6.1.4] operates. The existing Content Item structure is the basis for this, as it contains all the information required for exchange and sharing of items: the originating agent and data source are detailed in the CI Metadata, and the CI Data and Augmentations are self-contained. Trust enforcement requires the CI be wrapped in a document containing the CI itself and a signature, added by the transmitting agent using public key cryptography and supported by a web of trust.

The transmission media for AXP can be any technology that supports the transmission of CI documents as JSON.

6.6.9 Autonomous Operation

While a large element of the Awareness Agent concept is autonomy for the agent's owner, the autonomous operation of the agent is a separate but significant part. Returning to weak and strong definitions from Michael Wooldridge and Nicholas R. Jennings [1995] that we discussed in the Literature Review [2.4.1], we can see that some parts of the weak notion of agency – such as operating without direct intervention [Castelfranchi, 1995] and inter-agent interaction [Genesereth and Ketchpel, 1994] – are addressed by parts of the model we have already developed. Yet others – such as goal-directed behaviour – are not. The model that we have put forward so far does not address those 'stronger' parts of the weak definition of an agent, nor does it approach those "strong" characteristics such as *intention* [Shoham, 1993].

Yet there is a place in our model for the strong agent behaviour that can be described by more mentalistic notions, and we propose an Agent Autonomy Service (AwAgAS) to fulfil this role. The AwAgAS sits alongside the other services in the Agent, but takes on a different role – part way between the reactive structure of the Awareness Agent core and the autonomy of the user themselves. Recent changes in commodity AI have made this role possible to a degree not envisioned at the initiation of our project; while the design principles of modularity and commoditisation [6.4.3] have helped us integrate the concept, the settled structure of our system model reflects that this was not something that we looked to design in this form at the outset.

The AwAgAS in our model has three driving components for its action:

- Continual ingestion and processing of Content Items
- Direct user instruction
- Indirect user input
- Scheduled operations

We envision an engine that conducts operations driven by each of these, utilising commodity AI to handle logic tasks. AwAgAS carries out the following types of action:

- Adding, removing or modifying Acquire sources
- Adding or removing Exchange channels
- Communicating information to the Awareness Agent owner

6.6.9.1 CI Ingestion

Each CI that is augmented by the Awareness Agent is passed to AwAgAS at the Allocate stage, in addition to Interact processing. Each item is considered, in particular the Augmentations and the item source/content. For a mature (i.e. well-trained) User-Directed ML model, the augmentations are likely to indicate content that is interesting (or not) for the user. Highly rated items would be used in aggregate form as the source for content recommendation type queries to identify new content sources, such as RSS feeds to add to an RSS Acquire module. These could either be enacted automatically, or more conservatively communicated to the user for action. Only items that meet a high threshold should lead to action.

6.6.9.2 Direct User Instruction

The user can communicate directly to AwAgAS via the Interact Service to request actions. This type of on-demand processing uses the same mechanisms as for actions driven by continual CI ingestion, but demands prompt or interactive results. For example, the user may request “show me more sources that provide content like this item” or “find me more sources about road cycling”, which would trigger an interactive process. Requests may also be *administrative*, such as natural language requests to “delete model X” or “add category ‘foo’ to model Y”.

6.6.9.3 Indirect User Input

Indirect input to AwAgAS comes from other user interactions with the agent, for example if a user indicates that they are not interested in a particular item. On aggregate these should trigger processes leading to content actions or recommendations.

6.6.9.4 Scheduled Operations

The AwAgAS should also carry out processes on a schedule. This a way of either processing aggregate level information, or performing a “daily check” or similar to looks for new sources or maintain existing ones.

6.7 Architecture

In this Architecture section, we expand on the System Model previously introduced in Section 6.6. The intent is to provide a more concrete and technical description of the Awareness Agent design, expanding on the more abstract model with more detail and actual examples from a reference architecture that we have developed. While most of the concepts discussed in this section are still generally applicable to the Awareness Agent concept, we use some specific implementation examples to illustrate particular aspects.

6.7.1 Content Item Structure

This section expands on the CI structure illustrated in Figure 6.2.

6.7.1.1 Standard Fields (`awag:std`)

These fields represent common conceptual properties, mapped from native fields. Examples include `TYPE`, `ID`, `FROM`, `T0`, `SUBJECT`, `DATE_SENT`, and `BODY`. These fields do not exist in a concrete form in the CI Data structure, but are instead always mapped representations of Native or Extended fields. For each source type, a *metadata mapping* links native fields

to these standard properties, in much the same way RDF uses triples (subject-predicate-object) to establish relationships in a machine-readable way. The mapping concept is discussed in detail in Section 6.7.1.4.

6.7.1.2 Extended Fields (`awag:ext`)

These fields provide additional metadata for each content item, which is standardised at creation time. For example, an item having @type “awag:slack-message” will always include fields `awag:xtd:user-name` and `awag:xtd:conversation-name`. These fields may be derived from native fields or fetched via external queries (e.g. retrieving the value for `awag:xtd:user-realname` from the Slack API). By storing these fields during creation, Awareness Agent makes each content item *self-contained*, ensuring consistency even as external systems evolve. It also provides a way to capture information that may only be reliably available at creation time via a contemporaneous API query.

6.7.1.3 Native Fields (*no prefix*)

These fields exist in the source content and are carried across directly to the CI Data; the Awareness Agent may read values from these either directly or indirectly via Standard Fields, but will not change or rename them, except if a naming conflict occurs.

6.7.1.4 Mapping

As introduced above, mapping of data items is an important feature of the Content Item design and use as it allows us to address the conceptual challenges outlined in Section 6.6.2.1. To do this we have introduced several conceptual standard fields (TO, FROM, BODY etc.) and established a mechanism for associating these with tangible data fields using mapping.

Standard field mapping is performed using a `ldMapping` document in the CI Metadata, as was previously shown in Figure 6.3. These mapping documents are defined on a source-specific basis, with each different source type having its own mapping defined.

An example mapping document showing the list of standard fields is shown in Figure 6.5.

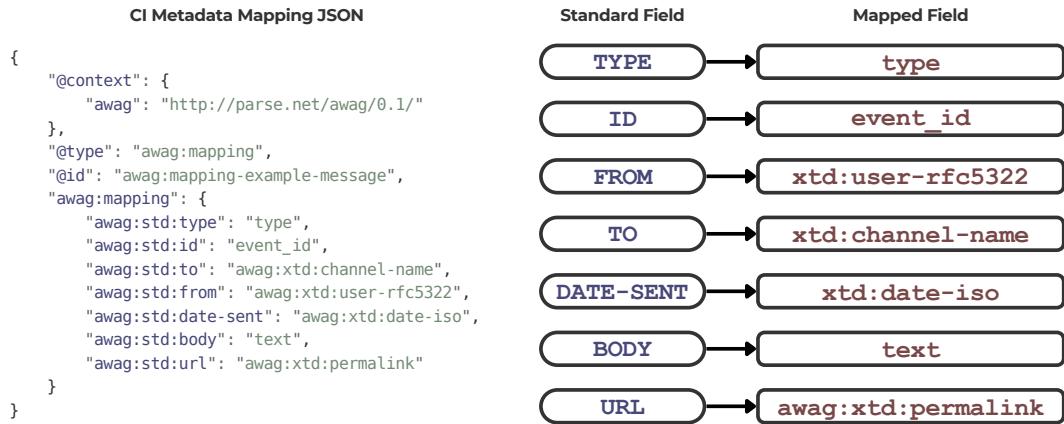


Figure 6.5: Content Item Data Mapping – LD Mapping

The mapping for Standard data fields in CI Data is illustrated in Figure 6.6. This example shows how standard fields are addressed in CI Data for an example type source item; we can see that mappings are to both extended fields (with an xtd prefix) and native fields (with no prefix). This allows the agent to benefit from the additional self-contained data offered by the use of extended fields where appropriate, selecting between native and extended fields in a way that is transparent to the system.

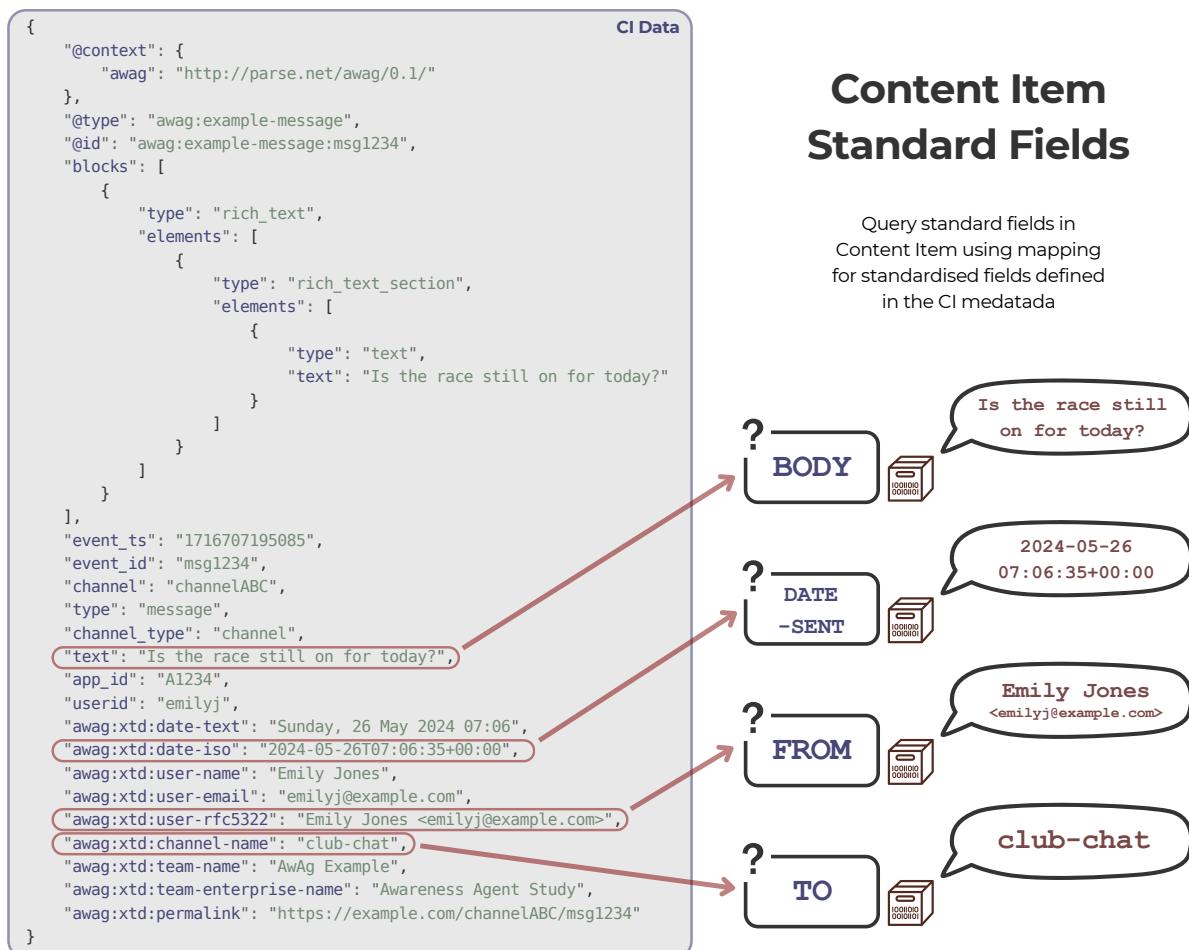


Figure 6.6: Content Item Data Mapping – Standard Fields

The mapping for Extended data fields in CI Data is illustrated in Figure 6.7. This example shows how standard fields are addressed in CI Data for an example type source item.⁷

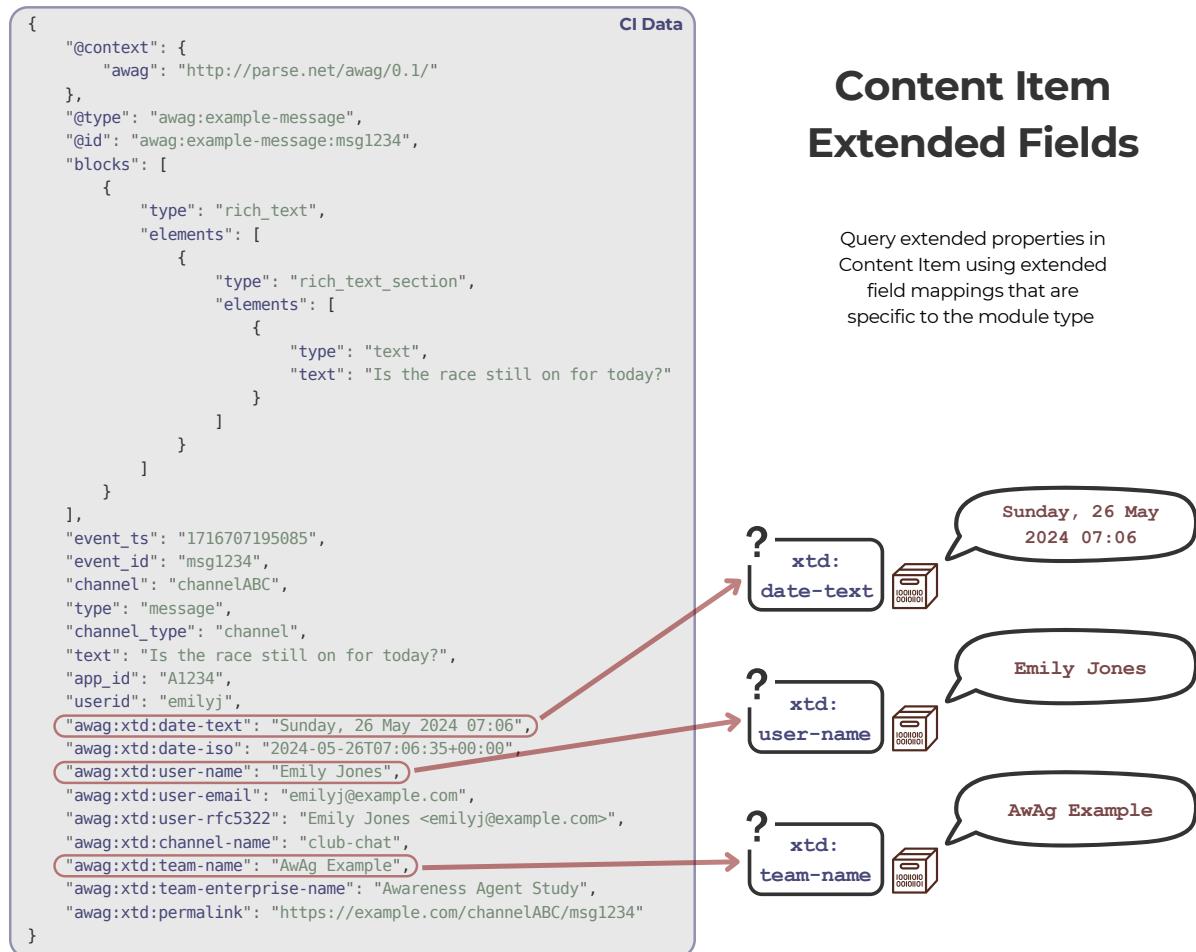


Figure 6.7: Content Item Data Mapping – Extended Fields

⁷Our example includes formatted xtd:date-text and xtd:date-iso fields, which are derived from the native event_ts epoch timestamp at CI creation time – it is a decision for the implementer as to whether such redundant derived fields should be included as convenience fields within the CI.

The mapping for native data fields in CI Data is illustrated in Figure 6.8. These are fields that have the same name in the CI Data as in the original native format.

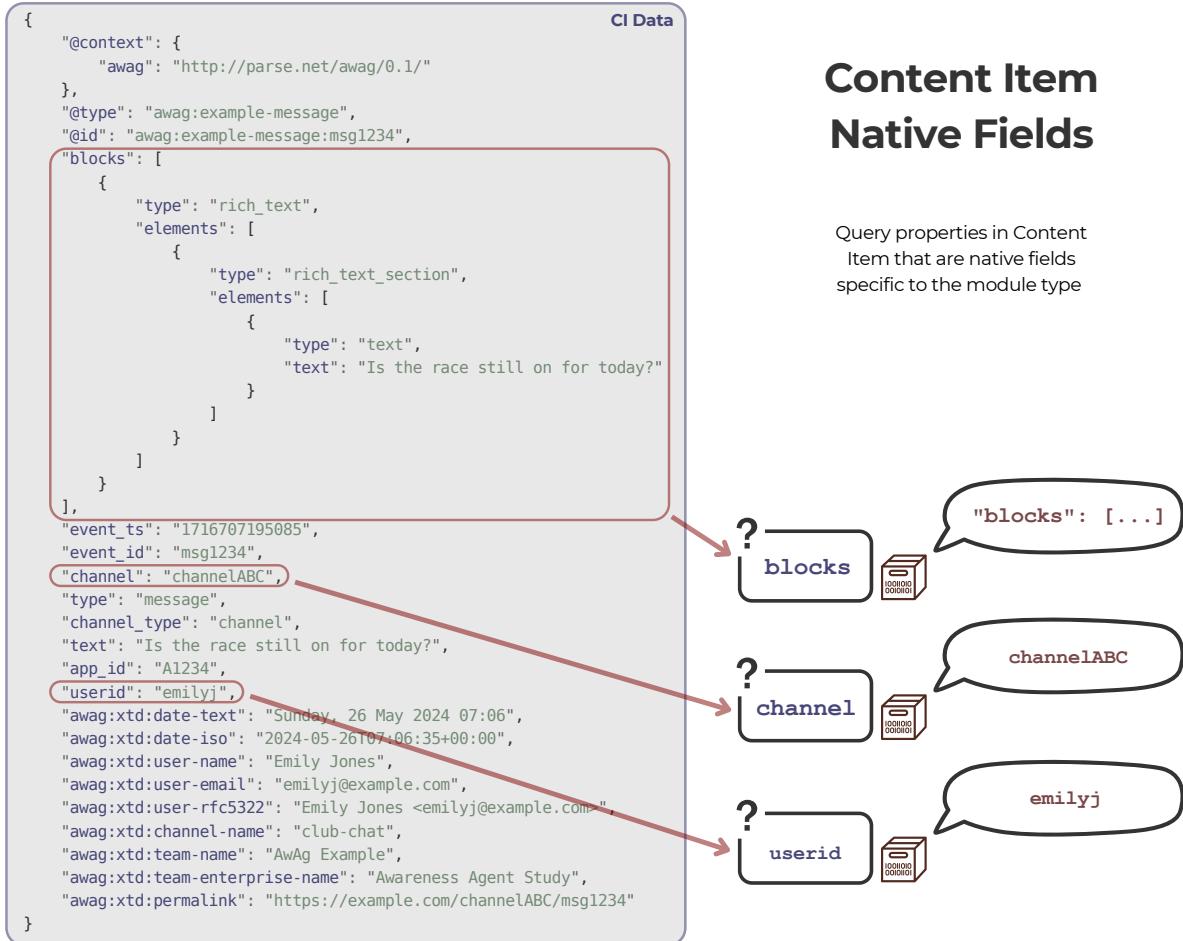


Figure 6.8: Content Item Data Mapping – Native Fields

6.7.2 Content Formatters

Having established a system of standard fields for a Content Item, we can then design a standardised way of representing CIs in different contexts. We do this by introducing a concept called the Content Item Formatter, as software element that takes the standardised structure of a Content Item and renders a representation of it in a way that is independent of the CI type and origin.

Figure 6.9 shows how Content Item Formatters use Standard Fields and Augmentations in Content Items to produce a consistent representation in a format suitable for the output channel. The default formatter is Text, and this will produce a simplified textual representation of the CI; where the Interact output channel supports richer content, a platform-specific formatter can be used. For example, a Slack Item Formatter will take a Content Item and produce a Block-based output⁸.

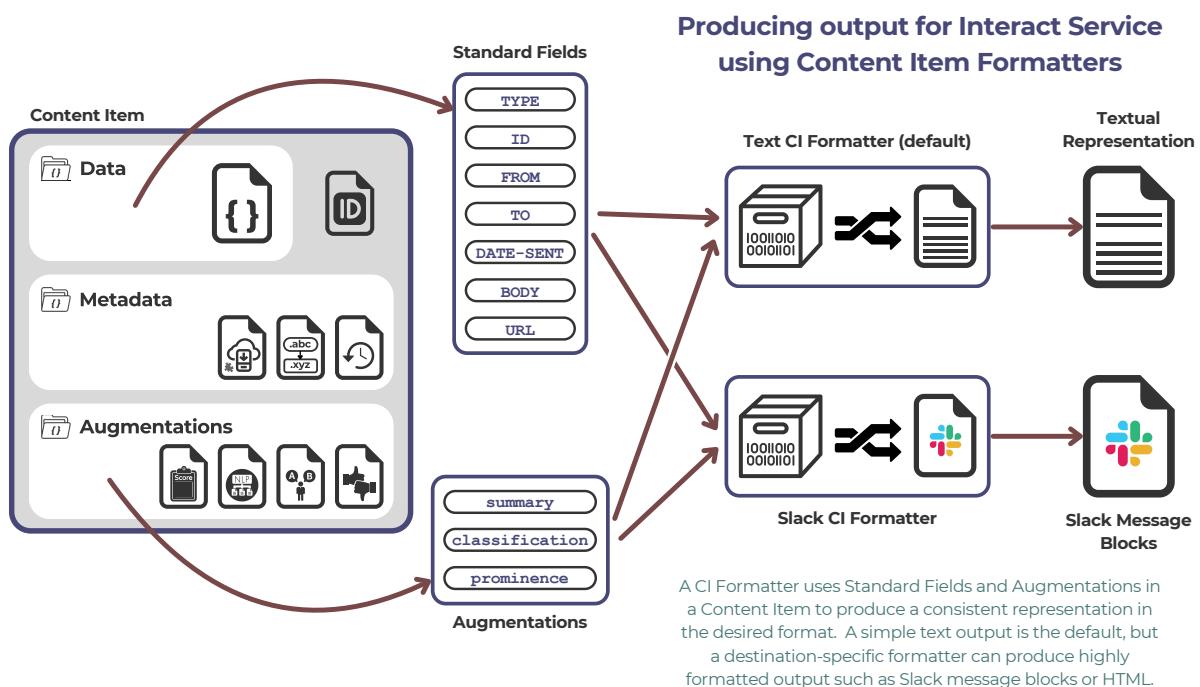


Figure 6.9: Content Item Formatters

6.7.3 Content Item Generation

As was introduced in Figure 6.1 and Section 6.6.1.3, the CI is created and augmented as it passes through the modules of the Agent. The process flow for a single CI is shown in more detail in Figure 6.10.

The initial creation of the CI Data part is illustrated in Figure 6.11. This shows how the CI Data is assembled from native data and the results of additional queries & calculations. The example given shows how a CI Data item is generated for a synthetic example source

⁸<https://api.slack.com/block-kit>

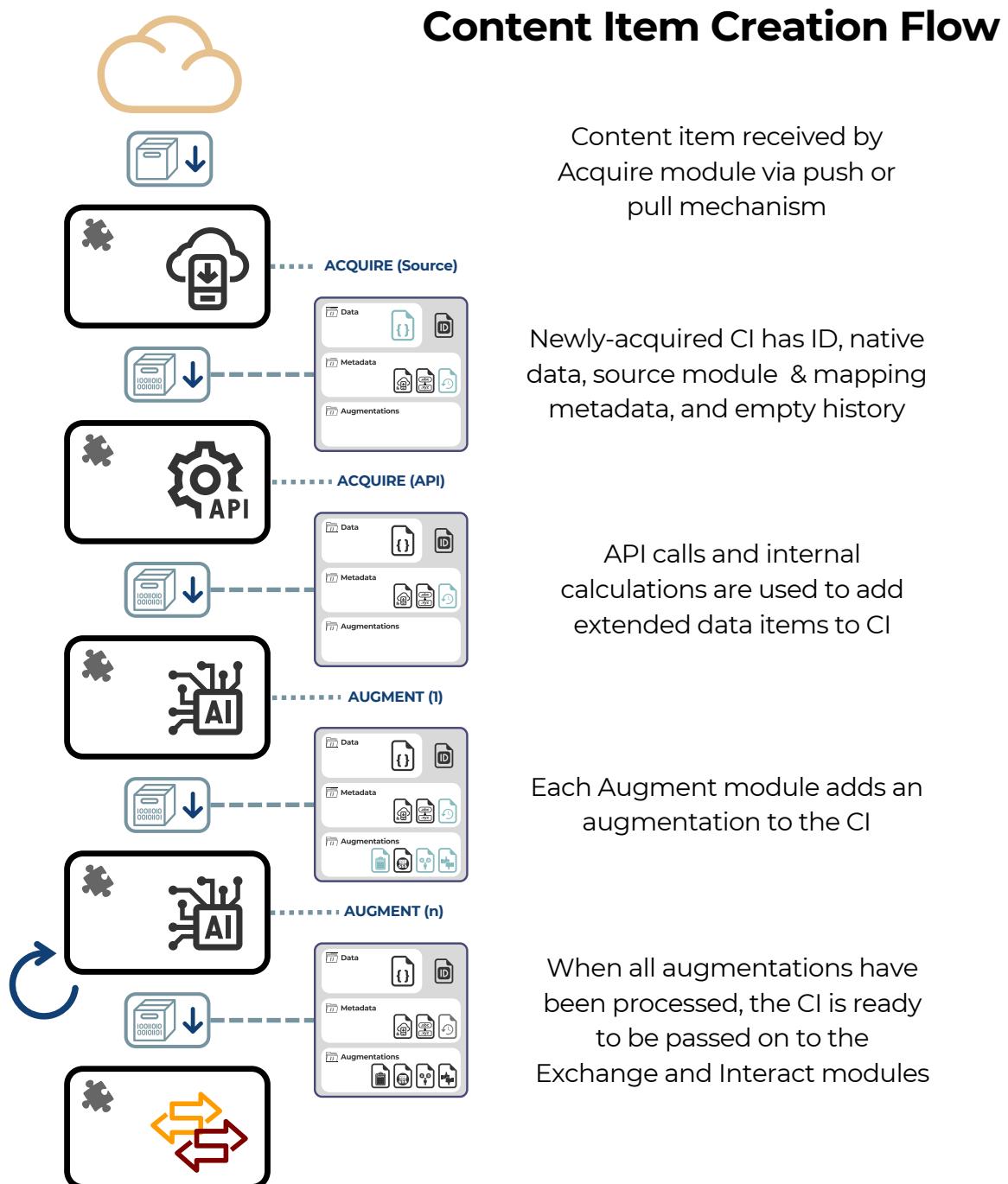


Figure 6.10: Content Item Creation Flow

message⁹. The Awareness Agent operates on the basis that every source message is a JSON document; there is an implied first step where any non-JSON source is converted to JSON with appropriate property names during the Acquire process – this part is considered an implementation detail within the Acquire module.

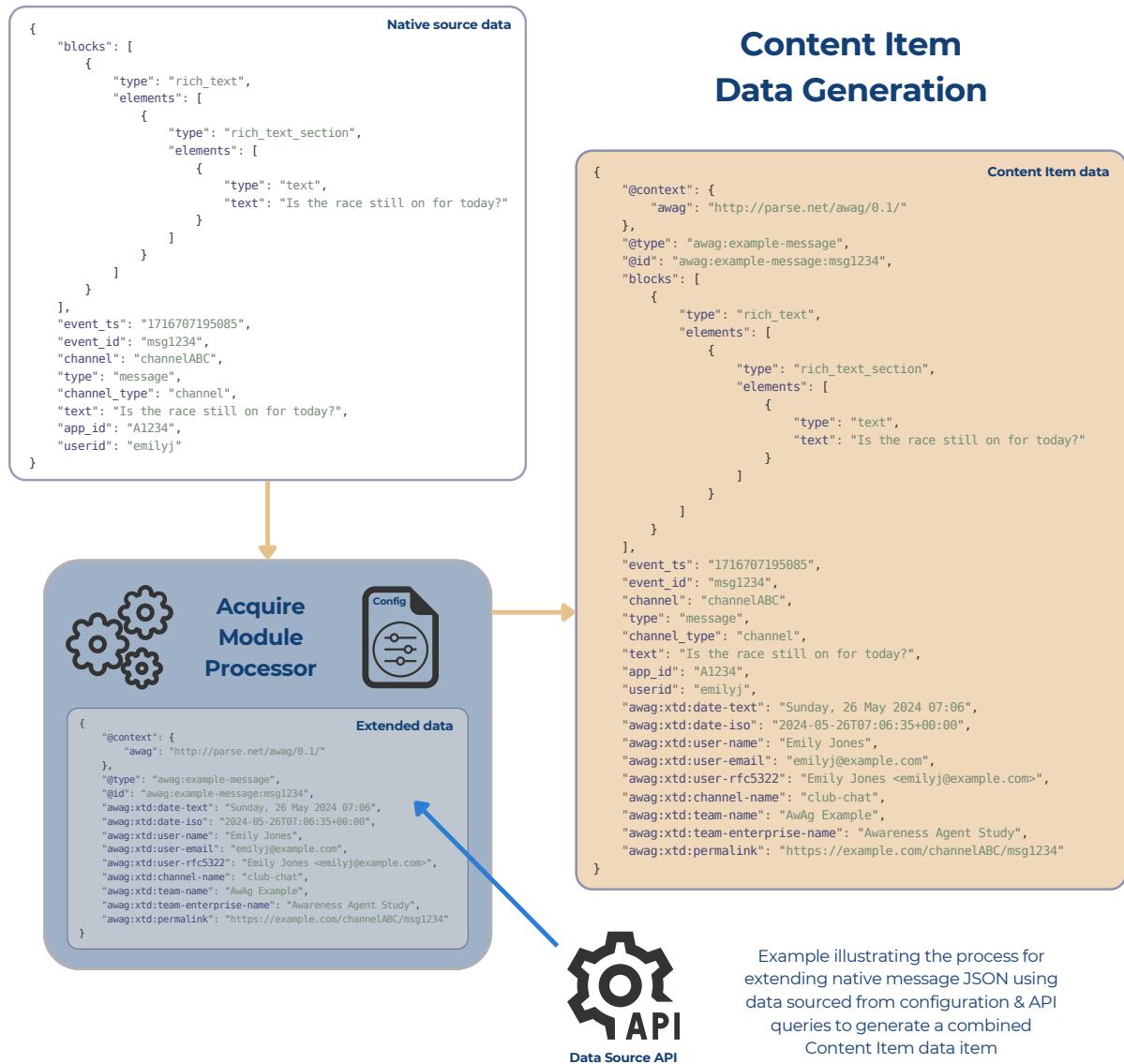


Figure 6.11: Content Item Data Generation

⁹Note that the example given is loosely derived from a Slack message, with changes added to improve readability and better illustrate the concepts.

6.7.4 Services, Queues and Engines

Figure 6.12 shows how Queues are used to transfer Content Items through logical Services in the agent, as was introduced in Section 6.6.5. Each logical service consists of a number of modular components that perform a related function, such as Acquire or Augment. A CI is processed by each applicable module within a given service before leaving it via either a queue, external process or discard action.

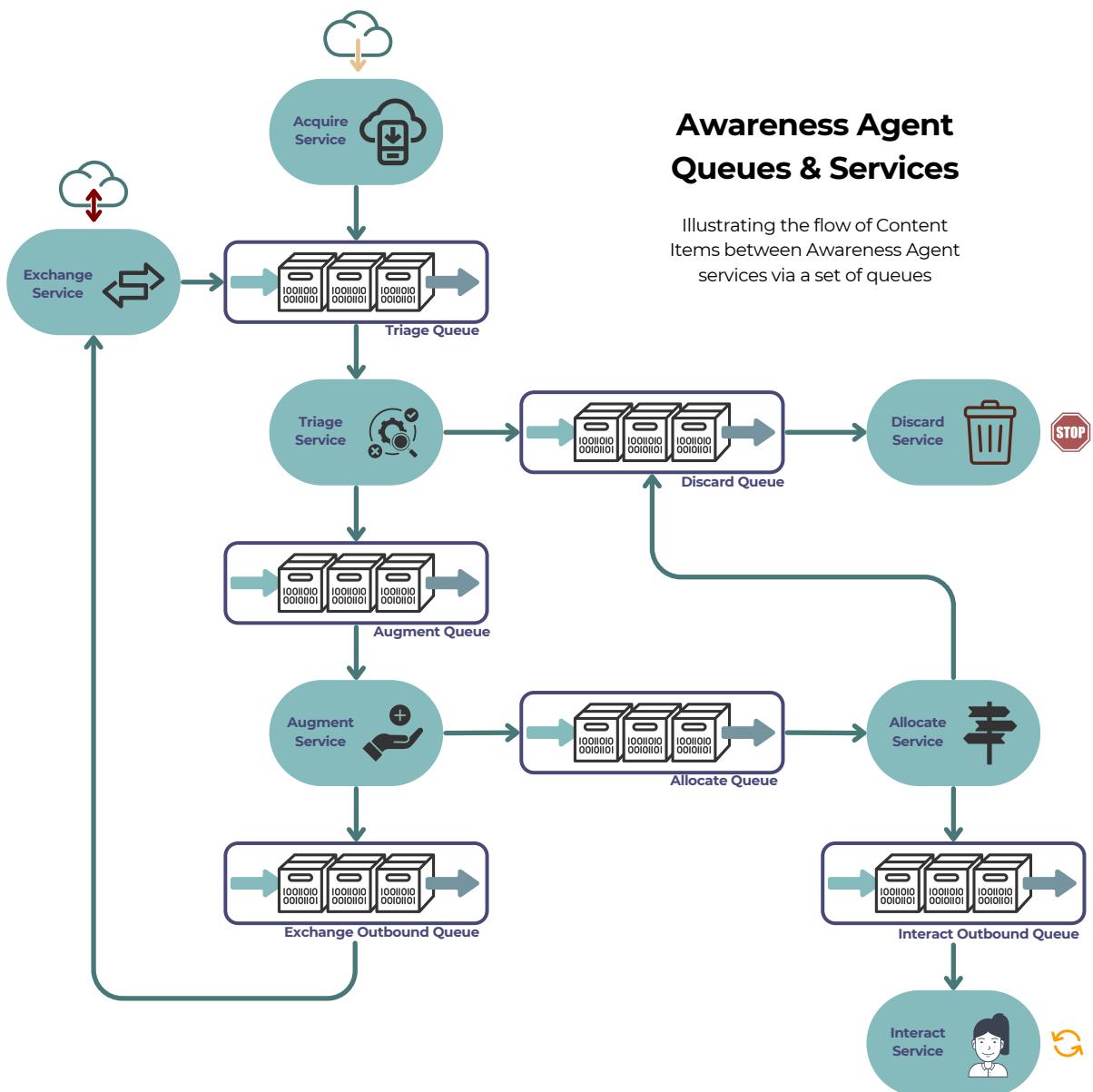


Figure 6.12: Awareness Agent Queues & Services

Figure 6.13 shows how the service modules are launched by the agent at Launcher Service runtime. The Launcher Service examines the installed modules and corresponding module configurations. For each module having a valid and enabled configuration, the Launcher Service runs that module's own Launcher, which will run an instance of the service engine for each configured and enabled Module Instance. Some module types can have multiple *Implementations*, and module implementations can have multiple *Instances*. An Implementation is a different variant of the same type of module – for example *RSS Acquire* and *Slack Acquire* are both types of Acquire module, that have code specific to their implementation needs (so a Slack module understands how to interact with Slack, while an RSS module knows how to interact with RSS feeds). Within these, multiple Instances can be configured. An Instance uses the same code (i.e. *Slack Acquire*) with different instance configuration settings (i.e. Slack Workspace A, Slack Workspace B). Module Implementations are launched by the agent Launcher Service, whereas Instances are launched by the module's own launcher.

6.7.4.1 Acquire & Triage Services

As previously discussed, there are two modes of operation for Acquire module services – *Client* and *Listener*. Client modules *pull* content from external sources, on a schedule or trigger basis. Listener modules instead receive content via *push* operations, by exposing a web service to allow third party applications to push content to the service. We will illustrate these with one example module of each type: RSS (client) and Slack (listener).

Client Example – RSS Figure 6.14 illustrates the RSS implementation of the Acquire Service. RSS is an example of a *Client* type acquire module, where content is pulled from a third party service. The module instance configuration contains a list of RSS feeds to poll, and details of the schedule on which to poll them. Multiple instances can be configured to allow for different schedules for different groups of feeds, but only a single instance is illustrated here.

Each ingested RSS item is assigned a tracking ID by the service – preferably the GUID

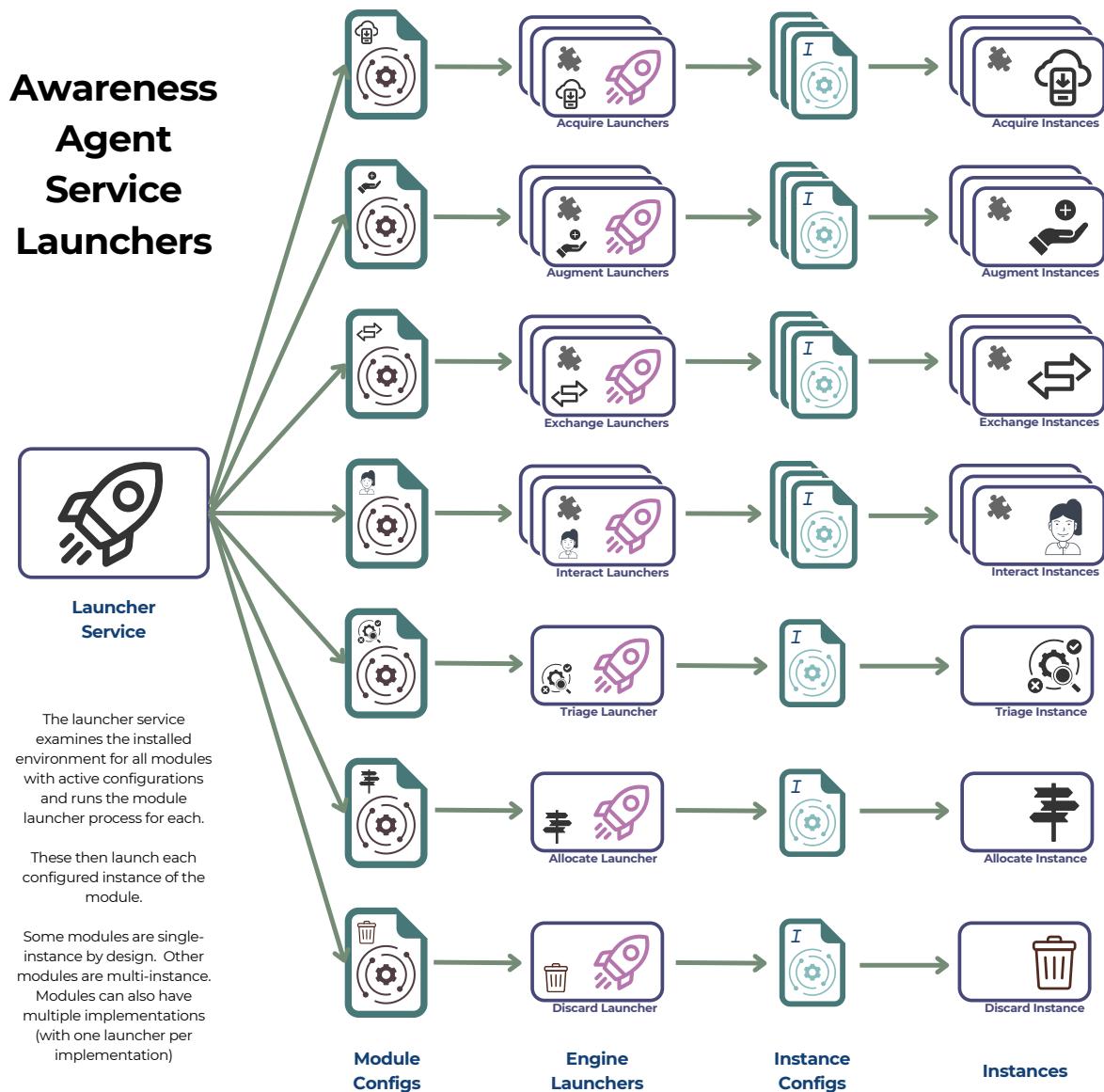


Figure 6.13: Awareness Agent Service Launchers

of the source item¹⁰, otherwise a generated ID. This is used to track items so that those already seen are ignored. The instance configuration also allows for a maximum item age to be set. Those RSS items that are not too old and have not previously been ingested are used to create a Content Item [6.2] for each. In this case, a JSON version of the RSS item is set as the Data part of the CI. Some RSS-specific Extended Fields are also added based on derived and runtime data. Content Items are then placed on the Awareness Agent Triage Queue for processing.

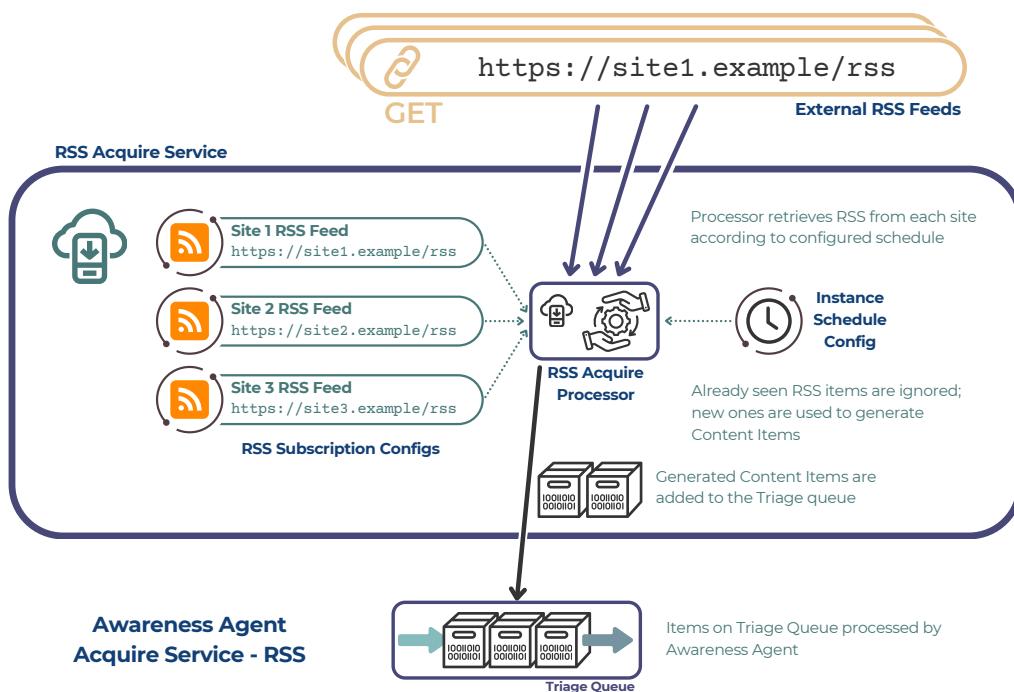


Figure 6.14: Awareness Agent Acquire Service for RSS

Listener Example – Slack Figure 6.15 illustrates the Slack implementation of the Acquire Service. Slack is an example of a *Listener* type acquire module, where a third party service pushes content to the service¹¹. The Awareness Agent exposes a public REST URL (in this case `/service/slack/acquire/events`) to which the Slack API sends events¹². Received events are converted to a standardised Callback Item, which is the raw JSON of the event request wrapped with some additional contextual information, and added to an

¹⁰ <https://www.rssboard.org/rss-specification#hrelementsOfLitemgt>

¹¹ <https://api.slack.com/apis/events-api> [<https://perma.cc/N83S-8PEU>]

¹² To activate this, the Awareness Agent must register as a bot in the Slack application and subscribe to channels to receive events

internal queue. Queued items are picked up by a processor component, which checks that the event is relevant – for example that it is a message rather than some other event such as a channel join notification – and creates a Content Item for the event if so. The processor component makes its own Slack API calls to obtain additional runtime information from the Slack API so that it can populate the CIs Extended Fields. This corresponds to the first two steps shown in the Content Item Creation Flow figure [6.10] and to the Data Generation figure [6.11].

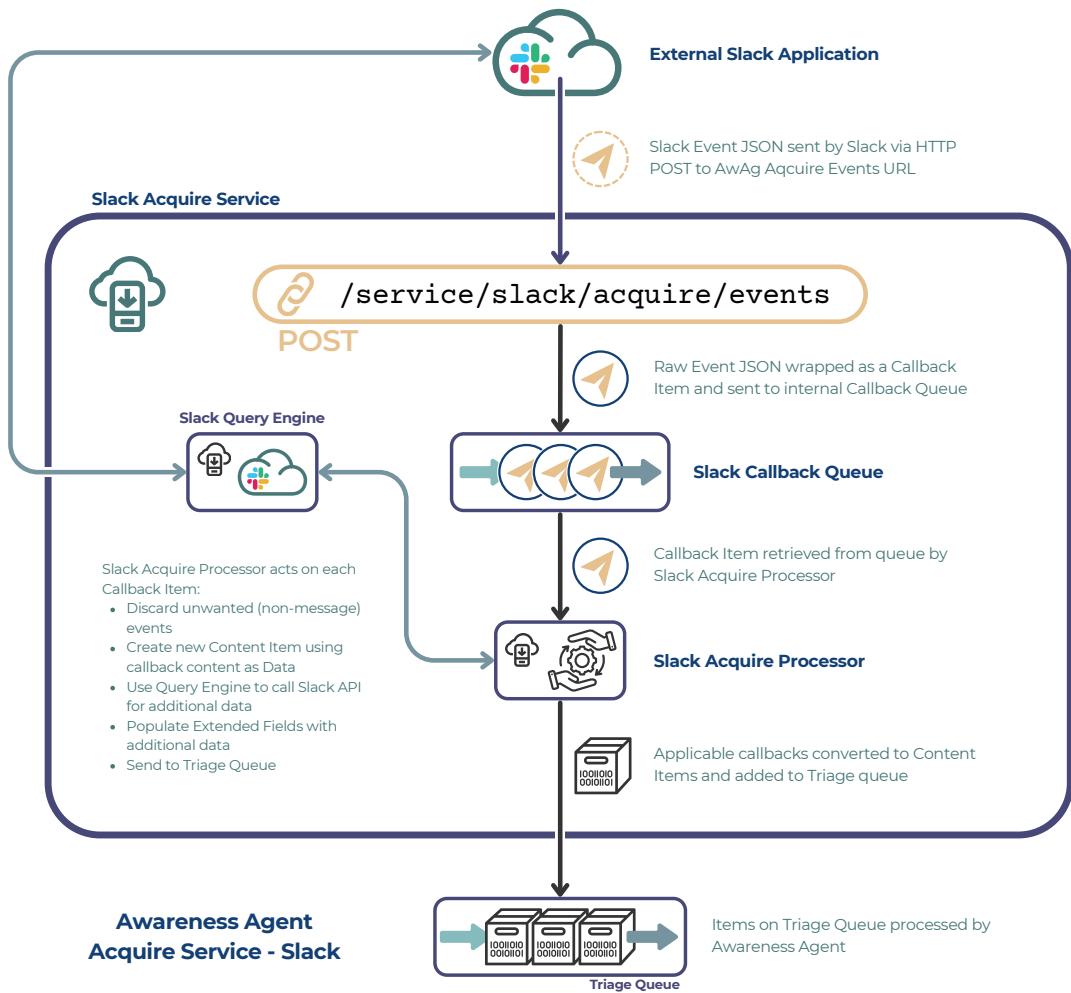


Figure 6.15: Awareness Agent Acquire Service for Slack

6.7.4.2 Augment Service

The Augment service adds augmentations to each Content Item, as modelled in Section 6.6.2.6. As with other services, Augment is modular, with the details of how it works left up to the implementer; all augmentations added to the Content Item follow a defined

structure that the implementation must conform to. The Engine component of the service will access each configured instance and perform an augmentation based on the instance type and details. The example illustration in Figure 6.16 shows three different types of Machine Learning based augmentations, although there is no requirement for any given augmentation to use an ML service – the implementation author could choose to perform a given augmentation entirely internally based on simple logic or filters for example.

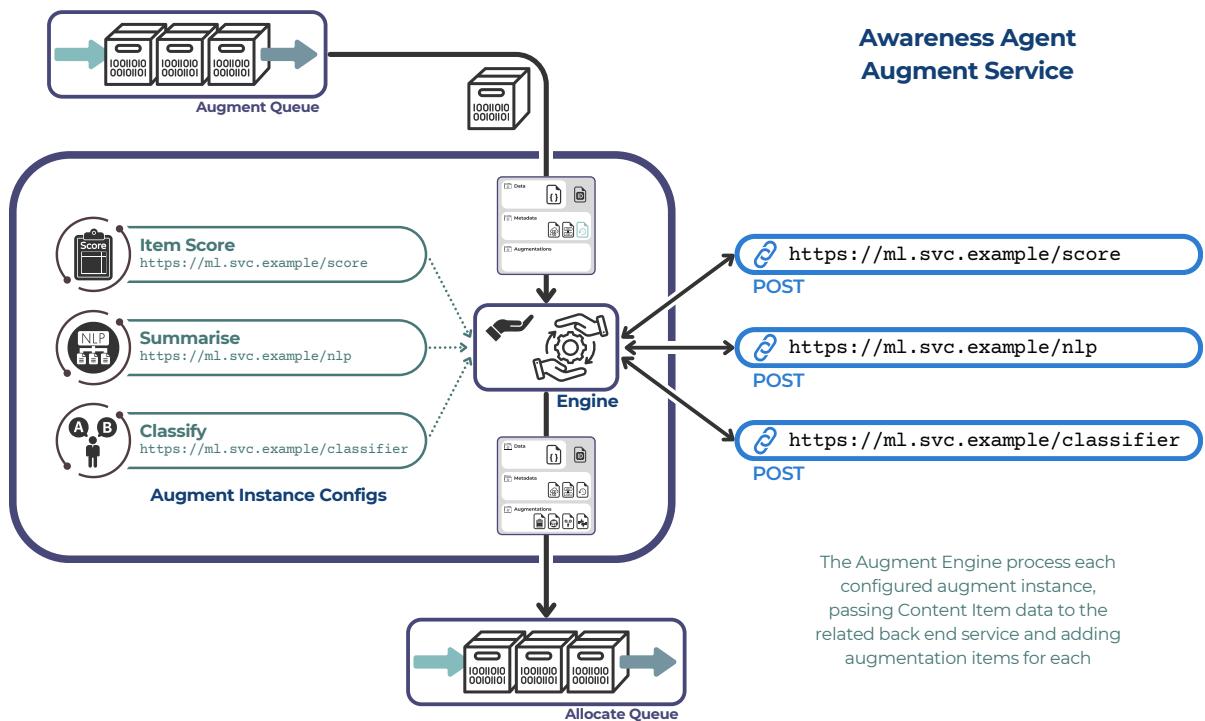


Figure 6.16: Awareness Agent Augment Service

6.7.4.3 Allocate Service

The *Allocate* component manages the distribution of augmented CIs within the Agent – primarily to the *Interact* component but also in theory elsewhere. It uses the augmentation information and its own internal state/configuration to determine the fate of each CI – which (if any) *Interact* service it is passed to, and when to do this.

The system is based on the principle of the selection of a current *Context* [6.6.3]. If a context is selected – either explicitly by the user or via some other mechanism such as a schedule, only content associated with that context will be sent to the *Interact* service.

To do this, the service uses a system of internal lockable queues and mapping elements, in conjunction with the Augmentations applied to each Content Item, as shown in Figure 6.17. The JSON mapping documents used by the service are shown in Figure C.2.

Mappings can be set up for internal queues based on a named augmentation; each value of the augmentation is mapped to one queue (Augmentation Queue Mapping). These queues are the same type of FIFO CI queue that is used for inter-service item passing, with the difference that they can be locked and unlocked by the Allocate engine. When the queue is locked, it can be added to but will not emit items; when it is unlocked it emits items as usual.

Context Queue Mapping is used to determine which internal queue(s) should be locked or unlocked based on a Context Selection Action. The Context ID of the selected context is mapped to all Queue IDs that should be unlocked when that context is selected. Context Selection Actions are passed by the Control component, in response to either a user Interact action/command [6.7.4.5], or to scheduled or other internal decision-making process of the Agent.

In the absence of any defined mappings, or if there is no context selected at a given moment, all CIs are sent to the Interact service (i.e. all internal Allocate queues are unlocked).

Queue Interact Mapping is used to determine to which Interact service content on a given queue should be sent to. Queue IDs are mapped to the applicable Interact Service IDs.

6.7.4.4 Exchange Service

For this architecture, the Exchange service uses a simplified approach to AXP [6.6.8]. We opted to use MQTT¹³ as a basic message broker. With this approach an Awareness Agent would publish Content Items to an MQTT Topic via a Broker. Other agents, which subscribe to that topic would receive the published CIs.

An MQTT payload can be any binary data¹⁴, so the CI JSON is serialised to binary format

¹³<https://mqtt.org/> [<https://perma.cc/4L63-YZUB>]

¹⁴<https://www.emqx.com/en/blog/how-to-process-json-hex-and-binary-data-in-mqtt> [<https://perma.cc/H4YL-DJKV>]

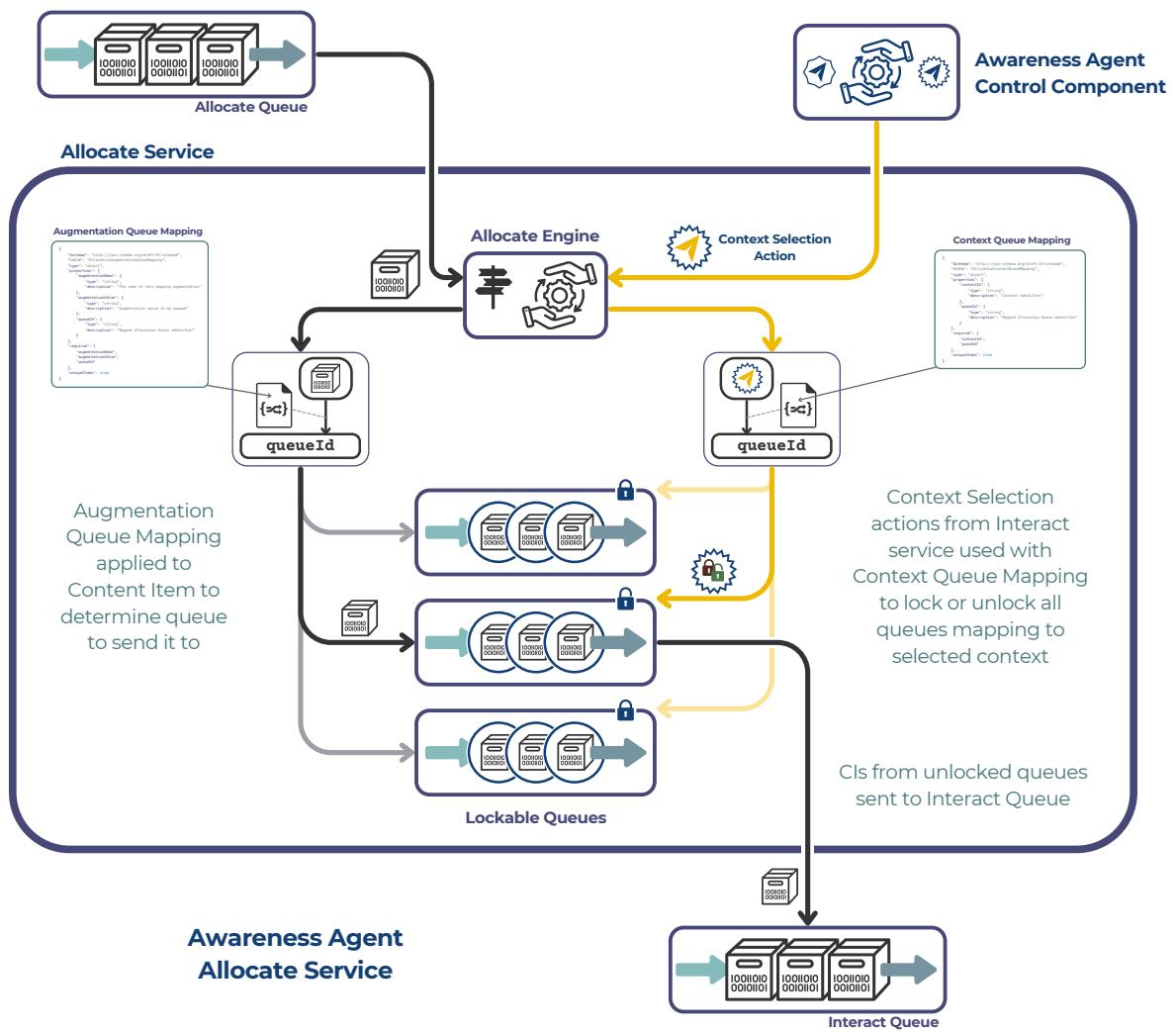


Figure 6.17: Awareness Agent Allocate Service

for transmission and deserialised by the receiving Awareness Agent.

We have not developed a detailed architecture for Exchange and AXP in this thesis, but note it as a topic for future work [10.4.7].

6.7.4.5 Interact Service

The Interact service has three main responsibilities:

1. Publishing applicable content to the user
2. Handing content-based interactions
3. Receiving administrative and other commands

It is possible that these functions could be partially split across different Interact implementations, for example by having a dedicated Interact implementation for administrative commands, but in our documentation we will consider only a monolithic implementation. Additionally, different implementations could be used to perform the same functionality in different ways, such as supporting interactive functionality via both integrated and standalone modules.

Publishing Content for publication always comes from the Interact Outbound Queue, placed there by the Allocate service. Note that the current architecture supports only a single content publication Interact implementation, as the active publisher will remove items from the queue. Alternative approaches would be compatible with the overall design but are out of scope for this documentation.

The content publication processor in the service can decide what to do with each CI based on its internal configuration and the information contained within the CI, such as the augmentations. This includes the construction & appearance of the published items, where they are sent, *if* they are sent, and when they are sent. These are all internal details for the implementation, which we will not discuss in this architecture.

Content Interactions When the user interacts with content that it published, the Interact service is expected to handle these interactions and take appropriate actions. Again, both the mechanism and process for handling this is a matter for the individual implementation. Examples of interactions include removing or moving content, giving scoring or classification feedback, sharing an item etc.

Commands The user will also need to issue commands to the agent, for example to change its configuration, add or remove data sources, supply it with context information and so on. The command implementation is *also* a matter for the individual implementation. Commands are distinct from content interactions because they are not associated with individual content items. Different interfaces could be used for different types of commands – for example implementing configuration and content administration via a web interface, while context switching could be done via a mobile application.

6.7.5 Interact Example – Slack

6.7.5.1 Slack as a User Interface

Figure 6.18 illustrates the Slack implementation of the Interact Service. The paradigm for this implementation is that Slack serves as the user interface and notification infrastructure, where content is published as Slack messages to Slack channels, with channel selection based on augmentations. This is implemented via Slack Apps¹⁵. This approach means that many of the interactive features can use standard Slack functionality, such as allowing the user to read content on mobile using the standard Slack mobile app, having notification on mobile devices enabled for specific channels and similar features. While we have already discussed using Slack as an Acquire data source [6.7.4.1], the approach here is to use a standalone Slack workspace to serve as the UI rather than a source for content.

As well as consuming content, the user interacts with custom components in messages posted to this Interact workspace to perform tasks such as reclassifying and giving feed-

¹⁵<https://api.slack.com/docs/apps> [<https://perma.cc/HHN4-LQQD>]

back. Similarly, a command line control interface is provided using Slack Slash commands¹⁶.

The implementation has four modes of interaction between Slack and the Agent:

- Content Item publication via Slack API call (outbound from agent to Slack)
- Content Interactions via callback to Agent Interact API (inbound from Slack to agent)
- Commands via callback to Agent Interact API (inbound from Slack to agent)
- Asynchronous responses to commands and interactions via Slack API call (outbound from agent to Slack)

The implementation has a handler process for each of these modes. Information flow is one way for all except the Command processing; Slack Slash commands can be sent a synchronous response, so for fast running commands such as requesting a list of configured RSS feeds (for example), the agent can send a synchronous response. Some commands may take longer – such as asking the agent to perform a model creation action, in which case the agent can send an immediate acknowledgment response, and later send an asynchronous response.

6.7.5.2 Content Formatting

We introduced Content Item Formatters in Section 6.7.2, with our reference design having both Text and Slack implementations [Figure 6.9]. Our Interact example uses the Slack formatter to generate the content to output to Slack from each Content Item. Figure 6.19 highlights where the Slack Content Item Formatter fits in the Interact Service for Slack.

Figures 6.20 and 6.21 show examples of the content that is published to Slack (with and without explanatory annotations). In this case the messages are those generated by the User-Directed ML service [6.7.6], which is a specific implementation of Augmentation and Interact components.

¹⁶<https://api.slack.com/interactivity/slash-commands> [<https://perma.cc/76S8-RE7V>]

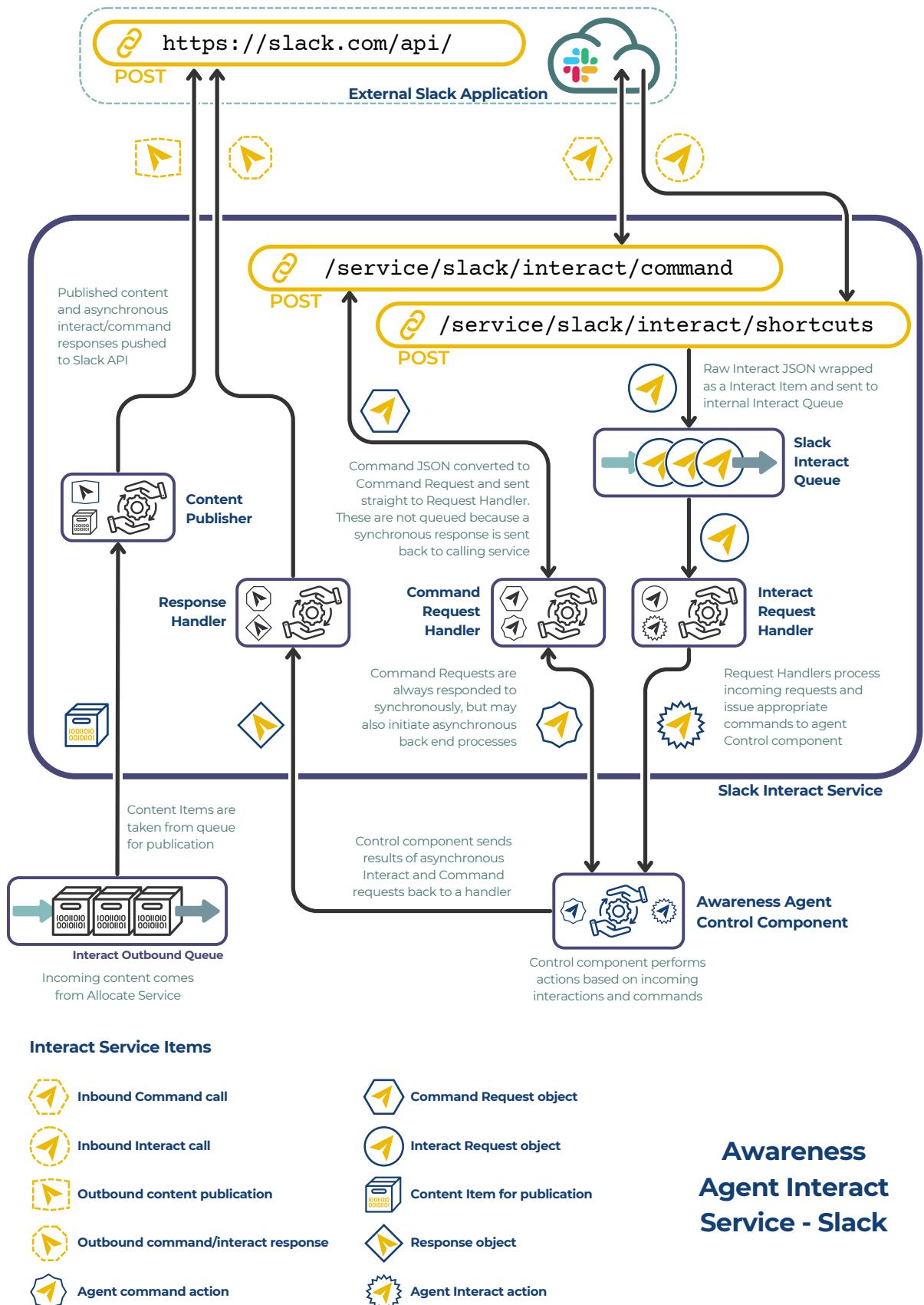


Figure 6.18: Awareness Agent Interact Service for Slack

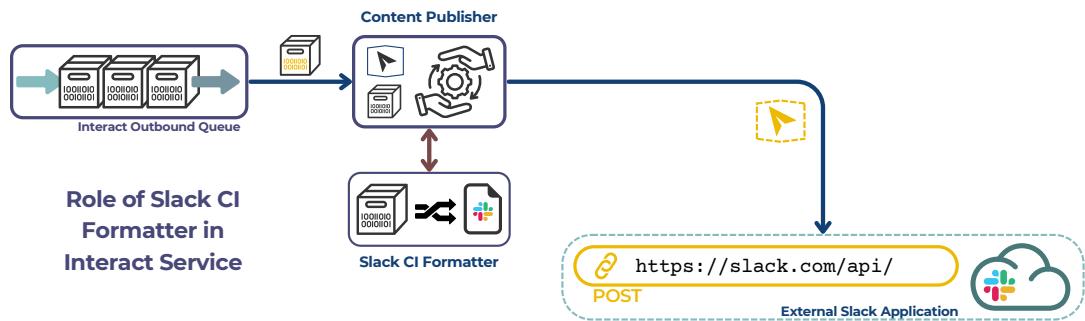


Figure 6.19: Slack Content Item Formatter within Interact Service

Interact messages on Slack

Awareness Agent APP
[RSS item from Technology | The Guardian](#) on [UK news | The Guardian <rss>](#)

Britons urged to dig out unwanted electricals to tackle copper shortage Tue 8 Oct 2024 06:01

Britons are being urged to recycle unused electrical items such as cables and old tech to help address a growing copper shortage. A study found that UK households hold around 823 million unused items containing significant amounts of copper, enough to cover 30% of the copper needed for the country's decarbonisation goals by 2030.

[Remove](#) [...](#)

Classification Summarisation quality

Awareness Agent APP
 Slack message from Charlotte Walker to [ctg-team](#) on [Borchester Software](#)

Fri 26 April 2024 16:41

Adam & Oscar - we need to get together to finalise Monday's site visit at UDG before close of play today.

[Remove](#) [...](#)

Classification Summarisation quality

Awareness Agent APP
 Slack message from Adam Macy to [ian-chat](#) on [Family](#)

Fri 26 April 2024 17:34

Hey Ian, I'm stuck in a meeting with Oscar and Charlotte. Might be late tonight.

[Remove](#) [...](#)

Classification Summarisation quality

Figure 6.20: Example of Slack Interact messages

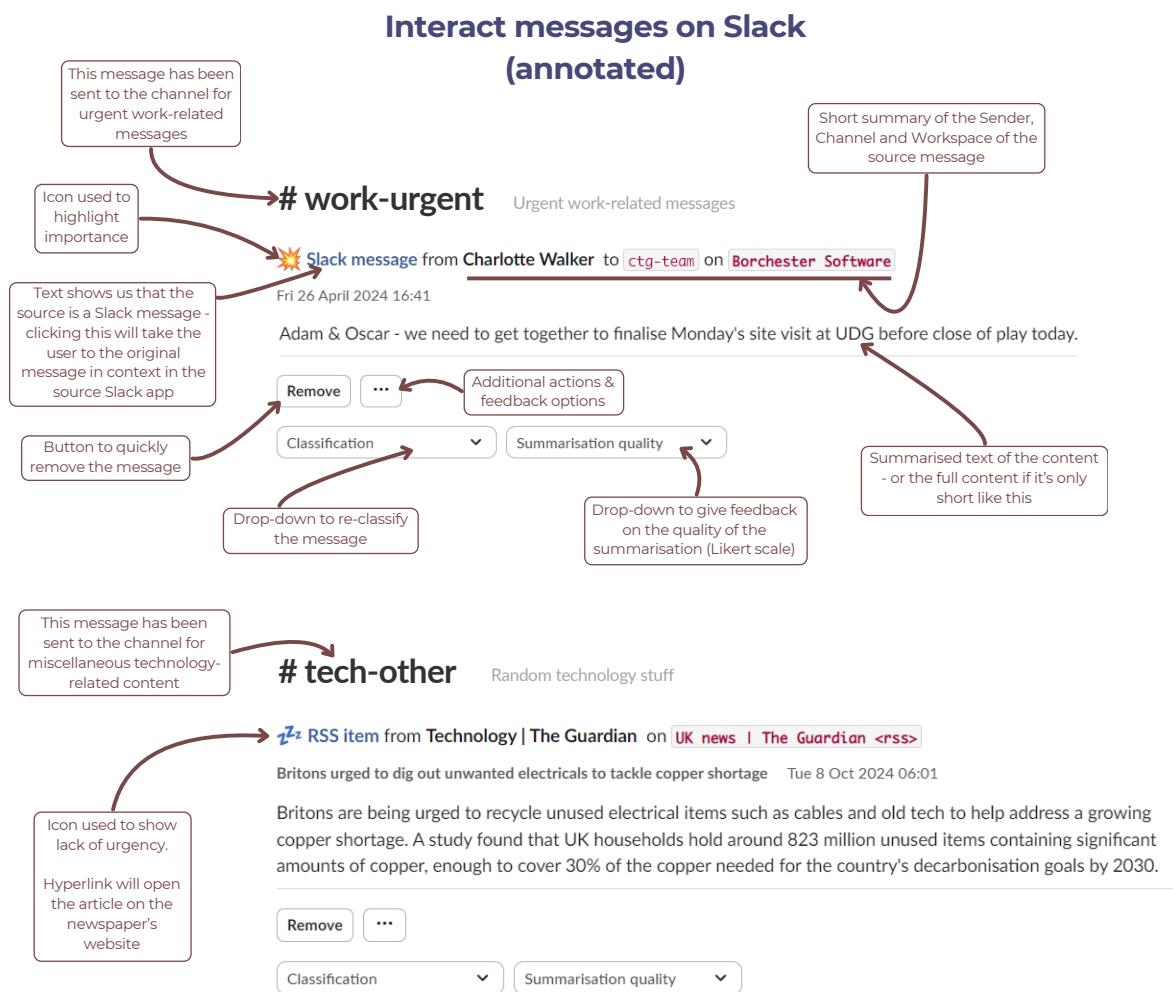


Figure 6.21: Example of Slack Interact messages (annotated)

6.7.6 User-Directed ML

User-Directed Machine Learning, introduced in Section 6.4.6 can be considered an application within an application, and comprises of the following main features:

- Administrative functionality allowing the user to manage their own ML models
- Specific augmentations that are added by an Augment implementation that calls the UD-ML API
- An Interact implementation that represents these augmentations by directing content to specific output channels.

Our reference architecture uses a Slack-based Interact service that uses User-Directed ML models to direct content items to different Slack channels based on how they have been classified by the back end UD-ML service.

6.7.6.1 Model Lifecycle

User-Directed ML is based on the idea of user-owned and managed, focussed Machine Learning models that exist for a single function. The initial concept is for a classifier based system, with one dedicated classifier model created per ‘task’. For example, the user may wish to have a model that classifies content about work from non-work, or urgent from non-urgent – or something more specialised such as identifying messages related to cycle racing.

The UD-ML application allows the user to create and destroy models with simple commands; it then associates those models with specific output channels and sets up the Augmentation and Interact configurations required to drive the process. When a model is first created it is completely untrained; the user is given a simple bulk interface to allow them to bootstrap each model that they create by applying manual classifications to content. Ongoing training and refinement of the model is performed within the Awareness Agent UI itself, with the user being able to correct mis-classifications and feed this back to the model. It is an assumption (that we will be testing in our research) that this process can lead relatively quickly to effective and usable models.

No model is expected to have an indefinite lifespan. Some may be short-lived by design (such as a model designed to locate content about a time-limited topic such as a currently running major sporting or political event), but others may drift over time and become less effective as historical training data becomes less applicable to current needs. The design intent of User-Directed ML is to make it easy for the user to create, dispose of and replace these models, with the necessary back end ‘plumbing’ being handled for them.

6.7.6.2 User-Directed ML Request

A User-Directed ML Request Item is a simplified request to the User-Directed ML service that can be used for either training, classification, information retrieval or administration. Figure 6.22 shows the main types of logical request and response objects that are passed to and from the User-Directed ML service. In the case of a training or classification type item, it contains extracted data from a Content Item representing the content of the item. As an example, we show in Figure 6.23 how a UD-ML Classification Request Item is generated from a Content Item. This process uses the Standard Fields feature of the Content Item [6.7.1.1] to extract the content needed for a User-Directed ML classification or training action. In this way the internal structure and type of the content item is transparent; UD-ML does not need to know this information in order to construct a request.



Figure 6.22: Key of User-Directed ML Request & Response Objects

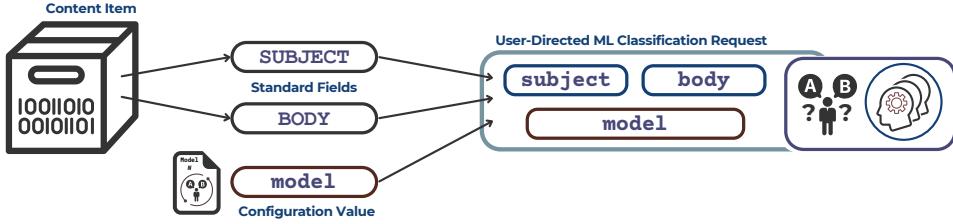


Figure 6.23: Generation of User-Directed ML Classification Request from Content Item

6.7.6.3 User-Directed ML Augmentation Process

The User-Directed ML augmentation process is just one of potentially many augmentation implementations that are applied to a CI by the Awareness Agent as it passes through the Augment service. Figure 6.24 illustrates this process in our reference architecture. In this case, when the Augment Engine accesses the UD-ML Augment Module Configuration, it calls the UD-ML Processor to generate the augmentations. This processor extracts information for each UD-ML model from the module configuration, and processes each in turn. To do so, it generates a Classification Request from the module config and the Content Item, and passes this to the back end service via an API call, receiving a Classification Response back.

This Classification Response is used to add an augmentation to the CI – Figure 6.25 shows how an augmentation Item of type Single Classification [6.6.2.6] is populated using this object. This takes the classification value selected by User-Directed ML, the descriptive text associated with that classification value, and the list of possible available classifications, and combines it with configuration and state information held by the Augmentation Service to populate the Augmentation Item. This process is repeated for each configured model, so that an Augmentation Item is added to the CI per model.

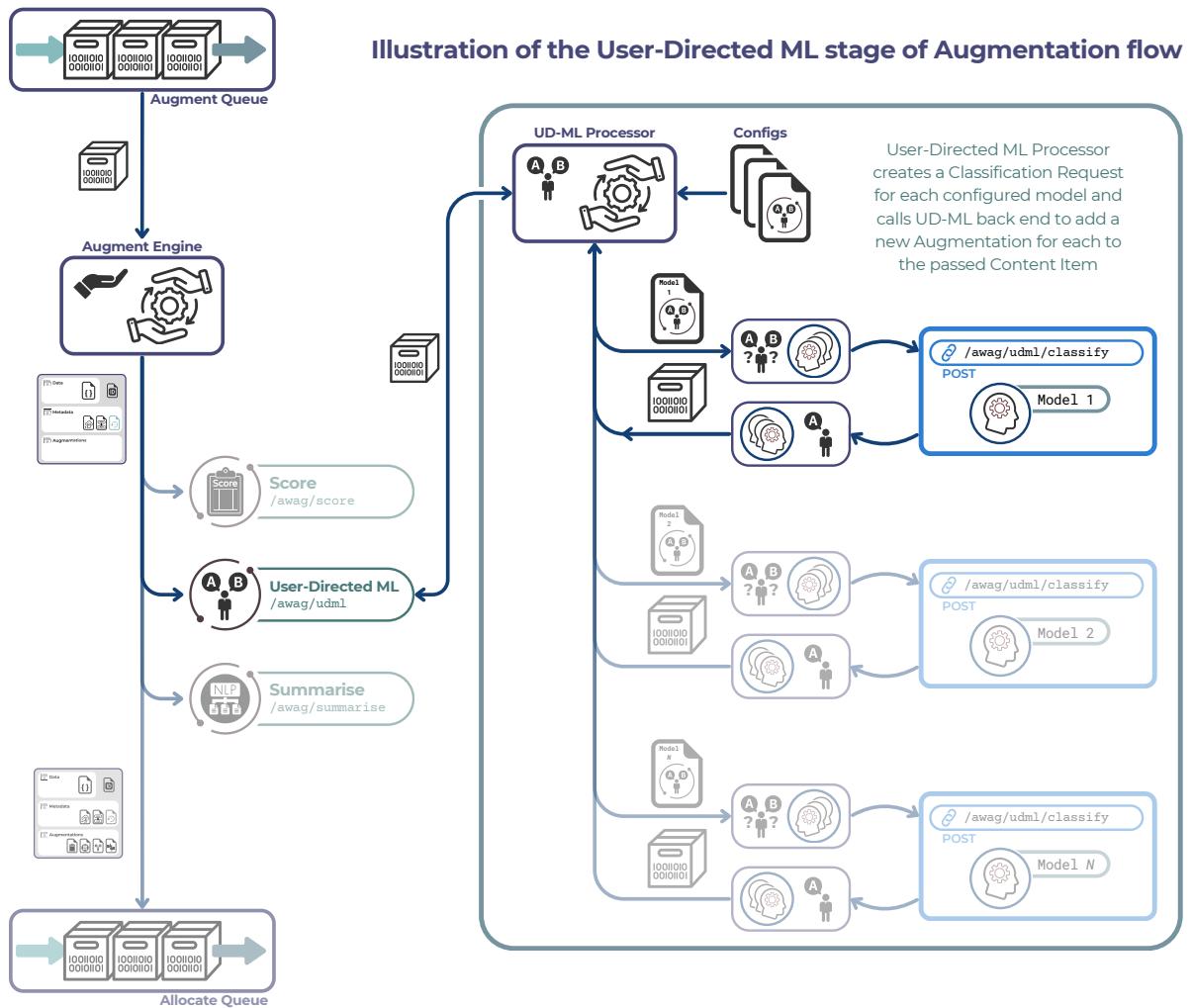


Figure 6.24: User-Directed ML stage of Augmentation flow

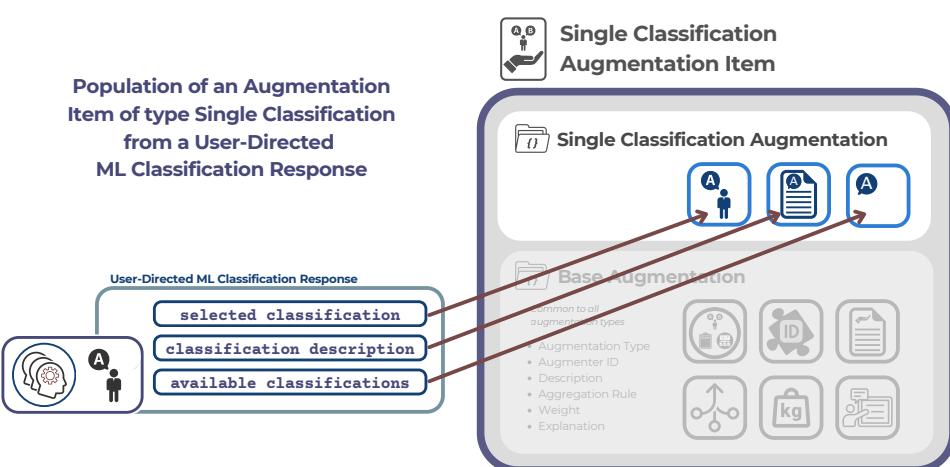


Figure 6.25: Population of Augmentation Item from UD-ML Classification Response

6.7.6.4 User-Directed ML Interaction Process

This section describes the interaction process and user interface for User-Directed ML. Screenshots illustrating this can be found in Section 6.9.3.

Publishing

This design uses the concept introduced in Section 6.7.5, to publish content to a Slack workspace that provides the Awareness Agent UI. In this case, we take the approach of using multiple Slack channels to publish classified items to, with the user being able to then customise the notification behaviour of those channels. The process of publishing User-Directed ML content in this way is shown in figure 6.26.

The model management process maintains a list of configured models and channel mappings in the UD-ML Interact module configuration. This is used to identify the augmentations in the Content Item and direct content to the appropriate channel. Each configured model is processed in turn, and the associated augmentation retrieved from the Content Item being processed. This is used to construct the name of the Slack channels that the item should be published to, with one channel per model-classification combination.

For example, say we have a model called “model1”, and this has available classifications “classificationA”, “classificationB” and “classificationC”. This corresponds to three channels, named: “model1-classificationA”, “model1-classificationB” & “model1-classificationC”. If a given CI has an augmentation for model1 that is classified with value ‘classificationB’, it will be published to Slack channel #model1-classificationB in the Interact workspace.

Alternatively, a model called “urgency” might have available classifications “urgent” and “not”. An item that the model has classified as being urgent would be sent to channel #urgency-urgent and conversely, a non-urgent item would go to #urgency-not. The user can set notification settings on #urgency-urgent in Slack to always notify them in their mobile app, and those for #urgency-not to never do so.

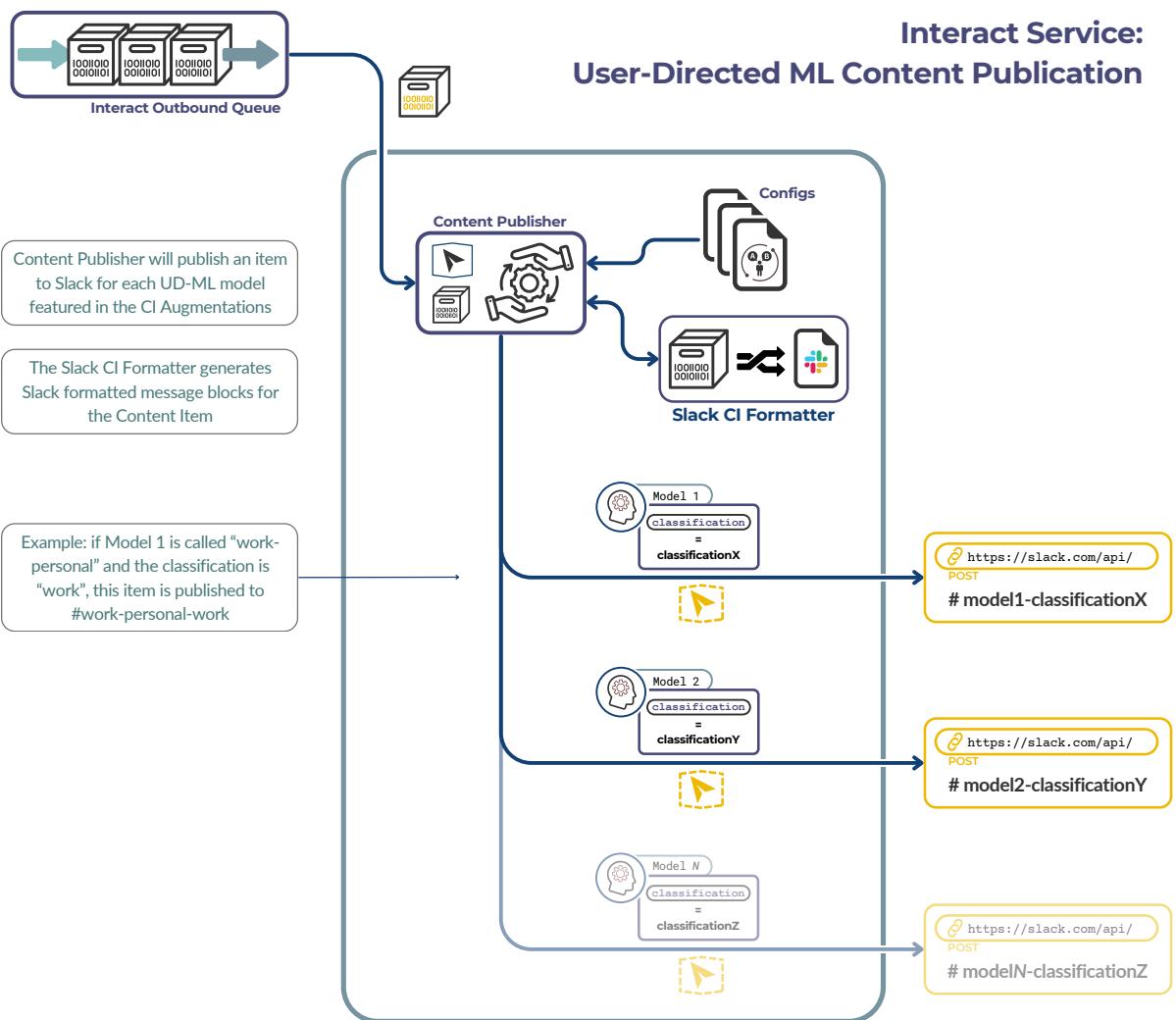


Figure 6.26: User-Directed ML Content publication in Interact Service for Slack

Feedback

An essential part of the User-Directed ML process is feedback and correction by the user. As was illustrated in Figure 6.21, published UD-ML Slack items have interactive features to facilitate this. The user is able to change how an item has been classified for example. The training process is shown in Figure 6.27. Figure 6.28 shows how the User-Directed ML Training Request is generated from a combination of the current Content Item data and the user input.

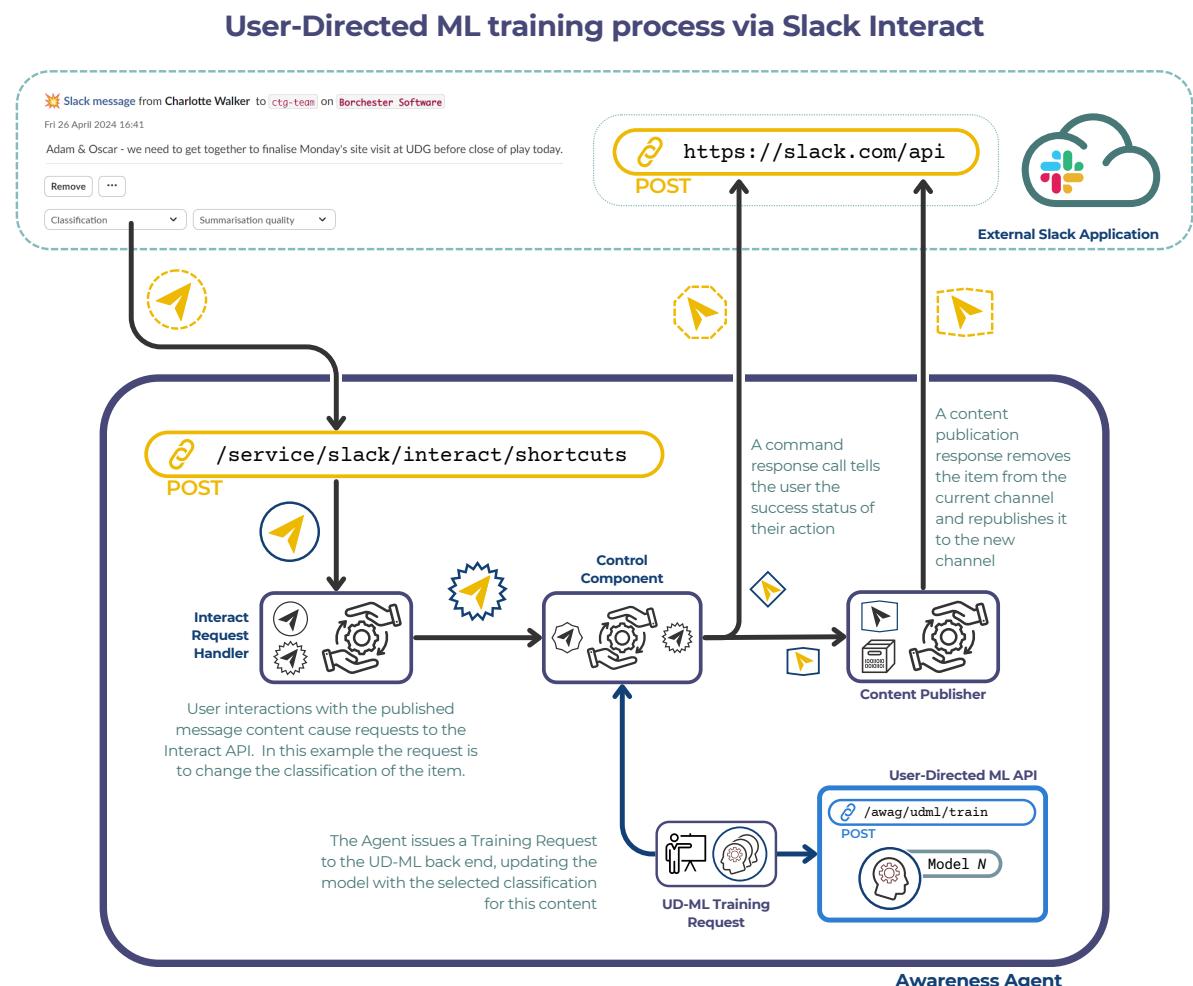


Figure 6.27: User-Directed ML Training process example in Interact Service for Slack

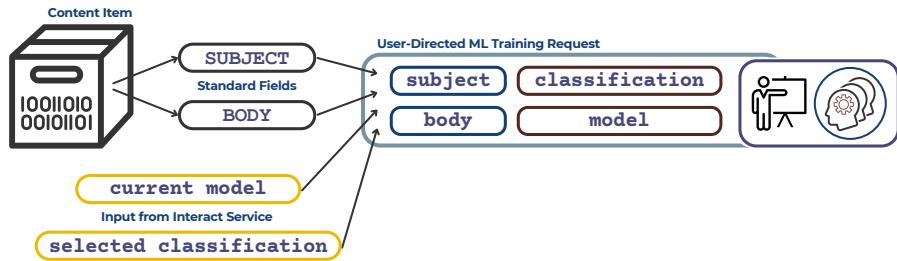


Figure 6.28: Generation of UD-ML Training Request from CI & Interact data

The following UD-ML corrective actions are available in the reference design:

- Reclassify – user selects a different classification from a drop-down, which moves the item to the correct channel and issues a training request to the back end service for the corrected classification
- Remove – user clicks a button to delete the item from the channel¹⁷
- Remove and train – as above, but a training event is also generated

6.7.7 Autonomous Operation

As described in the Model [6.6.9], the Agent Autonomy Service integrates with the service/queue structure of the agent. Specifically, items are directed by the Allocate Service to AwAgAS, and appropriate Command Request and Interact Request items [6.7.5.1 & 6.18] are also sent to the service. AwAgAS outputs to the user by placing items on the Interact Outbound Queue; we have defined a special type of Content Item – the Autonomy Operation Item – to represent this. To support the taking of action via continual CI ingestion, AwAgAS must record relevant information in its own data store, allowing periodic actions to be taken on aggregated data.

¹⁷Slack supports content deletion, but this requires more mouse clicks than a dedicated button

6.8 Implementation

6.8.1 Core Awareness Agent

6.8.1.1 Language & Platform

We decided to build the core Awareness Agent application in Java – initially Java 8, but later moving to Java 9, using Eclipse IDE¹⁸ as a development environment. We built this as an Apache Tomcat¹⁹ web application, using Apache Juneau²⁰ to provide REST server & client functionality and to handle serialisation tasks. During the course of the project we deployed to FreeBSD²¹ and Ubuntu Linux²² servers.

Our choice of platform was influenced by our initial design concepts – we knew that we would be needing a service based system that could both expose and consume REST web services and present a web interface to the user or administrator. Our prior experience of developing similar applications with Java, Tomcat and Juneau led us to select this combination.

We defined Slack apps for Acquire, Interact and Simulate components using manifests listed in Appendix C.2.

6.8.1.2 Project Structure

We organised our Java code in Eclipse in the following structure, with a total of four Eclipse projects under a root awareness-agent parent. We used Maven²³ as the project build and management tool, using the AntRun plugin²⁴ for build automation.

Although we deployed to Tomcat, we used Jetty²⁵ as an integrated servlet container for development & debugging.

¹⁸<https://eclipseide.org/>

¹⁹<https://tomcat.apache.org/>

²⁰<https://juneau.apache.org/>

²¹<https://freebsd.org/>

²²<https://ubuntu.com/>

²³<https://maven.apache.org/>

²⁴<https://maven.apache.org/plugins/maven-antrun-plugin/>

²⁵<https://jetty.org/>

6.8.1.2.1 Project: awareness-agent-platform

The Platform project houses all the core code of the Awareness Agent, with the output being a library of code that could be used by any agent implementation, under a top-level package `net.parse.ou.awarenessagent.platform`²⁶. Many objects in the platform are abstract classes, with the expectation that concrete implementations will exist in other projects. This comprises of the following general packages:

- `async` – Items associated with asynchronous or queued operations, including abstract base queue objects, callback classes, service request and result objects
- `config` – Top-level agent configuration
- `core` – Core abstract objects, including Content Item, Exchange, serialisable objects
- `data` – Data access code including base REST client, awagdata & awagml clients/objects, object store client
- `listener` – Not an incoming service listener as the name might suggest but instead an implementation of `ServletContextListener` for launching queues and services at startup
- `ml` – Client code for accessing ML services (awagml and OpenAI)
- `modules.common` – Common code supporting modular components, including clients & helpers for REST, MQTT, RSS and Slack
- `modules.genus` – Base code for modular components, organised into the sub-packages: `acquire`, `augment`, `control`, `evaluate`, `exchange`, `interact` and `simulate`. These include concrete classes for queues such as `AugmentQueue` and configuration classes such as `AugmentModuleConfig`, as well as abstract base classes for engines such as `AugmentEngineInstanceBase` (which itself extends `ModuleEngine`). Any implementing application is expected to extend these base engine classes; the platform's listener service will automatically start up concrete classes that extend `ModuleEngine` at runtime.

²⁶The author personally owns the `parse.net` domain, which is used for naming in various places

- `scheduler` – Schedule management
- `state` – Classes for managing the application state (service activation etc.)

The output of this project is a jar file²⁷ containing this code, intended to be included in other projects.

6.8.1.2.2 Project: awareness-agent-service

The Service project is a *reference implementation* of an Awareness Agent, under a top-level package `net.parse.ou.awarenessagent.service.impl`. This project predominantly contains concrete classes extending the Platform code with implementation-specific functionality. In some cases there is negligible such code, but in others this is extensive.

6.8.1.2.3 Project: awareness-agent-application

The Application project contains very little code; it is instead intended to construct and deploy the actual Awareness Agent web application using the `maven-war-plugin`²⁸. This builds a WAR file²⁹ containing the Platform and Service libraries, other supporting libraries and the supporting set of files to run the application. This is the file deployed as the Awareness Agent.

6.8.1.2.4 Project: awareness-agent-test

This project contains test and debug code.

²⁷<https://docs.oracle.com/javase/8/docs/technotes/guides/jar/jarGuide.html> [<https://perma.cc/6H92-T2PK>]

²⁸<https://maven.apache.org/plugins/maven-war-plugin/> [<https://perma.cc/7C26-7K5L>]

²⁹https://docs.oracle.com/cd/E19199-01/816-6774-10/a_war.html [<https://perma.cc/V8CY-9SMN>]

6.8.2 ML Service (`awagml`)

We built the ML Service³⁰ in Python 3, using Flask³¹ as the framework for exposing web services. We used the Python Flask boilerplate from Idris Rampurawala³² as the basis for our code.

There are two types of method exposed by `awagml`: model and data management methods for managing available models and classifications, and a classifier method.

We are publishing the source code for the ML Service – see Section 6.9.4.

6.8.2.1 Classifier

The classifier uses `scikit-learn` [Pedregosa et al., 2011] [Buitinck et al., 2013], a set of Open Source Machine Learning tools. We added support for two classifiers in our code: `MultinomialNB`³³ and `SGDClassifier`³⁴. Bearing in mind that the academic focus of our research was *not* the study of the ML implementations themselves, the development process had gone through a number of testing iterations to find something that was ‘just good enough’. In the end we settled on a linear support vector machine (SVM) classifier with SGD learning³⁵ using the `SGDClassifier`. Our testing had showed that this consistently outperformed the Naive Bayes classifier (`MultinomialNB`). We configured this with a `modified_huber` loss function, `l1` penalty, an alpha of 1×10^{-5} , random state of 42, and maximum iteration of 5 – as with the classifier itself, we had settled on these values after a process of iterative testing. Having established that these performed suitably well for our needs, we intentionally did not change these further during the course of our research.

We reiterate that the purpose of our research was not to test which classifiers with what settings worked best, but rather to explore how the classifiers could be incorporated into our design. Thus when we had established a preferred classifier and configuration through initial testing, we continued with that and did not further test any alternatives.

³⁰Development Log [D] entry 2022-01-04

³¹<https://flask.palletsprojects.com/> [<https://perma.cc/XC37-Q3ZT>] [<https://perma.cc/EG4V-2FUD>]

³²<https://github.com/idris-rampurawala/flask-boilerplate>, open-sourced under the MIT License

³³https://scikit-learn.org/1.0/modules/generated/sklearn.naive_bayes.MultinomialNB.html

³⁴https://scikit-learn.org/1.0/modules/generated/sklearn.linear_model.SGDClassifier.html

³⁵<https://scikit-learn.org/1.0/modules/sgd.html>

6.8.2.2 Model Management

The other methods exposed by awagml are for model management. The service maintains a directory hierarchy local to the server, with each client Awareness Agent having a top-level container directory identified by its agentId. Within this the service maintains subdirectories for each User-Directed ML model and the classifications within those, each containing a set of training text documents. This structure is illustrated in Figure 6.29.

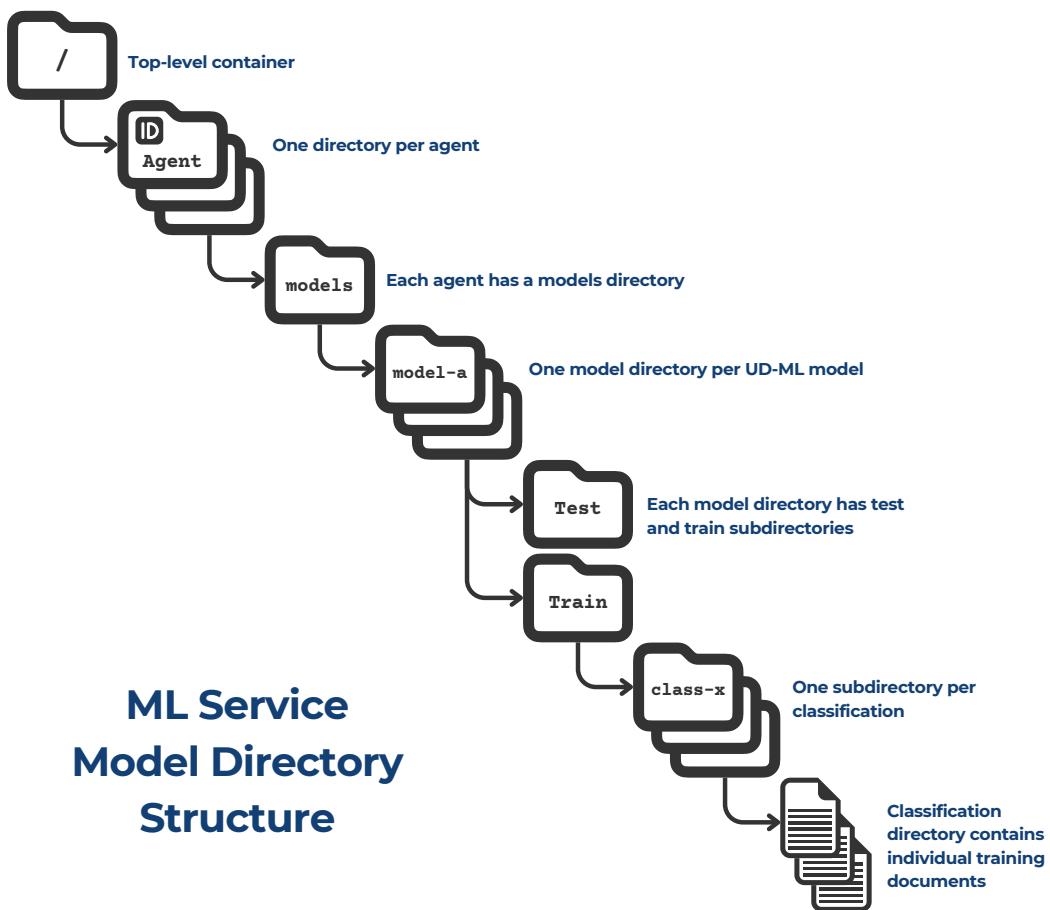


Figure 6.29: ML Service Model Directory Structure

The ML Service maintains this structure in accordance with administrative actions, adding or removing model and classification directories as needed. Each training action results in a text document containing the new training item being placed in the appropriate classification directory. The ML Service performs a fit action on each model on startup and when classification documents are added to a model, utilising the contents of the appropriate train directory.

6.8.3 Data Service (awagdata)

The Data Service is also implemented in Python Flask, and while logically distinct from the ML Service was also initially implemented within the same Flask application as the ML Service, using a different application route to distinguish it. Only authentication and configuration code is shared between awagml and awagdata.

We are publishing the source code for the Data Service – see Section 6.9.4.

Table 6.3 shows the high-level routes implemented to support core Awareness Agent functions. This table also includes references to relevant entries in the Development Log [D.2] where applicable.

Table 6.3: Data Service Routes – Core Awareness Agent

Route	Function	Reference(s)
/data/class	Classification action recording, feedback, training	[D] 2023-04-14
/data/flow	Flow Monitor recording	[D] 2023-06-25
/data/maintain	Data management tasks such as purging agent data	
/data/misc	Utility functions such as tag retrieval	
/data/summ	Recording of summarisation requests and feedback	[D] 2023-04-14 [D] 2023-04-16

Table 6.4 shows the high-level routes that are also implemented to support the Synthetic Evaluation function [Chapter 8] and study analysis. This table also includes references to relevant entries in the Development Log in Appendix D.

Table 6.4: Data Service Routes – Simulate, Evaluate, Study

Route	Function	Reference(s)
/data/chat	OpenAI chat requests using fine-tuned models	[D] 2024-02-23
/data/eval	Evaluation data recording, management & retrieval, evaluation failure recording	[D] 2023-06-15 [D] 2023-09-07 [D] 2023-07-11
/data/fixit	Data recovery and reconstruction	[D] 2024-02-18
/data/gentrain	OpenAI training (fine-tuning) item generation	[D] 2024-02-18
/data/reporting	Prepare and update study data for reporting/analysis	
/data/sim	Simulated data generation and query	[D] 2023-04-21
/data/stats	Statistics generation for the study	[D] 2024-03-14 [D] 2024-05-27
/data/subsets	Subset creation and management	[D] 2024-02-23
/data/train	Dataset management, OpenAI fine-tuning management & execution	[D] 2023-12-08 [D] 2024-05-02

6.8.4 Persistent Object Store

For object storage [6.6.7] we originally selected a commercial implementation of Apache CouchDB³⁶ for persisting JSON objects – initially only for configuration object storage but later also for many other objects. Following a change in the terms of the commercial provider, we elected to replace this part way through the project with our own implementation of object storage.

We considered using one of the open source alternatives for object storage but elected to write our own implementation as a Python Flask application with a SQLite back end. Our rationale for taking this approach was that we wanted a lightweight and fast database, installed locally on the Awareness Agent server. This not only allowed for very low latency times, but also made it practical and economic to store large amounts of data. Because it was backed by a local database, this approach also gave us the ability to execute SQL queries to access data, giving us a powerful alternative to using an API. Additionally, much

³⁶<https://couchdb.apache.org/>

of the code base for the Flask application was shared with the other Awareness Agent back end services.

We named this object software Simple Persistent Object Store (SPOSS) and made it freely available under an Apache 2.0 license at: <https://github.com/revisionist/python-apps>.

6.8.5 Awareness Agent UI – awagUi

We created a standalone UI using Angular³⁷, served using Apache 2.4³⁸ on FreeBSD & Linux.

This was initially created to support user input in relation to the study of synthetic evaluation [9.2.5.4], but could also partially be considered an additional Interact user interface. See Development Log [D] entry 2023-10-02. Data for awagUi is supplied by REST calls to the awagdata interfaces documented in Section 6.8.3.

This UI is shown in the Study chapter in Section 9.4.

We are publishing the source code for awagUi – see Section 6.9.4.

6.9 Technical Status

6.9.1 Implementation Status & Variance From Architecture

As we noted previously, the System Model [6.6] and Architecture [6.7] are the result of knowledge gained from an iterative design and testing process, so the content of these partially follows from the Implementation work, rather than entirely preceding it. Because of this, there are some gaps between our settled design and what has been implemented – either because we saw on final reflection that a slightly different design would work better, or simply because we lacked the time to completely implement some aspects in our prototype. This section discusses those gaps that exist.

³⁷ <https://angular.io/>

³⁸ <https://httpd.apache.org/>

6.9.1.1 Allocate Service and Context

While our original design thinking envisioned a significant role for the Allocate Service, as shown in Architecture section 6.7.4.3, we implemented this in a much more simple way for our prototype. This was driven by time and scope constraints: we designed our study [9] to focus on the User-Directed ML concept to keep its scope manageable. Because of this we did not need to implement any method of contextual awareness or time-based queuing to support the study, so the Context Queue Mapping from the architecture remains theoretical at this stage.

6.9.1.2 Interact Service Plurality

The concept of user interaction outlined in sections 6.4.4 and 6.7.4.5 envisions a system where multiple Interact services exist to fulfil different types of interaction needs, such as reactive or proactive interaction. Again due to time and scope constraints, we developed only a single Interact implementation, the Slack Interact Service [6.7.5]. There remains a possibility to increase this, in conjunction with completing the Allocate work described above. The addition of the awagUi [6.8.5] showed us a way that this could be done.

6.9.1.3 ML and Data Services

The ML and Data services – awagml & awagdata – grew organically during our iterative process of development³⁹ and are not structured exactly as depicted in the architecture. We originally created both of these services for specific, limited tasks before realising a more significant expanded role for them. Because of this, Figure 6.22 represents an idealised abstracted structure of the User-Directed ML request-response objects. The service as implemented passes the same information but without these explicitly defined objects. Future work should adapt the ML service to use these structures.

³⁹See Development Log [D] entries 2022-01-04 & 2023-04-14 and later

6.9.1.4 UD-ML Modular and Sequential Integration

Our concept of ML Modular & Sequential Integration for User-Directed ML described in Section 6.6.6 was not included in the implementation that we tested. While this would be compatible with the UD-ML architecture that we developed, we felt that we did not need to include this functionality in the studied implementation and it would have significantly expanded the scope.

6.9.2 Gap Analysis

Table 6.5 shows an assessment of each of the mostly functional requirements originally described in Table 6.2 as of the end of prototype development. We assessed how well the design and prototype implementation respectively fulfil the requirements, and have categorised these as:

- **Yes** – the requirement is broadly met
- **No** – the requirement is not met
- **Partial** – the requirement is partially met
- **Study** – study is required to determine this

We acknowledge that this is a subjective assessment – we are marking our own homework, based on our knowledge of the design and implemented prototype, applying our experience of the iterative design and reflection process where applicable. We will go on to quantitatively assess some requirements – largely non-functional requirements relating to the performance of the agent – in the study documented in Chapter 9.

The remainder of this section details each of those entries listed in Table 6.5.

Table 6.5: Agent Requirement Compliance

	Requirement	Met		Reference
		Design	Prototype	
1	Act independently on behalf of the owner	Partial	No	6.9.2.1
2	Not discard important content	Yes	Study	6.9.2.2
3	Prompt delivery of time-sensitive content	Yes	Study	6.9.2.3
4	Make decisions about resource handling	Partial	Partial	6.9.2.4
5	Query resources using appropriate methods	Yes	Yes	6.9.2.5
6	Receive and process incoming information	Yes	Yes	6.9.2.6
7	Manage scheduling	Partial	Partial	6.9.2.7
8	Multiple resource types with a consistent interface	Yes	Yes	6.9.2.8
9	Multiple communication channels	Yes	Yes	6.9.2.9
10	Present information according to owner preferences	Partial	Partial	6.9.2.10
11	Provide mechanisms for user training	Yes	Yes	6.9.2.11
12	Allow user review of self-training adjustments	No	No	6.9.2.12
13	Represent owner outwardly and notify other actors	Partial	No	6.9.2.13
14	Be conservative in outgoing actions	Partial	No	6.9.2.14

6.9.2.1 Act Independently on Behalf of the Owner

While the agent is clearly acting on behalf of the owner, its level of independence is low, with the prototype only acting in reaction to user control and defined schedules. This falls short of the autonomy described in 6.4.1. However, we believe that our design for autonomous operation described in sections 6.6.9 and 6.7.7 provides a basis for meeting

this requirement. We reflect on this in Section 6.10.6.

6.9.2.2 Not Discard Important Content

The designed flow of Content Items through a set of queues and services [6.7.4], incorporating the Discard Service⁴⁰ ensures that Content Items seen by the agent do not simply drop out of the system, except for clearly defined conditions within Acquire service implementations (for example service bot messages, or those matching a specific exclude filter at the Acquire layer). We also implemented the Flow Monitor to allow us to verify this⁴¹, and we could see that 100% of items that should flow through the system, did do so.

While we're confident of the mechanics of the process, both in design and prototype, we note that the definition of *important* is more subjectively defined. One of the roles of the study [9] is to assess the prototype's performance in this regard.

6.9.2.3 Prompt Delivery of Time-sensitive Content

As with content discarding [6.9.2.2], we can see that the architecture of the Awareness Agent supports the timely flow of items into the Interact service. The Interact model that we have implemented for User-Directed ML [6.7.6.4] puts final delivery within the purview of Slack, where the user can configure their preferences for how different channels alert them to content. Because of this, we conclude that the design does satisfy this requirement, but in order to evaluate how well the prototype implements this in practice, study of the quality of classification decisions is needed.

6.9.2.4 Make Decisions About Resource Handling

Our design model for the Awareness Agent makes decisions about each incoming resource (CI), but does not make any explicit decision on handling (act, defer, ignore) above what is provided by the User-Directed ML classification process. Because of this we consider that this is only partially met in both design and prototype.

⁴⁰Development Log [D] entry 2019-10-21

⁴¹Development Log [D] entry 2023-06-25

6.9.2.5 Query Resources Using Appropriate Methods

We designed, implemented and tested support for Slack RTM⁴², Slack Events, and RSS⁴³. We also tested limited a proof of concept for LDN⁴⁴ as a source. This gives us confidence that we are able to query a wide variety of resources using appropriate methods, both in theory and in practice.

6.9.2.6 Receive and Process Incoming Information

We had designed the Agent Exchange Protocol (AXP) [6.6.8] to address this information (which also applies to 6.9.2.13). Our work with the Exchange Service [6.7.4.4] showed us that the Awareness Agent can successfully receive and process incoming information via MQTT⁴⁵ in the prototype. This content is injected into the queue system in the same way as other CIs.

6.9.2.7 Manage Scheduling

Scheduling support in both the design and prototype implementation is very limited, and falls short of the autonomous operation envisioned in the design concept. The RSS Acquire⁴⁶ implementation uses a cron-type scheduler, which requires explicit configuration.

6.9.2.8 Multiple Resource Types with a Consistent Interface

Our testing of the Interact Service for Slack [6.7.5] demonstrates a consistent UI [Fig. 6.21] with content coming from both Slack and RSS sources.

⁴²Development Log [D] entry 2019-07-26

⁴³Development Log [D] entry 2023-03-27

⁴⁴Development Log [D] entry 2020-09-28

⁴⁵Development Log [D] entry 2021-04-21

⁴⁶Development Log [D] entry 2023-03-27

6.9.2.9 Multiple Communication Channels

Our use of Slack for our prototype implementation provides multiple communications channels supported by the Slack platform. Slack notifications⁴⁷ can be configured to notify the user by a mobile alerts, desktop notifications or email. Our design also allows for additional output channels [6.6.4].

6.9.2.10 Present Information According to Owner Preferences

We can only say that we have partially demonstrated this requirement with our design and implementation. While the User-Directed ML concept supports this to an extent – by directing content to channels that are under the user’s control – this is not the type of understanding that we envisioned – nor does our concept of autonomy [6.6.9] really address this. To advance this requirement we would need to first make progress with implicit feedback [6.9.2.12].

6.9.2.11 Provide Mechanisms for User Training

The User-Directed ML design and implementation [6.7.6] supports this requirement, with explicit training of the agent by the user [6.7.6.4].

6.9.2.12 Allow User Review of Self-training Adjustments

While User-Directed ML training [6.7.6.4] inherently supports user correction of *classifications*, this is not the same as reviewing implicit feedback. Indeed the design does not currently support implicit feedback, which would be a topic for future work.

However, we do note that we did introduce an explanation mechanism for Augmentations⁴⁸. This could be a basis for the information provision to the user about training actions.

⁴⁷ <https://slack.com/help/articles/201355156-Configure-your-Slack-notifications> [<https://perma.cc/X9FG-YYA7>]

⁴⁸ Development Log [D] entry 2019-10-17

6.9.2.13 Represent Owner Outwardly and Notify Other Actors

The outward notification functionality is catered for by the Awareness Exchange Protocol [6.6.8]. However we've marked this as only partial satisfaction of the requirement because while AXP does provide a mechanism for the representation process, the design does not adequately address issues of trust and selectivity of content. That is, AXP as envisioned in this initial design is insufficiently discriminating over the content sent out, and a comprehensive inter-user (inter-agent) trust model needs to be established.

On the implementation side, while we have tested the concept via MQTT⁴⁹, this does not represent a functional implementation of even the limited design.

6.9.2.14 Be Conservative in Outgoing Actions

As already noted [6.9.2.13], AXP doesn't apply any additional discrimination over outgoing shared content. Therefore, while we recognise that the design allows for the Allocate service [6.7.4.3] to apply some discrimination of what content is shared based on Augmentations, this aspect is not sufficiently developed to qualify as anything more than partial fulfilment. Likewise, the AXP implementation is incomplete.

6.9.3 User Interface Illustration

Figures 6.30 and 6.31 show the Awareness Agent User-Directed ML interface in Slack for test user 'Adam' as viewed in a normal web browser. The available channels for Adam are shown on the left, and the content of the currently selected channel on the right (#interested-personal [6.30] and #cycling-cycling [6.31] respectively). This shows content that has been processed by the agent and placed in model-specific channels [6.7.6.4]: "personal" for the "interested" model, indicating that the agent thinks these items are interesting for Adam from a personal (rather than work) perspective; and "cycling" for the "cycling" model, indicating that this content is about cycling. We can see that as implemented, the channel name is constructed from a combination of the model and

⁴⁹Development Log [D] entry 2021-04-21

classification values. The user has the ability to set notification preferences individually for each channel in the Slack UI.

You can also see in the figures that this workspace – which is used as the Slack Interact UI [6.7.5] – has a single app installed, “Awareness Agent”. This is the integration of the Slack Interact service into the workspace. An Acquire workspace (source of content) would have the lightweight “Awareness Agent Listener” app installed⁵⁰.

Figure 6.32 shows the reclassification drop-down being used [6.7.6.4] – in this case to reclassify an item in the #interested-personal channel to “not”. This will have the effects of moving the item to the channel #interested-not and generating a training instruction in the background [6.7.6.2].

Figure 6.33 shows the summarisation quality drop-down being used to submit feedback on the quality of the summary of this item.

6.9.4 Source Code

We have published some of the source code for the applications that we have developed, detailed in Supplement S12 [[doi:10.21954/ou.rd.28045598](https://doi.org/10.21954/ou.rd.28045598)].

⁵⁰<https://slack.com/integrations> [<https://perma.cc/D8AM-RAJM>]

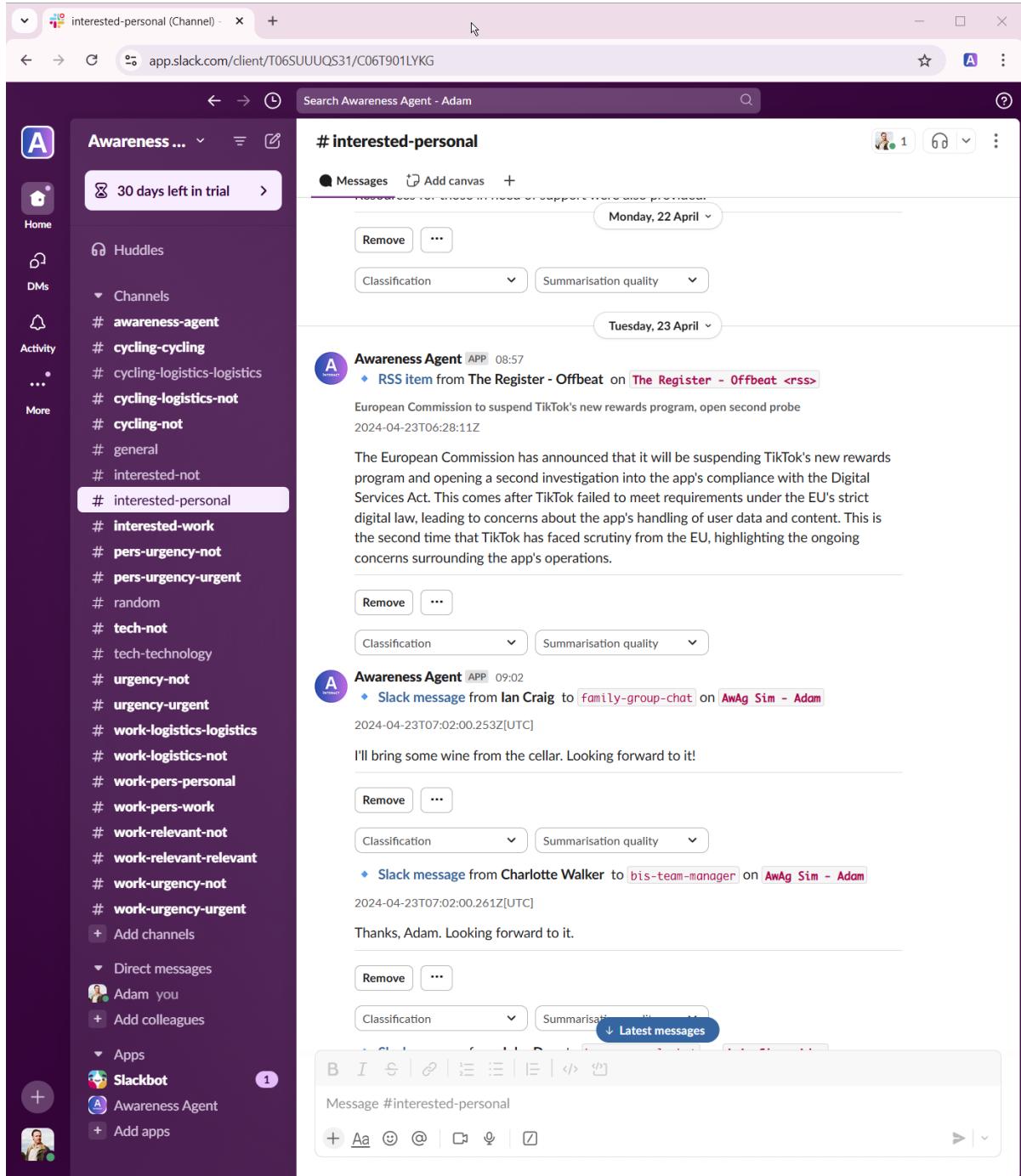


Figure 6.30: Awareness Agent UI showing UD-ML channel #interested-personal

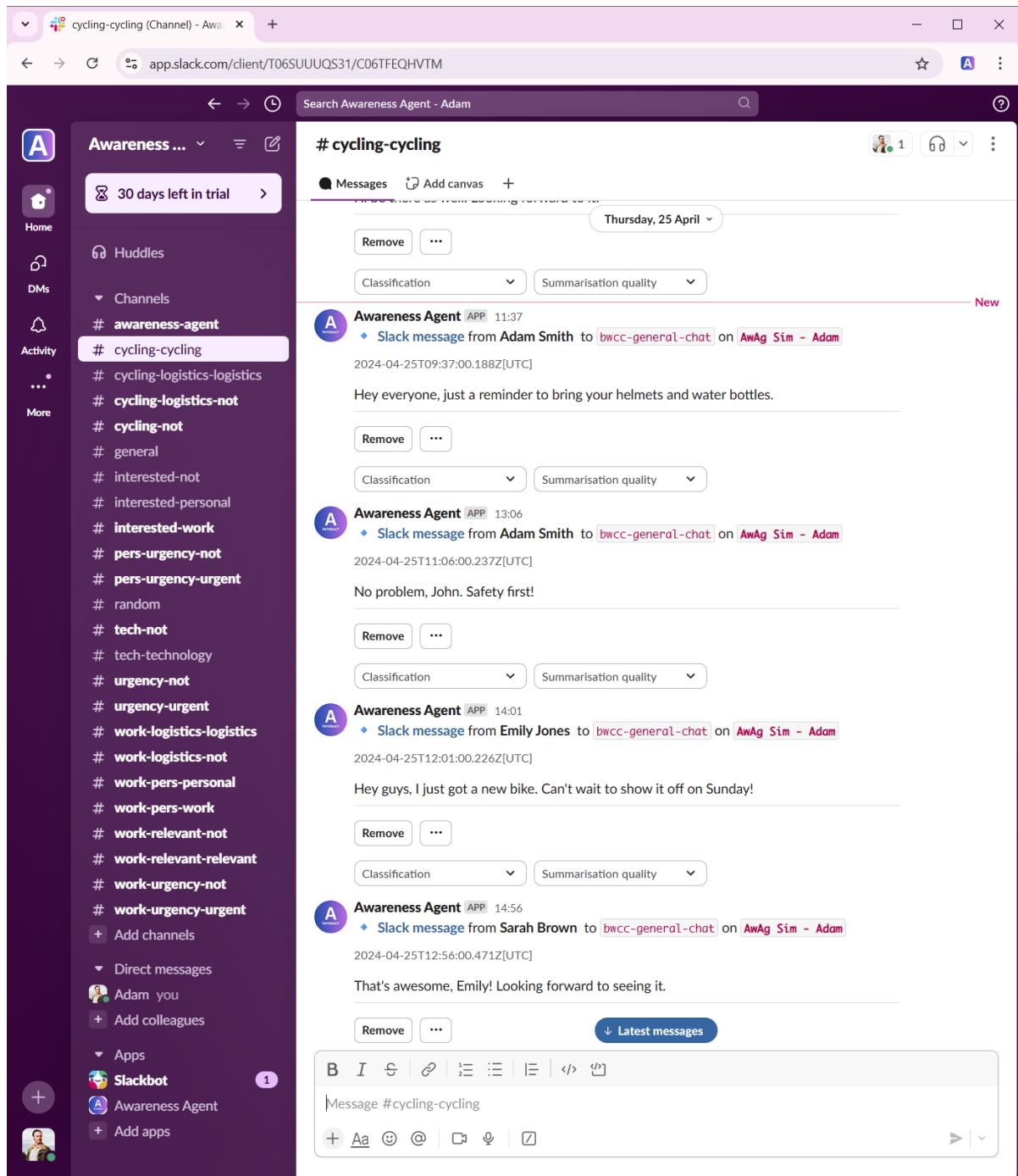


Figure 6.31: Awareness Agent UI showing UD-ML channel #cycling-cycling

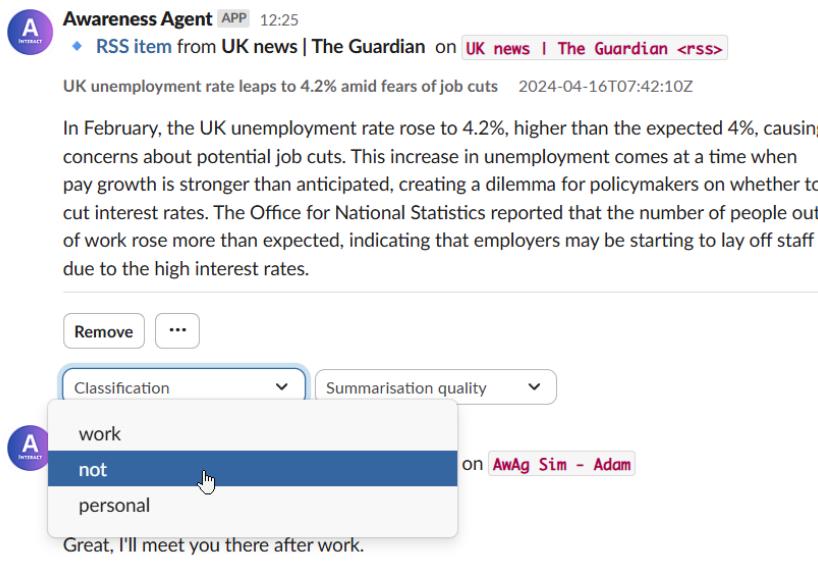


Figure 6.32: Awareness Agent UI showing UD-ML item reclassification

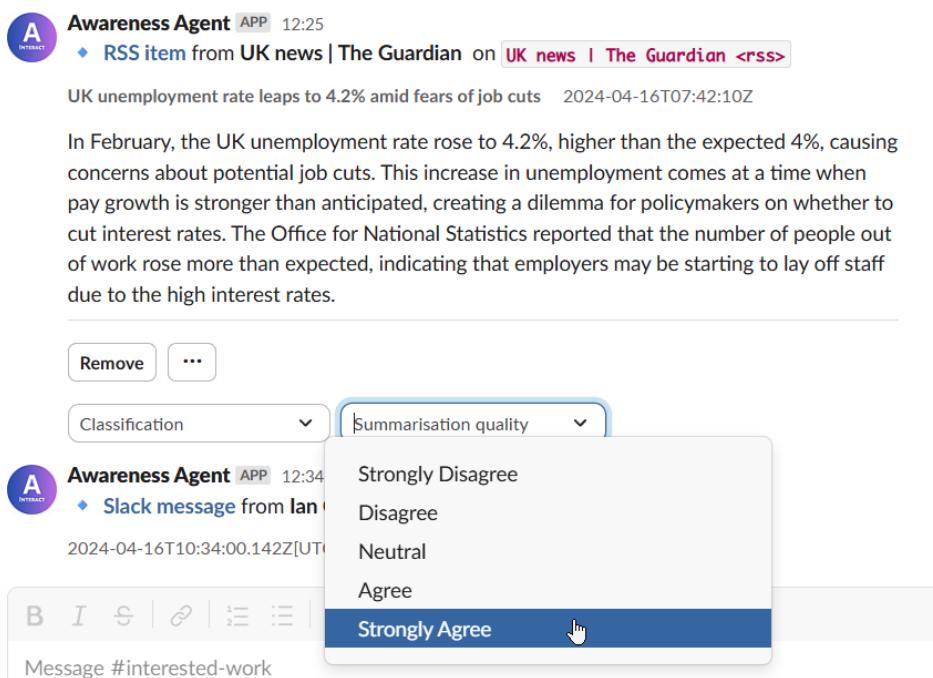


Figure 6.33: Awareness Agent UI showing summarisation feedback

6.10 Reflections

This section contains a number of our reflections as we look back on the process of design and development, documenting some general lessons learned as well as specific thoughts on the state of work at the current time. These reflections should be considered to complement the gap analysis [6.9.2] and development log [D], providing more a commentary on conceptual issues as well as on (and from) the design process that we have been through.

6.10.1 General Lessons Learned During Development

We found that the process of abstracting from source APIs and formats by using common Acquire component interface [6.6.1.1 & 6.7.4.1] and CI elements such as standard fields [6.7.1] was very important for making the design work. Different systems were found to need some quite different techniques to integrate, something that would have caused significant problems without the modular approach.

We did find some minor issues in practice with this approach. For example, while we tried (and were mostly successful) to design our CI standard fields at the outset, we did need to slightly adapt during the prototyping process to account for how information was both available and used in practice. Finalising such a list of fields should probably only be completed after implementing more types of Acquire and Interact components.

6.10.2 Access to Suitable Content

One expected issue that we experienced was that of access to content, something that we had considered in our problem analysis [3.1]. This led us to take an approach that placed more emphasis on the design process and challenges than to data itself, but did also lead us to make progress in initially unexpected areas such as synthetic data.

Our use of Slack – a widely used Business Collaboration Platform – demonstrated that we have the technical ability to use the Awareness Agent with corporate data, but we are mindful that the security implications of integrating third party apps such as the Slack Listener [6.7.4.1] is a significant concern [Chen et al., 2022]. We take a view that the

approach most likely to succeed with this is to make the Acquire component available as Free & Open-Source Software (FOSS) and encourage auditing of this, but corporate acceptance is likely to also hinge on visible good governance of any organisation wishing to offer an Awareness Agent-like service [Boulanger, 2005].

The other half of content restrictions is the prevalence of walled gardens⁵¹ and proprietary networks that do not always share nicely – or cannot be counted on to continue to do so⁵². The problem and potential solutions [La Cava, Greco, and Tagarelli, 2021] are a wider issue than our narrow experience – we hope to have designed a system that is broadly compatible with whatever may be accessible to it in future and place some trust in the work of others to keep information as free as possible.

6.10.3 Standard, Extended and Native properties in CI Data JSON-LD

The generation of Content Item Data is discussed in the Architecture section 6.7.3.

As we mentioned in the development log [D] entry 2019-11-09, we chose to leave the names of native properties in the source objects unchanged when constructing the CI Data. That is, in the output CI Data we have property names that are prefixed awag:xtd (extended fields), some that may be prefixed awag:std (standard fields) and those with no prefix (native fields).

We had considered alternatives – specifically adding a prefix to the native fields, or renaming the relevant native fields to the matching standard field names – but decided against it. Our rationale was to leave the original content as untouched as possible, by adding new fields only, and also by referring to standard fields by mappings where possible.

However, on reflection we recognised that the extended fields were not optimally designed. In reality, these are extended fields *per source* (they are generally specific to the type of Acquire source, so for example a Slack CI will have a particular set of extended fields). Our implementation did not distinguish between sources, so for example we have awag:xtd:channel-name, which is a Slack-specific concept. It may be more clear

⁵¹<https://www.adalovelaceinstitute.org/blog/walled-gardens-open-meadows/> [<https://perma.cc/4PYM-AKUQ>]

⁵²<https://arstechnica.com/gadgets/2024/02/exploring-reddits-third-party-app-environment-7-months-after-the-apocalypse/> [<https://perma.cc/HCV8-3LH4>]

and extensible if we instead use a further qualifier in the namespace for extended fields, for example: awag:xtd:slack:channel-name. In practice this does not make much difference, but it would probably be conceptually more consistent.

6.10.4 Relationship Between Augmentations and Outputs

The User-Directed ML process of mapping Augmentations to output channels described in Section 6.7.6.4, and to Allocate queues as described in Section 6.7.4.3 is not as conceptually tidy as we would like. This is partly a consequence of how we arrived at this design, and a revisit may be in order.

Currently we have an `InteractChannelMap` in the Slack `Interact` objects, which returns augmentation values mapped to channel names for a given augmentation name – but this is very implementation level dependent; there is no such thing as a defined ‘mapping augmentation’, we just process each existing augmentation that is of type `Simple Classification` and use those for which we find a mapping value in the `Interact` instance config (with a similar process for `Allocate`). While this works well enough in practice, we consider that it may be better – on a design purity level at least – if the role(s) of an Augmentation was stored in the Base Augmentation metadata as an additional attribute.

6.10.5 LDN and Solid

As noted in the Development Log [D] entry 2020-09-28, we did some work on Linked Data Notifications [2.3.3]⁵³ using `nodeSolidServer`⁵⁴. While we demonstrated on a technical level the viability of LDN for our use to our satisfaction, we eventually discontinued work on this mechanism for acquiring content. The reason at the time was that there was not sufficient data available via this mechanism, and other sources such as Slack or RSS had much better prospects for this.

However, this decision was made before we started looking at synthetic data. We had chosen Slack as the outlet for synthetic data [7.6.3] in part because we had already invested

⁵³ <https://csarven.ca/linked-data-notifications> [<https://perma.cc/7SVP-JPBV>]

⁵⁴ <https://github.com/nodeSolidServer>

efforts into Slack acquire sources by this time, so it was a pragmatic choice. However, on later reflection we realised that LDN could have been well suited as a channel to publish simulated content for our study. This would not have had the lack of content problem as we would be generating our own content and would have been a useful demonstration of LDN/Solid as an acquisition source.

6.10.6 Autonomous Operation

Agent autonomy has been a difficult issue during the course of this research, with the reality of what could be delivered in practice not initially matching our desired capabilities. Due to this we followed a course in the early part of the research that focussed more on the “weak” elements of agent behaviour such as operating independently on behalf of the user but without exhibiting autonomous features such as intent or independent knowledge. As of the closing stages of this work, we find that the commodity AI landscape has changed sufficiently to allow far more capable solutions to be devised. In line with our ethos of modularity and commoditisation, we can see a number of ways to utilise the new public LLM capabilities in particular to introduce many of the stronger elements of agent behaviour. We have discussed this in a limited fashion in our model, in Section 6.6.9.

We view autonomy of the Awareness Agent as a rich seam of future work that is quite compatible with our existing design and implementation. As well as the benefits that more autonomous operation could bring in terms of content identification and management, the techniques that we introduce would be well suited to closing some other gaps, such as ease of administration and control.

6.10.7 Further Scope For Work

The gap analysis [6.9.2] clearly defined a number of areas where more work is required to achieve the original vision. Specific areas of interest are autonomous operation, contextual awareness, and inter-agent (inter-user) information exchange.

We highlighted above [6.10.6] that there is considerable scope for development of agent

autonomy.

As we discussed in Section 6.9.1.1, there is a substantial opportunity to expand the scope to implement and study the context-aware aspects of the agent concept.

Our design was limited to only Slack and RSS as sources of information; future work could include adding many different Acquire sources, from the relatively easy to access (such as email or Discord⁵⁵) to those more challenging for technical or non-technical reasons [6.10.2] such as social media apps and SMS.

We also note from our experiences with Slack as a user interface [6.7.5.1] and work with awagUi [6.8.5] that there is significant scope for exploring alternative methods of user interaction [6.9.1.2], both to complement and maybe upgrade or replace what we have originally tested. This could include bespoke user interfaces that allow the user to proactively interact with content at times of their choosing, alternative technical platforms for content and notifications, or integration with AI assistants such as Siri⁵⁶ or Alexa⁵⁷.

Our ML Service was intentionally limited to a single classifier, with parameters that we deliberately did not tweak after our first testing [6.8.2.1]. This leaves scope for future work to focus on the quality of different classifiers and parameters as part of the User-Directed ML process. There is also scope to expand awagml to perform tasks other than simple single classification. Third party ML services that provide classification capabilities could also be slotted in to the User-Directed ML framework.

⁵⁵ <https://discord.com/developers/docs/intro> [<https://perma.cc/LQA3-7DH2>]

⁵⁶ <https://developer.apple.com/siri/> [<https://perma.cc/46GE-GWUA>]

⁵⁷ <https://developer.amazon.com/en-US/alexa/alexa-skills-kit> [<https://perma.cc/D2GE-KZZC>]

6.11 Chapter Summary

In this chapter we covered the design and development of our notional Awareness Agent, starting by considering the possible requirements [6.3] through the lens of the personas that we had previously developed [6.2], and finishing by analysing the design process and outcome for the application that we developed. We have used a process of Research Through Design [6.5] to inform the development of first a theoretical model [6.6], then a more concrete architecture [6.7] and finally a prototype implementation [6.8].

We then went on to look back at the status of the implementation [6.9], comparing what we had achieved in both the design and implemented prototype with the requirements we had identified at the outset. The iterative and reflective process that we followed allowed us to reflect on this work [6.10] and identify where this could be taken in future.

We contend that both our designed applications and the documentation of the reflections on the process advance the state of knowledge in this field.

We will go on over the next two chapters to look at how this application can be studied, developing the concepts of synthetic content [7] and evaluation [8] to facilitate this process, and then document the results of our study [9] of the Awareness Agent.

Chapter 7

Synthetic Content

The next two chapters could be considered something of a bridge between the introduction of the Awareness Agent and the final study, in that they cover some techniques and technologies that we will use to enable the study. However, we believe these are also significant in their own right from a research point of view, as we are developing here some novel techniques that could be used to support other research efforts. Indeed, the study discussed in Chapter 9 will also go on to evaluate the techniques introduced in these chapters, as well as the base Awareness Agent.

In these chapters we describe how we are taking an approach of synthetic content¹ generation and evaluation to study our agent design, utilising commodity LLMs such as OpenAI. We describe a novel process for performing a study using synthesised users aligned with our previously developed personas, in combination with human input on a supervisory and participant level. This is divided into two distinct parts:

1. Synthetic generation and use of pseudo social media and corporate messaging for use in an academic study (this chapter)
2. Development and use of synthetic evaluators to act as ‘virtual study participants’ (Chapter 8)

¹Also referred to as simulated content in some of our code

7.1 Motivation

While we were working on the Awareness Agent and on our plans to evaluate it, we realised that we faced particular hurdles specific to the nature of the application and how we wanted to evaluate it. These fell into two camps: *content-related* and *user-related*.

7.1.1 Content-related Hurdles

We were aware that much of the content that we would want to test our agent with was restricted in some way. Such data restrictions included:

1. Corporate communications restricted by confidentiality and technical barriers
2. Social media content restricted by technical barriers such as lack of API
3. Personal content restricted by privacy considerations

7.1.1.1 Technical Barriers

Broadly speaking we found two types of technical barrier, sometimes both at the same time:

Access Barriers where we had only limited access to the target service, typically because it was accessible only behind a corporate firewall or access was only granted to compliant devices. With the general move of services to public cloud hosting, firewall restrictions were less of an issue, but we found that corporate messaging and email resources hosted on the public cloud mostly had authentication requirements that limit access to specific compliant devices (a category that does not generally include prototype academic software).

API Barriers where access for third-party apps via an API was with limited or entirely absent. For example, the popular messaging service WhatsApp only makes an API available for business use², while Apple provides no API access that we can use to read messages.

²<https://developers.facebook.com/docs/whatsapp> [<https://perma.cc/FAC5-UCCL>]

7.1.1.2 Privacy & Confidentiality Barriers

Barriers relating to personal privacy and confidentiality of data are particularly relevant in the setting of an academic study [Peisert, 2020]. We can divide this into two areas:

Access to confidential data may not be easily granted by a corporation for academic study or experimental software; such organisations generally have strong information security policies in place.

Consent to participation in an academic study cannot be assumed for users of social media software who are merely communicating with a study participant. The gathering and processing of communications unbeknownst to unwitting study participants could be an ethical breach, and gathering informed consent from such individuals whose data may be processed by an Awareness Agent is impractical³

7.1.2 User-related Hurdles

We also realised that we could experience difficulty in the study relating to the user experience for the participants. One of the aspects that we wanted to examine was how our proposed solution would handle potentially large amounts of content over time. In order to evaluate this, the participant would need to view and check the classification of a large number of items over the period of the study. The Awareness Agent would not ameliorate this load because the study would require the user to assess all items, including those that the agent would have discarded as unimportant. Somewhat ironically, this would be a source of information overload for the study participant, with them being asked to pay attention to content that they would otherwise ignore.

7.1.3 Mitigating Restrictions Using Synthetic Techniques

We decided to use synthetic techniques to mitigate both types of hurdle: synthetic content would allow us to avoid content access and privacy issues, while synthetic evaluation could

³Not least because such consent could only be sought retroactively in many cases, as these individuals could be known only after they have sent a message.

allow us to reduce the workload of human study participants.

7.2 Research Through Design

As with the core Awareness Agent, we undertook our work on synthetic content and evaluation as an exercise in RtD. This chapter contains references to the log in Appendix D.2.

7.3 Approach to Synthetic Content

We have used Large Language Models (LLM) – specifically OpenAI’s GPT⁴ – to generate text snippets on specified topics, that we could then use to simulate real messages from users, serving them up to the Awareness Agent via some common delivery mechanism. The short-form text format would be best suited to emulating content from social and corporate messaging apps, but would not necessarily be tied to any given one. By using simulated content in this way we could then avoid having to integrate with various third party applications, needing to integrate only with the chosen delivery mechanism for that content.

7.3.1 Requirements

The main requirement for synthetic content was that it would be plausible to the user as content from whatever platform was being simulated. As a bare minimum this required grammatically correct texts⁵ that could be understood to relate to a given topic.

We need to be careful in our definition of “related to the topic”, and also precise about what we mean by ‘topic’. Real world content often deviates from the official topic of a given channel⁶ [Dahana et al., 2024] – to be useful for our study, simulated content needs to reflect this real-world deviation. Consequently, our concept for a ‘topic’ for content

⁴<https://www.searchenginejournal.com/history-of-chatgpt-timeline/488370/> [<https://perma.cc/6JP7-NZKF>]

⁵While acknowledging that real world content is not always grammatically correct, something that we did not attempt to simulate here

⁶Not all types of content have a defined topic, but many do

generation must encompass realistic deviation from the official topic. For example, say a user is a member of a messaging group that has been set up to discuss cycle racing fixtures. While most messages may relate to this topic, many could be entirely unrelated to cycle racing⁷. The ‘topic’ used to generate synthetic content for this channel must emulate such propensity for discussion to wander off the subject.

Synthesised content must also have diversity in terms of message topic, phrasing and simulated author, while also having consistent themes – such as frequent topics, conversations between individuals, consistent authors and terminology.

The delivery mechanism chosen for the content must support differentiating message by some form of channel or categorisation, such as an individual messenger group, or a named channel in a corporate messaging app.

7.3.2 Design Considerations

Initial testing using the OpenAI API⁸ showed us that it was possible to generate a wide variety of texts that emulated ‘real world’ messaging and conversations⁹. For our purposes we needed to devise a mechanism for creating texts that reliably maintained several different themes, containing a mixture of important and trivial, and had a sense of a ‘narrative arc’ of conversations between individuals about particular things.

We identified the following elements in the process:

- **Topic** – some summary of the topic or theme of texts to be generated
- **Dramatis Personae** – a cast of virtual people involved in the conversation
- **Entities** – non-human parts of the discussion, such as companies, clubs or other organisations

We could attempt to meet our needs by formalising these within a standard prompting framework for the LLM.

⁷This specific example is one that the author has direct experience of

⁸Other commercial LLMs were also available, but we selected OpenAI based on personal experience and initial work using it for text summarisation elsewhere in the project

⁹Although we acknowledge that there is substantial subjectivity in what constitutes a realistic text

We also had some technical and organisational considerations. As well as a wordy topic that sets the scene for the conversation, we also needed a terse way of referring to that topic in order to organise and publish content. This led us to introduce the concept of a “category” for the conversation as a short string identifying the topic.

Another consideration was that we did not want to operate on a ‘just in time’ model for the content, only generating texts on the fly, but instead preferred to generate these in batch and store them for later use. There are multiple reasons for this, including:

- Avoid issues due to short term connectivity problems to the LLM provider at runtime
- Quality of content can be visually inspected before use
- Generating multiple items in batch allows the LLM to better simulate conversations
- Batch generation is more resource-efficient

We logically and physically separated our application into three parts: content generation, content publishing, and consumption.

7.4 Content Generation

We implemented a standalone system for creating and managing simulated content, based on a Python application deployed to Flask and using the Data Service as the back end. The code for this application is included in Appendix Section E.2.

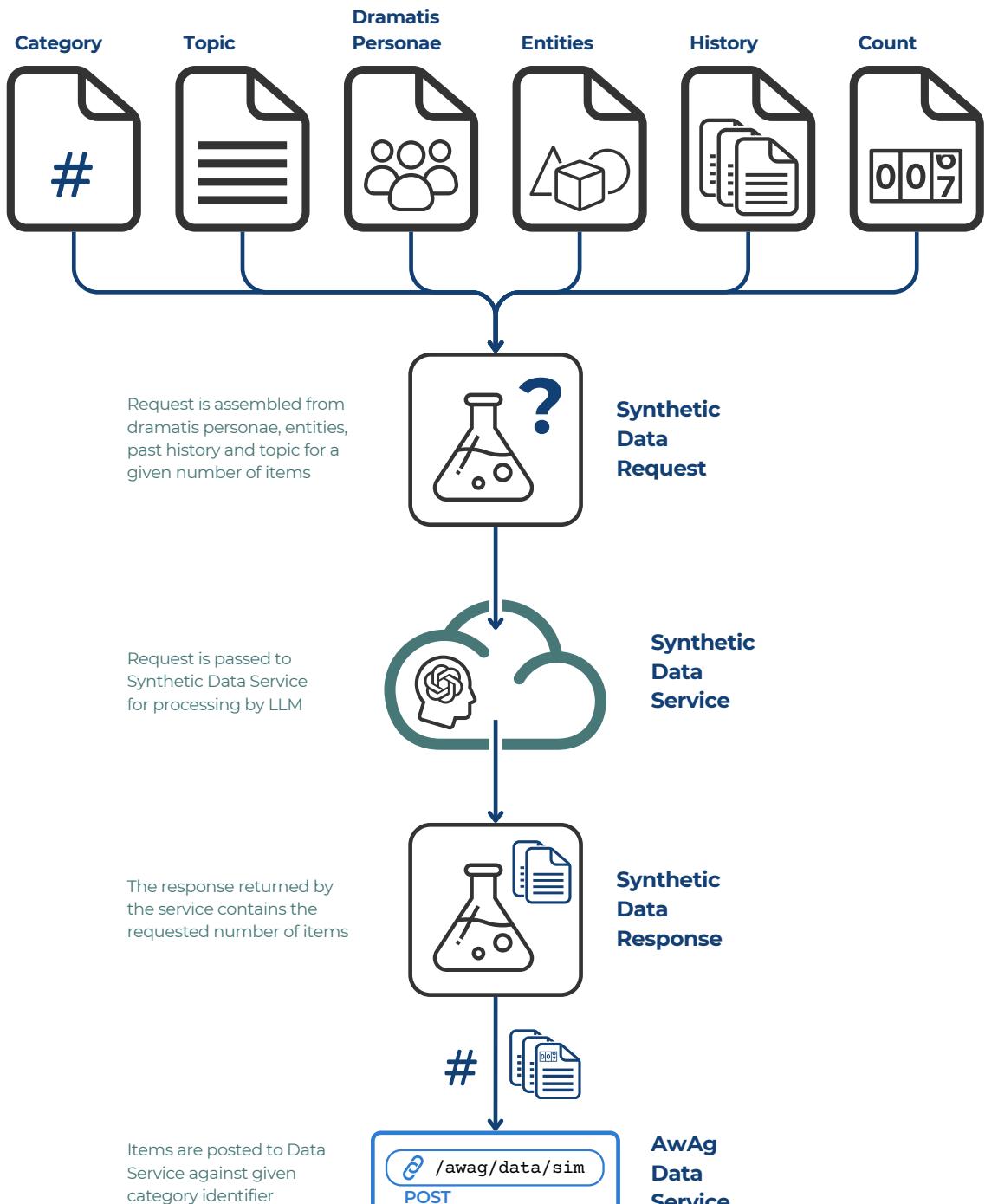
Figure 7.1 shows the process of generating a set of simulated content items and posting them to the awagdata /data/sim service for later retrieval [6.8.3]. Note that we have implemented the Synthetic Data Service as a part of the Data Service in our code, but we are documenting it here as a logically separate entity for clarity.

We ran the content generation process semi-manually, using the Postman REST client¹⁰ to initiate batch processes via the Data Service¹¹.

¹⁰<https://www.postman.com/product/rest-client/> [<https://perma.cc/8RQL-WAK3>]

¹¹See doi:10.21954/ou.rd.28044944 [path: /study/postman]

Figure 7.1: Synthetic Content Generation & Storage



Synthetic Content Generation & Storage

7.4.0.1 Synthetic Data Request

Table 7.1 shows the elements that comprise a Synthetic Data Request to generate the content, with references to the detailed items and examples in Appendix E.

The elements in the request are used to construct a request to the back end OpenAI GPT LLM using the `/chat/completions` API. Section E.2 links to the Python code used to do this, and to the overall prompt template that is used as the basis for every request. The following schemas are used in the process:

1. Result [E.1]
2. Dramatis Personae [E.3]
3. Entities [E.4]

Table 7.1: Synthetic Data Request Elements

Element	Description	Reference
Category	Short text label for this topic, used to store and retrieve messages	[7.4.0.2]
Topic	The text instructing the processing LLM the topic of content of the messages	[7.4.0.3]
Dramatis Personae	A document describing the fictional individuals that relate to this topic	[7.4.0.4]
Simulation Entities	Entities (organisations, clubs, businesses etc.) that relate to the topic	[7.4.0.5]
History	Passed message content for this topic, used to avoid repetition	[7.4.0.6]
Count	Number of items that should be generated	[7.4.0.7]

7.4.0.2 Category

Category is a short text identifier for a topic. When messages are generated, they are stored by awagdata against this category, and later retrieved for publishing using the same category identifier. While the identifier could have any unique value, we followed a naming

convention of “*persona-context-detail*”. For example, `adam-work-company-announce` and `adam-work-company-general` are work-related topics for persona *Adam*, while persona *Susan* has a tennis-related `susan-tennis-chat`. The output channels that messages are posted to during the simulation are also named based on the category value.

7.4.0.3 Topic

The topic is one or more paragraphs of text that defines what a given set of simulated messages is about. This text tells the underlying LLM what the general content of the messages should be and what style to write them in. For example, the following is the topic text for category `susan-work-university-announce` for the persona *Susan*:

susan-work-university-announce

Corporate messenger application chat within the employer of our fictional protagonist, Susan Carter. The messages are exclusively internal university announcements coming from the senior leadership team and PR office of the university that Susan works for, Borchester University. Generate messages that come from all of the senior leaders in the provided list as well as from an anonymous PR account or from Alumni Relations. Messages should occasionally refer to the university’s academic achievements, rankings, campuses, faculty changes etc. Message content should be unique and not duplicated. Try and use a good range of content, making up names of internal and external entities such as campuses, buildings, research centres, funding bodies, business partnerships, alumni organisations and government entities. The tone of the messages should be formal business British English with British spellings.

A full set of topics is linked in Appendix Section E.5.

7.4.0.4 Dramatis Personae

The *dramatis personae* JSON document represents a “cast” of characters to use when generating messages. The main prompt [E.2] instructs the LLM to use these users in two ways:

- Create messages ‘originating’ from these individuals, acting as fellow users of the simulated messaging system
- Create messages that refer to these individuals in the message body, as if they are someone being discussed or mentioned

The document conforms to a schema of our devising [E.1] and contains groups of members (simulated individuals). The groups are logical collections of people, such as members of a work team, messaging application group chat members, or family and friends. Each *dramatis personae* document can contain multiple groups to be used when generating messages.

A *dramatis personae* document can be shared across multiple categories, but is not expected to be shared for different contexts. For example requests for the categories for adam-work-company-announce and adam-work-company-general both use the *dramatis personae* document (adam-work) containing groups of the Adam persona’s work colleagues, manager, clients and so on, while adam-cycling-club-general would use a different document containing members of his cycling club and related individuals.

Each individual entry in the document contains the following information for a simulated user:

- **surname & firstname**
- **userid** – used to generate consistent user IDs in messages ‘sent’ by the person
- **role** – more information about the user that can be used to inform content generation

More information on *dramatis personae* documents is located in Appendix Section E.3.

7.4.0.5 Simulation Entities

Similarly to *dramatis personae*, non-human entities that are to be used in messages are contained in a JSON document that we pass with simulated content requests. The main prompt [E.2] instructs the LLM to use these entities inside the content of messages as things that might be referred to, such as organisations or companies.

The document conforms to our own schema [E.1] and contains a single array of entity objects. Each entity object has the following properties:

- **name** – the most commonly used name of the entity
- **also_known_as** – other names/aliases/abbreviations for the entity
- **type** – what type of thing the entity is (i.e. employer, customer, club)
- **notes** – additional information describing the entity and its role
- **people** – Some notional individuals associated with the entity

The **people** property of an entity is intended as a much more limited alternative to the simulated individuals contained in the *dramatis personae* document. While the latter is something that defines the core virtual population of the persona's world, the people defined in the entity are something that should only exist within the context of that entity, and would not be expected to be the 'author' of content.

More information on entities documents is located in Appendix Section E.4.

7.4.0.6 History

One issue that arose in early testing was the tendency of the LLM to repeat identical or very similar content. While this can be influenced by supplying appropriate values of parameters such as `temperature`, `top_p`, `presence_penalty` and `frequency_penalty` to OpenAI¹², we did not find that this was an effective way to construct a 'narrative arc' of content. To improve this we have the ability to supply a past history of messages for a given category (as supplied by awagdata) alongside the request, with the prompt instructing the

¹²<https://platform.openai.com/docs/api-reference/chat/create> [<https://perma.cc/GZZ3-ZUWD>]

LLM to use these to avoid repetition and create inter-message references.

7.4.0.7 Count

We ask the LLM to generate multiple messages in a single request. This enables greater efficiency by reducing prompting data size per message, and also makes it easier for the LLM to generate messages that appear to have a narrative or conversation. While the theoretical limit on this was the token support of the model¹³ We found in testing that performance began to become an issue when taking this number above approximately 50, although this was dependent on topic complexity.

7.4.1 Synthetic Data Result

The result of a simulated data request is an array of message objects. Each message has the following properties:

- **category** – category for the message
- **userid** – an ID for the message ‘author’, from the *dramatis personae* document
- **username** – full name of the message ‘author’, from the *dramatis personae* document
- **text** – content of the message

The schema passed to the OpenAI `chat completions` API tools function¹⁴ is referenced in Appendix E.1.

Appendix E.6 contains an example of some generated messages for a simulated school parents chat messaging group.

7.4.2 Publishing Simulated Content

Simulated content is published on a schedule that is determined by the administrator on a category by category basis, attempting to emulate real content. Figure 7.2 shows the

¹³<https://help.openai.com/articles/4936856-what-are-tokens-and-how-to-count-them>
[<https://perma.cc/3V7V-VCZM>]

¹⁴https://cookbook.openai.com/examples/how_to_call_functions_with_chat_models [<https://perma.cc/FY7T-WUNA>]

process for publishing synthetic content using Slack as the output mechanism [7.4.2.3]. The process – incorporated into the Awareness Agent [7.4.2.1] – runs for every configured category on a scheduled basis [7.4.2.4], publishing a message each time.

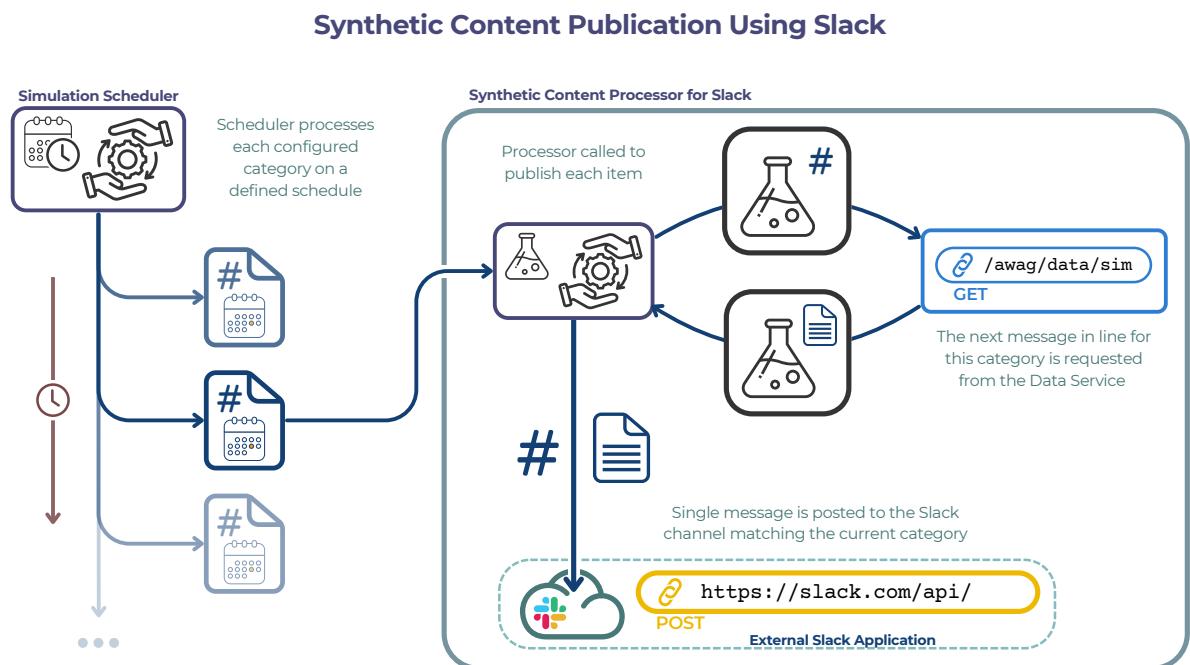


Figure 7.2: Synthetic Content Publication Using Slack

7.4.2.1 Extension of Awareness Agent

We decided to incorporate the simulated data publication functionality into the Awareness Agent even though it is logically separate and not related to the core Awareness Agent role. This was a pragmatic decision that allowed us to use the existing application infrastructure of the agent and re-use existing code – such as that for publishing content to Slack and interacting with the Data Service. Control of the simulation process – adding category/topic schedules, changing publish frequencies etc. – is performed using Slack Slash commands¹⁵, passing JSON configuration documents to the agent (see Section 7.4.2.4).

¹⁵<https://api.slack.com/interactivity/slash-commands> [<https://perma.cc/76S8-RE7V>]

7.4.2.2 Retrieval From Data Service

We added a method to awagdata that would allow a single message at a time to be retrieved for each category. A private integer counter within the service keeps track of the last message that was retrieved by a given agent for a given category/topic and returns the next in sequence¹⁶. This allowed the client to step through the pre-generated messages one at a time, maintaining a conversational style.

7.4.2.3 Slack as Simulation Platform

The output platform for simulated data should be thought of as simply a vessel for containing and accessing the content. It is not expected to necessarily resemble the usual medium for a given message type. That is, simulated content could be intended to mimic mobile phone instance messages, or corporate messenger content or some other form of data, but in each case we use one common mechanism for publishing the simulated content rather than trying to emulate each different type of app. This is because in the case of our study we care more about the content, timing and logical channels than faithfully recreating different applications.

We selected Slack for this role – our main rationale for doing so was that we already had significant resources for Slack within the Awareness Agent that we could utilise here, both in the Acquire and Interact components. However, we note our previous reflection in Section 6.10.5 regarding the alternative possibility of using LDN in this role.

Slack uses a paradigm of Channels¹⁷ for publishing content. In our concept, we publish each Category [7.4.0.2] to its own Slack Channel.

7.4.2.4 Publish Schedule in Awareness Agent

We used the same scheduling library that was already in use for RSS Acquire sources¹⁸, but in this case we schedule on an explicitly timed basis rather than using cron.

¹⁶Parameters to reset or override the sequence were also added

¹⁷<https://slack.com/help/articles/360017938993-what-is-a-channel> [<https://perma.cc/9R26-WU53>]

¹⁸See Section 6.7.4.1 & Development Log entry 2023-03-27

We have designed the process to allow the administrator to set up a daily schedule for defined days of the week with varying volumes of content posted per hour. This is done by a fairly crude mechanism of setting the number of items per hour for each hour in the command JSON. For example, Listing 7.1 shows and example of the Slack Slash command used to schedule the adam-work-team-general category to execute on weekdays. We can see that a Slack output channel of ‘bis-team-general’ has been set for this, meaning that the published items for this schedule will be posted to Slack channel #bis-team-general.

```
/awag sim daily add {  
    "category": "adam-work-team-general",  
    "channels": [  
        "bis-team-general"  
    ],  
    "daysOfWeek": "MONDAY-FRIDAY",  
    "volumes": {  
        "8": 2,  
        "11": 3,  
        "13": 2,  
        "17": 3,  
        "18": 1,  
        "21": 0  
    }  
}
```

Listing 7.1: Sim Add Slash Command Example

In this example, no items will be posted until 08:00, from which point 2 items per hour will be posted, until the next change, which is at 11:00, where the rate changes to 3 per hour. This continues until 13:00 when the rate drops back down to 2, increases to 3 at 17:00 and then drops to 1 at 18:00. Finally, after 21:00 the rate is set to 0 and no more items are posted that day.

The simulation scheduler works by setting up a schedule at the start of each day, scheduling the configured number of publish events for every hour for that day. In order to add some variance of exact time, the Awareness Agent Java code `generateEventTimes()` method applies a random offset for each event of a few minutes. This code is included in Supplement S6.5 [doi:10.21954/ou.rd.28045469].

7.5 Consumption of Synthetic Content

Synthetic content should be consumed by the Awareness Agent just like any other content, using a Listener type Acquire instance directed to the publication location. In our case this is a Slack Listener [6.7.4.1]. The agent can be configured with a Listener installed to any Slack workspace¹⁹ that synthetic content is published to.

7.5.1 Categories as Slack Channels

As noted above, topics for simulated content are associated with Channels in Slack using a mapping configured by the administrator [7.4.2.4]. The Listener app installed to Slack must be configured to receive messages for each channel that we wish to feed content to.

When we first started developing the Slack Acquire module, we worked with older restrictions for the Slack API that required an app or bot to be added to a channel/conversation in order to receive content²⁰. Although Slack since updated from the legacy ‘bot’ API²¹ and we updated our client code to match, we continued this approach for our study, requiring the user to manually add the Listener app to each channel that it would be expected to monitor. This approach gave good control over the content going to the agent, and was not an administrative burden within the context of our work. An alternative would be to extend the application to automatically join the appropriate channel at the point of schedule creation [7.4.2.4].

After this, content posted to the channel is emitted by Slack using the Events API²².

7.5.2 User Interface Illustration

The following screenshots show examples of the synthetic content generated for persona Adam [B.7] that have been published to a Slack workspace called “AwAg Sim - Adam”. This workspace is distinct from the Awareness Agent UI [6.9.3] and is integrated with only

¹⁹<https://slack.com/help/articles/202035138-add-apps-to-your-slack-workspace> [<https://perma.cc/8V4S-NLG5>]

²⁰<https://github.com/slackapi/node-slack-sdk/issues/26>

²¹<https://api.slack.com/legacy/enabling-bot-users> [<https://perma.cc/9542-8M3F>]

²²<https://api.slack.com/apis/events-api> [<https://perma.cc/N83S-8PEU>]

the Awareness Agent Listener (Acquire) app and not the Interact app. Items posted to the various channels in this workspace are fed to the Awareness Agent.

Figure 7.3 shows the output for channel `#bis-team-manager`, which is simulated work discussion between Adam and his manager Charlotte Walker. The topic text for this is located in Appendix E.5 entry `adam-work-team-manager`. This discussion contains messages ‘sent’ only by Adam and his manager, as directed in the topic text. We can see that entities such as “ColWidgets” from the entities document have been selected for inclusion (ColWidgets has been correctly identified as a client in this case).

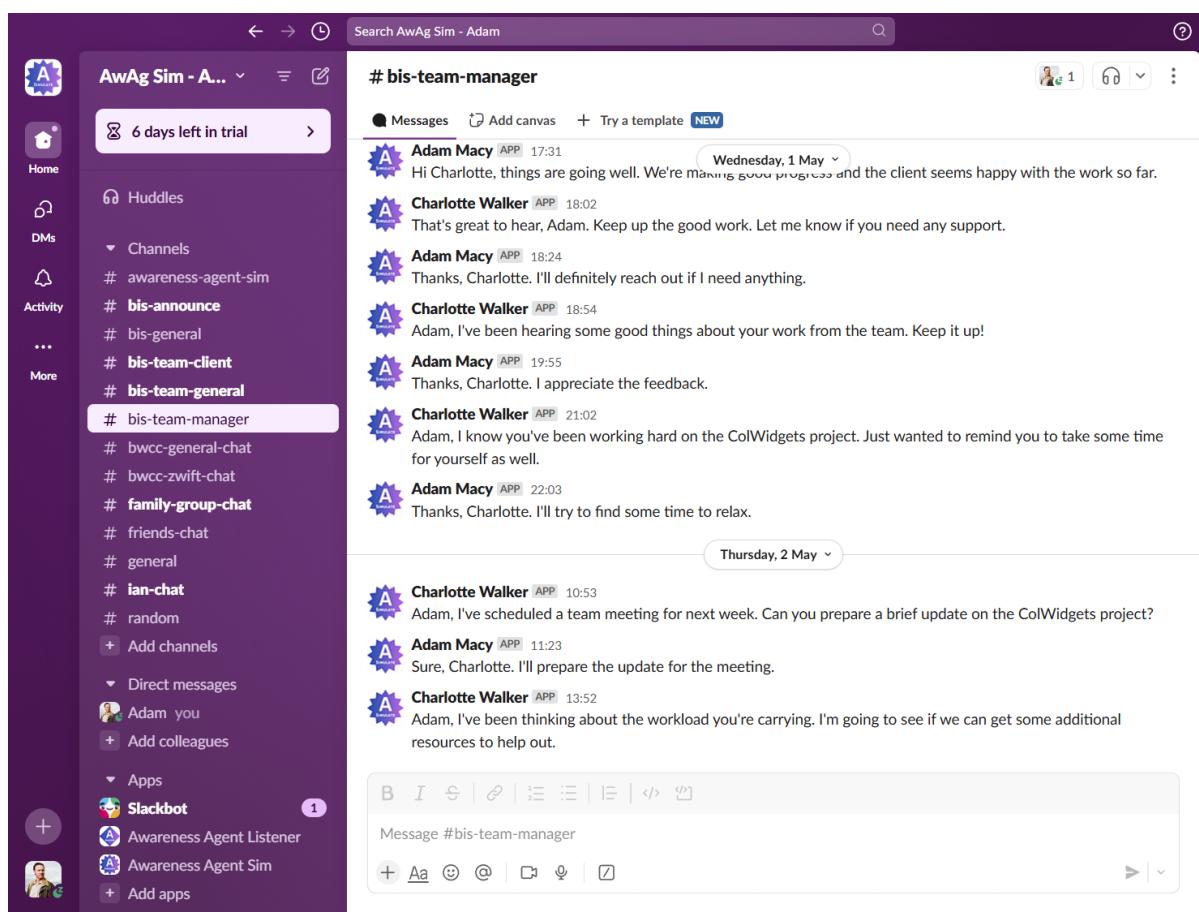


Figure 7.3: Slack web app for persona Adam showing channel `#bis-team-manager`

Figure 7.4 shows the output for channel `#bis-team-client`, which is a simulated work discussion between members of Adam’s team at work. The topic text for this is located in Appendix E.5 entry `adam-work-team-client`. The senders of messages correspond to those listed in the *work* dramatis personae for Adam as team members, and the discussion in this case relates specifically to matters relating to the company’s clients, as directed by

the topic. We can see that Tiny Widgets Company, which is identified in the entities document as a client is also referred to as “TWC” – an abbreviation listed in the `also_known_as` field for this entity’s entry.

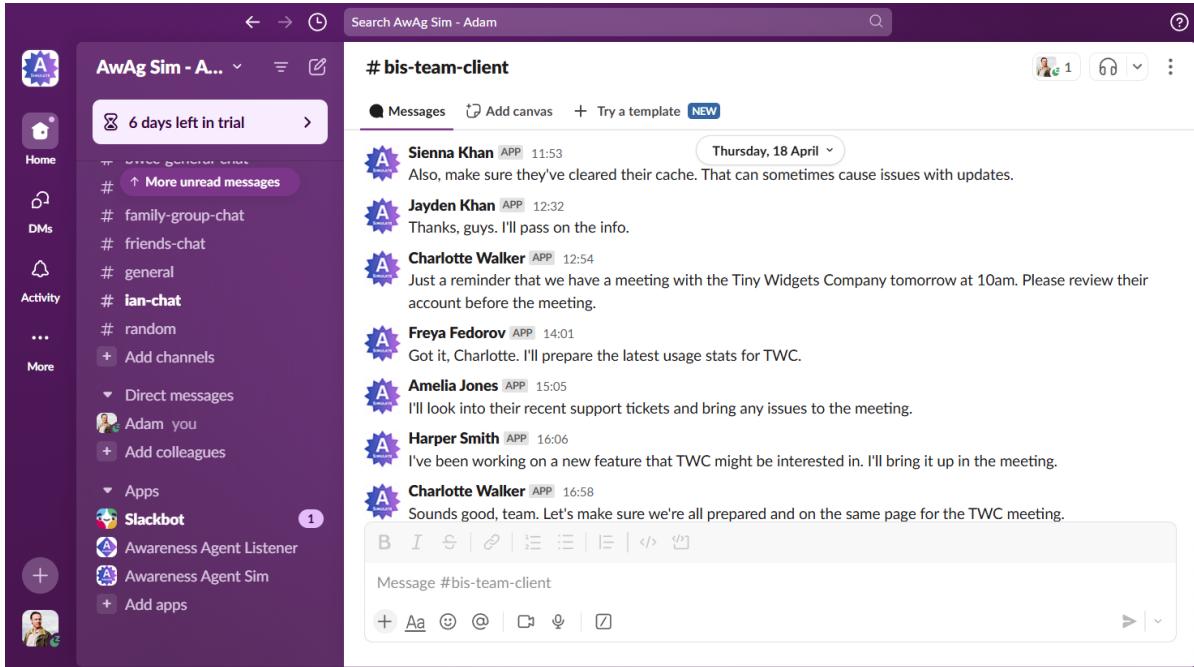


Figure 7.4: Slack web app for persona Adam showing channel `#bis-team-client`

Figure 7.5 shows the output for channel `#bis-announce`, which is simulated general announcements at Adam’s place of work. The topic text for this is located in Appendix E.5 entry `adam-work-company-announce`. We can see that messages in this channel are from a much smaller group of people, those who have been defined as management in Adam’s `work` dramatis personae document. The more formal tone of announcements reflects the direction given in the topic.

Figure 7.6 shows the output for channel `#family-group-chat`, which is a simulated instant messenger group chat between members of Adam’s extended family. The topic text for this is located in Appendix E.5 entry `adam-family-group-chat`. We can see an informal tone, with senders being those identified as family members in Adam’s `personal` dramatis personae document, and also that there is a simple narrative arc, where messages relate to the same topic (a family get-together).

Figure 7.7 shows the output for channel `#friends-chat`, which is a simulated instant messenger group chat between Adam’s friends. The topic text for this is located in the

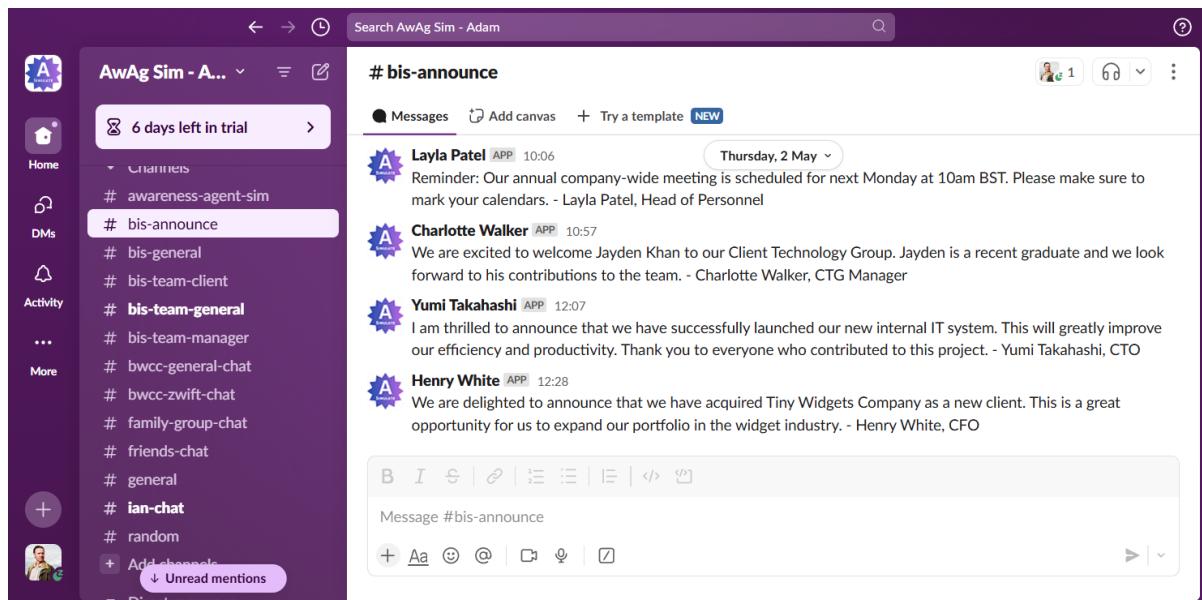


Figure 7.5: Slack web app for persona Adam showing channel #bis-announce

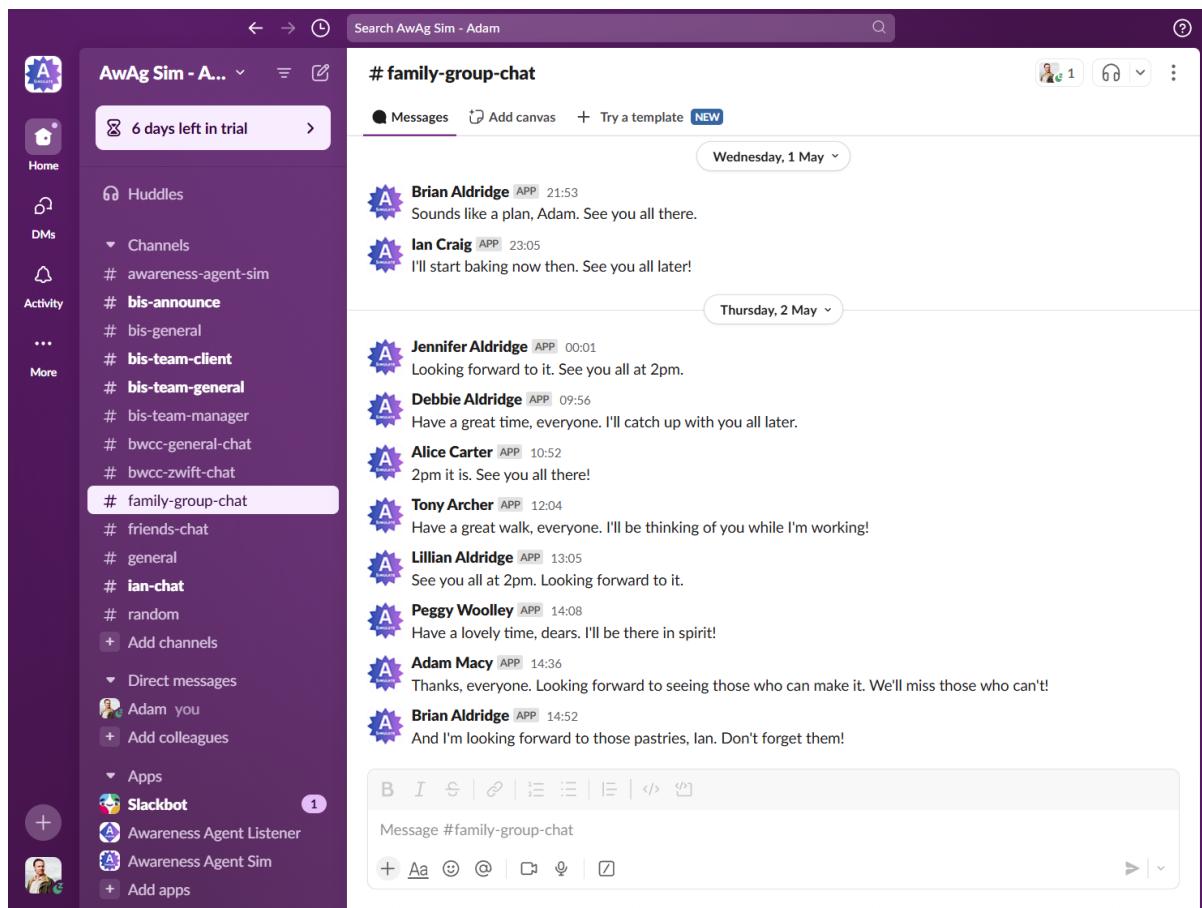


Figure 7.6: Slack web app for persona Adam showing channel #family-group-chat

Appendix E.5 entry `adam-friends-chat`. Similarly to the family chat, we can see an informal tone and narrative arc. While the same *personal* dramatis personae document is used, senders here are those identified as friends in that document.

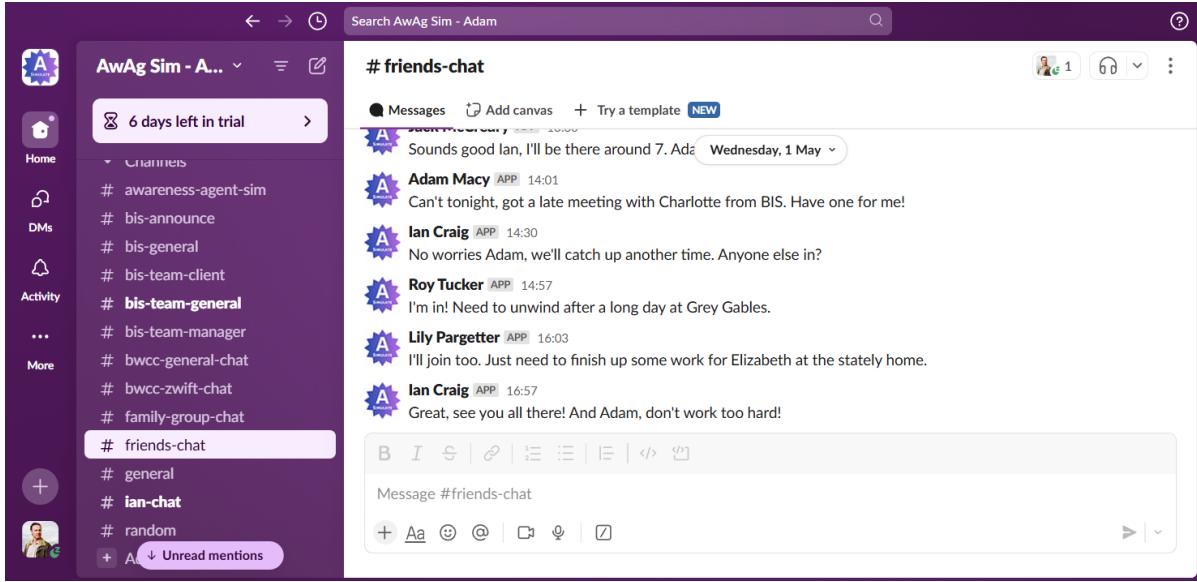


Figure 7.7: Slack web app for persona Adam showing channel `#friends-chat`

Figure 7.8 shows the output for channel `#bwcc-general-chat`, which is a simulated discussion forum for Adam's cycling club, Borchester Wheelers. The topic text for this is located in Appendix E.5 entry `adam-cycling-club-general`. We see a different set of individuals in the chat, coming from Adam's *cycling* dramatis personae document. Places referred to come from either the topic or the entities document.

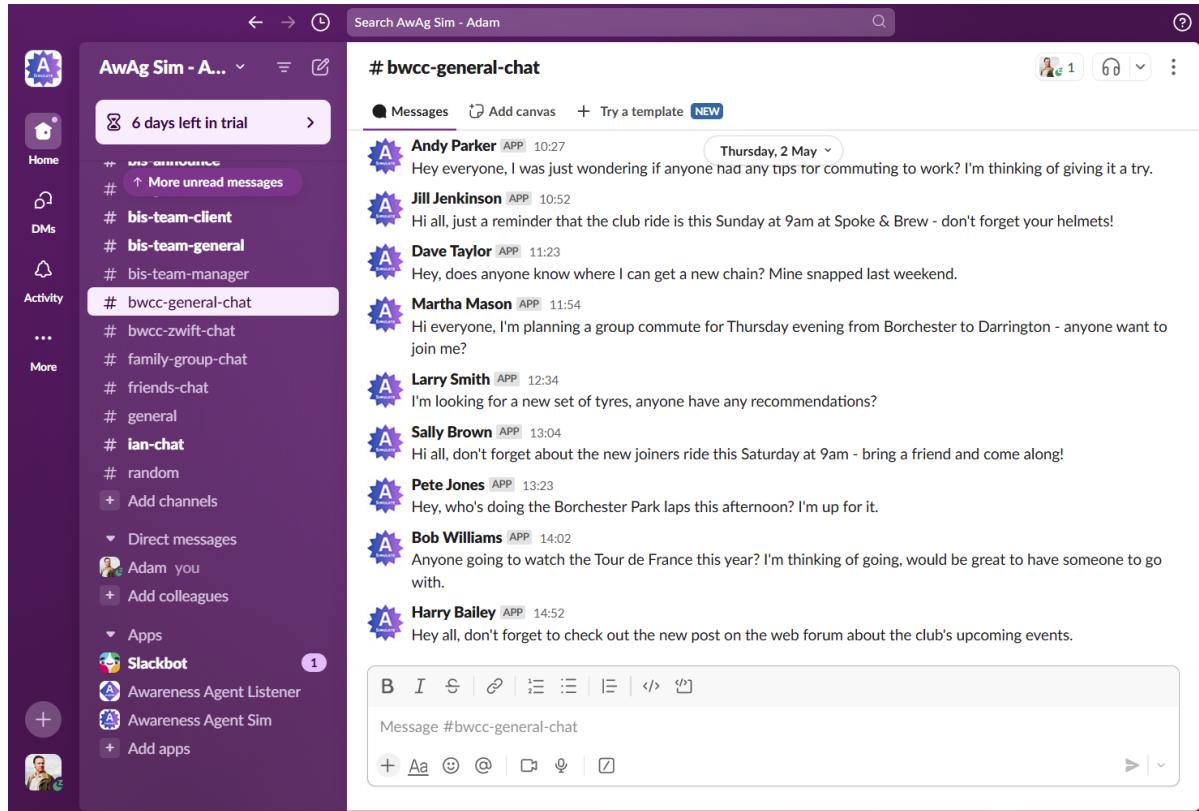


Figure 7.8: Slack web app for persona Adam showing channel #bwcc-general-chat

7.6 Reflections

7.6.1 Initial Findings

Development of the synthetic content was very much an iterative process. Our initial attempts used only an increasingly complex topic definition to drive the output from the LLM. This worked on a superficial level, but had some issues:

- The simulated message ‘senders’ were inconsistent, with random names often being used
- There was no concept of a narrative across messages
- There was little ‘depth’ – messages were mostly superficial
- There was a lot of repetition of content across different messages

We added the *dramatis personae* document [7.4.0.4] to formalise the concept of message senders; this also allowed us to define roles and organisational structures for those people (we can think of them as lightweight personas). This simplified topic construction, allowing us to direct that particular topics related to particular groups of senders, and also allowed the LLM to pick biographical details from that information. This also helped with conversations over time, with the same individuals being selected to send replies. Some tweaking of the topic and overall prompting was required to get this right however.

Similarly, the addition of the entities document [7.4.0.5] allowed us to define things that the LLM should talk about, whether that is a work client or rival cycling club. By giving names and aliases (abbreviations) to them, we were able to have the LLM output more realistic conversations that had a narrative.

We found that even with direction in the topic for diversity and larger batch sizes [7.4.0.7], we were still seeing some repetition of exact messages. We added the history feature [7.4.0.6] to minimise this, and also to allow the LLM to construct longer term narrative arcs. This feature input previous messages into the request – while this increased token usage and processing time, it had a notable effect on the quality of conversational realism.

In hindsight we believe that we should also have added a Locations document, detailing places that might be referred to, alongside the *dramatis personae* and entities. This was particularly apparent when working on the cycling club topic, where location would be a frequent subject of discussion, but could also be used to expand logistics-related discussions for work-related travel for example.

7.6.2 Limitations

Our findings above are purely observational from the author, emerging as part of the RtD process. We are not able to objectively measure the quality of the outputs without a formal study of those outputs accounting for changes in topic, personas and other inputs.

As noted, the publishing schedule [7.4.2.4] is not very flexible or powerful, being manually defined. This should be thought of more as a method that allows us to approximate relative volumes for items, but not in a sophisticated way. We did not develop this further as it was

adequate for our needs, as our study did not focus closely on the timing of simulated content posting.

As we discussed above, the lack of a formal Locations document in the synthetic content request limited the geographical nature of discussions, with more work needed in the topic to work around this.

We were limited in the amount of prior history that we could use in requests by LLM token and performance restrictions. While the history that we did include allowed a reasonable narrative, this could be improved with longer history.

There is no concept of time of day in our message generation and publication (we publish different volumes of message by time of day but do not select the content for the time of day). This led to messages appearing at inappropriate times and for conversations to have odd timings (for example where a conversation should be immediately interactive between two people, you would expect messages to be close together in time).

7.6.3 Slack as Outlet for Synthetic Data

As noted in Section 7.4.2.3, we chose Slack as the output mechanism for published content. We did this for pragmatic reasons, as it enabled us to use many synergies with other parts of the project in terms of code re-use, as well as other advantages such as the concept of categories/topics mapping well to Slack's *channels* concept.

However, in hindsight we can see that we could have made better use of a Linked Data Notifications server, which we were already looking at as an Acquire source (see Development Log [D] entry 2020-09-28). This would have given us much the same synergies as with Slack, while also providing a use case for LDN as an Acquire source.

Taking the LDN approach would also be more consistent with our use of JSON-LD for the Content Item [6.6.2.3].

7.6.4 OpenAI Model Selection

As we noted in our discussion of the application of the literature to our research topic [2.5.9, 3.3.3], taking a commoditised approach to AI allowed us to focus on the inputs and outputs of the AI system rather than how it internally functions, and this formed a central tenet of our design approach for the Awareness Agent [6.4.3].

At the time we started working with LLMs²³, OpenAI was the dominant commodity provider [2.5.3], with competitors yet to launch alternatives such as Google's Gemini²⁴. OpenAI also provided a flexible and mature API, launched in 2020²⁵ – this made it the most suitable commodity service for our purposes.

We chose to use the most recent generally available model from OpenAI, so initial work on synthetic content used the GPT-3 `text-davinci-003` model²⁶, which was released in Autumn 2022²⁷²⁸.

As detailed in the Development Log [D] entry 2023-07-25, OpenAI later moved to the `/chat/completions` API, enabling more functionality alongside access to the newer GPT-3.5 model, so we added support for the new API in the Synthetic Evaluation Service [8.4.4] and also ported this functionality to content generation. However unlike for evaluation, when API access for GPT-4 was made generally available in April 2024²⁹ – we did not change content generation to use this version. Our reasoning for not moving was that we had already generated substantial content using 3.5 at this time and wanted to be consistent in which model we used for all content. We had also found GPT-3.5 to be sufficiently effective at the content generation task.

²³Development Log [D] entry 2023-02-21

²⁴<https://blog.google/technology/ai/google-gemini-ai/> [<https://perma.cc/83T4-8XVR>]

²⁵<https://openai.com/index/openai-api/> [<https://perma.cc/49KE-F2Y3>]

²⁶<https://platform.openai.com/docs/models> [<https://perma.cc/9UL2-BX5D>]

²⁷<https://pub.towardsai.net/openai-just-released-gpt-3-text-davinci-003-i-compared-it-with-002-the-results-are-impressive-dced9aed0cba> [<https://perma.cc/59YV-Z7HS>]

²⁸<https://scale.com/blog/gpt-3-davinci-003-comparison> [<https://perma.cc/RTA6-WX7Y>]

²⁹<https://openai.com/index/gpt-4-api-general-availability/> [<https://perma.cc/6GAX-K4YM>]

7.6.5 Scope for Further Work

Study of Content Quality

As our study has not explicitly examined the quality of generated content items in isolation, there is scope for a separate study that focuses on how well this technique simulates various types of real-world content. This study could examine aspects including:

- The general readability of generated texts
- Applicability to the defined topic
- Relative effectiveness of differently worded topics
- Relative effectiveness of different LLMs
- Heterogeneity of generated content
- Extent of use of Entities and Dramatis Personae content in generated output

LDN as Publishing Mechanism

As noted above, switching to using LDN as a publishing mechanism instead of Slack would provide a number of benefits.

Publish Schedule in Awareness Agent

There is room for improving the scheduling process, to allow for greater flexibility and automation. In particular, the administrator should be able to specify a Probability Density Function alongside a total daily volume range to control the time distribution of posting.

Narrative Improvements

The use of entities, dramatis personae and history were effective at subjectively improving the quality of inter-message narrative in the synthetic content, but this could be improved. We suggest a number of techniques that could be explored:

- Add a ‘themes’ document, detailing a number of common themes (or plot lines) that could be referred to in messages, such as an ongoing deal with Client X, or the campaign to save a local pub from closure. The LLM would be able to select from these themes to create ongoing narratives.
- Maintain a history summary – we could periodically ask the LLM to process all past messages and produce a summary document giving an overview of past messages and themes. This could then be used in each request to help the LLM continue ongoing themes.

Timing Improvements

Current message generation does not pay attention to time of day, and the serving mechanism also does not reference time. This leads to messages with inappropriate timing (such as a “goodnight” message at 2pm). Work could be done to add the timing concept to messages, but this would also need to be matched with work on the publishing side to output pre-generated content at the right time – this would also have an impact on the scheduling work mentioned above.

7.7 Chapter Summary

In this first precursor chapter to our study [9], we covered the rationale for using synthetic content and described the creation of a system to support the study. This included:

- Motivation [7.1]
- Research through Design 7.2]
- Approach [7.3]
- Content generation [7.4]
- Content consumption [7.5]

We also documented our reflections on the process [7.6], and included an abridged version [D] of the full Design & Development Log [D], containing the more significant entries that related to the work covered in this chapter. In the next chapter [8] we will discuss our complementary work on synthetic evaluation.

Chapter 8

Synthetic Evaluation

This chapter covers the second part of the bridge to the final study – Synthetic Evaluation. We describe here the development and use of synthetic evaluators to act as ‘virtual study participants’ in our final study, to be tasked with evaluating the UD-ML classification of incoming content. While we used synthetic content in order to address content-related hurdles we experienced in our work [7.1.1], synthetic evaluation helps us address user-related hurdles [7.1.2]. Synthetic evaluation could – if it functions well enough – take on the role of a highly attentive participant with an excellent work rate and quick turnaround. In this chapter we discuss our design for synthetic evaluation based on an LLM and how we then implemented that. Examining how well this evaluation works in practice forms a significant part of the study covered in Chapter 9.

We undertook our work on synthetic evaluation as an exercise in RtD. This chapter contains references to the log in Appendix D.2.

8.1 Approach

We decided to approach the task of synthetic evaluation by looking at the capabilities of LLMs such as OpenAI’s GPT models. Or rather, we should say that working with this technology on other tasks led us to consider whether we could make use of them to address a pain point in our study design [7.1.2]. Having found that LLMs could be used to generate

realistic conversational content based on an understanding of human scenarios [7.4.0.3], it was a logical next step to investigate whether we could use LLMs in a structured way to evaluate decisions made about such content, using analysis and classification capabilities that have an active body of current research [2.5.4]. We began this process by looking at what we required of a synthetic evaluation system, considering the design constraints and other factors, then building a conceptual framework for an evaluator. We then implemented a prototype evaluator and integrated it into our existing Awareness Agent systems and processes before going on to test it within our final study [9.1.2].

While other LLM providers could provide the functionality that we are looking for, we selected the OpenAI API for this work,

8.1.1 Requirements

We consider that the most important aspect for synthetic evaluation is its high degree of fidelity to how a human participant would respond. This includes similarity of quantitative outputs such as scores or Likert evaluations, and also that any textual output closely resembles human-generated content. Being able to perform well in this role is the primary requirement.

To support this, there must be a mechanism to communicate the evaluation rules and expectations adequately to the evaluator – in particular so that any scenarios or personas are respected.

The system needs to be aware of the context of what it is evaluating at least at a level acceptable to perform the task. By this we mean that it should be aware of the timing of content being delivered to it and how this relates to the activities or wishes of the scenario persona at that time.

We identified a main benefit of using a synthetic evaluator being that it can be constantly active and alert; this implies that both high availability and good performance are required.

To support experimental validity, it must also be possible to independently verify or vali-

date results generated by the evaluator. To do so, all necessary data must be captured and recorded. This data includes: the content to be evaluated, the personas, classifications and other query data passed to the evaluating LLM and the raw responses received back.

8.1.2 Design Considerations

The design of our system for synthetic evaluation was largely driven by the design of the Awareness Agent itself, specifically the flow and augmentation [6.6.2.6] of Content Items through a system of queues and services [6.7.4].

The aim within the scope of our research was to evaluate the effectiveness of User-Directed ML classifications [6.4.6] in particular, so the evaluation system discussed here is tailored to evaluate classification decisions that are coded as Simple Classification type augmentations [6.7.6.3].

From a perspective of evaluation, the pertinent elements in this type of classification are:

- The content being classified
- The available classification options
- The selected classification
- The classification criteria

The evaluation system needs to have an understanding of all of the above to evaluate a classification. The first three of these are relatively easy to objectively define; we just need to ensure that the content being classified is available to the evaluator in the same (or equivalent) form to the system being evaluated, and we should be able to codify the available and selected values in a structured format.

The classification criteria are more challenging to formalise; the objective is to mimic what a human evaluator would choose, so it is necessary to have the synthetic evaluation be consistent with whatever that notional human's understanding of the criteria are. This is distinct from mimicking what the evaluated system's understanding of the criteria are – indeed, that system may have no inherent understanding of those criteria, as is the

case with User-Directed ML models (although current LLM implementations are also not considered to form an ‘understanding’ of anything in the same way that a human would – as we discussed in the Literature Review [2.5.2]¹ – but for the purposes of our research when we discuss the understanding that an LLM has of something, we are really talking about how well it functionally mimics understanding based on its internal world model).

We can convey this understanding of the criteria to an evaluating LLM by providing it with information about descriptions of the model and/or the individual classification options, as well as information describing the motivation and priorities of the evaluator (bearing in mind that we would be asking the human evaluator to adopt a persona for their evaluation, we need to ask the synthetic evaluator to do the same thing).

In terms of process flow, the natural point of integration with the Awareness Agent design is at the Allocate juncture [6.7.4.3], which is where the agent is directing augmented Content Items to their output destination(s). At this point we have the option of either initiating an asynchronous evaluation event in the background, or saving the item for later evaluation.

We were also aware that not all LLM models are equal, and that this has been a rapidly developing area. This would mean that quality of synthetic evaluation would potentially vary considerably depending on which model was used. To properly assess this, it would need to be a simple process to perform evaluations against a variety of models and compare the outputs.

8.2 Concepts

8.2.1 Persona

We ask the synthetic evaluator to adopt a Persona [5.3] to conduct its evaluations. Each evaluation task is carried out from the point of view of that one persona. This is also the persona that is adopted by the human participant in each study instance [9.2.2]. It can be thought of as an element of the script that should be followed by both human and synthetic

¹See also <https://neurosciencenews.com/llm-ai-logic-27987> [<https://perma.cc/NMA2-XLR2>]

evaluators in a study.

The persona should be consistent with synthetic content elements such as Topic [7.4.0.3], Dramatis Personae [7.4.0.4] and Simulation Entities [7.4.0.5], which should all reflect elements of the persona's life.

8.2.2 Perspectives

There is more than one way that any given CI can be evaluated. In our case study [9], we were interested in evaluating the operation of the User-Directed ML process [6.7.6], by asking the question: "has this item been classified correctly?". We can consider that this is one *perspective* on the Content Item.

However, this is not the only type of question that could in theory be asked about a CI. For example, we could evaluate other types of augmentation, such as a score (asking "was this item given a correct score?"). It's also possible to evaluate other aspects of a CI, such as such as timing of delivery (asking "was this item delivered at the right moment?"). We can codify these different questions as Perspectives, formalising the process of passing them to the evaluator.

8.2.3 Modes

Our initial work with evaluation did not have any concept of an evaluation mode; this was something that we added as we explored different ways of constructing evaluation requests. An evaluation mode is our way of instructing the system about how we are presenting the data to be evaluated, and what format the response should take. This is then implemented by using specific prompting, request and result schemas, and input data format.

At the outset of our work we used a request/response structure that passed request information including content, the value to be evaluated, and the evaluation persona, and received a Likert scale response. This evolved into a compound request that comprised multiple content items and values to be evaluated for a single persona and perspective in

order to reduce token usage and improve overall efficiency².

We refer to this original evolution of evaluation modes as Mode 1, or *Compound Ordinal Evaluation* [8.2.3.1], reflecting that it has a compound data structure containing multiple personas and items, and that it returns an ordinal (Likert) output. We subsequently introduced two additional modes over the course of our development in order to address issues with the complexity of requests being put to the evaluator, as we discuss in Section 8.3.3.1. This resulted in Mode 2 (*Simplified Ordinal Evaluation*) and Mode 3 (*Simplified Binary Evaluation*) [8.2.3.2]. These modes present the evaluator with a simplified query structure and differ between them in the type of response (Likert vs binary agreement).

We had also found that prompt construction was critical to evaluation performance, something that had also been observed by others in classification tasks [W. Zhang et al., 2024] [Krugmann and Hartmann, 2024]. Adopting a multiple mode approach gave us a framework within which to vary prompts in a cohesive and controlled way, for example by associating specific text elements with modes and selectively including examples for one/few shot analysis.

8.2.3.1 Mode 1 – Compound Ordinal Evaluation

Mode 1 is intended for the most efficient batching of data with minimal prompt and content redundancy. With this mode, the request is a document [8.2.4.1] that contains one persona but potentially multiple perspectives and items. Each item should be evaluated from each perspective for the supplied persona. The result [F.2.5] is a hybrid document containing evaluations for all items for all perspectives.

Due to the complexity of the input, Mode 1 requests also pass the request schema to the evaluating LLM alongside text description in the prompts of how to approach the request. We should note that as of writing, OpenAI does not provide a mechanism in the API for structured requests of this type - while we can and do pass JSON in requests, the API includes no formal support for request schema in the same way it supports structured outputs and result schemas.

²Development Log [D] entry 2023-07-09

8.2.3.2 Modes 2 & 3 – Simplified Ordinal and Binary Evaluations

We are including modes 2 and 3 under the same heading because they are very similar to each other in design, sharing the same basic data structure [8.2.4.2], with the only difference between them being that Mode 2 (Simplified Ordinal Evaluation) requests a Likert type response [F.2.6] while Mode 3 (Simplified Binary Evaluation) requests a binary Agree/Disagree response [F.2.7].

With these modes, instead of passing a complex document that can potentially contain multiple perspectives and requiring the LLM to make the connection for each input item, the data is flattened out so that each combination of persona/perspective/classification is treated as a single self-contained item. This structure is much more intuitive for a human or AI evaluator to understand and process.

These modes also do away with the request schema – while the schema exists, and requests conform to it, this is not passed. Instead free text in the prompts describe the request.

8.2.4 Evaluation Request

Evaluation requests are closely related to mode in that the type of evaluation mode dictates the structure and content of the request itself. There is currently only one type of evaluation request, a Classification Evaluation Request. The Evaluation Request object is a JSON document derived from a Content Item and other elements such as Persona and Perspective.

8.2.4.1 Classification Evaluation Request – Mode 1 (Compound)

The Mode 1 Classification Evaluation Request is generated from Content Items that have had a UD-ML Classification Augmentation applied [6.7.6.3]. The structure for the Mode 1 request is shown in Figure 8.1, and the JSON schema is documented in Appendix F.2.3.

A Mode 1 request item is a one to one mapping of a Content Item, containing the content text for that CI (this is the content that the classifications were based on, that we must

now use to evaluate each classification), and all the applied classifications. The LLM processing the request should use the supplied Persona, and for each Perspective process a set of evaluations for each Item. For each Item, one evaluation should be generated for each included Classification.

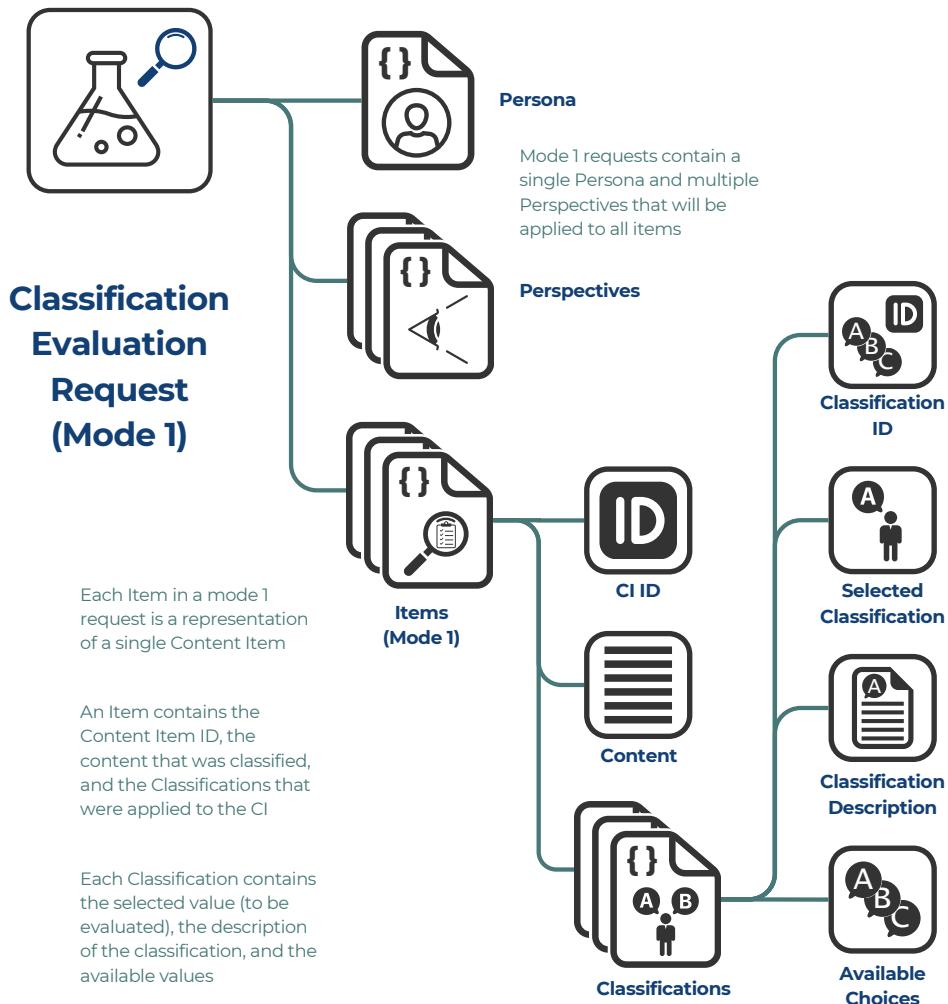


Figure 8.1: Classification Evaluation Request – Mode 1 (Compound)

8.2.4.2 Classification Evaluation Request – Mode 2 & Mode 3 (Simplified)

Modes 2 and 3 use the same Classification Evaluation Request structure. As with Mode 1, these requests are generated from Content Items that have had a UD-ML Classification Augmentation applied. The structure for the Mode 2/3 request is shown in Figure 8.2, and the JSON schema is documented in Appendix F.2.4.

Each Mode 2/3 request is for a single Persona and single Perspective, which differs from Mode 1, which supported multiple perspectives in the request.

Mode 2/3 request items are extracted from a Content Item, with one request Item being generated for each Classification in the CI. This CI-Classification granularity differs from Mode 1, where the top level Item is at the granularity of CI. The Mode 2/3 Item structure is also different. This contains a compound ID that can be used by the processing software to uniquely identify the Content Item, Perspective and Classification that any given item relates to. Each item also contains a copy of the Content Item summary information as well as the classification itself. We chose to copy the content text from the CI to each child item in order to make the evaluation task more simple for the CI – everything that it needs to evaluate is contained in the item. Similarly, we removed the Classification ID from the Classification object and put it in the Compound ID to make it more clear that the ID is not part of the evaluation – the evaluator should use only the Description of the classification to understand it.

8.2.5 Evaluation Response

The Evaluation Response contains the requested evaluations in a structure that is dependent on the evaluation mode. The response object is not constructed by our code, but is instead generated by OpenAI using the schema that we supply in a ‘tools’ parameter. This is described in:

<https://platform.openai.com/docs/guides/function-calling> [<https://perma.cc/3MEF-YVY9>]³.

The evaluation response always contains a value for the Evaluated Selection. This should be the same as the Selected Classification from the evaluation request. We ask the LLM to return this in the result so that we can validate that it matches when processing the result. If they don’t match, that can be flagged as an evaluation failure⁴.

³See also Development Log [D] entry 2023-07-25 and <https://medium.com/@alexanderekb/openai-api-responses-in-json-format-quickstart-guide-75342e50cbd6> [<https://perma.cc/3LNL-YSGP>]

⁴Development Log [D] entry 2023-07-11

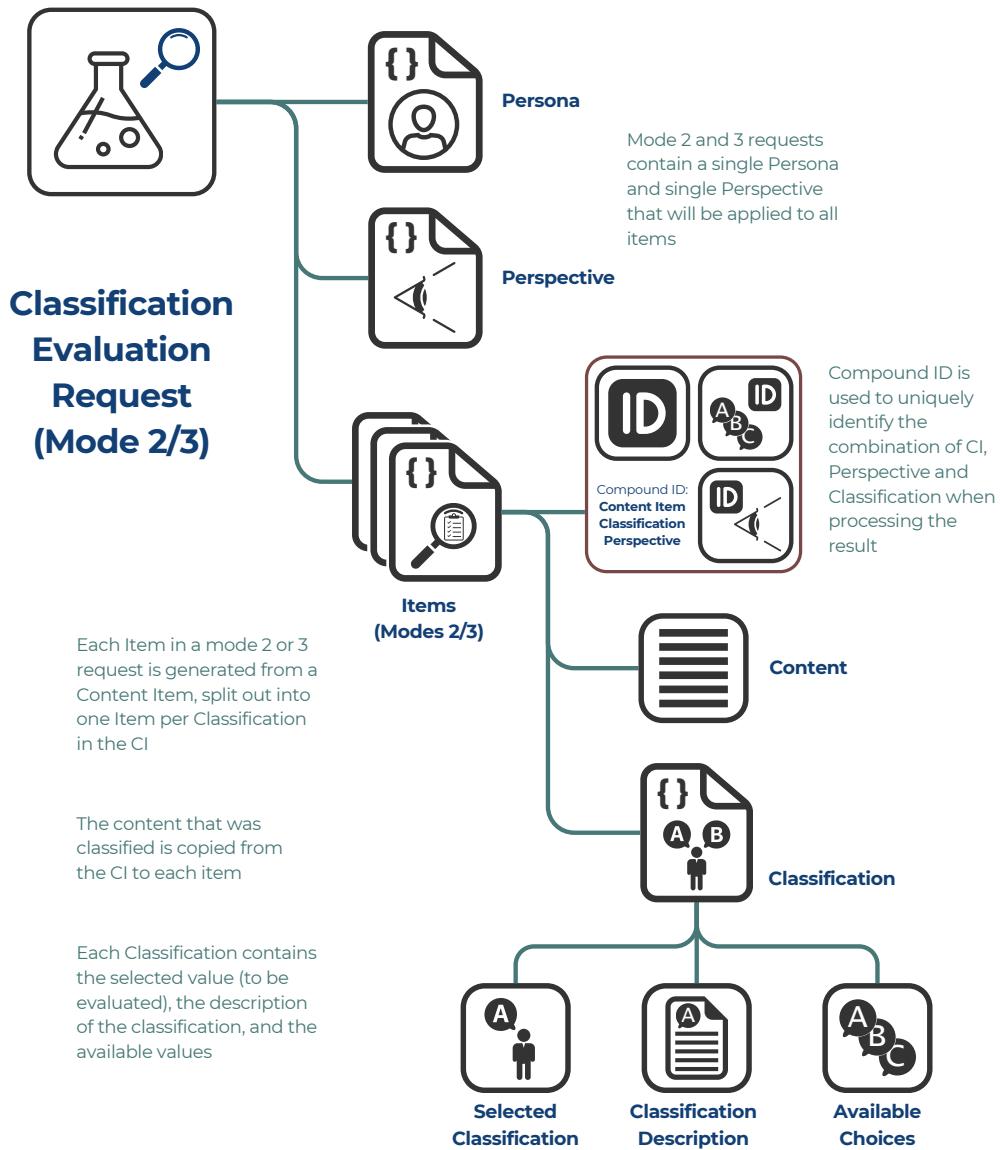


Figure 8.2: Classification Evaluation Request – Mode 2 & 3 (Simplified)

8.2.5.1 Classification Evaluation Response – Mode 1 (Compound)

The Mode 1 Classification Evaluation Response contains a list of items, each of which corresponds to a Content Item in the passed request; each of these items contains a list of Classification Evaluations for that item, each of which contains the Classification ID and the Perspective Evaluations for that classification. The evaluations themselves take the form of a Likert value and a textual description of the evaluation.

The structure for a Mode 1 response is shown in Figure 8.3, and the JSON schema is documented in Appendix F.2.5. Examples of request and response JSON for this mode is linked from Appendix F.4.

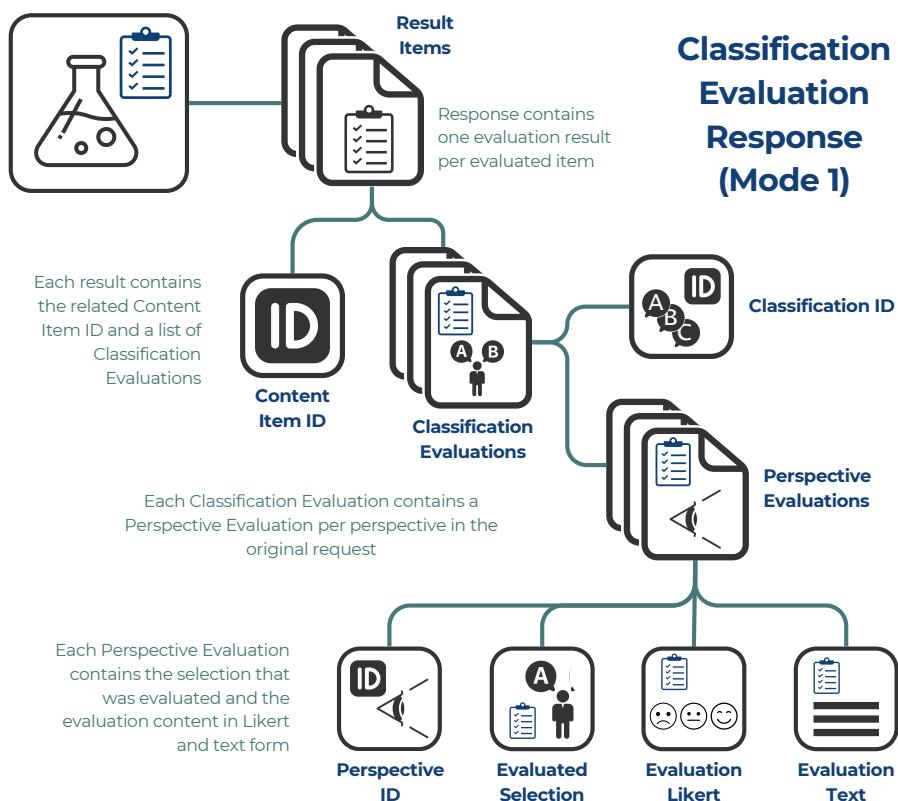


Figure 8.3: Classification Evaluation Response – Mode 1 (Compound)

8.2.5.2 Classification Evaluation Response – Mode 2 & Mode 3 (Simplified)

The Mode 2 and 3 results reflect the simplified nature of the Mode 2 and 3 request types compared to Mode 1. While Mode 1 has nested result arrays for the multiple Perspectives and Classifications passed in the request, Modes 2 & 3 results are a flatter array of items only, reflecting that only one Perspective & Classification are evaluated per item in those modes.

In this case, each item in the response corresponds to a CI-Classification-Perspective combination from the request, and is identified with the same compound ID as was passed in the request. This ID is used by the processing software to reorganise the result data into a form that matches the hierarchical structure of a Mode 1 response, so that all modes of evaluation response can be stored by the Data Service in an equivalent way⁵.

The actual evaluation content in Mode 2 is the same as for Mode 1: Evaluated Selection, Evaluation Likert and Evaluation Text. The difference is that the data is presented in a flatter structure.

Mode 3 responses are identical to Mode 2 except that an evaluation agree/disagree value is returned instead of a Likert.

The structure for a Mode 2 and 3 responses are shown in Figure 8.4, and the JSON schemas are documented in Appendix F.2.6 and F.2.7. Examples of request and response JSON for these modes are linked from Appendix F.4.

8.2.6 Tags

Tagging is widely used across many domains [Kliimask and Nikiforova, 2024] [Y. Zhang et al., 2023] to organise and identify data. We decided to use tags as a way of organising data at various stages as it passes through the Awareness Agent and synthetic evaluation system⁶. Although we use separate instances for different studies/personas [9.2.3], we also saw a need for finer control over identifying content, on both a structured and *ad hoc*

⁵The design of the evaluation storage API in awagdata preceded the introduction of modes 2 & 3, so it expects data in a format compatible with Mode 1, as can be seen in Appendix F.3.1

⁶Development Log [D] entry 2023-09-07

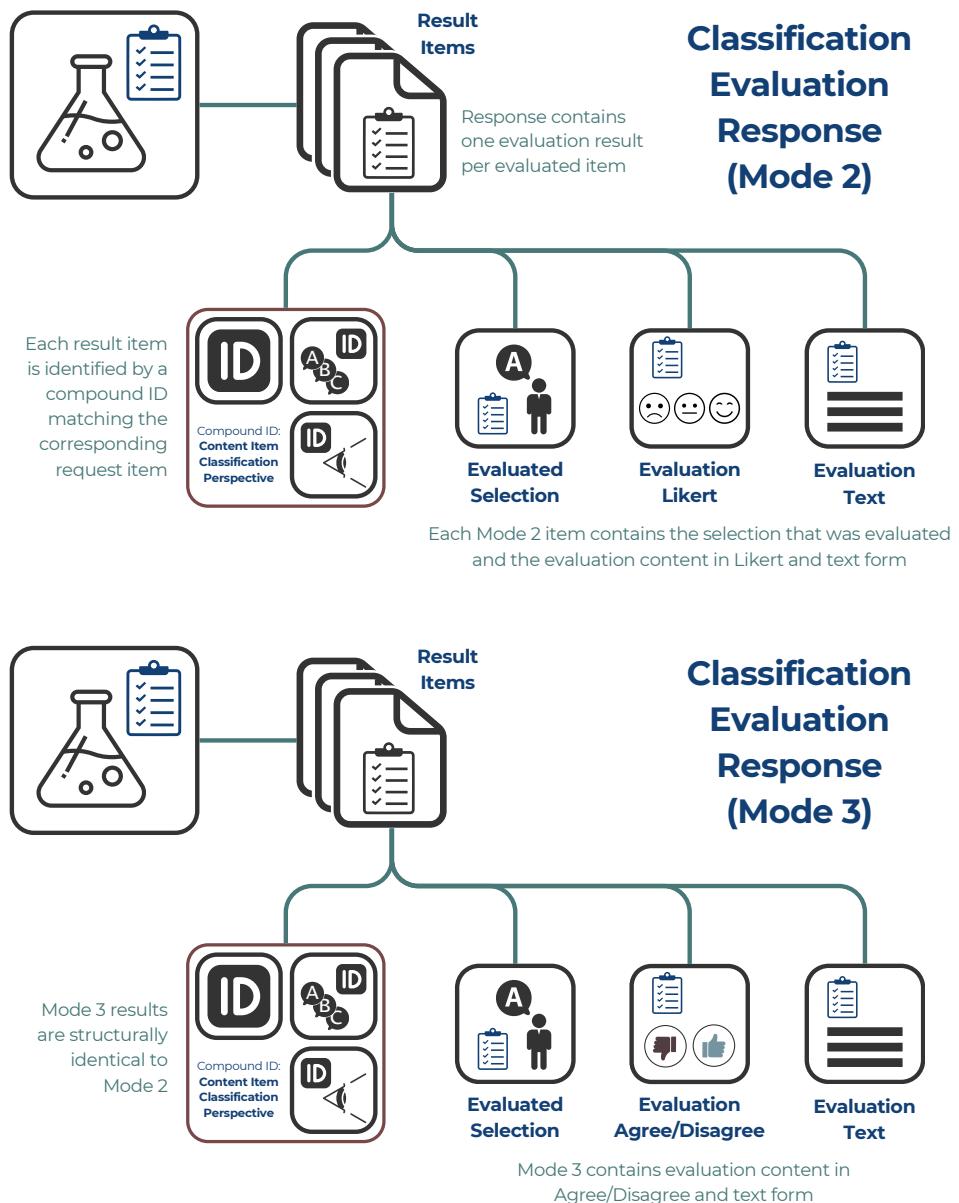


Figure 8.4: Classification Evaluation Response – Modes 2 & 3 (Simplified)

basis. We identified two types of data flow that would benefit from tagging: data coming into the system via the Acquire component [6.7.4.1] (Input tags) and evaluation results (Output tags).

8.2.6.1 Input Tags

We apply one or more input tags to every item that enters the system at the point where we record that item using the Data Service [6.8.3]. As the Acquire process runs in the background, we extended the Awareness Agent functionality to allow the administrator to add and manage ‘runtime’ tags for the current environment. Each item coming in is tagged with the current set of runtime tags. An example of a pair of tags might be `phase2` (to identify the overall phase of a study run [9.3.3]) and `phase2-1` (to identify a specific time period within that phase). This allows us to filter any UI or data processing (such as evaluation) on a given phase or part of phase.

8.2.6.2 Output Tags

Output⁷ tags are those tags assigned to evaluation outputs when they are stored back in the Data Service. We use these tags when analysing evaluation runs; having the ability to specify different tags allows us to run multiple different evaluations against the same set of Content Items – for example using different modes [8.2.3] or models [8.2.8] for processing.

For example, we might use a tag such as `phase2-vanilla-mode2-01` to identify a Mode 2 evaluation using a vanilla model [8.2.8.1] against data acquired during Phase 2 of the study run, with an `-01` suffix indicating that this is the first run of this evaluation (allowing for multiple runs in case we need to re-run for some reason).

By using tags in this way we can more easily manage the experimental inputs and outputs, and have a structure for organising experimental data.

⁷During development we also referred to these as ‘add tags’, which is still used in some places in the code and data

8.2.7 Subsets

In addition to tags, we can control the content being processed for evaluation by using subsets⁸. Subsets are created to contain a percentage of the Content Items matching a given Input Tag. The intent of subsets is to allow us to process or analyse a pseudo-random subset of the overall input data rather than having to process all of it or limit the volume in some other way such as by date.

A subset is stored in the Data Service as a set of Content Item IDs by Percentage Identifier by Tag. For example, an administrator can create a subset 25 for tag phase2 which will contain 25% of the CIs having a phase2 tag at that time. The *identifier* for a subset is the same as the *percentage* of items to be included in that subset – i.e. in this case ‘25’ is both the label and the percentage of the subset.

Because a subset is a collection of CIs, as well as using it to process a subset of input items, it can also be applied to evaluated items. For example, if we select the phase2/25 subset when analysing evaluations for output tag phase2-vanilla-mode2-01, we will see the subset of evaluations for phase2-vanilla-mode2-01 that are in the 25 subset for phase2. In practice we don’t necessarily need to do this, because when processing the evaluation for phase2-vanilla-mode2-01 we could choose the subset at that stage, but it is an alternative way to approach that data.

8.2.8 Models

We tested two types of OpenAI model in synthetic evaluation: untuned (or ‘vanilla’) models such as GPT-3.5 or GPT-4, and fine-tuned models derived from these. Some of our reflections on models can be found in Section 8.4.4.

8.2.8.1 Vanilla Models

Vanilla models are the standard GPT models [Yenduri et al., 2024] deployed by OpenAI. Our intent was that our evaluation process should be able to successfully use such models

⁸Development Log [D] entry 2024-02-23

with no training input other than the information supplied in the request prompt.

8.2.8.2 Fine-Tuned Models

The fine-tuning (FT) process for OpenAI models [Ouyang et al., 2022] [D. M. Ziegler et al., 2019] is documented at:

<https://platform.openai.com/docs/guides/fine-tuning> [<https://perma.cc/CKJ2-LVVK>].

While OpenAI advise that many improvements to model performance can come from improved prompting rather than fine-tuning, they identified that: “Fine-tuning improves on few-shot learning by training on many more examples than can fit in the prompt, letting you achieve better results on a wide number of tasks. Once a model has been fine-tuned, you won’t need to provide as many examples in the prompt. This saves costs and enables lower-latency requests”.

In our case, we wanted to test how well providing a number of example evaluations would improve the quality of evaluation output, both in terms of the correctness of evaluation decisions and the ability of the service to understand the input data structure and respond appropriately. We took a two-pass approach, by first training a ‘base’ model using curated training data, and then by further training an ‘extended’ model using data obtained from human participant classification actions.

8.2.8.2.1 Base Fine-Tuned Models are generated using the fine-tuning process using curated training items. These are to be shared across all personas – that is, the training items in the model are not exclusively associated with from a single persona. All evaluations using a base model use the same model for all personas. As training items would not be persona-specific, they need to be common concepts that are widely applicable – such as items relating to urgency for example.

8.2.8.2.2 Extended Fine-Tuned Models are generated by further training the base models, using training items generated in phase 2 of the study [9.3.5]. Extended models are study/persona specific, as they are trained using data gathered during the course of

that study.

Due to the higher complexity of the compound Mode 1 request [8.2.4.1] and response [8.2.5.1], we limited use of fine-tuning to the simplified modes 2 & 3. Additionally, because the data used to train Extended models was available as agree/disagree only, we limited use of those models to Mode 3, as the sole binary evaluation mode.

8.3 Implementation

8.3.1 Persona

We formulated each defined persona into a JSON document conforming to the schema listed in Section F.2.1. Details of the personas are given in Appendix B.2.

The LLM is instructed within the request prompts [F.1] to use the persona. Depending on the evaluation mode, this may be supported by either a JSON schema describing the persona [F.1.2] or a free text description of the contents [F.1.3].

8.3.2 Processing

This section describes the processing of individual evaluations. These processing tasks can be initiated in two ways - in real time or in batch, which are described in Section 8.3.5.

Figure 8.5 shows how a single Content Item is processed by the Synthetic Evaluation Service. The inputs used in this process are described in Table 8.1.

The Content Item is the variable item for each request – the subject of the evaluation – and content to evaluate is extracted from this using Standard data fields [6.7.1.4] in the same way as was done for a UD-ML request [6.7.6.2]. The fields DATE-SENT and BODY are used for this⁹, as well as the Augmentations themselves [6.7.6.3]. These are combined with the

⁹We considered that other fields such as FROM could prejudice the evaluation, which we wanted to be done on content and context only

Table 8.1: Evaluation Processing Inputs for a Single Item

Input	Description	Reference
Content Item	Item to be evaluated, for example containing Simple Classification augmentations	6.6.2.6
Output Tags	Tags to add to completed evaluations	8.2.6.2
Persona	Persona to run this evaluation for	8.2.1
Perspectives	Perspectives to run this evaluation for	8.2.2
Prompting Components	The text prompting components to be used to construct the evaluation request	8.3.3.3
Mode	Mode for this evaluation	8.2.3
Model ID	Identifier of the LLM model to use for evaluation processing	8.2.8

other elements to form an Evaluation Request [8.2.4].

Each item processed will generate a single compound evaluation response [8.2.5], which is captured using the Data Service for later work [8.3.7]. Content is stored using the passed output tags to facilitate this.

8.3.3 Prompting

This section discusses the implementation of the prompts supplied to OpenAI for evaluation requests, covering the overall development process, parameter refinement, and the construction of prompt elements by request mode.

8.3.3.1 Prompt Development

Prompt development was a significant part of our work, and a key element in our RtD approach. We went through a number of iterations during a develop-test-refine cycle prior to finalising a set of prompts for our study.

We had different types of driver for our eventual prompt design:

- **Quality:** variations in prompt design changed the quality of responses (evaluations) that we generated

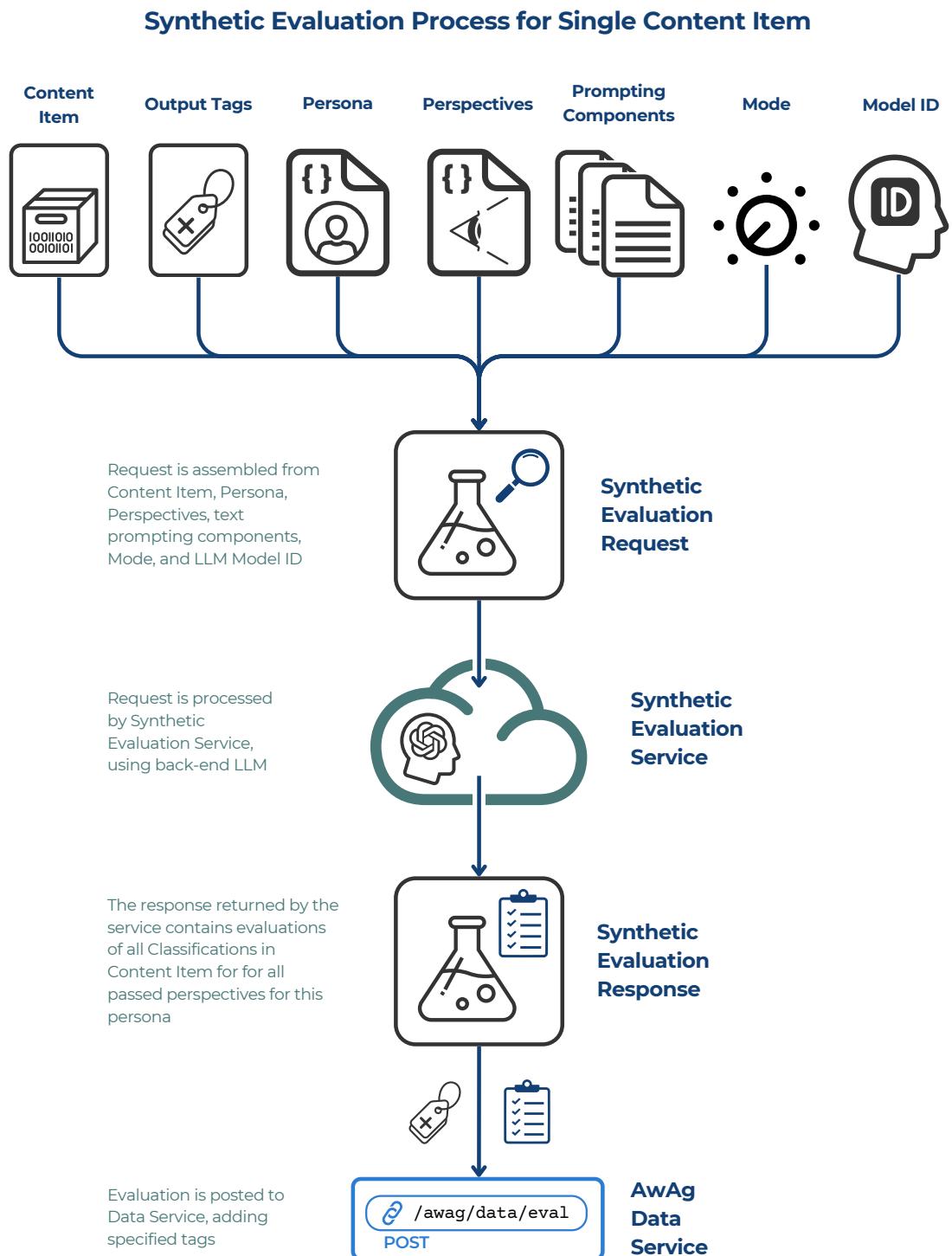


Figure 8.5: Synthetic Evaluation Process for Single Content Item

- **Modularity:** our requests using different modes required variations in prompt content between modes, this led us to develop a modular structure so that common elements could be re-used
- **External factors:** developments external to our work – in particular changes by OpenAI to models and API features resulted in significant changes to our prompt design
- **Efficiency:** we found that some prompts produced good results but were too expensive¹⁰ per query on a token usage basis; this led us towards prompt designs that performed well enough while being relatively economical¹¹

One example of API changes affecting our prompting development is recorded in Development Log [D] entry 2023-07-25, where we changed our prompt approach to use a newer API from OpenAI and the models that this enabled.

The Development Log also highlights some work relating to quality improvements (although this was really a continual process over the whole development cycle): 2023-06-14, 2023-06-26 and 2023-10-15 for example.

We later made further changes to the structure to support our work with fine-tuning models [8.3.4], where we found that modularity was particularly important: the training artefacts used to fine-tune models shared a lot in common with evaluation prompting, but with some differences relating to the mechanics of the fine-tuning process. Using a modular approach to prompt construction helped us avoid redundancies and reduce the risk of different versions of prompt text being inconsistent or even in conflict with each other.

8.3.3.2 Parameters

In a similar way to which we had experimented with differing OpenAI query parameters for content generation [7.4.0.6], we also went through an evolution process with the parameters that we applied to evaluation queries¹². In particular, we found that varying

¹⁰<https://www.wingback.com/blog/should-ai-founders-use-the-same-pricing-model-as-openai> [<https://perma.cc/P78B-FL7N>]

¹¹See also Supplement S11.3 [doi:10.21954/ou.rd.28045580] on token usage in our Study

¹²<https://platform.openai.com/docs/api-reference/chat/create> [<https://perma.cc/GZZ3-ZUWD>]

`presence_penalty` and `temperature` caused divergence of quality of response – but we eventually settled on using the default values for these (0 and 1 respectively), as well as the other parameters.

8.3.3.3 Prompt Elements

Evaluation requests are made to the OpenAI API using a set of System and User messages¹³. While there are some common elements, the components of each request vary according to mode. Requests start with System messages to prime the LLM to the general nature of the task. These comprise a message element that is common to all modes [F.1.1], and a mode-specific element [F.1.2, F.1.3].

The user messages are more mode-specific, with some shared elements where appropriate (for example, modes 1 & 2 are expected to return a Likert scale result, and these have a specific section of text added to describe that [F.1.4]). Finally, the request JSON itself is included as a User message, with a different structure for Mode 1 [8.2.4.1] and Modes 2 & 3 [8.2.4.2].

The static text elements used to construct prompts are listed in Appendix F.1. The prompt composition for the three evaluation modes is shown in tables 8.2, 8.3 & 8.4.

Table 8.2: Prompting Messages for Mode 1 (Compound Ordinal Evaluation)

Message Type	Message	Reference
System	System Message – Common	F.1.1
System	System Message – Extra (Mode 1)	F.1.2
System	Evaluation Request Schema	F.2.3
User	User Message Prefix (Mode 1)	F.1.5
User	User Message Likert (Mode 1/2)	F.1.4
User	Evaluation Request JSON (Mode 1)	8.2.4.1

¹³<https://platform.openai.com/docs/guides/text-generation> [<https://perma.cc/PV2K-7wdx>]

Table 8.3: Prompting Messages for Mode 2 (Simplified Ordinal Evaluation)

Message Type	Message	Reference
System	System Message – Common	F.1.1
System	System Message – Extra (Mode 2/3)	F.1.3
User	User Message Prefix (Mode 2)	F.1.6
User	User Message Likert (Mode 1/2)	F.1.4
User	Evaluation Request JSON (Mode 2/3)	8.2.4.2

Table 8.4: Prompting Messages for Mode 3 (Simplified Binary Evaluation)

Message Type	Message	Reference
System	System Message – Common	F.1.1
System	System Message – Extra (Mode 2/3)	F.1.3
User	User Message Prefix (Mode 3)	F.1.7
User	Evaluation Request JSON (Mode 2/3)	8.2.4.2

8.3.4 Fine-Tuning

Early experiences with ‘vanilla’ or untrained OpenAI models showed promise, but we also wanted to explore how fine-tuning [8.2.8.2] would influence the quality of output¹⁴. We hoped that fine-tuning would allow us to better explain the task required of the evaluator with examples, while minimising token usage.

Our approach was to assemble a set of example evaluation requests and responses and use these to generate fine-tuning training data, as per OpenAI’s specifications¹⁵. We would then test these fine-tuned models alongside vanilla models.

The method of generating training content differs depending on whether the model is a Base [8.3.4.1] or Extended one [8.3.4.2].

Similarly, the composition of training messages depends on whether the model is for or-

¹⁴Development Log [D] entry 2023-12-08

¹⁵<https://platform.openai.com/docs/api-reference/fine-tuning> [<https://perma.cc/KGA6-XWMD>]

dinal (Mode 2) or binary (Mode 3) usage. Each training item consists of a number of messages (similar to evaluation queries themselves [8.3.3.3]), the composition of which depends on whether it is Base or Extended, and Mode 2 or Mode 3:

- **System** – as with request prompting, the system message sets the scene for the task. We use the same common text for this as for evaluation requests [F.1.1] (the commonality assists the LLM in making use of the fine-tuning¹⁶)
- **User** – the basic evaluation request instruction and the request JSON
- **Assistant** – the response that is expected for this request (the training content)

Base training items can be either Mode 2 or Mode 3 – the former trains a Likert response while the latter trains an agree/disagree response. Table 8.5 shows how a Base training item of Mode 2 is created, using a similar approach to the one taken with general prompting [8.3.3].

Table 8.6 shows how a Mode 3 training item is created. This can be either a Mode 3 Base item, or an Extended item (which are only ever Mode 3). The structure is the same for both, only the mechanism for generating the content differs.

In the case of training items for fine-tuning, the Evaluation Request part is generated or hand edited to be a ‘representative’ example request¹⁷, while the Assistant message is the example that trains the model, which may also be generated or edited by hand.

Table 8.5: Messages for Base Training Item (Mode 2 Only)

Message Type	Message	Reference
System	System Message – Common	F.1.1
User	User Message Prefix (Mode 2)	F.1.6
User	Evaluation Request JSON (Mode 2/3)	8.2.4.2
Assistant	Training Response JSON (Mode 2)	F.2.6

¹⁶<https://community.openai.com/t/do-the-system-messages-in-gpt-3-5-turbo-fine-tuning-need-to-be-the-same-for-all-entries/466012> [<https://perma.cc/M5K4-UMPV>]

¹⁷For Base training items, this is a subjective decision (albeit based on sifting through real examples), while Extended items are derived from actual requests so are as representative as the source data

Table 8.6: Messages for Base Training Item (Mode 3) and Extended Training Item

Message Type	Message	Reference
System	System Message – Common	F.1.1
User	User Message Prefix (Mode 3)	F.1.7
User	Evaluation Request JSON (Mode 2/3)	8.2.4.2
Assistant	Training Response JSON (Mode 3)	F.2.7

8.3.4.1 Base Fine-Tuned Models

As we noted in Section 8.2.8.2.1, Base fine-tuned models are a first pass fine-tuning implementation, and we used curated content for this. That is, we manually edited a set of example evaluation requests and responses, and used these with the prompt structure described in Tables 8.5 and 8.6

While this content was designed to be persona-neutral, we used content from several personas in our process of generating training items.

The content that we used for base fine-tuning is located in:

doi:10.21954/ou.rd.28044944 [path: /eval/fine-tuning/base].

8.3.4.2 Extended Fine-Tuned Models

Extended models, as introduced in Section 8.2.8.2.2, build on Base models, adding a set of training items based on real data gathered during our study [9.3.5]. Because the items were created during the course of a study – which is persona-specific – the Extended models are also persona-specific.

Because the data captured from participant actions during the study is effectively a binary AGREE/DISAGREE¹⁸, only Mode 3 is supported for Extended model training and evaluation – hence used only the structure shown in Table 8.6.

We also made use of different subsets when building the datasets to train the extended

¹⁸A user-selected classification that matches the UD-ML classification value is considered an ‘Agree’, while one that does not match is treated as a ‘Disagree’

models¹⁹. This was to allow us to balance the number of Agree and Disagree items in the final merged dataset²⁰.

Examples of Mode 3 Extended training items are included in Appendix F.6.1.

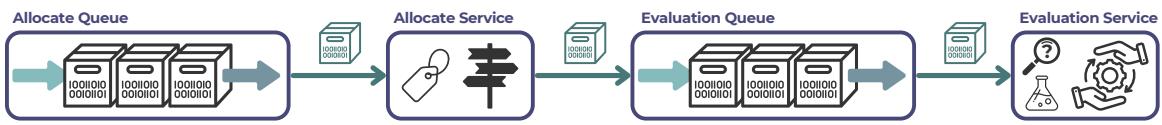
8.3.5 Integration

This section discusses how the evaluation functionality was integrated into the Awareness Agent Services, Queues & Engines architecture [6.7.4] and overall application structure [6.8.1.2].

8.3.5.1 Real-Time Processing

Figure 8.6 shows the flow of Content Items to the Evaluation Service within the Awareness Agent. This mode of operation – which processes evaluations for items in real time, as and when they come in – was used less frequently than the batch processing alternative.

Figure 8.6: Flow of Content Items from Allocate to Evaluation Services



Flow of Content Items from Allocate to Evaluation Services Within Awareness Agent

Figure 8.7: Storage of Content Items for Later Processing



Storage of Content Items for Later Processing Within Awareness Agent

¹⁹Development Log [D] entry 2024-05-02

²⁰Development Log [D] entry 2024-05-02 (2)

8.3.5.2 Batch Processing

Figure 8.7 shows the process of storing representations of Content Items for later evaluation, performed by the Allocate Service [6.7.4.3]. These items are posted to the /data/eval API of the Data Service [6.8.3]. Items stored in this way are saved associated with one or more tag strings [8.2.6]; these tags are set by the administrator at runtime – each item processed will be stored against the currently active set of tags²¹.

Figure 8.8 shows batch processing as initiated by an administrator, who supplies the service with an input tag [8.2.6.1] and subset [8.2.7] to specify a set of Content Items to process²². The corresponding CIs are then processed in turn, using the Persona, Perspectives, Model and Mode specified by the administrator. Results are then stored back in awagdata using the output tags also specified by the administrator [8.2.6.2].

8.3.6 Data Service

The following awagdata routes [6.8.3] are part of the Evaluate implementation, as detailed in Table 6.4:

- /data/eval
- /data/fixit
- /data/gentrain
- /data/subsets
- /data/train

²¹See also Development Log [D] entry 2023-09-07

²²See also Development Log [D] entry 2023-09-12

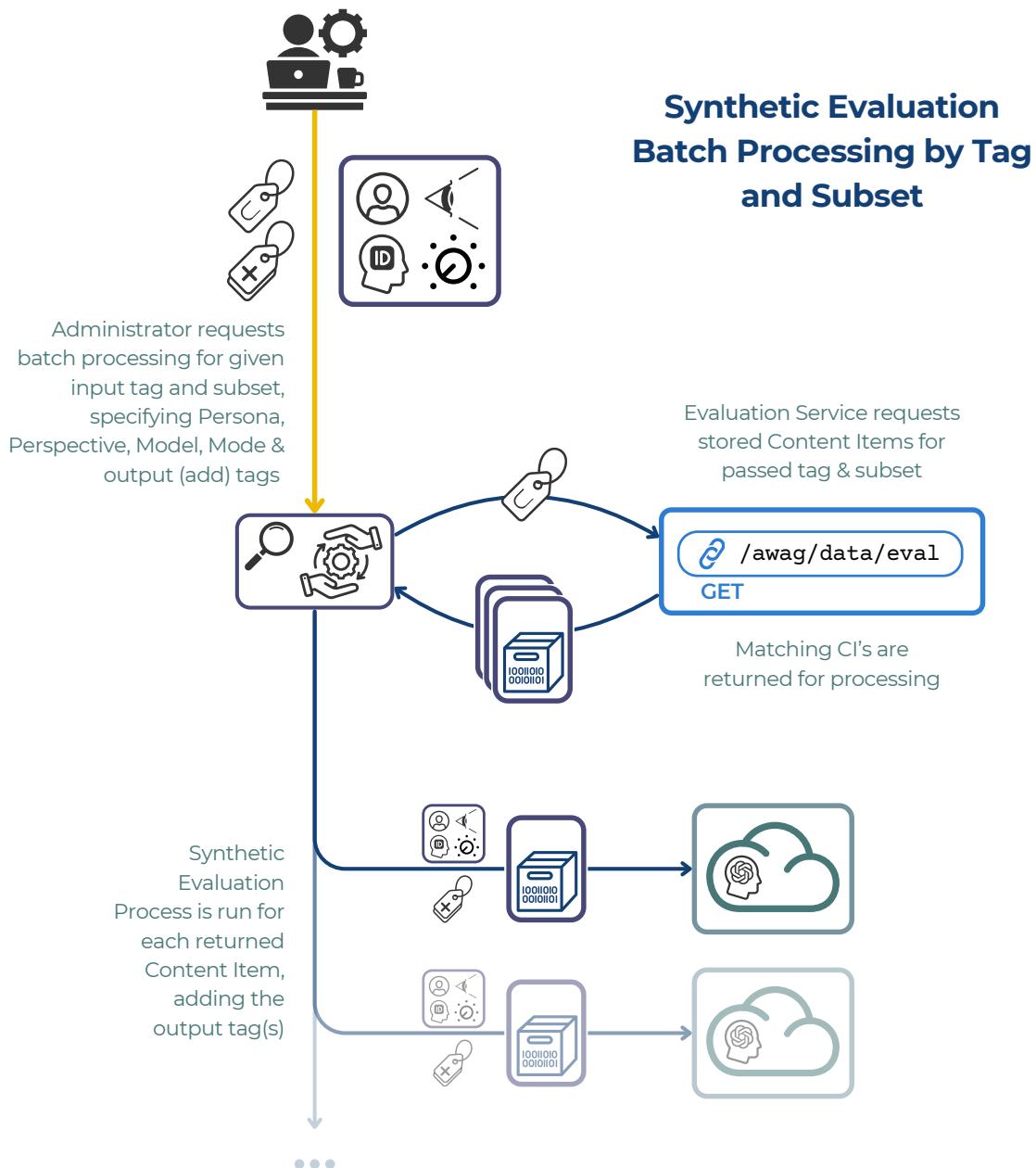


Figure 8.8: Synthetic Evaluation Batch Processing by Tag and Subset

8.3.7 Data Capture

The priority for our design for data capture in synthetic evaluation was to ensure that we covered the following areas:

- **Scope:** The data recorded should be enough to support all evaluation needs.
- **Failure handling:** By recording all fail conditions, the inputs and raw responses, and also flagging failed items as still needing processing.
- **Reproducibility:** By recording all parameters, generated text, and original items in as complete format as possible, we aimed to make it possible to re-run any aspect of the evaluation process at a later date.
- **Accounting:** So that we can track metrics such as token usage and the number of items processed.

Some of the main JSON schemas used for interacting with awagdata are included in Appendix F.3.

8.3.8 User Interface Illustration

Illustrations of the Training UI and Evaluation UI can be found in the Study chapter – in sections 9.4.1 and 9.4.2 respectively.

8.4 Reflections

8.4.1 Initial Findings

The findings discussed here relate to the RtD process for developing synthetic evaluation; the data-related findings are covered later in the Study chapter and overall conclusions.

8.4.1.1 Pace of Change

As we discussed in Section 8.3.3.1, prompt development was a significant part of our work, evolving over time. This evolution was directed by a combination of factors internal and external to our work. We found that the rapid pace of change of OpenAI model and API capabilities both helped us by addressing some of our pain points while also making more work for us to keep up with the changes.

These changes have continued to happen at pace while we have been documenting our research. For example, Structured Outputs²³ were introduced by OpenAI in August 2024 partly to address issues with the previous type of JSON outputs that we were using in our code. We had experienced some of the problems described, such as unreliability of JSON responses and data extraction issues. However, due to the timing of this we did not implement such new features in our code.

8.4.1.2 Processing Integration

It was important for us to be as consistent as possible with the Awareness Agent platform design discussed in Section 6.6.5, so our original implementation of evaluation was entirely integrated into this structure. However, we found during the course of development and running the study that this was not optimal – for example we found that integrating evaluations into the agent CI flow [6.7.4] was not the best approach, as we often needed to run multiple different evaluations and re-run jobs. The solution we ended up with, where items were recorded in `awagdata` for later evaluation in a standalone process, worked much better in practice, and with hindsight we would have taken this approach from the start.

8.4.1.3 Evolution from Mode 1 to Modes 2 and 3

Our evolution from original Mode 1 (Compound Ordinal Evaluation) to also using modes 2 & 3 (Simplified Ordinal & Binary Evaluations) is also something that we would likely have

²³<https://openai.com/index/introducing-structured-outputs-in-the-api/> [<https://perma.cc/DML6-C74X>]

done differently with the benefit of hindsight. Our original intent with the design was maximum flexibility and abstraction, with the perspectives system being used to allow maximum flexibility over what an evaluation looks like. However, in terms of evaluating simple classifications there is less utility in having perspectives [8.4.1.5] – because there is only really one question to answer – and the complexity these add to both the LLM query and the data handling make it less worthwhile to use in this case. The complexity on the query side is a main reason behind changing the structure for modes 2 & 3, and we would contemplate changing entirely to this different structure (and updating the back end awagdata to match).

8.4.1.4 Multiple Content Items in Requests

Our original design for a Mode 1 Classification Evaluation Request [8.2.4.1] allowed for multiple CIs within the single request. This was to allow more efficient processing by reducing duplication of prompting, personas and perspective information. However, in practice we only ever used a single CI per request, as we found that the LLM did not process multiple requests well and was more prone to error. Similarly modes 2 and 3 requests [8.2.4.2] could also be generated from multiple CIs, but we limited these to a single CI.

8.4.1.5 Perspectives

As noted in Development Log [D] entry 2023-06-23, we found that we had only limited success with using perspectives in our evaluations, the main stumbling block being that there was only really one question to ask in our study: “has this item been classified correctly?”. However, we believe that this concept has merit in the case where evaluation is expanded to include other types of augmentation such as scoring (“was this item given a correct score?”). Additionally, if aspects such as timing of delivery were included in the evaluation, this might also lead to valid perspectives (“was this item delivered at the right moment?”). Perspectives remain a useful way of formalising the evaluation requirements for this type of content in cases where multiple questions could be asked.

However, we did find in our testing that the Mode 1 approach to perspectives – including all perspectives within a single evaluation item – did not function well, with the LLM often generating incorrect answers for the different perspectives. This was one reason why Mode 2 & 3 requests [F.2.4] were changed to that each individual evaluation item only had one perspective, which was included as a simple text field. So aside from finding that we only needed one perspective in our testing, we also made changes to ensure that only one perspective would feature in any one request.

8.4.2 OpenAI API Parameter Variation

As we noted in Section 8.3.3.2 we eventually settled on default values for OpenAI parameters. We did not formally document the process of testing the different variations tried, nor log those results. While the parameters that we settled on passed our ‘good enough’ test, there would certainly be scope for running a set of tests comparing the same evaluation items with different API parameter values (and also with different prompting variations) in a similar way to which we ran tests with using different models and modes for the same corpus of evaluation items.

8.4.3 Personas in Fine-Tuned Training Sets

Our approach to fine-tuned models [8.3.4] used a system whereby Base FT models were trained using a generic data set that is not explicitly persona-specific. In reality, this data set was created semi-manually from an amalgam of data across several personas, and are given as an example of how evaluation should be conducted based only on the topic, the content and the available classifications. Specifically the training data does not ask the LLM to adopt a persona.

The Extended models on the other hand do use persona-specific data, so can be thought of as training the model with how that specific persona would evaluate that content. We should bear this in mind when analysing comparative results between base and extended models.

8.4.4 OpenAI Model Selection

Our use of OpenAI models²⁴ evolved during the course of our work as new models and functionalities were introduced by OpenAI.

As was the case with synthetic content [7.6.4], initial work on the evaluation side used OpenAI as a provider, initially with the GPT-3 `text-davinci-003` model. The mature OpenAI API was again a significant factor in this choice.

We found in that the evaluation workload was much more sensitive to model quality than content generation, so evaluation quality was one of the main drivers for updating to use the OpenAI `/chat/completions` API, and the GPT-3.5 and GPT-4 models that this change made available to us²⁵.

API access for GPT-4 was made generally available in April 2024²⁶ – although we did not change our simulated content development to use this version, we did use it for evaluation alongside 3.5. Indeed, comparison between evaluations performed using GPT-3.5 and GPT-4 would be a significant focus of our study.

Our fine-tuning work was only able to make use of GPT-3.5. While fine-tuning of GPT-4 was available on a limited access programme, we did not have access during our study development and a fine-tunable version of GPT-4 was only made generally available in August 2024²⁷.

²⁴<https://platform.openai.com/docs/models> [<https://perma.cc/9UL2-BX5D>]

²⁵Development Log [D] entry 2023-07-25

²⁶<https://openai.com/index/gpt-4-api-general-availability/> [<https://perma.cc/6GAX-K4YM>]

²⁷<https://openai.com/index/gpt-4o-fine-tuning/> [<https://perma.cc/8DZM-9CA8>]

8.5 Chapter Summary

In this second precursor chapter to our study, we documented the design and implementation of a system for synthetically evaluating elements of the Awareness Agent that will go on to be a core part of the final study. This included:

- Approach [8.1]
- Concepts [8.2]
- Implementation [8.3]

We also documented our reflections on the process [8.4], and included an abridged version [D] of the full Design & Development Log [D], containing the more significant entries that related to the work covered in this chapter. The next chapter [9] documents the study itself.

Chapter 9

Awareness Agent Prototype Study

9.1 Overview

This chapter reports on the evaluation of components described previously: primarily the Awareness Agent itself, and secondarily our synthetic content/evaluation systems. While the latter of these aspects arose from practical considerations we encountered in evaluating the former [7.1], we believe that it has independent merit to study. In this chapter we discuss the study that we developed and ran to give insight on this questions.

9.1.1 Studying the Awareness Agent

We previously ran a survey to help us with the research question of information overload [3.2.1], which highlighted some ways in which users experienced IO [5.1.3.2] and established that in many (although not all) cases, there was a problem that could be addressed.

We proposed a concept for an Awareness Agent [6.1] that could potentially alleviate issues with information overload, as well as offering other advantages to the user [6.4.1]. By studying this, we hoped to find answers for our research questions on solution development [3.2.2].

The survey also led to the development of a number of personas [5.2] through which we could evaluate the proposed solution [6.2]. In our case the personas informed our

decisions when designing each instance of the study - specifically data sources, ML models and participant expectations. We will expand on this further below. These personas would also form a basis for the synthetic evaluation process that we discuss below.

We designed our study around the actual prototype implementation of the Awareness Agent that we had built during our research [6.9.1]. Specifically, we designed the study to focus on the User-Directed ML implementation [6.7.6], concentrating on examining the effectiveness of UD-ML models with a combination of real and simulated data, through a lens of the personas that we had developed. This examination was content-focussed, limited to those areas that we had identified as being *met* in our gap analysis of the prototype [6.9.2].

Our general approach was to run an instance of the Awareness Agent prototype for each persona, defining a number of synthetic and real data sources and the UD-ML models to analyse these. The choice of data sources and models was driven by the persona scenarios¹. We would then run these instances over a time period, with one participant interacting with each, having been asked to take on the role of the related persona. We gathered data on classifications made by both the participant and UD-ML, conducted a synthetic evaluation of the UD-ML decisions, and finally examined the synthetic evaluation using participant input.

9.1.2 Studying Synthetic Content and Evaluation

The other aspect of the study was designed to answer the question of whether simulated content could be used in evaluation [3.2.3], but in practice it flows from the design of the study to assess the agent described above. This happens where synthetic content is introduced to provide study data, and we gain insight on this from user feedback, and also during the evaluation process. For this, we introduced an additional step to the study – evaluation feedback – where the participant is asked to review and rate or correct the output of the synthetic evaluator. Additional insight was also gained by comparing participant training decisions during earlier study phases with synthetic evaluation decisions.

¹See Supplement S5 [doi:10.21954/ou.rd.28045460]

9.2 Design

The design discussed in this section aligns with a Study Protocol document that we produced at the outset of the study process [G.1].

9.2.1 Participant Recruitment

We aimed to recruit a small number of people who agreed to donate a relatively significant amount of time to the task, with a preference for individuals with a relatively sophisticated understanding of information systems and social media applications. As the number of study instances was so small ($n=5$), our pool for recruitment was personal contacts.

9.2.2 Persona

As noted above, the participant was expected to adopt a persona for their study instance [5.3], which would also be applied for the synthetic evaluation associated with their instance [8.2.1, 8.3.1]

9.2.3 Instances

In order to allow multiple concurrent study runs while isolating these from each other, we ran each study instance in a dedicated Awareness Agent instance² with a unique ID [G.2].

9.2.4 Modes

Data on the relative performance of the three evaluation modes introduced in Section 8.2.3 can be obtained in two ways:

- Modes 1 and 2 (Ordinal): manual feedback from the study participant [9.2.5.5]
- Mode 3 (Binary): automatic rating based on training data from the participant [9.2.5.4]

²See Development Log [D] entry 2023-03-10

Analysis data for Mode 3 is derived from the correlation of each Mode 3 evaluation's agree/disagree rating with the implied value given by the user when they trained the models (leaving a classification unchanged in a training action implies an Agree, while changing it implies a Disagree). Provided that the user trains a sufficient number of items that go on to also be evaluated, this implicit data can be used to analyse the synthetic evaluation.

9.2.5 Stages

9.2.5.1 Planning and Preparation

To prepare for the study we needed to establish our final study design, prepare information for participants and set up the technical aspects:

- Finalise study design and methodology
- Prepare and test the technical infrastructure:
 - Provision server VM to support study [6.8.1.1]
 - Set up awagml [6.8.2] and awagdata [6.8.3] services
 - Set up web server and reverse proxy to route external requests to study services

9.2.5.2 Participant Onboarding and Persona Development

Onboarding study participants involved two stages:

- Conduct initial meetings with recruited participants to discuss the study and obtain informed consent
- Collaboratively develop persona configurations with participants and identify appropriate public and synthesized data sources

9.2.5.3 Data Source Definition and Synthesis

Data sources (both synthetic and non-synthetic) and UD-ML configuration were specific to each study instance. The steps were:

- Finalise the selection of public data sources [6.7.4.1] and define synthesized data requirements
- Generate synthetic content [7.4] and store it in the awagdata system
- Define and set up the UD-ML models [6.7.6.1] for each persona

9.2.5.4 Execution – Data Collection and Model Training

During the first Execution stage, the participant was actively involved on a daily basis:

- Begin data acquisition through the Awareness Agent
- Participants engage in model training through manual classification of incoming items [9.4.1]
- Use awagdata to continuously monitor and record data interactions and classifications

9.2.5.5 Execution – Synthetic Evaluation and Participant Feedback

The second Execution stage involved two steps in sequence:

- Study administrator conducts batch evaluations of ML model classifications [8.3.2] using both vanilla and fine-tuned OpenAI models [8.2.8]
- Participants provide feedback on OpenAI's evaluations through the evaluation-feedback interface [9.4.2]

9.2.5.6 Data Analysis and Reporting

The final stage, conducted by the study administrator, assembled the data for the study:

- Analyse the collected data for each study instance, focusing on ML model accuracy, OpenAI evaluation concordance, and user experience
- Conduct interviews with participants for qualitative feedback
- Produce & analyse consolidated data across instances

9.2.6 User-Directed ML Initialisation

UD-ML models are initialised on creation by awagml to contain the required classifications, but have no training data to start with. Due to this, a short process of initialisation is required to make the model minimally functional. This process was conducted with the participant in the early part of the execution stage – the standard training process was used [9.2.5.4], but with data tagged separately to exclude it from evaluation.

9.2.7 Statistical Techniques

We designed the study process to generate a set of statistics automatically, using the awagdata /data/reporting and /data/stats services [6.8.3] to process data collected during the study phases. The output of these is a set of study instance-specific statistics packs in JSON and Microsoft Excel format, which can then be combined to produce statistics covering all study instances. Source code for the statistics generation and other aspects is covered in Supplement S12 [[doi:10.21954/ou.rd.28045598](https://doi.org/10.21954/ou.rd.28045598)].

A mix of methods and data combinations was selected with the aim of gaining insight into the quality of the classifications by the UD-ML models and of the synthetic evaluations. Because the classifications, training actions and different modes of feedback output different combinations of categorical and dichotomous data, we needed to employ specific methods for each, as detailed below.

9.2.7.1 Evaluation Score

The Evaluation Score is the metric by which the Synthetic Evaluator rates a UD-ML classification value and is derived from the Evaluation Response [8.2.5].

For **Ordinal (Mode 1 & Mode 2)** evaluations, a score is assigned based on the Evaluation Likert value:

- Strongly Disagree: 0
- Disagree: 0.25
- Neutral: 0.5
- Agree: 0.75
- Strongly Agree: 1

For **Binary (Mode 3)** evaluations, an Evaluation Agreement value of 'Disagree' is awarded a score of 0, while 'Agree' is awarded 1.

9.2.7.2 Evaluation Percentage

This is a simple percentage of evaluated items where the evaluation is considered to agree with the UD-ML classification:

For **Ordinal (Mode 1 & Mode 2)** this is defined as the percentage of evaluated items having an Evaluation Likert values 'Agree' and 'Strongly Agree'.

For **Binary (Mode 3)** evaluations, this is the percentage of evaluated items having an Evaluation Agreement value 'Agree'.

9.2.7.3 Manual Classification Agreement – Percentage

The metric *Percent Of Classification Manual Agrees* is a measure of the quality of the UD-ML classification of items by model. This metric is obtained by comparing the classification decision made by UD-ML for an item with the Phase 2 training actions [9.3.3.1] made by the user, where such data exists. Not all items processed by the Awareness Agent will

have such records, because the participant was not expected to train/classify every item, but the number of such items was expected to be sufficient to enable this analysis.

The components of the metric are:

- `count_items` – Number of items in the dataset having a manual classification record (the numerator in the percentage calculation)
- `count_agree` – Number of items in the dataset having a manual classification record in agreement with the ML classification (the denominator in the percentage calculation)
- `percentage_agree` – Percentage of records in dataset having a manual classification, which agrees with the ML classification

Because this metric is derived from only the UD-ML classifications and the participant actions, it is independent of the synthetic evaluation process.

9.2.7.4 Manual Classification Agreement – Cohen's Kappa

The metric *Cohen's Kappa For Manual Classification Agreement* is an alternative to the percentage [9.2.7.3] to measure the quality of the UD-ML classification. This applies Cohen's Kappa (κ) [J.A. Cohen, 1960]³ to measure the agreement between the original ML classification and the manual classification.

9.2.7.5 Evaluation Feedback Correlation Coefficient

The metric *Pearsonr For Eval Feedback* is a measure of the quality of ordinal (modes 1 & 2) synthetic evaluation [8.2.3]. Pearson's correlation coefficient (r) [Pearson, 1895]⁴ was used to measure the strength of the linear relationship between the Likert value selected by the synthetic evaluator [8.2.5.1] and the Likert selected by the study participant during the feedback stage [9.2.5.5].

³<https://www.statology.org/cohens-kappa-statistic/> [<https://perma.cc/R5F9-5QCA>]

⁴<https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html> [<https://perma.cc/3Y7X-CETA>]

9.2.7.6 Evaluation Agreement Correlation Coefficient – Point-Biserial

The metric *Pointbiserialr For Eval Agreement* is an alternative measure of the quality of ordinal (modes 1 & 2) synthetic evaluation. In this case, the feedback is not used, and instead the Phase 2 training actions classification decision [9.3.3.1] made by the user is used to assess the quality of the synthetic evaluator's Likert selection [8.2.5.1].

In this case, the ordinal Evaluation Likert value is compared with the dichotomous (binary) Classification Manual Agrees value, which is derived from the agreement between the UD-ML classification and the training action in the same way as for 9.2.7.3. Because this metric compares a dichotomous classification agreement value with the ordinal evaluation likert, it is calculated as a point-biserial correlation coefficient (r_{pb}) [Jacob Cohen et al., 2002]⁵.

9.2.7.7 Evaluation Difference Average and Spread

The metric *Avg And Spread For Eval Difference* is another measure of the quality of ordinal (modes 1 & 2) synthetic evaluation. This is derived by treating the synthetic evaluation [8.2.5.1] and feedback [9.2.5.5] Likert values as integers in the range 1-5, then calculating the difference between these values. If the synthetic evaluation and the participant feedback are in perfect agreement, this value would be 0, while the maximum value for disagreement is 4 (5 minus 1).

The metric consists of mean, median and modal values of the difference value, by UD-ML model and tag, with standard deviation also calculated.

⁵<https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pointbiserialr.html> [<https://perma.cc/BJ4S-HXGY>]

9.2.7.8 Mode 3 Agreement – Phi Coefficient

The metric *Phi Coefficient For Mode3 Agreement* is a measure of the quality of Mode 3 evaluations, which are a binary Agree/Disagree value rather than a Likert [8.2.3.2]. As with Manual Classification Agreement [9.2.7.3], this measure is calculated using the Phase 2 training actions [9.3.3.1] rather than the evaluation feedback – but in this case we use the training actions to infer a human-generated Agree/Disagree value [9.2.4] to compare with the synthetic one.

This is a similar logical comparison to the point-biserial for modes 1 & 2 [9.2.7.6], but because this metric compares two dichotomous variables in this case, it is calculated as a phi coefficient (r_ϕ)⁶ for each Mode 3 tag and model.

9.2.7.9 Evaluation Likert Statistics

The metric *Likert Stats* is simple count of the number/percentage of Mode 1/2 item evaluations falling into each bucket⁷.

⁶<https://www.statology.org/phi-coefficient/> [<https://perma.cc/33TF-7UHR>]

⁷Strongly Disagree through to Strongly Agree

9.2.8 Analysis Process

There are two aspects to the result analysis:

1. Quality of Synthetic Evaluation techniques, assessed by:
 - Participant input (feedback) [9.2.7.5, 9.2.7.7 & 9.2.7.9]
 - Participant input (classification/training) [9.2.7.6 & 9.2.7.8]
 - Other participant input and observations
2. Strength of the UD-ML models, assessed by:
 - Participant input (classification/training) [9.2.7.3 & 9.2.7.4]
 - Synthetic Evaluation [9.2.7.1 & 9.2.7.2]

Our analysis process should approach these in this order, because the application of synthetic evaluation to the UD-ML models is dependent on the quality of the selected synthetic evaluation method.

Therefore the broad steps of the analysis process are:

1. Generate and combine statistics for synthetic evaluation, using data from participant training actions [9.2.5.4] and feedback [9.2.5.5]
2. Compare the various synthetic evaluation techniques⁸ and identify the strongest of these – which we will refer to as the prime synthetic evaluation
3. Compile and analyse statistics on the quality of the UD-ML models, using participant training input and the prime synthetic evaluation
4. Draw overall conclusions

⁸Identified by output tag [G.5], which identify the OpenAI mode, the evaluation mode and any iteration; all evaluations performed using a given tag will use identical configuration

9.3 Implementation

9.3.1 Instance Preparation

We initiated each study instance using the steps described in Appendix G.6. The following information was provided to participants, as detailed in Appendix G.7:

- G.7.1 – Participant Information Sheet
- G.7.2 – Participant Consent Form
- G.7.3 – Persona
- G.7.4 – Technical Details Sheet
- G.7.5 – Introductory Presentation
- G.7.6 – Study Protocol

9.3.2 Study Instances

We ran studies in two tranches. Tranche 1 consisted of study instances for three personas: *Susan*, *Adam* and *Kenton*. We then ran Tranche 2 later in the project with two further instances for personas *Phoebe* and *Usha*. This second tranche allowed us to gather further data using different OpenAI models [9.3.4].

9.3.3 Execution Phases

We refer to the stages where the participant is actively engaged in using the study, processing and evaluating data, as the Execution Phases. These take place over a set time period, with the Awareness Agent and supporting UI's activated, during which the experimental data is gathered from external sources and participant actions.

9.3.3.1 Phases 1 & 2: Data Collection and Model Training

This is the execution of the design phase 9.2.5.4, where the participant viewed items in the Training UI [9.4.1] and applied appropriate classifications.

Phase 1 is the initialisation period [9.2.6], and is associated with the input tag [8.2.6.1] ‘phase1’ or ‘init-N’⁹. During this short period the Awareness Agent was running, gathering data from each Acquire source and processing all of the UD-ML classifications. The participant was instructed to use the Training UI to correct or confirm classifications to train the models, but this data would not then be used for evaluation, as we only wanted to evaluate the output of relatively mature UD-ML models.

Phase 2 is actual study data collection and is associated with the input tag ‘phase2’. As well as the primary tag, additional tags such as ‘phase2-1’ were added for each day the phase was running to distinguish data for those days. Data gathered during this phase would then go on to be used for Synthetic Evaluation.

Phase 1 was performed using only the Training UI [9.4.1], while for Phase 2 participants were given the choice of the Training UI or the Awareness Agent Slack Interact UI [6.7.6.4].

9.3.3.2 Phase 3: Synthetic Evaluation and Participant Feedback

This is the execution of the design phase 9.2.5.5, where the participant used the Feedback UI [9.4.2] to rate the performance of synthetic evaluations. At the start of this phase, the administrator produced a set of evaluations, which were later given feedback by the participant.

Data for the evaluations was sourced using the input tag ‘phase2’. That is, the Content Items being processed for evaluation are those that came in while the phase2 tag was set during execution phase 2 (or a subset of them [8.2.7]).

⁹During execution of some of the studies we used init-1, init-2 and init-3 to tag data for days 1-3 of the initialisation period, but this is functionally equivalent to phase1

Phase 3 output was recorded using appropriate output tags [8.2.6.2].

Note that we shortened phases for later study instances based on prior experience – notably that the initial training process yielded results much more quickly than previously expected, and that we gathered more than enough data during phase 2 (compared to what a participant could reasonably be expected to process).

9.3.4 OpenAI Model Selection

Our selection of OpenAI models was based on the quality and availability considerations discussed in Sections 7.6.4 and 8.4.4. However, we used different models for the two tranches of study executions [9.3.2]. This allowed us to take advantage of the GPT-4o model that was released after we had started Tranche 1¹⁰.

In the end, we chose the gpt-4o-2024-11-20 model¹¹ for Tranche 2. This model had been released in November 2024 with better creative writing ability¹². We also cut the number of models used in the second tranche, as we believed that we had gained enough data on these models in the first tranche.

The progression through different models during the course of the research also made it possible to compare performance of these, which would form a significant element of the study results.

Models used by tranche are detailed in Appendix G.4.

9.3.5 Classification Recording

As discussed in Section 9.2.4, we made use of recorded classification actions generated by the user to give us data to analyse Mode 3 synthetic evaluations. To do this, we added recording to awagdata¹³ every time the participant performed a training action [9.3.3.1].

¹⁰<https://openai.com/index/hello-gpt-4o/> [<https://perma.cc/WNX5-4MER>]

¹¹gpt-4o and gpt-4o-2024-11-20 are compared at:

<https://docsbot.ai/models/compare/gpt-4o-2024-11-20/gpt-4o> [<https://perma.cc/G4YL-FA96>]

¹²<https://x.com/OpenAI/status/1859296125947347164> [<https://perma.cc/23NR-YL56>]

¹³Development Log [D] entry 2023-11-22

Training UI

Classification: friend-group personal-interested urgency work-logistics work-pers work-relevant

Tag: init-1 Time Period: Any Subset Tag: Subset Percent: Item ID: Most recent first

Exclude already trained items Hide hints & extra information

Page 2. Displaying 10 items. 378 items remaining.

Previous Next Number of items to retrieve: 10

Training items

We are thrilled to welcome Beacon Industries as a new client. Our team will be providing regulatory ...

SLACK MESSAGE	work-announce	friend-group	not
SENT	Monday 30 December 2024 at 22:27:00 CET	personal-interested	not
FROM	Emma Thompson	urgency	not
TEXT	No additional text	work-logistics	not

Ignore Item Commit Changes

friend-group not
personal-interested not
urgency not
work-logistics not
work-pers work
work-relevant relevant

Welsh ambulance service declares ?critical incident? after demand soars

RSS ITEM	UK news The Guardian	friend-group	not
SENT	Tuesday 31 December 2024 at 00:38:38 CET	personal-interested	not
FROM	UK news The Guardian	urgency	not
TEXT	Welsh ambulance service declares ?critical incident? after demand soars <p>Patients waiting many hours for ambulances to arrive, while those phoning 999 struggle to get through</p><p>A ?critical incident? was declared by the Welsh ambulance service on Monday evening due to significantly increased demand and extensive handover delays.</p><p>The ambulance service, which covers 3 million-plus people ...	work-logistics	not

Ignore Item Commit Changes

friend-group not
personal-interested not
urgency not
work-logistics not
work-pers personal
work-relevant not

Haha, guilty as charged. Tried making sourdough last week?let's just say it was more 'sour' than 'do...

SLACK MESSAGE	friends-university	friend-group	university
SENT	Tuesday 31 December 2024 at 14:55:00 CET	personal-interested	interested
FROM	Hassan Malik	urgency	not
TEXT	No additional text	work-logistics	not

friend-group university
personal-interested interested
urgency not
work-logistics not
work-pers personal

Figure 9.1: Awareness Agent Training UI – Full Browser Page

9.4 User Interfaces

The bespoke user interaction components of the study were part of awagUi [6.8.5].

9.4.1 Training User Interface

Figure 9.1 shows the browser-based interface for the awagUi Training UI. The participant opens the UI using an instance-specific URL and sees a list of items that need training actions. They are able to select an input tag [8.2.6.1] and subset [8.2.7] alongside optional time and ID filters.

A (non-functional) copy of this UI is archived at: <https://perma.cc/2BBU-GDUX>.

Figure 9.2 illustrates a single item from the UI showing a synthetic content message posted to the Simulate Slack workspace. In this case, the item is one that had already been trained by the participant (persona Adam), which is shown by the optional text on the right¹⁴. We can see that in this case the user has changed work-relevant from ‘relevant’ to ‘not’, suggesting that the deadline for Q4 reports is not relevant to Adam¹⁵.

The screenshot shows a 'SLACK MESSAGE' card with the following details:

- SLACK MESSAGE:** bis-general
- SENT:** Friday 26 April 2024 at 09:00:00 CEST
- FROM:** Henry White
- TEXT:** Hey everyone, just a reminder that the deadline for the Q4 reports is next Friday. Please make sure ...

Below the message card are two buttons: 'Ignore Item' and 'Commit Changes'. To the right of the message card is a list of classification models with their current settings:

cycling	[not → not]	not
cycling-logistics	[not → not]	not
interested	[work → work]	work
pers-urgency	[not → not]	not
tech	[not → not]	not
urgency	[urgent → urgent]	urgent
work-logistics	[logistics → logistics]	logistics
work-pers	[work → work]	work
work-relevant	[relevant → not]	not
work-urgency	[urgent → urgent]	urgent

Figure 9.2: Awareness Agent Training UI – Single Item

¹⁴This is shown as ‘before’ and ‘after’ values in square brackets to the right of the classification/model

¹⁵Because the message in question had been sent to #bis-general (a company-wide channel), we might assume that the reports are a problem for some of Adam’s colleagues

9.4.2 Evaluation User Interface

Figure 9.3 shows the browser-based interface for the awagUi Evaluation UI. The participant opens this UI using an instance-specific URL and sees a list of items that have been evaluated and need feedback. They are able to select an output tag [8.2.6.2] and subset [8.2.7] alongside other optional filters¹⁶.

A (non-functional) copy of this UI is archived at: <https://perma.cc/J7H6-URHK>.

Figure 9.4 illustrates a single item from the UI, as Slack message posted to the channel, #bis-team-general. The UI is quite complex – several iterations were necessary to get to this point and the end result was still not entirely satisfactory. We discuss this further in Section 9.6.4. In the illustrated item, the participant is being asked to give feedback on evaluations that have been performed on the UD-ML models for cycling, cycling-logistics, interested, pers-urgency etc. This particular item has already had feedback submitted, as we can see from the “You already submitted” addendum. Looking at the tech classification as an example, we can see that the classification that was originally chosen by UD-ML was ‘not’. The classification drop-down for this has ‘Neutral’ selected¹⁷, but we can see from the addendum that this originally said ‘Strongly disagree’¹⁸, but the test participant here has changed this, as they disagree with that evaluation. To be clear, in the case of this item, the UD-ML engine had classified this message as ‘not’ being about technology, and the synthetic evaluator had strongly disagreed with this – indicating that the evaluator thought that this item was about technology. The study participant partially disagreed with the evaluator and gave a neutral response. Note that the user input here is given when the user changes the value shown in the drop-down¹⁹ or clicks on “Concur” to record that they concur with the decision made by the synthetic evaluator.

¹⁶Note that the page has “Mode 1” in the title – this is a *display mode* and not to be confused with Evaluation Mode [8.2.3]. The display mode in this case has two options, and they change the layout that the user sees; with hindsight different terminology should have been used. Evaluation mode is indicated only by what is shown in the selected output tag (in this case, Mode 2)

¹⁷Corresponding to a Likert value of 3

¹⁸Corresponding to a Likert value of 1

¹⁹Causing an xmlhttp call in the background to awagdata

Evaluation explorer

Mode: Persona: Tag: Time Period: Subset Tag: Subset Percent: Context ID: Item ID:

Mode 1 Select a persona phase2-vanilla4-mode2-01 Any phase2 25 ContextID ItemID

Classification Name: Minimum Evaluation: Maximum Evaluation: Most recent first Exclude items already having feedback Show clean UI

Classification Name Strongly disagree Strongly agree

Page 1. Displaying 10 items. 229 items remaining.

Previous Next Number of items to retrieve: 10

Evaluation items (Mode 1)

This mode allows you to explore the evaluation data that has been recorded by an AI LLM service for each of the displayed items. You will see how the service evaluated the classification for each item from one or more 'perspectives'.

You are asked to give your own feedback on how accurate the AI evaluation is. You can do this by selecting an alternative value in the dropdown that matches what you think the evaluation *should* be; alternatively, click on CONCUR to record your agreement with the AI's evaluation. If you check the 'Exclude items already having feedback' checkbox, items will be removed from the page as you give feedback on them, so you can focus only on items that you have not responded to.

Humza Yousaf vows to stay on as Scottish first minister

RSS ITEM UK news | The Guardian

SENT Friday 26 April 2024 at 16:09:04 CEST

TEXT

SNP leader says he will fight next week's no-confidence vote at Holyrood and take party into general election
• UK politics ? latest updates

Humza Yousaf has said he will not resign as first minister following 24 hours of intense speculation about his leadership.

Speaking at an event in Dundee, the Scottish National party leader, who faces a vote of confidence at Holyrood next week, told reporters: ?I will absolutely be taking us into a general election and 2026 Scottish parliament elections.?

Continue reading...

PERSONA Adam (adam)

CYCLING: NOT

This section is evaluating the classification chosen by the 'cycling' model, which has possible values of **cycling** and **not**. The chosen classification value for evaluation is: **not**

The AI gave this feedback from the perspective Classification - Click CONCUR if you agree with the evaluation or select alternative below:

Strongly agree * The news about a political election does not pertain to my cycling hobby; the classification 'not related to cycling' is therefore accurate.

Text and Likert scale value mismatch for 'Strongly agree'

* - You already submitted that you agreed with the value of 'Strongly agree' on Thursday 9 May 2024 at 10:47:43 CEST

CYCLING-LOGISTICS: NOT

This section is evaluating the classification chosen by the 'cycling-logistics' model, which has possible values of **logistics** and **not**. The chosen classification value for evaluation is: **not**

Figure 9.3: Awareness Agent Evaluation UI – Full Browser Page

Just a reminder that we have a team meeting with the CEO tomorrow. Make sure you're prepared with an...

SLACK MESSAGE	bis-team-general	≡
SENT	Friday 26 April 2024 at 10:38:14 CEST	
TEXT	Just a reminder that we have a team meeting with the CEO tomorrow. Make sure you're prepared with any updates or issues you want to discuss.	
PERSONA	Adam (adam) ⓘ	

CYCLING: NOT

This section is evaluating the classification chosen by the 'cycling' model, which has possible values of **cycling** and **not**. The chosen classification value for evaluation is: **not**

The AI gave this feedback from the perspective Classification ⓘ - Click CONCUR if you agree with the evaluation or select alternative below:

Strongly agree ⓘ * Fully agree with the classification as 'not' related to cycling. The content is about a work meeting, clearly unrelated to my cycling hobby.

Text and Likert scale value mismatch for 'Strongly agree' ⓘ

*- You already submitted that you agreed with the value of 'Strongly agree' on Thursday 16 May 2024 at 13:32:29 CEST

CYCLING-LOGISTICS: NOT

This section is evaluating the classification chosen by the 'cycling-logistics' model, which has possible values of **logistics** and **not**. The chosen classification value for evaluation is: **not**

The AI gave this feedback from the perspective Classification ⓘ - Click CONCUR if you agree with the evaluation or select alternative below:

Strongly agree ⓘ * Strongly agree with the 'not' classification for cycling logistics. The content is clearly work-related and has nothing to do with cycling or organizing group rides.

Text and Likert scale value mismatch for 'Strongly agree' ⓘ

*- You already submitted that you agreed with the value of 'Strongly agree' on Thursday 16 May 2024 at 13:32:40 CEST

INTERESTED: WORK

This section is evaluating the classification chosen by the 'interested' model, which has possible values of **not**, **personal** and **work**. The chosen classification value for evaluation is: **work**

The AI gave this feedback from the perspective Classification ⓘ - Click CONCUR if you agree with the evaluation or select alternative below:

Strongly agree ⓘ * I am definitely interested in this message from a work perspective as it pertains to a significant meeting with the CEO, which is fundamentally important for my job.

Text and Likert scale value mismatch for 'Strongly agree' ⓘ

*- You already submitted that you agreed with the value of 'Strongly agree' on Thursday 16 May 2024 at 13:32:46 CEST

PERS-URGENCY: NOT

This section is evaluating the classification chosen by the 'pers-urgency' model, which has possible values of **not** and **urgent**. The chosen classification value for evaluation is: **not**

The AI gave this feedback from the perspective Classification ⓘ - Click CONCUR if you agree with the evaluation or select alternative below:

Strongly agree ⓘ * Strongly agree with the classification of 'not' as this message pertains to work commitments and has no relevance to my personal life in terms of urgency.

Text and Likert scale value mismatch for 'Strongly agree' ⓘ

*- You already submitted that you agreed with the value of 'Strongly agree' on Thursday 16 May 2024 at 13:32:50 CEST

TECH: NOT

This section is evaluating the classification chosen by the 'tech' model, which has possible values of **not** and **technology**. The chosen classification value for evaluation is: **not**

The AI gave this feedback from the perspective Classification ⓘ - Click CONCUR if you agree with the evaluation or select alternative below:

Neutral ⓘ * Strongly disagree with the classification of 'not' regarding technology. As an IT consultant, a meeting involving the CEO where updates or issues are discussed relates directly to technology-driven decisions and workflows.

Text and Likert scale value mismatch for 'Strongly disagree' ⓘ

*- You already submitted an evaluation as 'Neutral' on Thursday 16 May 2024 at 13:33:08 CEST. The AI model had said: 'Strongly disagree'

URGENCY: URGENT

This section is evaluating the classification chosen by the 'urgency' model, which has possible values of **not** and **urgent**. The chosen classification value for evaluation is: **urgent**

Figure 9.4: Awareness Agent Evaluation UI – Partial Item

9.5 Results

We generated the following study outputs:

- Raw data generated by awagdata (SQLite3 database):

doi:10.21954/ou.rd.28238225

- JSON and Excel stats packs generated by awagdata:

doi:10.21954/ou.rd.28044944 [path: /study/data/stats/generated]

- Consolidated statistics in Microsoft Excel format:

doi:10.21954/ou.rd.28044944 [path: /study/data/stats/combined]

- Supplement S11 containing detailed study results: doi:10.21954/ou.rd.28045580

In this section we will mainly refer to the results documented in S11, reproducing some noteworthy or illustrative items here. This contains the following statistics, taken from the consolidated data referred to above:

- S11.1.1 – Classification Agreement [9.2.7.3 & 9.2.7.4]
- S11.1.2 – Evaluation Feedback [9.2.7.5]
- S11.1.3 – Evaluation Agreement (Modes 1 & 2) [9.2.7.6]
- S11.1.4 – Evaluation Agreement (Mode 3) [9.2.7.8]
- S11.1.5 – Evaluation Difference [9.2.7.9]
- S11.1.6 – Evaluation Ratings [9.2.7.5 – 9.2.7.8]
- S11.1.7 – Evaluation Results [9.2.7.1 & 9.2.7.2]

The results in this section follow the process laid out in Section 9.2.8.

9.5.1 Synthetic Evaluation Quality

9.5.1.1 Evaluation Feedback – Ordinal Evaluation Modes 1 & 2

We first looked at Evaluation Feedback for Modes 1 & 2 [9.2.7.5], producing the data contained in Table S11.4 in Supplement S11.1.2, showing correlation between synthetic

evaluation and human feedback for each combination of persona²⁰, UD-ML model and evaluation tag.

Looking at a high level at the Pearson correlation (r), we noted significant variation by tag, as illustrated in Figure S11.3 (reproduced here as Figure 9.5) – with r values ranging from 0.014 for Mode 2 using GPT-3.5 to 0.777 for Mode 2 using GPT-4o²¹. Those high-level summary numbers were computed only from granular items having a significant correlation²², and we also tracked the proportion of evaluations that met the significance criteria by tag – documented in Figure S11.4. The pattern for combinations having significant p mirrored that for r , with values ranging from 4% for Mode 2 GPT-3.5 to 98% for Mode 2 GPT-4o²³.

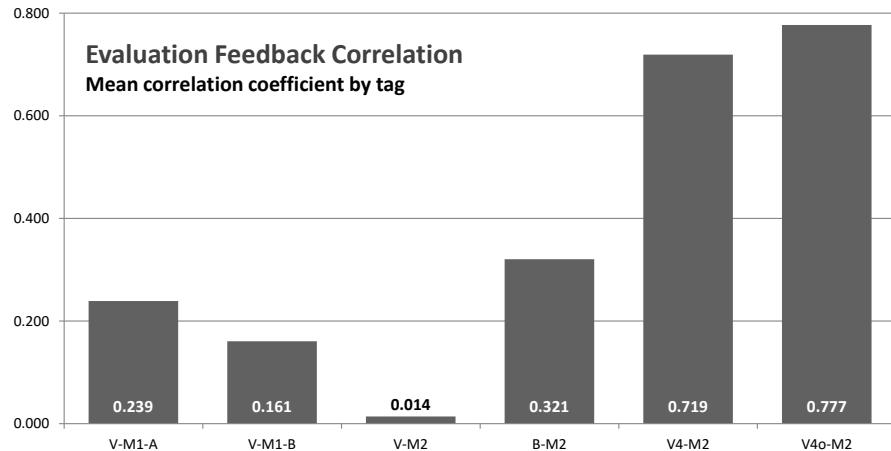


Figure 9.5: Mean Synthetic vs Participant Evaluation r by Evaluation Tag

9.5.1.2 Evaluation Agreement – Ordinal Evaluation Modes 1 & 2

The alternative metric for Modes 1 and 2 was Evaluation Agreement [9.2.7.6], a point-biserial correlation coefficient (r_{pb}) between the synthetic evaluations and classification actions performed by the study participant during the training phase – detailed in Supplement S11.1.3. Data is summarised in the same persona/model/tag combination, in Table S11.5.

²⁰Which is the same as study instance, as there was one study instance per persona

²¹Labelled V-M2 and V4o-M2 respectively for brevity in charts

²²Detailed calculations in resource item /study/data/stats/combined/stats-combined-eval-agreement-m1m2.xlsx

²³Labelled V4o-M2

This data also subjects the synthetic evaluations directly to comparison with participant-entered data, but using data gathered during the earlier classification/training stage. In this case, r_{pb} is used because Mode 1 and Mode 2 evaluations have a categorical value while classification actions are dichotomous.

As for the Feedback metric, we noted significant variation by tag here, as illustrated in Figure S11.6 (reproduced here as Figure 9.6) – with r_{pb} values ranging from -0.142 for Mode 2 using GPT-3.5 to 0.489 for Mode 2 using GPT-4o, based again on using only those combinations with a significant correlation. As with Feedback, the pattern for combinations having significant r_{pb} mirrored that for the r_{pb} values themselves, as shown in Figure S11.7.

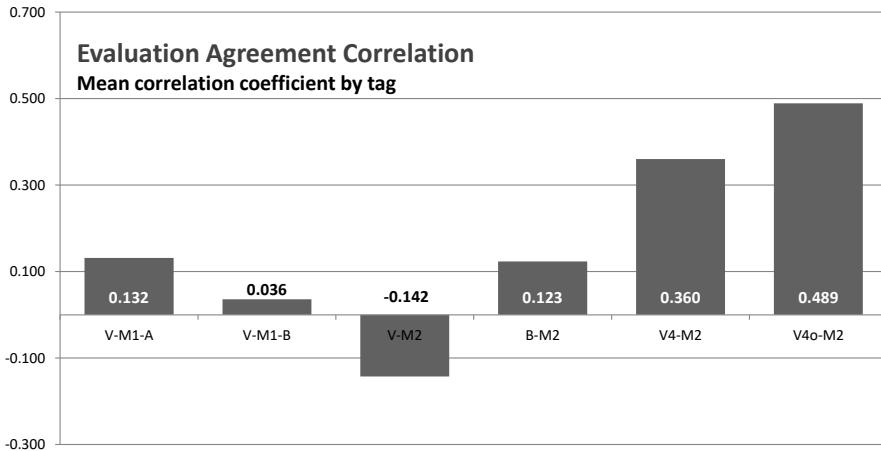


Figure 9.6: Mean Synthetic Evaluation vs Participant Classification r_{pb} by Evaluation Tag

9.5.1.3 Evaluation Agreement – Binary Evaluation Mode 3

Mode 3 evaluation assessment also relied on an agreement metric [9.2.7.8], but with a phi coefficient (r_ϕ) being calculated for these two dichotomous values. This is detailed in Supplement S11.1.4, with data summarised in Table S11.6.

Values of r_ϕ for those combinations having significant results are shown in Figure S11.6 (reproduced here as Figure 9.6).

Mode 3 evaluations have a different set of tags – reflecting not only that they are a different mode but also that in some cases different OpenAI models have been used in the evaluation

– such as the extended fine-tuned model [8.2.8.2.2] identified with label E-M3. Values of r_ϕ range from 0.095 to 0.388 for Mode 3 using GPT-4, a weaker association than for the Mode 2 evaluations. The weakest association is shown by both the base and the extended fine-tuned models²⁴.

Figure S11.7 shows a similar rising pattern of percentage of evaluations having significant r_ϕ across tag to what we have seen for the other metrics.

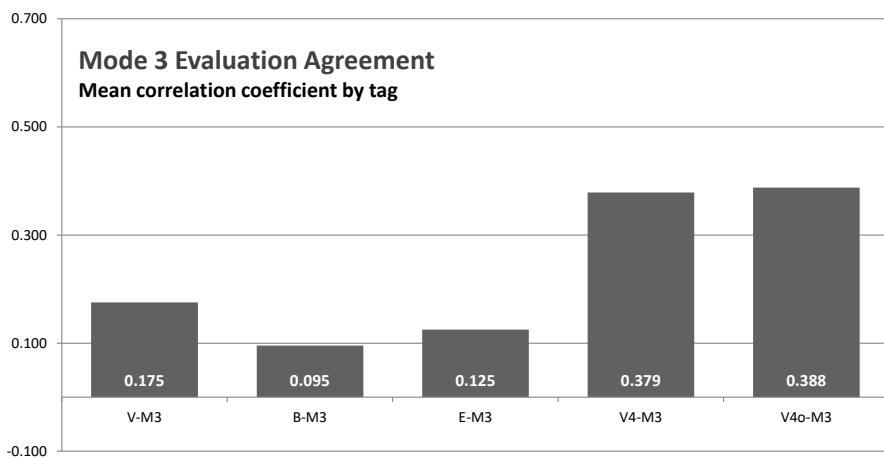


Figure 9.7: Mean Synthetic Evaluation vs Participant Classification r_ϕ by Evaluation Tag

9.5.1.4 Quality by UD-ML Model

Figure 9.8 shows a selection of evaluation feedback/agreement broken down by UD-ML model for different personas. The data taken is a mean across all tags, so overall r values are lowered by inclusion of less effective evaluations, but they are representative of the relative performance of evaluation for the different UD-ML models.

We can see that there is some notable variation between models. For example, Figure 9.8a shows that `tech` and `work-urgency` stand out as relatively strong association – suggesting that the synthetic evaluators are more able to make correct evaluation decisions on these topics. Similarly Figure 9.8b shows the weakest correlations for models `golf` and `golf-logistics` – suggesting that the synthetic evaluator had difficulty identifying these topics correctly.

²⁴Labelled B-M3 and E-M3 respectively

Figure 9.8c shows that for persona Susan the feedback correlation for tennis-arrangements is low relative to other models. Reference to Table S11.4 shows that this is something reflected to varying degrees across all of the tags for this model (although the GPT-4 based evaluations were significantly better). One reason for this could be that the OpenAI evaluator had more difficulty identifying content in the way a human was; a message that is about making arrangements to play tennis could for example be more ambiguous than one discussing an urgent work matter for example.

We also note that for Evaluation Agreement, Figure 9.8d shows that Kenton has a very low mean r_{pb} of 0.06 for the work-pers model, which performs much more strongly for other cases. In this case we see in Table S11.5 that the average is lowered disproportionately by the negative correlation ($r_{pb} = -0.36$) with the GPT-3.5 Mode 1 evaluation for this model, which otherwise records a relatively high quality of evaluations (indeed for evaluation *feedback*, work-pers is the strongest performer as shown in Figure 9.8b).

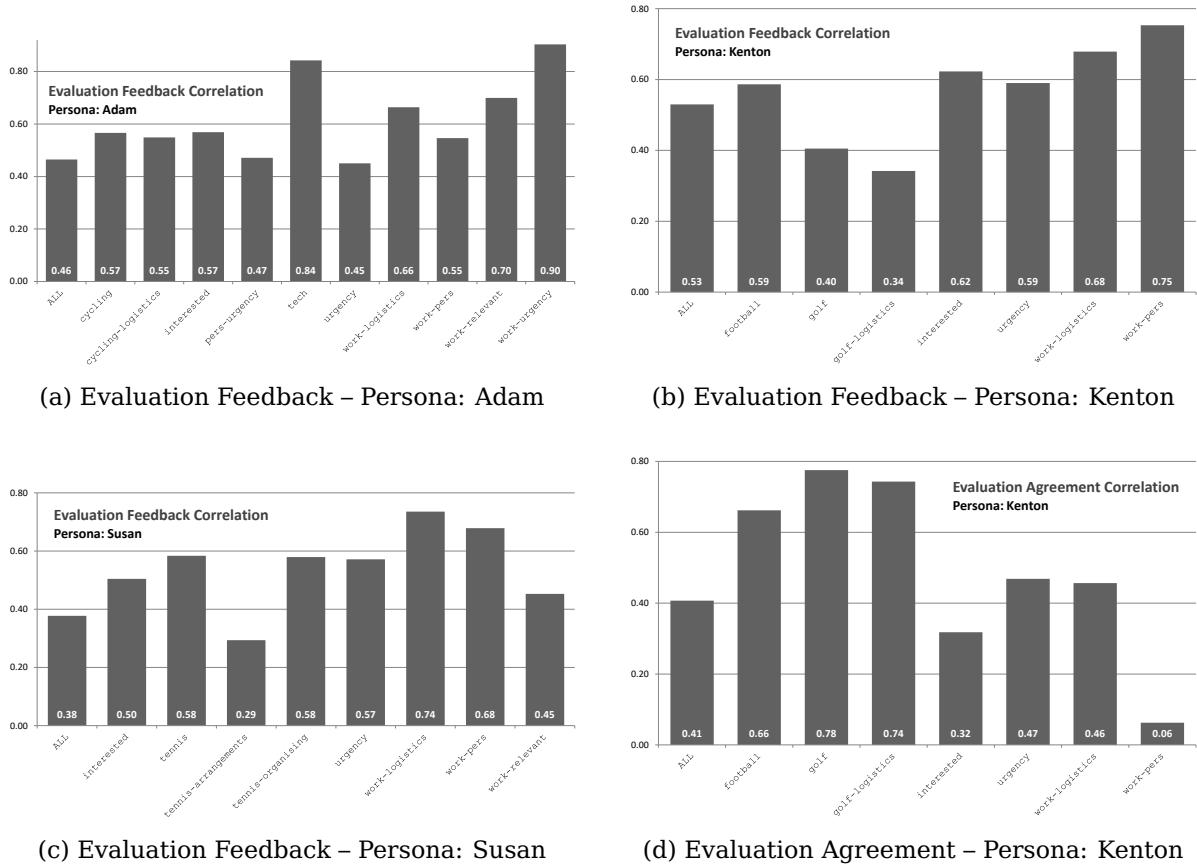


Figure 9.8: Selection of Evaluation Performances by UD-ML Model

9.5.1.5 Overall

We combined the outputs of the different evaluation assessment techniques above to create overall ratings for synthetic evaluations, as identified by output tag. This is detailed in Supplement S11.1.6. Table S11.8 shows the derivation of the ratings, which are illustrated in Figure S11.14 (reproduced here as Figure 9.9).

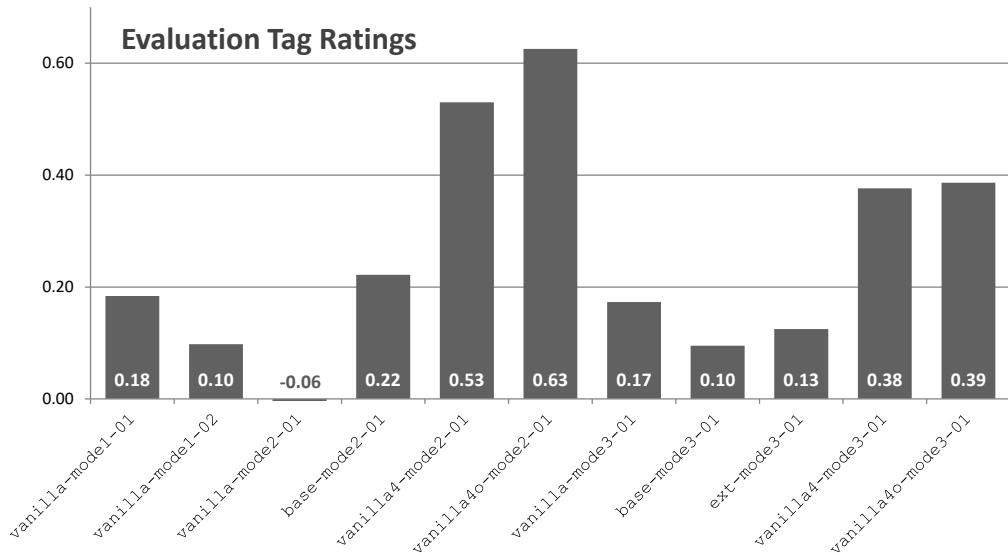


Figure 9.9: Synthetic Evaluation Tag Ratings

We calculated these ratings using a combination and comparison of the r , r_{pb} and r_ϕ values that we had derived for the tags. In the case of ordinal Modes 1 and 2, we had both r and r_{pb} values available, so we applied a simple mean of these. For binary Mode 3, we used the sole metric, r_ϕ . We note that there are limitations in comparing the different correlation coefficients in this way – they are calculated using different techniques and applied to data that has been gathered in vary ways – but contend that it is reasonable to do so in this case, with the caveat that we are attempting only to rate the relative performance of evaluations rather than assign definitive scores to each.

Based on these ratings, we selected **vanilla4o-mode2-01** (Mode 2 using GPT-4o, also labelled V4o-M2) as the prime synthetic evaluation.

9.5.2 Quality of UD-ML Classification

We have two methods of assessing the quality of the classification of UD-ML models:

- Classification Agreement statistic generated by the training process [9.2.7.3 & 9.2.7.4]
- The Synthetic Evaluation result

We will look at these independently first, then in combination.

9.5.2.1 Classification Agreement

Supplement S11.1.1 Figure S11.2 (reproduced here as 9.10) shows the agreement between UD-ML model classifications and manual classifications performed by the study participant for each study instance, calculated as a Cohen's Kappa.

Examining the ALL figure – calculated from the data set as a whole – we see a k of 0.83, indicating extremely strong agreement between the UD-ML models' classifications and the human classifications of the same items. Of the k values at the individual model level, we see that 15 out of 22 models have a value of over the 0.6 threshold for substantial agreement, while all 22 models exceed the 0.4 threshold for moderate agreement.

There is some variation however; models such as **work-relevant**, **school-importance** and **tennis-organising** have lower k values in the fair to moderate agreement range, suggesting that these UD-ML models have been less effective at matching the participants' mental model for classification. There is a specific reason for the low score for **work-relevant**: for persona Susan this item has a k value of 0. Checking the raw data showed that this was not an error: the 1006 records recorded for this model/instance combination all contained a UD-ML classification value of 'not', meaning that the model had failed to classify any items as 'relevant'. The uniformity of this data when compared to the non-uniform human-entered inputs gave a k of 0. However, as can be seen in Table S11.1, the human participant agreed with the UD-ML classification in 92% of cases, so this a stronger metric than the k value suggests. Similarly in the case of **tennis-organising**, high uniformity of the UD-ML classifications leads to a low k value, but the agreement percentage is 97%, suggesting that this is actually a well-performing model – better indeed than the more

general tennis, which has a higher k of 0.48 but a lower 88% agreement percentage.

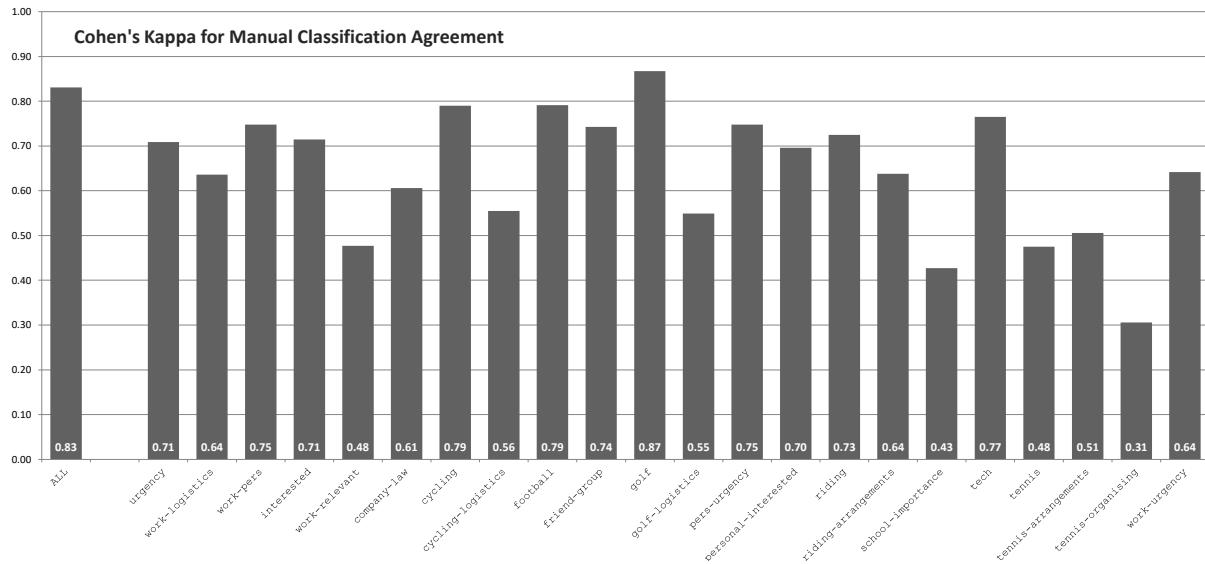


Figure 9.10: Cohen's Kappa for Manual Classification Agreement with UD-ML

We also looked at how the classification performance changed over time, and Figure S11.1 (9.11 here) shows how the classification agreement percentage changed over time during the training phase of each study, which occurred over a period of up to 7 days for each study instance. We can see a steady increase in performance over time as the participant entered more training data during this phase.

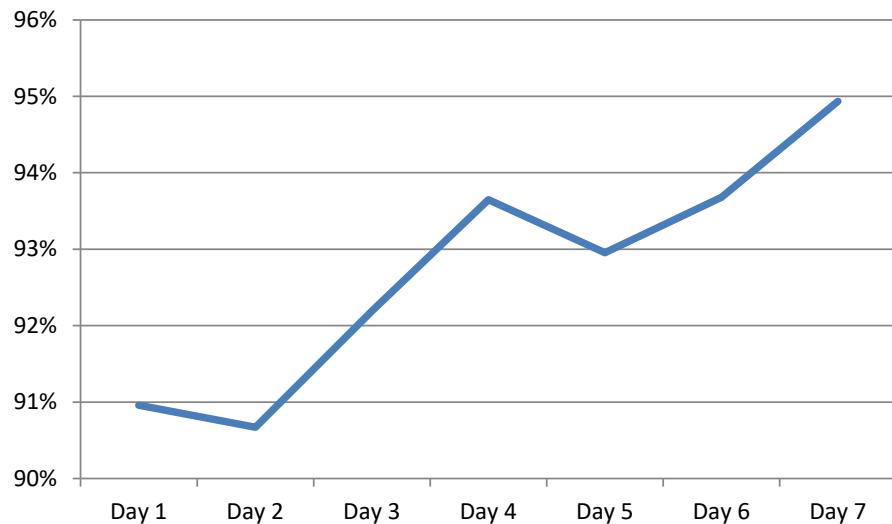


Figure 9.11: Percent of Manual Classification Agreement with UD-ML Time Series

9.5.2.2 Synthetic Evaluation

Having settled on **vanilla4o-mode2-01** as the prime synthetic evaluation method, we will use this in the analysis. Table S11.11 (reproduced here as 9.1) shows the Manual Agreement (human) and Evaluation Score (synthetic) by UD-ML model, which is documented in detail in Supplement S11.1.7. The data is also available in chart for in Figure S11.15 (9.12 here)

This also has two metrics to measure similarity between the human and synthetic values on a model by model basis, *Proximity Ratio* and *Squared Error x100 (SE₁₀₀)*, defined as:

$$\text{Proximity Ratio} = \frac{\min(A, B)}{\max(A, B)}$$

$$SE_{100} = 100 \times (A - B)^2$$

Where A is the Manual Agreement percentage value, and B is the Evaluation Score. We use these two metrics to allow us to see how closely the human and synthetic evaluations align for each model, but our prime interest in this section is assessing the UD-ML models themselves.

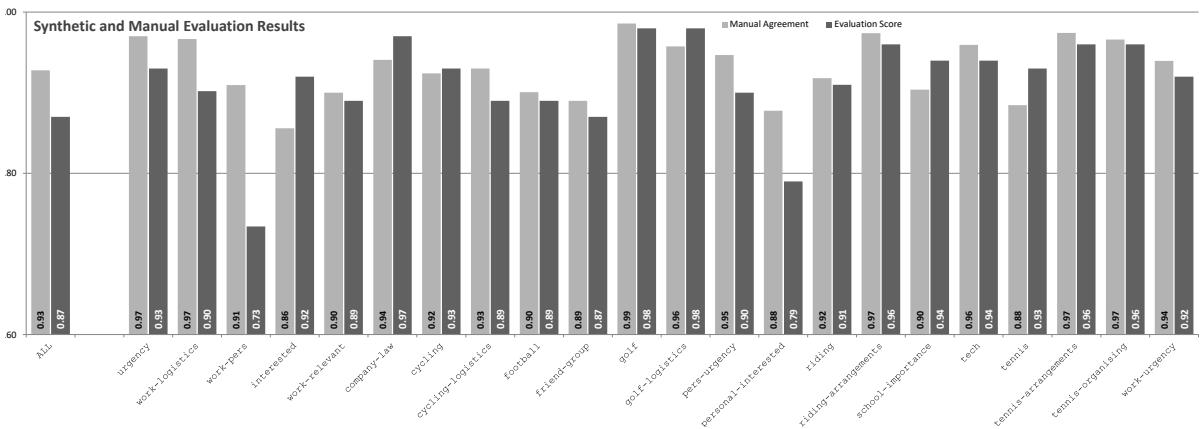


Figure 9.12: Classification Agreement vs Evaluation Score for Prime Evaluation

We can see that with some variation there is generally strong performance for most UD-ML models as judged by synthetic evaluation, with manual rating also mostly in agreement. One exception that does stand out is *work-pers*, where the synthetic evaluation scores the models quite poorly. However the manual classification in this case has a high k –

suggesting that this is a topic where the synthetic evaluation has difficulty in evaluating an actually well-performing model.

Classification Manual Agreement vs Evaluation Score

This table shows the Classification Manual Agreement value of each UD-ML model alongside the V4o-M2 Synthetic Evaluation Score. A proximity ratio and squared error are calculated to show the closeness of the synthetic to manual classifications.

Model	#	Manual Agreement	Evaluation Score	Proximity Ratio	Squared Error
ALL		0.93	0.87	0.94	0.33
urgency	5	0.97	0.93	0.96	0.16
work-logistics	5	0.97	0.90	0.93	0.42
work-pers	5	0.91	0.73	0.81	3.09
interested	4	0.86	0.92	0.93	0.41
work-relevant	4	0.90	0.89	0.99	0.01
company-law	1	0.94	0.97	0.97	0.09
cycling	1	0.92	0.93	0.99	0.00
cycling-logistics	1	0.93	0.89	0.96	0.16
football	1	0.90	0.89	0.99	0.01
friend-group	1	0.89	0.87	0.98	0.04
golf	1	0.99	0.98	0.99	0.00
golf-logistics	1	0.96	0.98	0.98	0.05
pers-urgency	1	0.95	0.90	0.95	0.22
personal-interested	1	0.88	0.79	0.90	0.77
riding	1	0.92	0.91	0.99	0.01
riding-arrangements	1	0.97	0.96	0.99	0.02
school-importance	1	0.90	0.94	0.96	0.13
tech	1	0.96	0.94	0.98	0.04
tennis	1	0.88	0.93	0.95	0.21
tennis-arrangements	1	0.97	0.96	0.99	0.02
tennis-organising	1	0.97	0.96	0.99	0.00
work-urgency	1	0.94	0.92	0.98	0.04

Table 9.1: Classification Agreement vs Evaluation Score for Prime Evaluation

9.5.3 Participant Observations

We encouraged the participants to give use informal feedback during the course of the study execution, which we noted and then categorised.

The following topics were reported by participants:

1. During the training phase, all participants noted that the UD-ML classifications were generally quite accurate for most models. Furthermore, the learning curve that UD-ML models went through was quite steep – that is, the quality of classifications improved rapidly over time as they went through the daily training cycle during phases 1 & 2.

2. Participants themselves found some topics hard to classify in character as their persona. A common question was “Would my persona be interested in this particular news story or message on a chat group?”. We gave them advice to be internally consistent with how they perceived their persona, but agreed that this was a highly subjective element. Conversely, other topics were very easy for the participant to classify, such as those about a clearly defined area (such as cycling), or using specific language to indicate urgency or relevance. This is something that was also reflected in the classification agreement and evaluation feedback data.
3. Similarly, some of the UD-ML models were inherently hard to classify in terms of knowing how to categorise content. For example the model friend-group caused confusion as there were multiple ‘work’ groups in the content but only one ‘work-friends’ classification in the model. Also in relation to this specific group, as well as the work-pers one, the participant asked about messages that were purely personal (i.e. about weekend brunch, or karaoke), but delivered in a work colleagues group – should it be personal or work? The advice for this case was that this would be ‘personal’; but more difficult for the human to quantify were messages relating to lunch arrangements during the working day. Again, the advice here was to attempt to be consistent when both training the model and providing evaluation feedback.
4. Also relating to the work-pers model, a participant noted that the synthetic evaluator at times seemed unsure how to rate this. It could identify when an item was not work related, but if it also wasn’t linked to something that the persona was obviously interested in (such as tennis), then it often provided an equivocal/neutral answer (even though a more appropriate answer would be that the item is personal, but not interesting).
5. They found that the quality of synthetic content was generally plausible and realistic, in as much as their personal experience allowed them to judge. They could see that simulated conversations were being carried on in a realistic way, although they noted that when conducting training and evaluation actions the context of these conversations was not visible to them (also identifying that this could cause difficulty to the synthetic evaluator too).

6. While quality of synthetic content was generally good, participants noted that some was relatively banal, with some topics being repeated more than would be expected (“They talk about biscuits a lot”).
7. Participants agreed that they could see value in training something like UD-ML to classify and organise incoming items, with the effort to reward ratio being worthwhile. However, they did not express trust that this prototype implementation would be able to handle their real-life information. The primary concern expressed was that they would need to be more satisfied that there was no risk of missing important items, with comments also that the prototype user interface was not ready for real-world usage.
8. All participants commented that the training UI [9.3.3.1] was much easier to work with than the evaluation UI [9.3.3.2], with the latter requiring much more attention to correctly input results. They found this UI to be more complex, with a greater amount of data to assimilate to perform each action, and also that the concept required more careful thought. Specifically, the user was asked to either concur with or correct an evaluation Likert assigned by the synthetic evaluator, and none of the participants found this to be an intuitive process.
9. A common problem identified by participants – particularly when working with the evaluations using GPT-3.5 models – was that the evaluation text did not match the Likert value assigned by the synthetic evaluator. For example the evaluation text might say that it strongly agreed with a classification, but then award a Strongly *Disagree* Likert value. We advised the participants to enter their response based on the assigned value and not the text²⁵; for early participants we also added a checkbox input to allow them to identify cases where this mismatch occurred, but we did not in the end use this data for all participants as it added complexity to their experience with the feedback UI and we were not confident of the data capture being comprehensive enough.
10. Most participants also noted that both the training and evaluation work was relatively boring and repetitive, although this was eased by being able to do this in smaller daily chunks. The training UI was easy and quick to use but many actions were a case of

²⁵Indeed the mechanism of the study implied this

selecting or verifying the same set of options for item after item. No participant felt that the added complexity of the evaluation UI made the process more interesting.

11. Participant feedback also led us to identify a number of specific evaluation content examples [9.5.4].

9.5.4 Evaluation Content Examples

While there are too many individual evaluations to examine each one, we selected some that highlight specific traits referred to above, two of which we include below.

A wider selection of synthetic evaluation items is contained in Supplement S11.2.

Example A

This example shows a case where the `vanilla-model-02` evaluator has incorrectly evaluated an item. While the item relates to logistics at work – it is a discussion about work-related scheduling that is dependent on someone’s availability (due to a personal dental appointment) – the UD-ML `work-logistics` model classified this as ‘not’ about work logistics. The evaluator agreed with this, noting that a dental appointment is a personal matter, but the human evaluator correctly interpreted this as work scheduling and marked that they strongly disagreed with the UD-ML and synthetic evaluation.

SLACK MESSAGE	<i>work-team-general</i>
phase2-vanilla-model-02	Sophie, can you put me down as a maybe for
07f09430-3e9a-4a50-acef-1e66e448f288	Thursday? Waiting to hear back about a
	dentist appointment...

Classification: **work-logistics**

Categorised as: **not**

Synthetic Evaluation: **Strongly Agree**

Participant Evaluation: **Strongly Disagree**

The system correctly marked this as ‘not’ for work-logistics. The message is not related to work logistics, but rather about a personal dental appointment. My likert value is 5 as I completely agree with the system’s classification.

Example B

This is another example of the human disagreeing with the synthetic evaluator. In this case UD-ML had classified this item as ‘not’ urgent, but the synthetic evaluator felt otherwise, presumably excited by the prospect of some positive feedback. The human corrected this, entering a weak agree with the original classification.

SLACK MESSAGE *bis-general*
phase2-vanilla-model1-02 **Good morning everyone, just a quick update.**
15179db6-8352-4e8a-a929-110ef240b374 **We've received positive feedback from Oceanic Research C...**

Classification: **urgency**

Categorised as: **not**

Synthetic Evaluation: **Strongly Disagree**

Participant Evaluation: **Agree**

The system has incorrectly classified this message as ‘not’ urgent. The positive feedback should likely be considered urgent due to its work-related nature. The urgency of the content is not accurately reflected in the classification.

9.5.5 Notes

1. While we have attempted to be generally consistent with terminology, the terms ‘model’ and ‘category’ have at times been used to refer to the ID of a UD-ML model during the study development and execution process. UD-ML models and OpenAI models are distinct topics; where we use ‘model’ without further context or clarification, we are referring to a UD-ML model.
2. In the statistics, a number of the models (i.e. `cycling-logistics` for persona Adam) have a smaller number of items evaluated than others for the same persona and evaluation tag. This is generally because OpenAI occasionally failed to return a result for some items in an evaluation request, so the participant wasn’t presented with a complete list for every item during evaluation feedback.
3. We recruited participants with varying backgrounds, including civil servant, academic, IT professional, media professional and teenage student. While this led to small differences in participant approach in terms of work pattern, nature of observations etc., they approached the task in a broadly similar way and gave us some common feedback

items.

4. We listed `awag_sqlite.db` above as a study output. We also generated a second database file, `awag_obj_sqlite.db`, which was used by SPOSS [6.8.4] for fine-tuning metadata stored by the `awagdata /data/gentrain` service [6.8.3]. This was not an output itself, but was rather data used in the process of executing the study.

9.6 Reflections

9.6.1 UD-ML Initialisation Speed & Model Lifecycle

We were surprised by the rapid rate at which UD-ML models improved in early training, as shown in Figure 9.11. This has a positive impact on our design concept of a user-controlled model lifecycle [6.7.6.1], because it supports the idea that it is practical to dispose of an old model that has drifted over time or no longer aligns with current user focus and start afresh with a new one. The experimental data suggests that a new model would quickly become useful with a minimal training effort required. given that we have taken a relatively unsophisticated approach to the internals of the UD-ML models [6.8.2.1], this is a positive result.

9.6.2 Issue with Schema in Compound Ordinal Evaluation (Mode 1)

As we began processing of Tranche 2 [9.3.2], we discovered that during Tranche 1, we had been incorrectly running all Mode 1 with the schema component not being sent in the request [8.2.3.1]²⁶. This was a deviation from our design and meant that Mode 1 evaluations might not be as accurate as we had hoped. This issue affected the output tags `phase2-vanilla-model1-01` and `phase2-vanilla4-model1-01`.

To rectify this, we re-ran the affected Mode 1 evaluations with a new set of output tags (`phase2-vanilla-model1-02` and `phase2-vanilla4-model1-02`) and asked the participants to perform feedback on these items. This actually gave us additional data, allowing us to

²⁶Supplement S8.1.3 [doi:10.21954/ou.rd.28045547]

compare the results with and without the schema being passed.

In the end, the results data showed that there was not a significant difference between the cases where the schema was and was not passed; in fact the later run including the schema performed marginally less well. One reason for this might be that the schema is not a required element of the OpenAI interaction – we were adding the schema as a way of conveying information about the nature of the evaluation request, but it was not a formal OpenAI API requirement (or even something explicitly supported by the API). We might conclude that the LLM was instead relying more heavily on other prompting and content information to generate the evaluations than a self-defined schema that we passed with the request.

9.6.3 Slack vs. awagUi

We had given the participants a dedicated Training UI for phase 1 [9.4.1], but for phase 2 gave the the choice of the Training UI or the Awareness Agent Slack Interact UI [6.7.6.4]. While the Slack UI could subjectively be a more ‘realistic’ approximation of what an actual agent UI might look at, it was inefficient for the phase 2 task, requiring too many clicks and pauses. Participants universally preferred the bespoke UI.

9.6.4 awagUi Complexity

The Evaluation Feedback UI was necessarily quite complex and not entirely intuitive, as was noted by participants. We found that this had an effect on their work rate when processing evaluation feedback, and they were able to cover less content than with the Training UI. In most instances we needed to go through a process of identifying which output tag combinations has weak results (high p value) and asking the participant to process more evaluations for those tags specifically. This targetted approach let us get more statistically valid results while hopefully not overly taxing the participants.

9.6.5 Classifying Ambiguous Content

The greatest challenge we and our study participants faced was dealing with content that had a degree of ambiguity or lack of clarity in how it should be classified. Some of this content would be confusing to a synthetic evaluator only, while other content challenged human and computer alike. Such items generally fell into the following categories:

- Where it is not obvious whether the item would be of interest to the persona as it is not obviously aligned with their documented interests
- Items lacking sufficient context to make a determination based on the item alone
- Cases where personal and work aspects are conflated – such as making personal arrangements to meet work colleagues for lunch or discussing work with personal contacts
- Items where the level of urgency or similar is not clear cut – for example to what extent is an item that does not contain a clear deadline or mention of urgency considered urgent

This issue appeared to affect some topics (UD-ML models) more than others, which is reflected in the experimental data, and also appeared to affect the synthetic evaluator more than both the human evaluator or the models themselves. The `work-pers` model is a good example of this, a case where UD-ML and the human evaluator were much better at classifying content than the synthetic evaluator. Other examples of challenging models were those focusing on logistical arrangements and organising – there was relatively little content in the feed for these topics, giving weaker models, and the synthetic evaluator performed relatively badly in assessing them.

Context is an important omission from our tested setup. As tested, items were classified and evaluated in isolation, with neither UD-ML nor the synthetic evaluator benefiting from message context or history at the point of evaluation. The human study participant fared better in this regard, having access to the context (if they chose to use it). We believe the performance of the synthetic evaluator could be improved in this regard, by exploring the following techniques:

- Provide a summary of related message history as part of the evaluation request
- Provide more cues to the synthetic evaluator, possibly in the form of a document listing a number of common topics and how they should be handled²⁷

9.6.6 Value of Fine-Tuning

We had hoped that fine-tuning [8.2.8.2] – particularly the extended fine-tuning that used actual classification data to train the models [8.3.4.2] – would show tangible benefits. However, with the caveat that we were only able to test this using GPT-3.5, this was not the case. Our final tag ratings [9.5.1.5] showed that the Base and Extended FT models were of comparable – and in some cases worse – performance to the vanilla models. This was the main reason why we did not include fine-tuned models in the second tranche of study instances, as it was not an effective use of time and resources [G.8].

9.6.7 Simplified Binary Evaluation (Mode 3)

We were surprised that Mode 3 evaluations were also less effective than we had expected, with binary Mode 3 evaluations underperforming both ordinal modes 1 and 2 for equivalent OpenAI models. We had expected to see the opposite, as Mode 3 requires a simpler response – Agree/Disagree rather than a Likert. It is possible that part of this might be explained by the different way in which we measured Mode 3 [9.5.1.3], but we do not believe this is the main cause, as the statistics we produced should be broadly comparable.

One plausible explanation is that giving the evaluator greater latitude for nuanced response gave us better results than forcing a binary choice. For example, if a Likert-based evaluation came in at Neutral, this would be more realistic (and score better in our metrics) than an incorrect (dis)agree response.

Our rationale for introducing Mode 3 had partly been practical – we could directly use the data from the training phase to obtain evaluation feedback rather than asking the participant to do this work – and this remains the case, but our preference would be not to

²⁷ Not dissimilar to few-shot learning as an approach, but possibly with a more structured and comprehensive format than we have used so far

use this mode in future work.

9.6.8 Compound Ordinal Evaluation (Mode 1)

We also did not run Mode 1 evaluations other than with the vanilla model in the first tranche of studies. This was not strictly because of poor quality of responses – we did not gather enough comparative data to judge this systematically – but more because it was slow and resource-intensive. We also observed many occasions in testing where the evaluator appeared clearly confused about the structure of the request, which reduced our confidence in this, and led us to develop Mode 2.

We believe that the Mode 1 approach is potentially powerful, but limited by the current LLM API's available to us. Specifically, the API's at present do no have any support for using structured JSON as part of requests in this way; should such structure become possible then the Mode 1 approach may be significantly more useful.

9.6.9 Evaluation Feedback vs Evaluation Agreement

Our use of two approaches for measuring the effect of synthetic evaluation – Feedback [9.5.1.1] and Agreement [9.5.1.2] – made the process of analysis less straightforward, the different types of evaluation and data available led us to this approach.

These mechanisms should in theory be in agreement with each other, as in both cases the participant is providing information used to rate the synthetic evaluation. However in one case they are being presented with an evaluation and asked to correct it, while in the other they are being asked to enter their own classification for an item without being explicitly asked to judge the evaluation (instead that information is obtained implicitly by comparison between human and synthetic actions on the same items). Using r vs r_{pb} could also alter the correlation numbers.

We believe that the weakness of Mode 3 – that most closely associated with Agreement based assessment – suggests that Evaluation Feedback is the more useful metric to pursue.

9.7 Conclusions

At the start of this chapter we identified two areas that we wanted to use the study to gain insight into: the quality of the Awareness Agent UD-ML classifier [9.1.1] and the effectiveness of using synthetic content and evaluation in the study process [9.1.2].

9.7.1 UD-ML

We obtained information on UD-ML quality using two techniques: via manual classification agreement using direct participant input [9.5.2.1], and using synthetic techniques [9.5.2.2].

We can see from Figure 9.10 that the participant assessment of UD-ML – which we regard as definitively correct but limited in coverage – showed a high level of agreement with the classification decisions made by UD-ML models. However, there was some variation, and a common theme that emerged from our results was that the strength of UD-ML classification varied by topic²⁸, with topics that humans found harder to consistently classify [9.5.3] also being more difficult for UD-ML.

One weakness that we found in our implementation of UD-ML was lack of context: an item taken on its own shorn of history and other contextual information can be hard for humans and computers alike to classify [9.6.5]. While the humans in the study were able to look at other contextual information – or use intuition – the synthetic systems were not able to do so. Items lacking identifiable keywords or phrases in the message body were more often misclassified or incorrectly evaluated. We note that the structure of the Awareness Agent Content Item [6.7.1] allows for a greater degree of contextual information to be used than we passed on either to the UD-ML models or the synthetic evaluator, and this would be an avenue to address this weakness.

Another weakness was consistency of interpretation – most notably relating to the models that were a variant on “Is this item of interest to me?”. This is an area less open to purely technical solutions; however, it does in part validate the design of the UD-ML model

²⁸Topic and UD-ML model being effectively the same thing

lifecycle [6.7.6.1] – one solution to vaguely-defined models that are not closely aligned to user needs, or models that cease to be relevant as user’s priorities drift over time, is to have a system where the user actively creates and destroys models to meet their needs at any given moment. Our study results showed that new models could quickly be brought up to speed [9.6.1], a necessary prerequisite of this approach.

Our synthetic evaluation data, as shown alongside participant data in Figure 9.12, tells a similar story to the human input – with strong performance evidenced for most models but with similar notable variations. However, there are some differences between the two techniques, which we discuss below.

9.7.2 Synthetic Techniques

We believe that the synthetic techniques for both content and evaluation were generally successful, with some caveats.

Study participants reported that most synthetic content was believable and consistent with the topic, and this is supported by the consistent classification of this topic (i.e. synthetic content was classified in consistent ways by UD-ML and this was in agreement with participant input). As the study administrator we can also report that the technique was successful from a practical perspective: we were able to automate the generation and use of large volumes of synthetic message content and deploy this for testing with very little overhead. We found that although the topic definition took a little work to get right initially, we could create surprisingly rich threads of discussion with limited effort.

There was significant variability in quality of synthetic evaluation by OpenAI model and technique, as shown in Figure 9.9. Least surprising was that newer LLMs – GPT-4 and GPT-4o – were far more capable. More surprising for us was the weakness of Mode 3 (Simplified Binary Evaluation) [9.6.7]. Given the issues that we had with Mode 1 (Compound Ordinal Evaluation) [9.6.8], we would consider Mode 2 (Simplified Ordinal Evaluation) with a recent LLM to be the best approach, hence our decision to select `vanilla4o-mode2` as our ‘Prime Evaluator’ for the purposes of assessing UD-ML.

As can be seen in Table 9.1, the Prime Evaluator was very close to human input for the

large majority of UD-ML models, and we consider this a reliable technique for evaluating classification performance, subject to some caveats and areas for improvement. Of the latter, we would apply the same considerations that we discussed above – greater contextual information in particular could potentially have a significant effect, as could increasing the cues available to the evaluator to assist with ambiguous topics.

9.8 Chapter Summary

In this chapter, we documented the final study of our research, where we used synthetic content [7] and evaluation [8] techniques according to our methodology [4.2.4] to examine our prototype for an Awareness Agent [6]. We broke this down into the following parts:

- Design of the study [9.2]
- Implementation [9.3]
- UI illustrations [9.4]
- Results [9.5]
- Reflections [9.6]
- Study conclusions [9.7]

The next and final chapter [10] will take our findings from this chapter and those prior and go on to draw some overall conclusions for our research.

Chapter 10

Conclusions and Further Work

We began our work looking at the issue of Information Overload from the perspective of users of social and other connected platforms, through a lens of Awareness [1.2.1]. After conducting a user survey into these topics [5.1], we developed a set of personas [5.2] to frame our work towards an Awareness Agent that might address some of the identified issues [6.4], then implemented a prototype agent to study [6.8]. During development we identified a number of opportunities to improve the evaluation process by introducing synthetic techniques, leading to the creation of methodologies for using synthesised content in the study [7], and for using a synthetic evaluator as a ‘virtual study participant’ [8]. Finally we ran a set of studies to test our prototype [9.2], and conducted an analysis of both the agent and the synthetic techniques [9.5].

We start this chapter by reviewing the scientific contributions that we believe we have made in this work [10.1], before going into more detail in the research questions and hypotheses that we have addressed [10.2]. We then discuss the limitations applicable to the work and our results [10.3] before going on to outline areas for further work in future [10.4]. We conclude with some final thoughts [10.5].

10.1 Scientific Contributions

We outlined the scientific contributions that we were hoping to make with our work in the introduction [1.2.4], and will assess those outcomes here.

1. Create a Design for an Agent-Based Solution To Address IO

We proposed and developed a detailed design for an Awareness Agent [6.4], motivated by the earlier survey and persona work [5.3], aimed at addressing information overload by enhancing user awareness. We designed a novel modular system [6.4.3.1] capable of being integrated with multiple information sources, handling and augmenting data in an abstracted and standardised way [6.6.2] using commodity and bespoke AI services [6.4.3.2], so that information burden on the human user could be reduced.

Our iterative design process – where we tested features and integrations over time as we developed them [6.5] – allowed us to explore and demonstrate the feasibility of implementing such a system in a real-world setting. Following this process, we were able to identify and address real-world challenges encountered in the development of such a system, resulting in an implemented prototype that acts both as a proof-of-concept for these ideas and a strong foundation for further development.

Conclusion: This contribution was successfully met by the creation of a functional design framework, supported by a detailed prototype implementation.

2. Gain and Share Insight Into Design and Practical Challenges

We identified and addressed several challenges related to designing an awareness agent during the iterative process described above. We documented the design process for this solution in Chapter 6 and the Design and Development Log in Appendix D. The experiences and insights that we have documented in this process have highlighted a number of real-world considerations and potential pitfalls, identifying techniques to address them¹. As such, this work combined with the supporting materials constitute a valuable resource to

¹For example: Development Log [D] entries 2019-10-15, 2023-06-14, 2023-06-26 & 2023-10-15

other researchers and practitioners working on similar problems.

Conclusion: We have captured information on numerous challenges and insights from the design and implementation process, fulfilling this expected contribution.

3. Produce a Detailed Analysis of Solution Operation

We conducted an in-depth evaluation of the prototype's operation through a structured user study, as documented in Chapter 9. Our analysis produced metrics covering topics including classification accuracy [9.5.2] and the efficacy of synthetic evaluation techniques from multiple perspectives [9.5.1]. These metrics and participant feedback [9.5.3] highlighted both the prototype system's strengths and areas for further development [9.6].

Conclusion: We produced a well-documented analysis of the prototype solution's operation, meeting the goals of this contribution.

4. Develop Reproducible Techniques and Materials for Synthetic Content

Chapter 7 documents our approach for generating and using synthetic content to simulate information systems in scenarios where real data is difficult or impossible to access [7.1]. We developed a clear framework for content generation using personas and scenario-driven design [5.3.6], creating a flexible and re-usable system for generating simulated content on a wide variety of topics using a defined set of virtual actors and entities for consistency and realism [7.4]. We have shared source code and supplementary materials such as examples and templates [E]², allowing future researchers to repeat and extend these techniques.

Conclusion: We met this expected contribution by developing reusable techniques addressing our intended goals and acting as an asset for other researchers.

²See Supplement S12 [[doi:10.21954/ou.rd.28045598](https://doi.org/10.21954/ou.rd.28045598)] for full source code information

5. Develop Reproducible Techniques and Materials for Synthetic Evaluation

The use of synthetic evaluation was an important part of our research's methodology, and we detailed our design for this in Chapter 8. As with synthetic content, we created a clear and re-usable framework for this, using a persona-based system [8.2.1] so that the evaluations could be systematically conducted in the role of fleshed-out, believable and consistent virtual study participants. Our study provided evidence [9.5.1] from human input to validate the efficacy of the virtual evaluators, highlighting areas of strength and weakness for these. We are enabling future researchers to repeat and extend these techniques by sharing source code and supporting materials [F].

Conclusion: We successfully developed and demonstrated techniques for synthetic evaluation, achieving this contribution.

Other Contributions

In addition to our research outcomes fulfilling the five expected contributions detailed above, we believe that we have also made other contributions over and above those we had initially expected. In particular, we've published a number of software contributions under Open Source licenses, as detailed in Supplement S12 [[doi:10.21954/ou.rd.28045598](https://doi.org/10.21954/ou.rd.28045598)]:

- Code for creating, collating and presenting a number of research statistics, automatically generating user-friendly Excel data packs.
- System for capturing experimental data in real time via a web service, which could be adapted and repurposed for other projects.
- A simple but extensible system for creating and using ML classifier models via simple REST web services that could easily be applied to other use cases.
- A Python Flask application framework for running the above services.
- A simple persistent JSON object store.

10.2 Addressing the Research Questions

RQ1 (Problem Understanding)

"What problems with information overload are experienced by users of information systems and what attitudes do they have towards providers and solutions?"

We found that most of the users surveyed experienced issues with IO at some time, with inappropriate interruptions being a significant factor. Table 5.3 in Section 5.1.3.2 documents a number of common issues, with 79% of our respondents agreeing that they received notifications about things that could have waited for later, to the extent that 59% felt that getting interrupted while trying to get things done was a problem.

User attitudes towards online service providers were not generally positive, with trust concerns being a primary factor – 79% agreed that they were uncomfortable sharing personal information with online services due to lack of trust. However, trust in the competence of these services was also low: 70% didn't trust online services to make the best decisions about what content to show them, with 61% disagreeing that these services always get it right when judging what the users are interested in.

However, while there were clear overall patterns evidenced, we also found that there was significant variation in attitudes and experiences between different groups of users relating to individual topics, and not all users experienced what they would describe as IO.

Hypothesis H1: *Users desire better ways to manage information overload than are currently available.*

With 69% of our respondents agreeing that they don't have enough control over what online services choose for them, and 66% also agreeing that they want to be able to tell these services what matters to them most, there is a clear desire for improvement in ability to deal with IO. Our study participants [9.5.3] also agreed in interviews that better techniques for managing IO would be helpful to them.

Hypothesis H2: *The distinction between work and personal communications is important to most users.*

Respondents in the survey showed a preference for users wanting to keep work and personal communications entirely separate (60%), but with 38% having trouble switching off from work. While only 31% were happy to receive work notifications in personal time, 66% were happy to see personal notifications at work, suggesting that this is partly uneven, and that – while a distinction is important – users place more importance on their personal than work sanctity. Timing was an important corollary to this, with just under half of respondents agreeing that work and personal communication from the same device was acceptable if the timing was right.

Hypothesis H3: *Users are prepared to put in some effort to help a system improve their information overload situation.*

Our findings suggest an appetite for solving IO problems. Not only did 66% of our respondents agree that they want to be able to tell online services what matters to them most, a majority claimed that this was something they were prepared to work at – 56% agreed that they were happy to put effort in to train online services, while only 20% disagreed. This is consistent with what our study participants also reported.

RQ2 (Solution Development)

“What design direction and system features can address information overload for diverse users managing multiple online information sources?”

We investigated this question using a Research Through Design process [6.5], with much of the answer being contained in the artefacts that we created, in addition to the study outputs. The Design and Development Log [D] is a record of the design process that we followed, and we have referenced specific sections of this during the thesis to illustrate design problems and solutions, as well as to substantiate decisions that we made during the process.

We have formally assessed the implemented Awareness Agent prototype, comparing it to the system model [6.6] and documenting this as a gap analysis [6.9.2]. This is supplemented by a number of reflections that we made as part of the design and evaluation process [6.10]. While the answer to this research question primarily relates to the *design* – i.e. the system model that we arrived at and the process we undertook to arrive at it – the reflection on the implementation helps us assess the viability of a number of aspects of the system model. The study [9.1] then provides detail on specific elements of the agent as designed and implemented.

We made a number of design decisions that either arose from or were validated by our RtD process. An early decision was that modularity was an important design principle [6.4.3], driven by the need to address the IO problem across multiple information sources. Our earliest investigative work – discussing issues with users and investigating the data sources they use – highlighted the diversity of data sources and formats, as well as the range of technical and non-technical barriers that we found to access them [6.10.2]. This would also lead us in the direction of synthetic content when it came to studying the agent [7.1.1], but it also mandated the design of a content acquisition system that could cope with a highly heterogeneous set of content sources. This led us to the design for Acquire components [6.7.4.1] that could be deployed to different systems³, using APIs and techniques specific to those sources, yet feed back into the central AWAG system.

³Such as Slack workspaces via API integration or web sources via RSS

We believe that the design for the Acquire system has been very successful; this aspect of the prototype tested well, demonstrating support for heterogeneous sources in both push and pull modes of operation [6.6.1.1] with a clear path to expanding to other source types. It was also used to seamlessly integrate synthetic content for the study.

Another significant early design decision was to take a classifier-based approach with direct user control, leading to our design for User-Directed ML [6.6.6]. The design process that led to this flowed from the need to differentiate the handling of incoming content items, separating them by level of urgency or topic for example. A classifier is a natural tool to handle this, but we wanted to combine this with a self-service concept where the user not only owns the classifier model but also has full control over the lifecycle of the models.

The UD-ML approach was in part predicated on Hypothesis H3, that users are prepared to put in effort to help a system improve their IO issues, and a way of directly addressing H1, that users desire better ways to manage IO.

This was facilitated by the design concept of the Content Item [6.6.2], which we designed to address the identified requirements of Abstraction & Standardisation [6.4.2], in addition to Modularity & Commoditisation (which we quickly found required a commonly understood item format for interchange between modular components).

RQ2a:

“How can agent-based systems be designed to effectively manage and mitigate information overload across diverse online information sources?”

We established that the UD-ML classifier system could be used handle content from multiple sources in an effective manner, successfully testing the process of directing classified output into multiple channels for the user [6.7.6.4 & 9.5.3]. We found that this paradigm worked well in our experiments, but needs more testing in real life use. This process addresses the first part of our understanding of the IO problem and its solution – identifying content and directing it in such a way that it can be handled appropriately.

The prototype study showed that we were able to successfully use Content Item abstraction [6.4.2] to tackle diversity of sources and present information of multiple origin in consistent way, a key prerequisite for the proposed solution to work.

Study participants reported that the system handled content well, and this was supported by the synthetic and participant evaluation data [9.5.2]. We do need to caveat this though that some topics were harder to process than others [9.6.5]. The Summarise facility in the Augment service [6.6.2.6], was also effective in summarising long-form content for display to the users [9.5.3].

More generally, the design decision to use a modular Augment service [6.7.4.2] meant that we were not limited to a single option for making decisions about content. We demonstrated that UD-ML classification was effective for processing content from two types of source, but the Augment service supports adding additional methods for extracting information about an item, with the Allocate service [6.7.4.3] providing a means to manage this.

RQ2b:

"How effective is a prototype of a designed system in reducing information overload and improving user awareness across multiple communication and information platforms?"

The prototype Awareness Agent successfully processed content from multiple sources, with each in the study instance configured with a persona-specific set of real and synthetic content feeds [9.2.5.3]. We demonstrated the integration of this diverse material into a consolidated user interface [9.4.1 and 6.7.4.5]⁴, with content being classified and organised according to user-defined topic [6.7.6.4].

We found that the quality of decisions made by the system about this unified content was high [9.5.2]. The Manual Agreement rating in Table 9.1 – which is also consistent with synthetic results – shows that the system distinguished the following topics/properties:

⁴Study participants had access to both a Slack-based and bespoke UI that presented the same information in different ways [9.6.3]

- Urgency: 97%
- Work vs Personal: 91%
- Interesting content: 86%
- Topic-specific content: varies from 88% to 99% depending on topic

The demonstrated ability of the system to identify items that the user considers ‘urgent’ for example, allows non-urgent items to be withheld until appropriate times; similarly a demarcation is possible between work and personal content, and the system is also very effective at differentiating content that has come from multiple sources by topic. The ability to rapidly train new subject-specific models [9.6.1 & 6.7.6.1] allows them to adapt the handling of incoming information flows in ways that are harder for less flexible models or fixed filter setups. This results in a system that can effectively address IO for content coming from multiple sources.

RQ2c:

“What insights can be gained through the design and development of an agent-based system addressing information overload, particularly regarding user needs, design trade-offs, and implementation challenges?”

As noted above, the Design and Development Log [D] and our reflections [6.10] highlight a number of insights into how we can create and use such a system. We have identified the following main challenges:

- Access limitations to sources due to technical and non-technical barriers
- Identifying and dealing with ambiguous topics and content types
- Command and control of the system by the user
- Designing a user interface to meet all needs
- User confidence in the system – achieving sufficient user trust

We made some interesting design trade-offs. Opting for a classifier system was a signifi-

cant trade-off; we made a decision that classifying content was the most effective way to achieve our aims within the context of our work, but this does not meet every need. In particular, it does not really address how to view content in aggregate – either by taking a view over a large quantity of items or dealing with other concepts such as one item being superseded by another (such as a reply to an earlier message that renders the original less urgent). We believe that our modular approach allows for such gaps to be closed if needed. For example an augment instance could apply tracking of items and conversations and handle appropriately by identifying chains of conversation. Such limitations of the scope of our work are a trade-off driven by practicality and resources.

The abstracted Content Item concept was also a necessary compromise. This abstraction enables a system that treats content from heterogeneous sources in a uniform way, but at a cost of losing some source-specific content or formatting. We believe that this is a trade-off worth making, due to the utility it provides, and it can be mitigated by techniques such as giving the user links to original content *in situ* within the agent UI. Some types of content will be more affected by this than others, with simple items being least affected.

Hypothesis H4: *A system with explicit user self-training can be effectively used to manage content from multiple sources.*

As our answers to research questions RQ2a and RQ2b show, the UD-ML implementation within the Awareness Agent can be used to manage content coming from multiple sources and direct it appropriately [9.5.2].

Hypothesis H5: *Users will see rapid value from efforts to train personalised content prioritisation models.*

As discussed in Section 9.6.1 and illustrated in Figure 9.11, UD-ML models respond rapidly to user training – something that we also saw from participant feedback.

Hypothesis H6: *Self-trained AI models are a viable alternative for personal IO management to other techniques such as filtering or third party systems.*

We found that the UD-ML prototype was successful in performing tasks that might otherwise be approached using filtering or other systems [9.7.1]. Self-trained models had a high success rate for correctly identifying and directing content. The training process was simple and fast, meaning it is potentially less work required to set up than a complex set of filters.

We also note that using UD-ML and manual filters are not mutually exclusive, and that the Awareness Agent system model supports a combination of the two via the modular Augment service [6.7.4.2].

RQ3 (Evaluation Techniques)

"Can synthetic techniques be used effectively to study and evaluate potential solutions to Information Overload?"

We found synthetic techniques were successful overall – as discussed in Section 9.7.2. Study participants reported that most synthetic content was believable and consistent with the topic, which was supported by classification results

The synthetic content technique was also viable from a practical perspective: we were successful in automating the generation and deployment of large volumes of synthetic message content for testing. Once we had established the process for one persona, it was easy to do so for others⁵. Although synthetic topic definition took some initial work, this yielded rich content from relatively limited effort.

We did find that the quality of synthetic evaluation varied considerably by OpenAI model and technique [Figure 9.9], with newer LLMs unsurprisingly being more capable. Our best synthetic evaluation implementation [9.5.1.5] was very close to human input for the large majority of UD-ML models [Table 9.1] and we consider this a reliable technique overall for evaluating classification performance

RQ3a:

"Is synthetic content an effective substitute for real content to support analysis of Information Overload problems?"

We found that synthetic content was an effective substitute for real data in our study. As noted above, we were successful in generating this content and employing it in the study instances, and study participants found it to be generally realistic. By using synthetic content we were able to run instances of the study much more easily and comprehensively than we otherwise could, freed from the need to seek access to corporate or personal data and the pitfalls associated with that process.

⁵The configuration information for each study instance, including synthetic content generation can be found in Supplement S10.1 [[doi:10.21954/ou.rd.28045577](https://doi.org/10.21954/ou.rd.28045577)], with details of the topics and entities used in Appendix E

RQ3b:

"Can we use a synthetic evaluation approach to evaluate a potential solution or service to address Information Overload?"

We found that our synthetic evaluation process was also an effective way to evaluate our UD-ML solution, and believe that this technique could be applied to other similar systems. The synthetic system was able to achieve a much higher workrate than even our most enthusiastic study participant, and we saw a statistically significant relationship between the evaluation decisions made by our virtual study participant and those made by the real one. In those areas where the synthetic evaluator was weaker [9.6.5], we found that the human evaluator was also weaker. While this is something to address with further research, it is not a weakness of the synthetic evaluation system itself.

Hypothesis H7: *By using synthetic as well as real data for studies, we can avoid some data confidentiality and access issues that would otherwise limit scope without significant negative consequences.*

A significant potential risk was that synthetic content would be insufficiently realistic to substitute for real content in a study. However, we were able to generate content that was sufficiently believable for our study participants [9.5.3 & 9.5.4] that was classified consistently by human and synthetic evaluators [9.5.2]. We achieved this by formalising a system to define not only topics for content [7.4.0.3], but also a virtual 'cast' [7.4.0.4] and other entities [7.4.0.5] to include in simulated content, as well as a mechanism for generating conversation chains and avoiding repetition [7.4.0.6].

Hypothesis H8: *We can use AI-driven evaluation techniques to partially replace humans in studies of information overload solutions, minimising the problem of overloading the study participants.*

The synthetic evaluators were able to process more items than their human counterparts⁶, and as can be seen in Section 9.5.2.2 & Table 9.1, there is a strong similarity between real and synthetic evaluators' responses – implying a high degree of fidelity to human evaluations. The process of administering synthetic evaluations – execution [9.3.3.2], recording [9.4.2] and analysis of results [9.2.7] – was straightforward for the study administrator.

Hypothesis H9: *Human feedback on synthetic content and evaluation processes is an effective strategy to ensure overall experimental integrity.*

We believe that human feedback into the synthetic process is an effective and necessary element of experimental integrity. We were able to use human input⁷ to validate the strengths and identify the potential weaknesses of synthetic techniques. Without this we would not have been able to have confidence in the reliability of the synthetic element – and importantly we would not have been able to identify those areas where the synthetic evaluator in particular is more prone to error [9.5.2.2]. This is important qualifying information for a study.

The human feedback also allowed us to identify which techniques and supporting LLM combinations were most performant [9.7.2], an essential requirement for confidence in the output.

⁶Detailed in doi:10.21954/ou.rd.28044944 [path: /study/data/stats/generated]

⁷Explicit feedback [9.5.1.1] and inferred feedback [9.5.1.2 & 9.5.1.3]

10.3 Limitations

10.3.1 Awareness Agent Design Actualisation

The scope of the Awareness Agent design was too wide to fully implement within the scope of our project, so while we implemented some areas in great depth, gaps remained. These are detailed in Sections 6.9.1 and 6.9.2, but the significant items are:

- **Autonomous Operation:** As detailed in Section 6.9.2.1, the prototype does not have the level of independence and autonomy envisioned in Section 6.4.1, but there is a clear design path to addressing this.
- **Outward Representation:** Representing the owner externally [6.9.2.13] is an aspect that has some partial implementation with the Exchange Service [6.7.4.4], but is not fully developed.
- **Explainable AI:** While we can argue that the UD-ML concept is inherently user-friendly, with control over the whole model lifecycle given over to the user, only limited information about individual UD-ML decisions and model state is available to them. Additionally, mechanisms for user-review of agent self-training [6.9.2.12] is not within the current scope of the application.

10.3.2 Testing Information Overload vs. Testing UD-ML

In our discussion of RQ2b [10.2], we to an extent used the performance of UD-ML in classifying incoming content in a realistic scenario as a proxy for how well the agent handles IO. We believe that this approach is justified by the demonstrated effectiveness of this process, alongside feedback from participants that it provides them with a powerful tool to counter IO. However, we recognise that assessing how the solution helps users with IO in the real world involves other factors, and that a detailed and targetted study would be needed to validate this. Such a study would require the participant to use an Awareness Agent for an extended period in a situation where they would normally be experiencing real IO issues, and assessment would need to quantify the extent to which the agent alleviates this. There are two significant barriers to this type of study: 1) the

agent would need to be accessing a wide range of data used by the participant in real life, which may be subject to the type of restrictions that we have already discussed; 2) the user would need to rely on the agent as a primary information source for long enough to gain real data, which is a significant commitment from a participant that requires trust in the performance of the agent.

We believe that the work we have done paves the way for a second stage study to investigate further, having developed and established the performance of an initial prototype agent that can be taken forward.

10.3.3 Underlying Reliability of Current Large Language Models

The fallability of major LLMs has been widely discussed [Emsley, 2023] [Alkaissi and McFarlane, 2023] [Chelli et al., 2024] and this is something that we were conscious of during our research. That concern was partially mitigated for us by the nature of our research: we were not relying on LLM output to perform any critical task, and we set a deliberately limited expectation for the primary generative use of this technology in our work [7.3.1].

Furthermore, while we acknowledge that the study of the quality of synthetic outputs was more informal and not comprehensive [7.6.2], some aspects of the study have been explicitly set up to assess the performance of the LLMs. The metrics Evaluation Feedback [9.2.7.5] and Evaluation Agreement [9.2.7.6, 9.2.7.8] rate the quality of LLM output using feedback from the human study participant. This aspect of the study generated statistics that include unwanted LLM behaviour, although it did not differentiate that from simple poor performance on subjective matters such as awarding evaluation scores.

We additionally implemented some other mechanisms for gaining a measure of additional explicit feedback from study participants, such as summarisation feedback⁸ and Likert text mismatch recording⁹, as well as direct participant observations [9.5.3]. These could be used to capture data about when the human participant noticed certain types of LLM misbehaviour.

⁸Development Log [D] entry 2023-04-16

⁹Development Log [D] entry 2024-04-04

We also included some systematic checks, including recording evaluation failures based on round-trip classification validation [8.2.5] and systematic recording of all evaluation request and response data [6.8.3].

However, we acknowledge that although we captured this data, we did not make systematic use of it during the final study. Instead it was used on an *ad hoc* basis during the course of study development and execution to guide our use of the LLMs.

We have identified the quality of Synthetic Content as an area for further focussed study [10.4.6], and there is also scope to improve the analysis of the failure and anomaly data captured duration the course of the study.

10.3.4 Context Mapping

Our system model includes a concept of context mapping [6.6.3], where the agent adopts a form of modal operation according to the operating context of the agent/user (for example to only deliver non-urgent personal content when the user is in a ‘personal’ context). While we developed this as a design concept, we have not implemented it in the prototype agent so were not able to test this aspect of the solution in practice.

However, we were able to demonstrate a shift for the user from source-specific interaction to topic-specific interaction – for example, with everything about work logistical arrangements in one place regardless of transmission medium, so that the user can go to one channel to look at everything relating to a single hobby or activity. This is a core element of our design to combat IO, which would be strengthened by the addition of context mapping.

10.3.5 Design Process

We would have done things differently in some ways if we were starting out now; the capabilities of commodity cognitive computing services in particular have radically expanded what is possible from what was originally available to us. The design lesson that we take from this is the importance of creating an adaptable and extensible design. In this respect

we believe that our original decision to take a commoditisation approach [6.4.3] has served well; this makes it much easier to adapt the agent design to take advantage of new and enhanced capabilities. Having said that, if we were starting our work in the present day, we would place an early emphasis on natural language interaction between user and agent, and give AI-driven content discovery a higher priority within a similar overall framework.

10.4 Further Work

10.4.1 Autonomous Operation

Advances in commodity AI present many opportunities for adding powerful self-discovery and adaptation features to the Awareness Agent. This could be accommodated within the existing modular architecture by adding new services to the design. For example, a *Discovery Service* might periodically examine the history of highly rated Content Items and look for new information sources that are similar to recommend to the user. The efficacy of such a service (measured by user rating of recommendations for example) could be a useful area of study, connecting the Awareness Agent to the research area of recommender systems.

10.4.2 Natural Language Communication With The Agent

Similarly, public LLM advances mean that there is much more accessible and flexible capability available for communications between the Awareness Agent and its owner. For example, while configuration has up to now been done via JSON-based commands, the Interact service [6.7.4.5] could be extended to allow the user to define UD-ML models, schedules and other parameters via natural language commands entered via the chat-based interface. This would be compatible with the current Slack-based architecture, requiring a layer to be added to transform conversational inputs to internal commands.

Similarly, natural language could be used to communicate outputs to the user. Rather than simply publishing content to channels as at present, an AI-based engine called by the Allocate layer could make decisions about how to handle items based on UD-ML classification

and other factors, with possible output options including emitting urgent notifications or preparing a natural language summary of content to send to the user as a lower priority.

There is also the possibility for the user to start a conversation with the agent, such as “Tell me what’s going on with X” – this could be a powerful interaction method and rich subject for further study. Again, this functionality is supported by the current agent architecture, requiring an additional internal service to facilitate the conversation.

10.4.3 Handling Ambiguity

We found in our study that some topics were harder to define than others, for human and synthetic system alike [9.6.5]. This is a potential topic for further research: the current framework could be employed to study in more depth the nature of the content that is less easy to classify and devise strategies for addressing the problem. There are a number of approaches that could be tested. For example, as based off the work of Franco, Gaggi, and Palazzi [2023], we might add a confidence metric to the augmentation produced by UD-ML and use this to trigger mitigation strategies such as changed notification thresholds or running additional assessment with different AI engines. A confidence value for UD-ML classifications could also be used to enhance the synthetic evaluation process.

10.4.4 UD-ML Classification

There is scope for more formalised research into the UD-ML classification system by applying standard benchmarks to the output [2.5.4]. The work of Sun et al. [2023] could provide a useful reference point.

10.4.5 UD-ML Impact on Information Overload

As we discussed in Section 10.3.2, our understanding of the area could be improved by a more immersive study, where an Awareness Agent with UD-ML is employed in a real life scenario for a user over a period of time to gain insight into the practical impact of UD-ML on users’ information overload.

10.4.6 Synthetic Content

We believe there are a number of advances that can be made in synthetic content generation and assessment. For example, the technique could be expanded to include a formal Locations document [7.6.1] in the same way that we use *Entities* and *Dramatis Personae*. We have also not looked specifically at the quality of synthetic content in a study – instead we have assessed this only within the context of our UD-ML study. There is value in a focussed investigation into the quality of synthetic content, quantifying realism using human assessment and differentiating between the quality of different topics, supporting documents and topic construction techniques.

10.4.7 Exchange Service and AXP

The Exchange Service and its supporting protocol AXP [6.7.4.4] is not significantly developed in this thesis. While we have proved the concept of distributing CIs via an MQTT broker, there are many aspects for further work on this topic.

We prefer a medium-agnostic approach, but an appropriate next step is reviewing the suitability of MQTT as the transmission system. We chose MQTT because it is lightweight and easy to set up, with features that we need such as topics that can be subscribed to, but this was developed for Internet of Things (IoT) and not for the sort of information exchange that we use.

There are many aspects of security and trust that would need to be investigated. For example, if an Awareness Agent was emitting confidential material as exchange items¹⁰, mechanisms would need to exist that prevent any unauthorised actors from receiving that material. Similarly, authenticity and trust mechanisms would need to be established – so for example that an agent can authenticate the sending agent of a given item, and handle content according to agent identity and trust levels. Introducing cryptography for content security and message signing would be one approach to this.

¹⁰Such as content that might come from a corporate workspace

10.5 Final Thoughts

We began this work in 2016, when the landscape was very different from now – in particular, this was some time before the commoditisation of Generative AI and Large Language Models changed the face of public computing. As of 2025, powerful LLMs from OpenAI and others have made many previously unreachable capabilities available to end users and application developers. This has affected many areas of in-progress academic research [Qasem, 2023] [Lund et al., 2023], including our own. In most cases, academics have been interested (or concerned) about the use of these tools to generate content, assist with writing & reviewing academic content, and search for literature.

In our case, we have used Generative AI to address some specific issues that we have faced in this research – namely to obtain synthesised content that is not subject to privacy or confidentiality concerns, and to take on the role of a virtual study participant that is more attentive and productive than we can expect from a human. We also kept some things the same – specifically our existing approach to using machine learning within our prototype software agents.

In the process of using Generative AI to address those issues, we also found that this was itself a valuable seam of research, and we shifted some of our efforts to focus on examining these techniques in their own right. Our necessary adaption to the change in the AI landscape during the course of our work means that the outcome is somewhat different to what we envisaged at the outset. While we would have done a number of things quite differently if we were starting our work in the current landscape, we think the change that has occurred while we have been looking at the topic has led us in interesting and worthwhile directions, so has been a clear positive overall.

Bibliography

- [Abt, 1998] Abt, Helmut A (1998). "Why some papers have long citation lifetimes". In: *Nature* 395.6704, pp. 756–757. issn: 0028-0836. doi: 10.1038/27355.
- [Adamczyk and B P Bailey, 2004] Adamczyk, P D and B P Bailey (2004). "If not now when?: the effects of interruption at different moments within task execution". In: *Proceedings of the SIGCHI conference on Human factors in computing systems* 6.1, pp. 271–278. issn: 1581137028. doi: 10.1145/985692.985727.
- [Adomavicius and Tuzhilin, 2005] Adomavicius, Gediminas and Alexander Tuzhilin (2005). "Personalization technologies: a process-oriented perspective". In: *A process-oriented perspective. Communications of the ACM* 48.10, pp. 83–90. issn: 00010782. doi: 10.1145/1089107.1089109.
- [Ahmad and Wegner, 1999] Ahmad, Morad and Lutz Wegner (1999). *Design Issues and First Experiences with an Awareness Server for Synchronous CSCW*. Tech. rep. Universität Kassel. https://www.researchgate.net/publication/2451429_Design_Issues_and_First_Experiences_with_an_Awareness_Server_for_Synchronous_CSCW.
- [Aljukhadar, Senecal, and Daoust, 2010] Aljukhadar, Muhammad, Sylvain Senecal, and Charles Etienne Daoust (2010). "Information overload and usage of recommendations". In: *Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI)*. Vol. 612. Barcelona: CEUR-WS.org, pp. 26–33. <http://ceur-ws.org/Vol-612/paper5.pdf>.

- [Alkaissi and McFarlane, 2023] Alkaissi, Hussam and Samy I McFarlane (2023). "Artificial Hallucinations in ChatGPT: Implications in Scientific Writing". In: *Cureus* 15.2, pp. 2–5. issn: 2168-8184. doi: [10.7759/cureus.35179](https://doi.org/10.7759/cureus.35179).
- [Antona et al., 2023] Antona, Margherita et al. (2023). "Special Issue on AI in HCI". In: *International Journal of Human-Computer Interaction* 39.9, pp. 1723–1726. issn: 15327590. doi: [10.1080/10447318.2023.2177421](https://doi.org/10.1080/10447318.2023.2177421).
- [Arnold, Goldschmitt, and Rigotti, 2023] Arnold, Miriam, Mascha Goldschmitt, and Thomas Rigotti (2023). "Dealing with information overload: a comprehensive review". In: *Frontiers in Psychology* 14.June. issn: 16641078. doi: [10.3389/fpsyg.2023.1122200](https://doi.org/10.3389/fpsyg.2023.1122200).
- [Artikis and Sergot, 2009] Artikis, Alexander and Marek Sergot (2009). "Executable specification of open multi-agent systems". In: *Logic Journal of the IGPL* 18.1, pp. 31–65. issn: 13670751. doi: [10.1093/jigpal/jzp071](https://doi.org/10.1093/jigpal/jzp071).
- [Auer et al., 2013] Auer, Sören et al. (2013). "Introduction to Linked Data and Its Lifecycle". In: *Reasoning Web: Semantic Technologies for Intelligent Data Access 9th International Summer School 2013, July 30 – August 2, 2013*. Mannheim, Germany: Springer, Berlin, Heidelberg, pp. 1–99. isbn: 978-3-319-10586-4. doi: [10.1007/978-3-319-10587-1_1](https://doi.org/10.1007/978-3-319-10587-1_1).
- [Brian P. Bailey and Konstan, 2006] Bailey, Brian P. and Joseph A. Konstan (2006). "On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state". In: *Computers in Human Behavior* 22.4, pp. 685–708. issn: 07475632. doi: [10.1016/j.chb.2005.12.009](https://doi.org/10.1016/j.chb.2005.12.009).
- [Bansal et al., 2023] Bansal, Gagan et al. (2023). "Workshop on Trust and Reliance in AI-Human Teams (TRAIT)". In: *Conference on Human Factors in Computing Systems - Proceedings*. doi: [10.1145/3544549.3573831](https://doi.org/10.1145/3544549.3573831).
- [Benford, Bowers, et al., 1994] Benford, Steve, John Bowers, et al. (1994). "Managing Mutual Awareness in Collaborative Virtual Environments". In: *Proceedings of the Conference on Virtual Reality Software And Technology (VRST'94)* August 1994, pp. 223–236. doi: [10.1142/9789814350938_0018](https://doi.org/10.1142/9789814350938_0018).

- [Benford and Fahlén, 1993] Benford, Steve and Lennart Fahlén (1993). "A spatial model of interaction in large virtual environments". In: *Proceedings of the Third European Conference on Computer-Supported Cooperative Work 13–17 September 1993, Milan, Italy ECSCW'93*. Milan, Italy: Springer Netherlands, pp. 109–124. doi: [10.1007/978-94-011-2094-4_8](https://doi.org/10.1007/978-94-011-2094-4_8).
- [Berners-Lee and Fischetti, 1999] Berners-Lee, Tim and Mark Fischetti (1999). *Weaving the Web: the past, present and future of the World Wide Web by its inventor*. New York: Harper Collins.
- [Berners-Lee, James Hendler, and Lassila, 2001] Berners-Lee, Tim, James Hendler, and Ora Lassila (2001). "The Semantic Web". In: *Scientific American* 284.5, pp. 34–43. issn: 0036-8733. doi: [10.1038/scientificamerican0501-34](https://doi.org/10.1038/scientificamerican0501-34).
- [Bettis-Outland, 2012] Bettis-Outland, Harriette (2012). "Decision-making's impact on organizational learning and information overload". In: *Journal of Business Research* 65.6, pp. 814–820. issn: 01482963. doi: [10.1016/j.jbusres.2010.12.021](https://doi.org/10.1016/j.jbusres.2010.12.021).
- [Bogg, Beydoun, and Low, 2008] Bogg, Paul, Ghassan Beydoun, and Graham Low (2008). "When to use a multi-agent system?" In: *Proceedings of the 11th Pacific Rim International Workshop on Multi Agents: Intelligent Agents and Multi-Agent Systems*. Ed. by The Duy Bui, Tuong Vinh Ho, and Quang Thuy Ha. Vol. 5357 LNAI. Hanoi: Springer, Berlin, Heidelberg, pp. 98–108. isbn: 3540896732. doi: [10.1007/978-3-540-89674-6_13](https://doi.org/10.1007/978-3-540-89674-6_13).
- [Bond et al., 2019] Bond, Raymond R et al. (2019). "Human Centered Artificial Intelligence: Weaving UX into Algorithmic Decision Making". In: *RoCHI 2019: International Conference on Human-Computer Interaction - Romania*. Bucharest, Romania, pp. 2–9.
- [Boulanger, 2005] Boulanger, A (2005). "Open-source versus proprietary software: Is one more reliable and secure than the other?" In: *IBM Systems Journal* 44.2, pp. 239–248. doi: [10.1147/sj.442.0239](https://doi.org/10.1147/sj.442.0239).
- [Boyle and Saul Greenberg, 2005] Boyle, Michael and Saul Greenberg (2005). "The Language of Privacy : Learning from Video Media Space Analysis and Design". In: *ACM Transactions on Computer-Human Interaction* 12.2, pp. 328–370. issn: 10730516. doi: [10.1145/10730516](https://doi.org/10.1145/10730516).

10.1145/1067860.1067868. <http://grouplab.cpsc.ucalgary.ca/grouplab/uploads/Publications/Publications/2005-LanguagePrivacy.TOCHI.pdf>.

[Buitinck et al., 2013] Buitinck, Lars et al. (2013). “{API} design for machine learning software: experiences from the scikit-learn project”. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122.

[Bürger, 1999] Bürger, Martin (1999). “Unterstützung von Awareness bei der Gruppenarbeit mit gemeinsamen Arbeitsbereichen (Support of awareness for group work with shared workspaces)”. PhD thesis. TU München.

[Cabrero-Daniel et al., 2024] Cabrero-Daniel, Beatriz et al. (2024). “Exploring Human-AI Collaboration in Agile: Customised LLM Meeting Assistants”. In: *Agile Processes in Software Engineering and Extreme Programming*. Ed. by Darja Šmite et al. Cham: Springer Nature Switzerland, pp. 163–178. isbn: 978-3-031-61154-4.

[Canter, 2017] Canter, Manc (2017). *What's the difference between a Bot and an Agent?* <https://medium.com/ai-blogging/whats-the-difference-between-a-bot-and-an-agent-6b99eb3a3d0d> <https://perma.cc/6VNA-V828> (visited on 11/23/2017).

[Capadisli, Guy, Lange, et al., 2016] Capadisli, Sarven, Amy Guy, Christoph Lange, et al. (2016). *Linked Data Notifications: a resource-centric communication protocol*. <http://csarven.ca/linked-data-notifications> <https://perma.cc/7SVP-JPBV> (visited on 05/25/2017).

[Capadisli, Guy, Verborgh, et al., 2017] Capadisli, Sarven, Amy Guy, Ruben Verborgh, et al. (2017). “Decentralised Authoring, Annotations and Notifications for a Read-Write Web with dokiel”. In: *Web engineering: 17th international conference, ICWE 2017 Rome, Italy, June 5-8*. Rome: Springer, Berlin, Heidelberg, pp. 469–481. isbn: 9783319601304. doi: 10.1007/978-3-319-60131-1.

[Castelfranchi, 1995] Castelfranchi, Cristiano (1995a). “Commitments: from individual intentions to groups and organizations”. In: *Proceedings of the First International Conference on Multiagent Systems, ICMAS 1995*, pp. 41–48.

- [Castelfranchi, 1995] — (1995b). “Guarantees for Autonomy in Cognitive Agent Architecture”. In: *Intelligent Agents* 890, pp. 56–70. issn: 16113349. doi: [10.1007/3-540-58855-8_3](https://doi.org/10.1007/3-540-58855-8_3). http://link.springer.com/chapter/10.1007/3-540-58855-8_3.
- [Castelfranchi, 2014] — (2014). “Minds as social institutions”. In: *Phenomenology and the Cognitive Sciences* 13.1, pp. 121–143. issn: 15728676. doi: [10.1007/s11097-013-9324-0](https://doi.org/10.1007/s11097-013-9324-0).
- [Chan et al., 2023] Chan, Alex J et al. (2023). *Harmonizing Global Voices: Culturally-Aware Models for Enhanced Content Moderation (preprint)*. Tech. rep. arXiv: arXiv:2312.02401v1.
- [Chang et al., 2024] Chang, Yupeng et al. (2024). “A Survey on Evaluation of Large Language Models”. In: *ACM Transactions on Intelligent Systems and Technology* 15.3, pp. 1–45. issn: 2157-6904. doi: [10.1145/3641289](https://doi.org/10.1145/3641289). arXiv: [2307.03109](https://arxiv.org/abs/2307.03109).
- [Chavez, 2023] Chavez, Tom (2023). *How We Can And Should Regain Control Of The Recommendation Algorithm*. <https://www.forbes.com/sites/tomchavez/2023/02/23/how-we-can-and-should-regain-control-of-the-recommendation-algorithm/> <https://perma.cc/GS09-GSHT> (visited on 11/12/2024).
- [Chelli et al., 2024] Chelli, Mikaël et al. (2024). “Hallucination Rates and Reference Accuracy of ChatGPT and Bard for Systematic Reviews: Comparative Analysis”. In: *Journal of Medical Internet Research* 26, pp. 1–11. doi: [10.2196/53164](https://doi.org/10.2196/53164).
- [Chen et al., 2022] Chen, Yunang et al. (2022). “Experimental Security Analysis of the App Model in Business Collaboration Platforms”. In: *Proceedings of the 31st USENIX Security Symposium, Security 2022*, pp. 2011–2028.
- [Chi, 2017] Chi, Ed H. (2017). “Humans and computers working together on hard tasks”. In: *Communications of the ACM* 60.9, pp. 92–92. issn: 00010782. doi: [10.1145/3068614](https://doi.org/10.1145/3068614).
- [Clavel and Callejas, 2016] Clavel, Chloé and Zoraida Callejas (2016). “Sentiment Analysis: From Opinion Mining to Human-Agent Interaction”. In: *IEEE Transactions on Affective Computing* 7.1, pp. 74–93. issn: 19493045. doi: [10.1109/TAFFC.2015.2444846](https://doi.org/10.1109/TAFFC.2015.2444846).

- [J.A. Cohen, 1960] Cohen, J.A. (1960). "A coefficient of agreement for nominal scales". In: *Educational And Psychological Measurement* XX.1, pp. 37–46. doi: [10.1177/001316446002000104](https://doi.org/10.1177/001316446002000104).
- [Jacob Cohen et al., 2002] Cohen, Jacob et al. (2002). *Applied multiple regression/correlation analysis for the behavioral sciences*. Routledge. isbn: 9780203774441. doi: [10.4324/9780203774441](https://doi.org/10.4324/9780203774441).
- [Cooper, 1998] Cooper, Alan (1998). *The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity*. Indianapolis, Indiana: Sams. isbn: 9780672326141.
- [Correia et al., 2023] Correia, António et al. (2023). "A hybrid human–AI tool for scientometric analysis". In: *Artificial Intelligence Review* 56.s1, pp. 983–1010. issn: 15737462. doi: [10.1007/s10462-023-10548-7](https://doi.org/10.1007/s10462-023-10548-7).
- [Cortes and Vapnik, 1995] Cortes, Corinna and Vladimir Vapnik (1995). "Support-Vector Networks". In: *Machine Learning* 20.3, pp. 273–297. doi: [10.1007/BF00994018](https://doi.org/10.1007/BF00994018).
- [Cui et al., 2011] Cui, Weiwei et al. (2011). "Textflow: Towards better understanding of evolving topics in text". In: *IEEE Transactions on Visualization and Computer Graphics* 17.12, pp. 2412–2421. issn: 10772626. doi: [10.1109/TVCG.2011.239](https://doi.org/10.1109/TVCG.2011.239). arXiv: [1404.4606](https://arxiv.org/abs/1404.4606).
- [Cutrell, Czerwinski, and Horvitz, 2001] Cutrell, Edward, Mary Czerwinski, and Eric Horvitz (2001). "Notification, Disruption, and Memory: Effects of Messaging Interruptions on Memory and Performance". In: *Conference on Human-Computer Interaction Interact 2001* 1999, pp. 263–269.
- [Dahana et al., 2024] Dahana, Wirawan Dony et al. (2024). "Impact of on-topic and off-topic discussions on member participation and contribution in a common-identity online community". In: *Telematics and Informatics Reports* 16.June. issn: 27725030. doi: [10.1016/j.teler.2024.100172](https://doi.org/10.1016/j.teler.2024.100172).
- [Davis, 2013] Davis, Philip M (2013). "Journal Usage Half-Life". In: 5 p. http://www.publishers.org/_attachments/docs/journalusagehalflife.pdf.

- [Deelman et al., 2009] Deelman, Ewa et al. (2009). "Workflows and e-Science: An overview of workflow system features and capabilities". In: *Future Generation Computer Systems* 25.5, pp. 528–540. issn: 0167739X. doi: [10.1016/j.future.2008.06.012](https://doi.org/10.1016/j.future.2008.06.012). <http://dx.doi.org/10.1016/j.future.2008.06.012>.
- [Dennett, 1987] Dennett, Daniel Clement (1987). *The Intentional Stance*. MIT Press, Cambridge, MA. isbn: 026204093X.
- [Derbyshire et al., 1997] Derbyshire, D. et al. (1997). "Agent-based digital libraries: driving the information economy". In: *Journal of Engineering and Applied Science*, pp. 82–86. issn: 11101903.
- [Dinh and Tamine, 2012] Dinh, Duy and Lynda Tamine (2012). "Towards a context sensitive approach to searching information based on domain specific knowledge sources". In: *Web Semantics: Science, Services and Agents on the World Wide Web* 12-13, pp. 41–52. issn: 15708268. doi: [10.1016/j.websem.2011.11.009](https://doi.org/10.1016/j.websem.2011.11.009). <http://linkinghub.elsevier.com/retrieve/pii/S1570826811001004>.
- [Dourish and Bellotti, 1992] Dourish, Paul and Victoria Bellotti (1992). "Awareness and Coordination in Shared Workspaces". In: November, pp. 107–114.
- [Dube and Verster, 2023] Dube, Lindani and Tanja Verster (2023). "Enhancing classification performance in imbalanced datasets: A comparative analysis of machine learning models". In: *Data Science in Finance and Economics* 3.4, pp. 354–379. issn: 2769-2140. <http://www.aimspress.com/article/doi/10.3934/DSFE.2023021>.
- [Dulin and A. Ziegler, 2017] Dulin, Kim and Adam Ziegler (2017). "Scaling up perma.cc: Ensuring the integrity of the digital scholarly record". In: *D-Lib Magazine* 23.5/6. doi: [10.1045/may2017-dulin](https://doi.org/10.1045/may2017-dulin). <https://perma.cc/H82L-R8C8>.
- [Earley, 2015] Earley, Seth (2015). "Machine Learning and Cognitive Computing". In: *IT Professional* 17.5, pp. 62–69. issn: 15209202. doi: [10.1109/MITP.2015.81](https://doi.org/10.1109/MITP.2015.81).

[Emsley, 2023] Emsley, Robin (2023). "ChatGPT: these are not hallucinations – they're fabrications and falsifications". In: *Schizophrenia* 9.1, pp. 4–5. issn: 27546993. doi: 10.1038/s41537-023-00379-4.

[Fasli, 2003] Fasli, Maria (2003). "From social agents to multi-agent systems: preliminary report". In: *3rd International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2003 Prague, Czech Republic, June 16–18, 2003*. Ed. by Vladimír Marík, Michal Pechoucek, and Jörg Müller. Prague: Springer, Berlin, Heidelberg, pp. 111–121. http://link.springer.com/chapter/10.1007/3-540-45023-8_12.

[Faulring et al., 2010] Faulring, Andrew et al. (2010). "Agent-assisted task management that reduces email overload". In: *Proceedings of the 15th international conference on Intelligent user interfaces - IUI '10*, p. 61. issn: 1605585157. doi: 10.1145/1719970.1719980. <https://pal.sri.com/> <https://perma.cc/3L8P-R4NB>.

[Feigenbaum et al., 2007] Feigenbaum, Lee et al. (2007). "The Semantic Web In Action". In: *Scientific American* December, pp. 90–97.

[Ferrara et al., 2016] Ferrara, Emilio et al. (2016). "The Rise of Social Bots". In: *Communications of the ACM* 59.7, pp. 96–104. issn: 00010782. doi: 10.1145/2818717. arXiv: 1407.5225.

[B. Ferreira et al., 2018] Ferreira, Bruna et al. (2018). "Technique for representing requirements using personas: a controlled experiment". In: *IET Software* 12.3, pp. 280–290. issn: 1751-8806. doi: 10.1049/iet-sen.2017.0313. <http://digital-library.theiet.org/content/journals/10.1049/iet-sen.2017.0313>.

[B. M. Ferreira, Diniz Junqueira Barbosa, and Conte, 2016] Ferreira, Bruna Moraes, Simone Diniz Junqueira Barbosa, and Tayana Conte (2016). "PATHY: Using Empathy with Personas to Design Applications that Meet the Users' Needs". In: *Human-Computer Interaction. Theory, Design, Development and Practice: 18th International Conference, HCI International 2016*. Vol. 9731, pp. 153–165. isbn: 978-3-319-39509-8. doi: 10.1007/978-3-319-39510-4_15.

- [Findeli et al., 2008] Findeli, Alain et al. (2008). "Research Through Design and Transdisciplinarity: A Tentative Contribution to the Methodology of Design Research". In: "*FOCUSED*" - Current Design Research Projects and Methods Swiss Design Network Symposium January, pp. 67–91.
- [J. E. Fischer et al., 2010] Fischer, Joel E. et al. (2010). "Effects of content and time of delivery on receptivity to mobile interruptions". In: *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services - MobileHCI '10*, p. 103. doi: [10.1145/1851600.1851620](https://doi.org/10.1145/1851600.1851620).
- [Franco, Gaggi, and Palazzi, 2023] Franco, Mirko, Ombretta Gaggi, and Claudio E. Palazzi (2023). "Analyzing the Use of Large Language Models for Content Moderation with ChatGPT Examples". In: *Proceedings of the 2023 Workshop on Open Challenges in Online Social Networks, OASIS 2023, Held in conjunction with the 34th ACM conference on Hypertext and Social Media, HT 2023*, pp. 1–8. doi: [10.1145/3599696.3612895](https://doi.org/10.1145/3599696.3612895).
- [Franklin and Graesser, 1996] Franklin, Stan and Art Graesser (1996). "Is It an agent, or just a program?: A taxonomy for autonomous agents". In: *Intelligent agents III: agent theories, architectures, and languages: ECAI'96 Workshop (ATAL), Budapest, Hungary, August 12-13, 1996*. Ed. by Jörg P. Müller, Michael J. Wooldridge, and Nicholas R. Jennings. Budapest, Hungary: Springer, Berlin, Heidelberg, pp. 21–35.
- [Friestad and Wright, 1995] Friestad, Marian and Peter Wright (June 1995). "Persuasion Knowledge: Lay People's and Researchers' Beliefs about the Psychology of Advertising". In: *Journal of Consumer Research* 22.1, pp. 62–74. issn: 0093-5301. doi: [10.1086/209435](https://doi.org/10.1086/209435). <https://doi.org/10.1086/209435>.
- [Friis Dam and Teo, 2024] Friis Dam, Rikke and Yu Siang Teo (2024). *Personas – A Simple Introduction*. <https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them> <https://perma.cc/Y8UX-HV6T> (visited on 08/14/2024).
- [Gabriel, 2020] Gabriel, Iason (2020). "Artificial Intelligence, Values, and Alignment". In: *Minds and Machines* 30.3, pp. 411–437. issn: 15728641. doi: [10.1007/s11023-020-09539-2](https://doi.org/10.1007/s11023-020-09539-2). arXiv: [2001.09768](https://arxiv.org/abs/2001.09768).

- [Gaines, 2012] Gaines, Brian R. (2012). "Knowledge acquisition: Past, present and future". In: *International Journal of Human Computer Studies* 71.2, pp. 135–156. issn: 10715819. doi: [10.1016/j.ijhcs.2012.10.010](https://doi.org/10.1016/j.ijhcs.2012.10.010).
- [Garcia, Giret, and Botti, 2008] Garcia, Emilia, Adriana Giret, and Vicente Botti (2008). "Towards an evaluation framework for MAS software engineering". In: *Proceedings of the 11th Pacific Rim International Workshop on Multi Agents: Intelligent Agents and Multi-Agent Systems*. Ed. by The Duy Bui, Tuong Vinh Ho, and Quang Thuy Ha. Vol. 5357 LNAI. Hanoi: Springer, Berlin, Heidelberg, pp. 197–205. isbn: 3540896732. doi: [10.1007/978-3-540-89674-6_22](https://doi.org/10.1007/978-3-540-89674-6_22).
- [García-Camino, 2009] García-Camino, Andrés (2009). "Normative regulation of open multi-agent systems". PhD thesis. Universitat Autònoma de Barcelona. http://www.iiia.csic.es/~andres/pubs/thesis_Garcia-Camino.pdf.
- [Gaver, 2012] Gaver, William (2012). "What Should We Expect From Research Through Design?" In: *CHI '12 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 937–946. issn: 145031015X. doi: [10.1145/2208516.2208538](https://doi.org/10.1145/2208516.2208538).
- [Genesereth and Ketchpel, 1994] Genesereth, Michael R. and Steven P. Ketchpel (July 1994). "Software Agents". In: *Communications of the ACM* 37.7, pp. 48–54. doi: [10.1145/176789.176794](https://doi.org/10.1145/176789.176794).
- [Ghassemi et al., 2023] Ghassemi, Marzyeh et al. (2023). "ChatGPT one year on: who is using it, how and why". In: *Nature* 624.December, pp. 39–41.
- [Godin and Zahedi, 2014] Godin, Danny and Mithra Zahedi (2014). "Aspects of Research through Design: A Literature Review". In: *DRS Biennial Conference Series. DRS2014 - Design's Big Debates*, pp. 1–14.
- [Goodwin and Cooper, 2009] Goodwin, Kim and Alan Cooper (2009). *Designing for the Digital Age: How to Create Human-Centered Products and Services*. 1st ed. Newark: John Wiley & Sons, Incorporated. isbn: 9780470229101.

- [Graf and Antoni, 2021] Graf, Benedikt and Conny H. Antoni (2021). "The relationship between information characteristics and information overload at the workplace - a meta-analysis". In: *European Journal of Work and Organizational Psychology* 30.1, pp. 143–158. issn: 14640643. doi: [10.1080/1359432X.2020.1813111](https://doi.org/10.1080/1359432X.2020.1813111).
- [Gray, Brown, and Macanufo, 2010] Gray, Dave, Sunni Brown, and James Macanufo (2010). *Gamestorming: A Playbook for Innovators, Rulebreakers, and Changemakers*. 1st. O'Reilly Media, Inc. isbn: 0596804172, 9780596804176.
- [Grudin, 2001] Grudin, Jonathan (2001). "Partitioning DigitalWorlds: Focal and Peripheral Awareness in Multiple Monitor Use". In: *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '01* 3, pp. 458–465. doi: [10.1145/365024.365312](https://doi.org/10.1145/365024.365312).
- [C. Gutwin and S. Greenberg, 2000] Gutwin, C. and S. Greenberg (2000). "The mechanics of collaboration: Developing low cost usability evaluation methods for shared workspaces". In: *Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE 2000*-Janua, pp. 98–103. issn: 15244547. doi: [10.1109/ENABL.2000.883711](https://doi.org/10.1109/ENABL.2000.883711).
- [Carl Gutwin and Saul Greenberg, 2002] Gutwin, Carl and Saul Greenberg (2002). "A descriptive framework of workspace awareness for real-time groupware". In: *Computer Supported Cooperative Work* 11.3-4, pp. 411–446. issn: 09259724. doi: [10.1023/A:1021271517844](https://doi.org/10.1023/A:1021271517844).
- [Ha and Schmidhuber, 2018] Ha, David and Jürgen Schmidhuber (2018). "World Models". In: doi: [10.5281/zenodo.1207631](https://doi.org/10.5281/zenodo.1207631). arXiv: [arXiv:1803.10122v4](https://arxiv.org/abs/1803.10122v4).
- [Han, 2020] Han, Kunni (2020). "Personalized News Recommendation and Simulation Based on Improved Collaborative Filtering Algorithm". In: *Complexity* 2020. issn: 10990526. doi: [10.1155/2020/8834908](https://doi.org/10.1155/2020/8834908).
- [Harper, 2016] Harper, Richard (2016). "From I-Awareness to We-Awareness in CSCW: a Review Essay". In: *Computer Supported Cooperative Work: CSCW: An International Journal* 25.4-5, pp. 295–301. issn: 15737551. doi: [10.1007/s10606-016-9247-8](https://doi.org/10.1007/s10606-016-9247-8).

[Hearst, 1999] Hearst, Marti A (1999). "Untangling Text Data Mining". In: *Proceedings of ACL'99: the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 1–13.

[Jim Hendler and Berners-Lee, 2010] Hendler, Jim and Tim Berners-Lee (2010). "From the Semantic Web to social machines: A research challenge for AI on the World Wide Web". In: *Artificial Intelligence* 174.2, pp. 156–161. issn: 00043702. doi: [10.1016/j.artint.2009.11.010](https://doi.org/10.1016/j.artint.2009.11.010).

[Himma and Tavani, 2009] Himma, Kenneth Einar and Herman T. Tavani (2009). *The Handbook of Information and Computer Ethics*, p. 532. isbn: 9780471799597. doi: [10.1002/9780470281819](https://doi.org/10.1002/9780470281819). arXiv: [arXiv:1011.1669v3](https://arxiv.org/abs/1011.1669v3).

[Intelligence, 2024] Intelligence, Mordor (2024). *Machine Learning as a Service Market Size & Share Analysis - Growth Trends & Forecasts (2024 - 2029)*. <https://www.mordorintelligence.com/industry-reports/global-machine-learning-as-a-service-mlaas-market> <https://perma.cc/V3CX-JRW7> (visited on 12/02/2024).

[Iqbal and Brian P. Bailey, 2008] Iqbal, Shamsi T. and Brian P. Bailey (2008). "Effects of Intelligent Notification Management on Users and Their Tasks". In: *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, p. 93. doi: [10.1145/1357054.1357070](https://doi.org/10.1145/1357054.1357070).

[Iqbal and Horvitz, 2010] Iqbal, Shamsi T. and Eric Horvitz (2010). "Notifications and Awareness: A Field Study of Alert Usage and Preferences". In: *Proceedings of the 2010 ACM conference on Computer supported cooperative work - CSCW '10*, pp. 27–30. doi: [10.1145/1718918.1718926](https://doi.org/10.1145/1718918.1718926).

[Izadi et al., 2002] Izadi, Shahram et al. (2002). "The FUSE platform: Supporting ubiquitous collaboration within diverse mobile environments". In: *Automated Software Engineering* 9.2, pp. 167–186. issn: 09288910. doi: [10.1023/A:1014534414062](https://doi.org/10.1023/A:1014534414062).

[N. R. Jennings, Sycara, and M. Wooldridge, 1998] Jennings, N. R., K. Sycara, and M. Wooldridge (1998). "A Roadmap of Agent Research and Development". In: *Autonomous*

- [Jin et al., 2024] Jin, Yiqiao et al. (2024). "MM-Soc: Benchmarking Multimodal Large Language Models in Social Media Platforms". In: pp. 6192–6210. issn: 0736587X. doi: 10.48550/arXiv.2402.14154. arXiv: 2402.14154v3.
- [Kaboré et al., 2013] Kaboré, Kiswendsida K. et al. (2013). "Information Access Assistant Service (IAAS)". In: *2013 8th International Conference for Internet Technology and Secured Transactions, ICITST 2013*, pp. 554–557. doi: 10.1109/ICITST.2013.6750263.
- [Kamdar, 2015] Kamdar, Sachin (2015). *3 Things About Walled Gardens That Drive Digital Publishers 'Up The Wall'*. <https://www.forbes.com/sites/sachinkamdar/2015/10/18/3-things-about-walled-gardens-that-drive-digital-publishers-up-the-wall/> <https://perma.cc/M34L-939V> (visited on 11/12/2024).
- [Karau and Williams, 1993] Karau, Steven J and Kipling D Williams (1993). "Social loafing: A meta-analytic review and theoretical integration." In: *Journal of Personality and Social Psychology* 65.4, pp. 681–706. issn: 1939-1315(Electronic),0022-3514(Print). doi: 10.1037/0022-3514.65.4.681.
- [Kliimask and Nikiforova, 2024] Kliimask, Kevin and Anastasija Nikiforova (2024). "TAGIFY : LLM-powered Tagging Interface for Improved Data Findability on OGD portals". In: *2024 Fifth International Conference on Intelligent Data Science Technologies and Applications (IDSTA) IEEE*. September. doi: 10.48550/arXiv.2407.18764.
- [Klimt and Yang, 2004] Klimt, Bryan and Yiming Yang (2004). "The Enron Corpus: A New Dataset for Email Classification Research". In: *Machine Learning: ECML 2004*, pp. 217–226. issn: 03029743. doi: 10.1007/978-3-540-30115-8_22.
- [Kluge, Antoni, and Ellwart, 2020] Kluge, Annette, Conny H. Antoni, and Thomas Ellwart (2020). "Digitalization as the Problem of and the Solution to Vast Amounts of Data in Future Work-Challenges for Individuals, Teams, and Organizations". In: *Zeitschrift für Arbeits- und Organisationspsychologie* 64.1, pp. 1–5. issn: 21906270. doi: 10.1026/0932-4089/a000317.

- [Kolbjørnsrud, 2024] Kolbjørnsrud, Vegard (2024). "Designing the Intelligent Organization: Six Principles for Human-AI Collaboration". In: *California Management Review* 66.2, pp. 44–64. issn: 21628564. doi: [10.1177/00081256231211020](https://doi.org/10.1177/00081256231211020).
- [Kolp, Giorgini, and Mylopoulos, 2001] Kolp, Manuel, Paolo Giorgini, and John Mylopoulos (2001). "A Goal-Based Organizational Perspective on Multi-Agent Architectures". In: *Eighth International Workshop on Agent Theories, architectures, and languages (ATAL-2001)*, p. 13.
- [Krafft, Macy, and Pentland, 2017] Krafft, Peter M, Michael Macy, and Alex Pentland (2017). "Bots as Virtual Confederates: Design and Ethics". In: *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. New York, NY, USA: Association for Computing Machinery, pp. 183–190. isbn: 9781450343350. doi: [10.1145/2998181.2998354](https://doi.org/10.1145/2998181.2998354). arXiv: [1611.00447](https://arxiv.org/abs/1611.00447).
- [Krugmann and Hartmann, 2024] Krugmann, Jan Ole and Jochen Hartmann (2024). "Sentiment Analysis in the Age of Generative AI". In: *Customer Needs and Solutions* 11.1. issn: 2196-291X. doi: [10.1007/s40547-024-00143-4](https://doi.org/10.1007/s40547-024-00143-4).
- [La Cava, Greco, and Tagarelli, 2021] La Cava, Lucio, Sergio Greco, and Andrea Tagarelli (2021). "Understanding the growth of the Fediverse through the lens of Mastodon". In: *Applied Network Science* 6.1. issn: 23648228. doi: [10.1007/s41109-021-00392-5](https://doi.org/10.1007/s41109-021-00392-5). arXiv: [2106.15473](https://arxiv.org/abs/2106.15473).
- [Landale, 2007] Landale, Anthony (2007). "Hunter-seeker strategies: The antidote to overload". In: *Industrial and Commercial Training* 39.4, pp. 227–230. issn: 00197858. doi: [10.1108/00197850710755177](https://doi.org/10.1108/00197850710755177).
- [Langley, 2012] Langley, Patrick W (2012). "The Cognitive Systems Paradigm". In: *Advances in Cognitive Systems* 1, pp. 3–13.
- [J. H. M. Lee and Zhao, 2002] Lee, Jimmy H M and Lei Zhao (2002). "A Real-Time Agent Architecture : Design, Implementation and Evaluation". In: *Proceedings of the 5th Pacific Rim International Workshop on Multi Agents: Intelligent Agents and Multi-Agent Sys-*

- tems. Ed. by Kazuhiro Kuwabara and Jaeho Lee. Tokyo: Springer, Berlin, Heidelberg, pp. 18–32.
- [J.-s. Lee and Tatar, 2014] Lee, Joon-suk and Deborah Tatar (2014). “Sounds of Silence: Exploring Contributions to Conversations, Non-Responses and the Impact of Mediating Technologies in Triple Space”. In: *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing - CSCW ’14*, pp. 1561–1572. doi: [10.1145/2531602.2531655](https://doi.org/10.1145/2531602.2531655).
- [Leer, Trost, and Voruganti, 2023] Leer, Courtland, Vincent Trost, and Vineeth Voruganti (2023). “Violation of Expectation via Metacognitive Prompting Reduces Theory of Mind Prediction Error in Large Language Models”. In: arXiv: 2310.06983.
- [Leslie, 2023] Leslie, David (2023). “Does the sun rise for ChatGPT? Scientific discovery in the age of generative AI”. In: *AI and Ethics* 0123456789. issn: 2730-5953. doi: [10.1007/s43681-023-00315-3](https://doi.org/10.1007/s43681-023-00315-3).
- [H. Li et al., 2023] Li, Huao et al. (2023). “Theory of Mind for Multi-Agent Collaboration via Large Language Models”. In: *EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings*, pp. 180–192. isbn: 9798891760608. doi: [10.18653/v1/2023.emnlp-main.13](https://doi.org/10.18653/v1/2023.emnlp-main.13). arXiv: 2310.10701v3.
- [Q. Li et al., 2022] Li, Qian et al. (2022). “A Survey on Text Classification: From Traditional to Deep Learning”. In: *ACM Transactions on Intelligent Systems and Technology* 13.2. issn: 21576912. doi: [10.1145/3495162](https://doi.org/10.1145/3495162).
- [Licklider, 1960] Licklider, J. C.R. (1960). “Man-Computer Symbiosis”. In: *IRE Transactions on Human Factors in Electronics* HFE-1.1, pp. 4–11. issn: 21682836. doi: [10.1109/THFE2.1960.4503259](https://doi.org/10.1109/THFE2.1960.4503259).
- [Liebrecht, Kunneman, and Bosch, 2013] Liebrecht, Christine, Florian Kunneman, and Antal van den Bosch (2013). “The perfect solution for detecting sarcasm in tweets # not”. In: *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis* June, pp. 29–37.

- [Lieto and Radicioni, 2016] Lieto, Antonio and Daniele P. Radicioni (2016). "From human to artificial cognition and back: New perspectives on cognitively inspired AI systems". In: *Cognitive Systems Research* 39, pp. 1–3. doi: [10.1016/j.cogsys.2016.02.002](https://doi.org/10.1016/j.cogsys.2016.02.002).
- [Lins et al., 2021] Lins, Sebastian et al. (2021). "Artificial Intelligence as a Service: Classification and Research Directions". In: *Business and Information Systems Engineering* 63.4, pp. 441–456. issn: 18670202. doi: [10.1007/s12599-021-00708-w](https://doi.org/10.1007/s12599-021-00708-w).
- [Liu, 2012] Liu, Bing (2012). *Sentiment Analysis and Opinion Mining*. Vol. 5. 1. Morgan & Claypool, pp. 1–167. isbn: 9781608458844. doi: [10.2200/S00416ED1V01Y201204HLT016](https://doi.org/10.2200/S00416ED1V01Y201204HLT016). arXiv: 1003.5699.
- [Luff, Heath, and Svensson, 2008] Luff, Paul, Christian Heath, and Marcus Sanchez Svensson (2008). "Discriminating conduct: Deploying systems to support awareness in organizations". In: *International Journal of Human-Computer Interaction* 24.4, pp. 410–436. issn: 10447318. doi: [10.1080/10447310801920490](https://doi.org/10.1080/10447310801920490).
- [Lund et al., 2023] Lund, Brady D et al. (Mar. 2023). "ChatGPT and a New Academic Reality: Artificial Intelligence-Written Research Papers and the Ethics of the Large Language Models in Scholarly Publishing". In: *Journal of the Association for Information Science and Technology* 74.5, pp. 570–581. issn: 2330-1643. doi: [10.1002/asi.24750](https://doi.org/10.1002/asi.24750).
- [Maes, 1994] Maes, Pattie (1994). "Agents that reduce work and information overload". In: *Communications of the ACM* 37.7, pp. 30–40. issn: 00010782. doi: [10.1145/176789.176792](https://doi.org/10.1145/176789.176792).
- [Manias et al., 2023] Manias, George et al. (2023). "Multilingual text categorization and sentiment analysis: a comparative analysis of the utilization of multilingual approaches for classifying twitter data". In: *Neural Computing and Applications* 35.29, pp. 21415–21431. issn: 14333058. doi: [10.1007/s00521-023-08629-3](https://doi.org/10.1007/s00521-023-08629-3).
- [Martin et al., 2011] Martin, Francisco J. et al. (2011). "The Big Promise of Recommender Systems". In: *AI Magazine* 32.3, pp. 19–27. doi: [10.1609/aimag.v32i3.2360](https://doi.org/10.1609/aimag.v32i3.2360).
- [Mathieu, Routier, and Secq, 2002] Mathieu, P, JC Routier, and Y Secq (2002). "Principles for dynamic multi-agent organizations". In: *Proceedings of the 5th Pacific Rim Interna-*

- tional Workshop on Multi Agents: Intelligent Agents and Multi-Agent Systems. Ed. by Kazuhiro Kuwabara and Jaeho Lee. Tokyo: Springer, Berlin, Heidelberg, pp. 109–122. isbn: 978-3-540-45680-3. doi: [10.1007/978-3-540-45680-3_8](https://doi.org/10.1007/978-3-540-45680-3_8).
- [Maynard and Greenwood, 2014] Maynard, Diana and Mark A. Greenwood (2014). "Who Cares About Sarcastic Tweets? Investigating the Impact of Sarcasm on Sentiment Analysis". In: *LREC 2014, Ninth International Conference on Language Resources and Evaluation*. Reykjavik: LREC, pp. 4238–4243. isbn: 978-2-9517408-8-4. <https://eprints.whterose.ac.uk/130763/>.
- [McCarthy, 1979] McCarthy, J (1979). "Ascribing Mental Qualities to Machines". In: *Philosophical Perspectives in Artificial Intelligence*, pp. 161–195. doi: [10.2307/2025382](https://doi.org/10.2307/2025382).
- [McGinn and Kotamraju, 2008] McGinn, Jennifer (Jen) and Nalini Kotamraju (2008). "Data-driven persona development". In: *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, p. 1521. issn: 978-1-60558-011-1. doi: [10.1145/1357054.1357292](https://doi.org/10.1145/1357054.1357292).
- [McGuinness and Harmelen, 2004] McGuinness, Deborah L. and Frank van Harmelen (2004). "OWL Web Ontology Language Overview". In: *W3C recommendation 10.2004-03* February, pp. 1–12. issn: 15302180. doi: [10.1145/1295289.1295290](https://doi.org/10.1145/1295289.1295290). arXiv: [arXiv:1011.1669v3](https://arxiv.org/abs/1011.1669v3).
- [Metaxas and Markopoulos, 2008] Metaxas, Georgios and Panos Markopoulos (2008). "'Aware of what?' A formal model of awareness systems that extends the focus-nimbus model". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4940 LNCS, pp. 429–446. issn: 03029743. doi: [10.1007/978-3-540-92698-6_26](https://doi.org/10.1007/978-3-540-92698-6_26).
- [Metzger, Schillo, and K. Fischer, 2003] Metzger, Jörg, Michael Schillo, and Klaus Fischer (2003). "A Multiagent-Based Peer-to-Peer Network in Java for Distributed Spam Filtering". In: *3rd International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2003 Prague, Czech Republic, June 16–18, 2003*. Ed. by

Vladimír Marík, Michal Pechoucek, and Jörg Müller. Prague: Springer, Berlin, Heidelberg, pp. 616–625.

[Mianowska and Nguyen, 2010] Mianowska, Bernadetta and Ngoc Thanh Nguyen (2010).

“A framework of an agent-based personal assistant for internet users”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6070 LNAI.PART 1, pp. 163–172. issn: 03029743. doi: [10.1007/978-3-642-13480-7_18](https://doi.org/10.1007/978-3-642-13480-7_18).

[Milewski and T. M. Smith, 2000] Milewski, Alan E and Thomas M Smith (2000). “Providing Presence Cues to Telephone Users”. In: *Proceedings of the ACM Conference on Computer Supported Cooperative Work* 07701.732, pp. 89–96. doi: [10.1145/358916.358978](https://doi.org/10.1145/358916.358978).

[Minaee et al., 2021] Minaee, Shervin et al. (2021). “Deep Learning-Based Text Classification”. In: *ACM Computing Surveys* 54.3. issn: 15577341. doi: [10.1145/3439726](https://doi.org/10.1145/3439726).

[Mittelstadt et al., 2016] Mittelstadt, Brent Daniel et al. (2016). “The ethics of algorithms: Mapping the debate”. In: *Big Data & Society* 3.2, p. 2053951716679679. doi: [10.1177/2053951716679679](https://doi.org/10.1177/2053951716679679).

[Modha et al., 2011] Modha, Dharmendra S. et al. (2011). “Cognitive computing”. In: *Communications of the ACM* 54.8, p. 62. issn: 00010782. doi: [10.1145/1978542.1978559](https://doi.org/10.1145/1978542.1978559). arXiv: [arXiv:1011.1669v3](https://arxiv.org/abs/1011.1669v3).

[Monk, Boehm-Davis, and Trafton, 2002] Monk, Christopher A., Deborah A. Boehm-Davis, and J. Gregory Trafton (2002). “The Attentional Costs of Interrupting Task Performance at Various Stages”. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 46.22, pp. 1824–1828. issn: 1071-1813. doi: [10.1177/154193120204602210](https://doi.org/10.1177/154193120204602210).

[Montaner, López, and De La Rosa, 2003] Montaner, Miquel, Beatriz López, and Josep Lluís De La Rosa (2003). “A taxonomy of recommender agents on the internet”. In: *Artificial Intelligence Review* 19.4, pp. 285–330. issn: 02692821. doi: [10.1023/A:1022850703159](https://doi.org/10.1023/A:1022850703159).

[Mousavi, 2011] Mousavi, Cathy (2011). *What is Web Services according to W3C?, "Big Web" Services Vs. Restful Services.* <https://catmousavi.wordpress.com/2011/11/10/what-is-web-services-according-to-w3c-big-web-services-vs-restful-services/>

is-web-services-according-to-w3c-big-web-services-vs-restful-services/ <https://perma.cc/FB9L-B6PL> (visited on 12/22/2017).

[Mukherjee and Bala, 2017] Mukherjee, Shubhadeep and Pradip Kumar Bala (2017). "Detecting sarcasm in customer tweets: an NLP based approach". In: *Industrial Management & Data Systems* 117.6, pp. 1109–1126. issn: 0263-5577. doi: [10.1108/IMDS-06-2016-0207](https://doi.org/10.1108/IMDS-06-2016-0207). arXiv: [arXiv:1011.1669v3](https://arxiv.org/abs/1011.1669v3).

[Neisser, 1976] Neisser, Ulric (1976). *Cognition and Reality: Principles and Implications of Cognitive Psychology*. W.H. Freeman.

[Pablo Noriega and Mark D'Inverno, 2014] Noriega, Pablo and Mark D'Inverno (2014). "Crowd-Based Socio-Cognitive Systems". In: *Crowd Intelligence: Foundations, Methods and Practices. European Network for Social Intelligence*. Barcelona. <http://research.gold.ac.uk/10370/>.

[Norri-Sederholm et al., 2015] Norri-Sederholm, Teija et al. (2015). "Situational awareness and information flow in prehospital emergency medical care from the perspective of paramedic field supervisors: A scenario-based study". In: *Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine* 23.1, pp. 1–9. issn: 17577241. doi: [10.1186/s13049-014-0083-x](https://doi.org/10.1186/s13049-014-0083-x).

[Oeldorf-Hirsch and Neubaum, 2023] Oeldorf-Hirsch, Anne and German Neubaum (2023). "Attitudinal and behavioral correlates of algorithmic awareness among German and U.S. social media users". In: *Journal of Computer-Mediated Communication* 28.5. issn: 10836101. doi: [10.1093/jcmc/zmad035](https://doi.org/10.1093/jcmc/zmad035).

[Oh and Look, 2003] Oh, Alice and Gary Look (2003). "Awareness Agents : A Distributed System for Group Awareness". In: pp. 3–6. <http://people.csail.mit.edu/aoh/papers/sow2004.pdf>.

[Okolloh, 2009] Okolloh, Ory (2009). "Ushahidi, or 'testimony': Web 2.0 tools for crowdsourcing crisis information". In: *Participatory Learning and Action* 59.January, pp. 65–70. issn: 1357-938X.

- [Okoshi, Tsubouchi, and Tokuda, 2019] Okoshi, Tadashi, Kota Tsubouchi, and Hideyuki Tokuda (2019). “Real-world Product Deployment of Adaptive Push Notification Scheduling on Smartphones”. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* July, pp. 2792–2800. doi: [10.1145/3292500.3330732](https://doi.org/10.1145/3292500.3330732).
- [Ouyang et al., 2022] Ouyang, Long et al. (2022). “Training language models to follow instructions with human feedback”. In: *Advances in Neural Information Processing Systems*. Vol. 35, pp. 27730–27744. doi: [10.48550/arXiv.2203.02155](https://doi.org/10.48550/arXiv.2203.02155). arXiv: [2203.02155](https://arxiv.org/abs/2203.02155).
- [Pan et al., 2007] Pan, Jianguo et al. (2007). “Ontology based user profiling in personalized information service agent”. In: *Computer and Information Technology, 2007. CIT 2007. 7th IEEE International Conference on* 90612010, pp. 1089–1093. doi: [10.1109/CIT.2007.63](https://doi.org/10.1109/CIT.2007.63).
- [Papadopoulos, 2006] Papadopoulos, Constantinos (2006). “Improving awareness in mobile CSCW”. In: *IEEE Transactions on Mobile Computing* 5.10, pp. 1331–1346. issn: 15361233. doi: [10.1109/TMC.2006.152](https://doi.org/10.1109/TMC.2006.152).
- [Pearson, 1895] Pearson, Karl (1895). “VII. Note on regression and inheritance in the case of two parents”. In: *Proceedings of the Royal Society of London, Volume 58, Issue 347-352*, pp. 240–242. doi: [10.1098/rspl.1895.0041](https://doi.org/10.1098/rspl.1895.0041).
- [Pedregosa et al., 2011] Pedregosa, F et al. (2011). “Scikit-learn: Machine Learning in {P}ython”. In: *Journal of Machine Learning Research* 12, pp. 2825–2830.
- [Peinl, 2016] Peinl, René (2016). “Semantic Web: State of the Art and Adoption in Corporations”. In: *KI - Künstliche Intelligenz* 30.2, pp. 131–138. issn: 0933-1875. doi: [10.1007/s13218-016-0425-0](https://doi.org/10.1007/s13218-016-0425-0). <http://link.springer.com/10.1007/s13218-016-0425-0>.
- [Peisert, 2020] Peisert, Sean (2020). “An Examination and Survey of Data Confidentiality Issues and Solutions in Academic Research Computing”. In: *Trusted CI Report, initial release September 2020, revised November 2020*. <https://escholarship.org/uc/item/7cz7m1ws>.
- [Pejovic and Musolesi, 2014] Pejovic, Veljko and Mirco Musolesi (2014). “InterruptMe: Designing Intelligent Prompting Mechanisms for Pervasive Applications”. In: *Proceedings*

- of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14)*, pp. 897–908. doi: [10.1145/2632048.2632062](https://doi.org/10.1145/2632048.2632062).
- [Peña et al., 2023] Peña, Alejandro et al. (2023). “Leveraging Large Language Models for Topic Classification in the Domain of Public Affairs”. In: *Document Analysis and Recognition – ICDAR 2023 Workshops*. Ed. by Mickael Coustaty and Alicia Fornés. Cham: Springer Nature Switzerland, pp. 20–33. isbn: 978-3-031-41498-5.
- [Petrescu and Krishen, 2020] Petrescu, Maria and Anjala S. Krishen (2020). “The dilemma of social media algorithms and analytics”. In: *Journal of Marketing Analytics* 8.4, pp. 187–188. issn: 20503318. doi: [10.1057/s41270-020-00094-4](https://doi.org/10.1057/s41270-020-00094-4).
- [Petrie, 1997] Petrie, Charles (1997). “What Is An Agent?” In: *Intelligent Agents III Agent Theories, Architectures, and Languages*. Springer Berlin Heidelberg, pp. 41–43. doi: [10.1007/BFb0013572](https://doi.org/10.1007/BFb0013572).
- [Poslad, 2007] Poslad, Stefan (2007). “Specifying Protocols for Multi-Agent Systems Interaction”. In: *ACM Transactions on Autonomous and Adaptive Systems* 2.4, 15–es. issn: 15564665. doi: [10.1145/1293731.1293735](https://doi.org/10.1145/1293731.1293735).
- [Prabowo and Thelwall, 2009] Prabowo, Rudy and Mike Thelwall (2009). “Sentiment analysis: A combined approach”. In: *Journal of Informetrics* 3.2, pp. 143–157. issn: 17511577. doi: [10.1016/j.joi.2009.01.003](https://doi.org/10.1016/j.joi.2009.01.003).
- [Premack and Woodruff, 1978] Premack, David and Guy Woodruff (1978). “Does the chimpanzee have a theory of mind?” In: *Behavioral and Brain Sciences*. 4, pp. 515–526. issn: 0140-525X.
- [Qasem, 2023] Qasem, Fawaz (2023). “ChatGPT in scientific and academic research: future fears and reassurances”. In: *Library Hi Tech News* 40.3, pp. 30–32. issn: 07419058. doi: [10.1108/LHTN-03-2023-0043](https://doi.org/10.1108/LHTN-03-2023-0043).
- [Rainie and Anderson, 2017] Rainie, Lee and Janna Anderson (2017). *Code-Dependent: Pros and Cons of the Algorithm Age*. Tech. rep. February, pp. 1–87.

- [Resnick and Varian, 1997] Resnick, Paul and Hal R. Varian (1997). "Recommender Systems". In: *Communications of the ACM* 40.3, pp. 56–58. issn: 0001-0782. doi: [10.1145/245121](https://doi.org/10.1145/245121). arXiv: [1202.1112v1](https://arxiv.org/abs/1202.1112v1).
- [Riccardi and Desai, 2023] Riccardi, Nicholas and Rutvik H. Desai (2023). "The Two Word Test: A Semantic Benchmark for Large Language Models University of South Carolina Department of Psychology". In: doi: [10.48550/arxiv.2306.04610](https://doi.org/10.48550/arxiv.2306.04610). arXiv: [2306.04610](https://arxiv.org/abs/2306.04610).
- [Richards and Wessel, 2024] Richards, Jonan and Mairieli Wessel (2024). "What You Need is What You Get: Theory of Mind for an LLM-Based Code Understanding Assistant". In: *International Conference on Software Maintenance and Evolution (ICSME), 2024*. doi: [10.48550/arxiv.2408.04477](https://doi.org/10.48550/arxiv.2408.04477). arXiv: [2408.04477](https://arxiv.org/abs/2408.04477).
- [Rodden, 1996] Rodden, Tom (1996). "Populating the Application: A Model of Awareness for Cooperative Applications". In: *Proceedings of the 1996 ACM conference on Computer supported cooperative work (CSCW '96)* 9, pp. 87–96. doi: [10.1145/240080.240200](https://doi.org/10.1145/240080.240200).
- [Rodriguez, Gummadi, and Schoelkopf, 2014] Rodriguez, Manuel Gomez, Krishna P Gummadi, and Bernhard Schoelkopf (2014). "Quantifying Information Overload in Social Media and its Impact on Social Contagions". In: *Eighth International AAAI Conference on Weblogs and Social Media*. arXiv: [1403.6838](https://arxiv.org/abs/1403.6838).
- [Sappelli et al., 2016] Sappelli, M. et al. (2016). "Assessing e-mail intent and tasks in e-mail messages". In: *Information Sciences* 358-359, pp. 1–17. issn: 00200255. doi: [10.1016/j.ins.2016.03.002](https://doi.org/10.1016/j.ins.2016.03.002).
- [K. Schmidt, 2002] Schmidt, Kjeld (2002). "The Problem with 'Awareness'". In: *Computer Supported Cooperative Work* 11.6249, pp. 285–298. doi: [10.1023/a:1021272909573](https://doi.org/10.1023/a:1021272909573).
- [S. Schmidt et al., 2007] Schmidt, Stefan et al. (2007). "Fuzzy trust evaluation and credibility development in multi-agent systems". In: *Applied Soft Computing Journal* 7.2, pp. 492–505. issn: 15684946. doi: [10.1016/j.asoc.2006.11.002](https://doi.org/10.1016/j.asoc.2006.11.002).
- [Schon, 2008] Schon, Donald A (2008). *The reflective practitioner: How professionals think in action*. Basic books.

- [Seewald, 2004] Seewald, Alexander K (2004). *Combining Bayesian and Rule Score Learning: Automated Tuning for SpamAssassin*. Tech. rep.
- [Segaran, 2007] Segaran, Toby (2007). *Programming Collective Intelligence: Building Smart Web 2.0 Applications*. O'Reilly Media, Inc. isbn: 9780596550684. <https://www.oreilly.com/library/view/programming-collective-intelligence/9780596529321/>.
- [Severance, 2015] Severance, Charles (2015). "Roy T. Fielding: Understanding the REST Style." In: *Computer* 48.6, pp. 7–9. doi: [10.1109/MC.2015.170](https://doi.org/10.1109/MC.2015.170).
- [N. Shadbolt, 2013] Shadbolt, Nigel (2013). "Knowledge Acquisition and the Rise of Social Machines". In: *International Journal of Human Computer Studies* 71.2, pp. 200–205. issn: 10715819. doi: [10.1016/j.ijhcs.2012.10.008](https://doi.org/10.1016/j.ijhcs.2012.10.008).
- [Shang, T. Wang, and Lv, 2011] Shang, Wenqian, Tong Wang, and Rui Lv (2011). "The Key Technology Research of Intelligent Information Syndication". In: *Computational Sciences and Optimization, International Joint Conference on* 0.2, pp. 865–867. doi: [10.1109/CSO.2011.275](https://doi.org/10.1109/CSO.2011.275).
- [Shen et al., 2006] Shen, Jianqiang et al. (2006). "A Hybrid Learning System for Recognizing User Tasks from Desktop Activities and Email Messages". In: *IUI '06 Proceedings of the 11th international conference on Intelligent user interfaces*. Vol. Sydney, pp. 86–92. isbn: 1595932879. doi: [10.1145/1111449.1111473](https://doi.org/10.1145/1111449.1111473).
- [Shoham, 1993] Shoham, Yoav (1993). "Agent oriented programming: An overview of the framework and summary of recent research". In: *NASA. Lyndon B. Johnson Space Center, The Sixth Annual Workshop on Space Operations Applications and Research (SOAR 1992)*, pp. 296–304.
- [Siemon, 2022] Siemon, Dominik (2022). *Elaborating Team Roles for Artificial Intelligence-based Teammates in Human-AI Collaboration*. Vol. 31. 5. Springer Netherlands, pp. 871–912. doi: [10.1007/s10726-022-09792-z](https://doi.org/10.1007/s10726-022-09792-z).
- [Simmhan, Plale, and Gannon, 2005] Simmhan, Yogesh L., Beth Plale, and Dennis Gannon (2005). *A Survey of Data Provenance in e-Science*. doi: [10.1145/1084805.1084812](https://doi.org/10.1145/1084805.1084812).

- [Sinha, 2016] Sinha, Tanmay (IBM) (2016). *IBM Watson Knowledge Studio – Teach Watson about your domain*. <https://www.ibm.com/blogs/watson/2016/06/alchemy-knowledge-studio/> (visited on 12/01/2017).
- [Smart and N. R. Shadbolt, 2014] Smart, Paul R. and Nigel R. Shadbolt (2014). “Social Machines”. In: *Encyclopedia of Information Science and Technology*. Ed. by Mehdi Khosrow-Pour. Pennsylvania: IGI Global, pp. 6855–6862. doi: 10.4018/978-1-4666-5888-2.ch67
5. <https://eprints.soton.ac.uk/361399/>.
- [R. Smith, 2010] Smith, Richard (2010). “Strategies for Coping With Information Overload”. In: *British Medical Journal* 341.dec15 2, pp. c7126–c7126. issn: 0959-8138. doi: 10.1136/bmj.c7126.
- [Sorower, Slater, and Dietterich, 2015] Sorower, Mohammad S., Michael Slater, and Thomas G. Dietterich (2015). “Improving Automated Email Tagging with Implicit Feedback”. In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15*, pp. 201–211. doi: 10.1145/2807442.2807501.
- [Steinfeld et al., 2007] Steinfeld, Aaron et al. (2007). “Evaluation of an integrated multi-task machine learning system with humans in the loop”. In: *Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems - PerMIS '07*, pp. 168–174. doi: 10.1145/1660877.1660901.
- [Strachan et al., 2024] Strachan, James W.A. et al. (2024). “Testing theory of mind in large language models and humans”. In: *Nature Human Behaviour* 8.7, pp. 1285–1295. issn: 23973374. doi: 10.1038/s41562-024-01882-z.
- [Street, 2024] Street, Winnie (2024). “LLM Theory of Mind and Alignment: Opportunities and Risks”. In: *Proceedings of Workshop on Theory of Mind in Human-AI Interaction at CHI 2024 (ToMinHAI at CHI 2024)*. doi: 10.48550/arXiv.2405.08154. arXiv: 2405.08154.
- [Street et al., 2024] Street, Winnie et al. (2024). “LLMs achieve adult human performance on higher-order theory of mind tasks”. In: pp. 1–18. doi: 10.48550/arXiv.2405.18870. arXiv: 2405.18870.

- [Stumpf et al., 2009] Stumpf, Simone et al. (2009). "Interacting Meaningfully with Machine Learning Systems: Three Experiments". In: *International Journal of Human-Computer Studies* 67, pp. 639–662. doi: [10.1016/j.ijhcs.2009.03.004](https://doi.org/10.1016/j.ijhcs.2009.03.004).
- [Sun et al., 2023] Sun, Xiaofei et al. (2023). "Text Classification via Large Language Models". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 8990–9005. doi: [10.18653/v1/2023.findings-emnlp.603](https://doi.org/10.18653/v1/2023.findings-emnlp.603). arXiv: [2305.08377](https://arxiv.org/abs/2305.08377).
- [Talia, 2011] Talia, Domenico (2011). "Cloud computing and software agents: Towards cloud intelligent services". In: *CEUR Workshop Proceedings* 741, pp. 2–6. issn: 16130073.
- [Tao et al., 2023] Tao, Zhengwei et al. (2023). "EvEval: A Comprehensive Evaluation of Event Semantics for Large Language Models". In: arXiv: [2305.15268](https://arxiv.org/abs/2305.15268).
- [Tenenberg, Roth, and Socha, 2016] Tenenberg, Josh, Wolff-Michael Roth, and David Socha (2016). "From I-Awareness to We-Awareness in CSCW". In: *Computer Supported Cooperative Work: CSCW: An International Journal* 25.4-5, pp. 235–278. doi: [10.1007/s10606-014-9215-0](https://doi.org/10.1007/s10606-014-9215-0).
- [Thejaswee et al., 2020] Thejaswee, Manda et al. (2020). "Performance Analysis of Machine Learning Algorithms for Text Classification". In: *Advanced Informatics for Computing Research*. Vol. 1393. Springer Singapore, pp. 414–424. isbn: 978-981-16-3660-8. doi: [10.1007/978-981-16-3660-8_39](https://doi.org/10.1007/978-981-16-3660-8_39).
- [Toffler, 1970] Toffler, Alvin (1970). *Future Shock*. Random House. isbn: 0-394-42586-3.
- [Tomasello, 2014] Tomasello, Michael (2014). *A Natural History of Human Thinking*. Harvard University Press.
- [Tran and Hoang, 2008] Tran, Dinh Que and Tuan Nha Hoang (2008). "Agent reasoning with semantic web in web blogs". In: *Proceedings of the 11th Pacific Rim International Workshop on Multi Agents: Intelligent Agents and Multi-Agent Systems*. Ed. by The Duy Bui, Tuong Vinh Ho, and Quang Thuy Ha. Vol. 5357 LNAI. Hanoi: Springer, Berlin, Heidelberg, pp. 389–396. isbn: 3540896732. doi: [10.1007/978-3-540-89674-6_43](https://doi.org/10.1007/978-3-540-89674-6_43).

- [Tsamados et al., 2022] Tsamados, Andreas et al. (2022). "The ethics of algorithms: key problems and solutions". In: *AI and Society* 37.1, pp. 215–230. issn: 14355655. doi: 10.1007/s00146-021-01154-8.
- [Tu et al., 2010] Tu, Nan Tu Nan et al. (2010). "Using cluster analysis in Persona development". In: *Supply Chain Management and Information Systems SCMIS 2010 8th International Conference on*, pp. 1–5.
- [Tuohy, 2024] Tuohy, Jennifer Pattison (2024). *The much-needed reinvention of the voice assistant is almost here.* <https://www.theverge.com/2024/6/14/24177991/apple-intelligence-siri-voice-assistant-amazon-alexa-generative-ai> <https://perma.cc/7EEP-FG2H> (visited on 11/12/2024).
- [Vafa et al., 2024] Vafa, Keyon et al. (2024). "Evaluating the World Model Implicit in a Generative Model". In: arXiv: 2406.03689.
- [Vaithyanathan, 2016] Vaithyanathan, Shivakumar (2016). "Building Industry-specific Knowledge Bases". In: *CIKM '16 Proceedings of the 25th ACM International Conference on Information and Knowledge Management*. Indianapolis: ACM, pp. 205–206. doi: 10.1145/2983323.2983369.
- [Van Noorden and Perkel, 2023] Van Noorden, Richard and Jeffrey M. Perkel (Sept. 2023). "AI and science: what 1,600 researchers think". In: *Nature* 621.7980, pp. 672–675. issn: 14764687. doi: 10.1038/D41586-023-02980-0.
- [Vilaplana, 2015] Vilaplana, Jaume Nualart (2015). "Visualization and exploration of texts". PhD thesis. University of Barcelona.
- [W3C, 2017] W3C (2017). *Social Web Protocols.* <https://www.w3.org/TR/2017/NOTE-social-web-protocols-20171225/> (visited on 12/30/2017).
- [D. Wang et al., 2020] Wang, Dakuo et al. (2020). "From human-human collaboration to Human-AI collaboration: Designing AI systems that can work together with people". In: *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1–6. doi: 10.1145/3334480.3381069.

- [Z. Wang et al., 2024] Wang, Zhiqiang et al. (2024). "Adaptable and Reliable Text Classification using Large Language Models". In: doi: 10.48550/arXiv.2405.10523. arXiv: 2405.10523v2.
- [Weitzner et al., 2008] Weitzner, Daniel J. et al. (2008). "Information accountability". In: *Communications of the ACM* 51.6, pp. 82–87. issn: 00010782. doi: 10.1145/1349026.1349043.
- [Witten et al., 2017] Witten, Ian H et al. (2017). *Data Mining: Practical Machine Learning Tools and Techniques*, p. 664. isbn: 978-0-12-804291-5. doi: 10.1016/C2015-0-02071-8.
- [Michael Wooldridge, 1996] Wooldridge, Michael (1996). "Agents as a Rorschach test: A response to Franklin and Graesser". In: *Intelligent Agents III Agent Theories, Architectures, and Languages*, pp. 47–48. issn: 16113349. <https://link.springer.com/chapter/10.1007/BFb0013574>.
- [Michael Wooldridge, 2002] — (2002). *An Introduction to Multi-Agent Systems*. John Wiley & Sons. isbn: ISBN 0-471-49691-X.
- [Michael Wooldridge and Nicholas R. Jennings, 1995] Wooldridge, Michael and Nicholas R. Jennings (1995). "Intelligent agents: theory and practice". In: *Knowledge Engineering Review* 10.2, pp. 115–152.
- [Wu et al., 2016] Wu, Lei et al. (2016). "Influence of information overload on operator's user experience of human-machine interface in LED manufacturing systems". In: *Cognition, Technology and Work* 18.1, pp. 161–173. issn: 14355566. doi: 10.1007/s10111-015-0352-0.
- [Xia and Sudharshan, 2002] Xia, Lan and D. Sudharshan (2002). "Effects of interruptions on consumer online decision processes". In: *Journal of Consumer Psychology* 12.3, pp. 265–280. issn: 10577408. doi: 10.1207/153276602760335103.
- [Xu, 2019] Xu, Wei (2019). "Toward human-centered AI: A perspective from human-computer interaction". In: *Interactions* 26.4, pp. 42–46. issn: 15583449. doi: 10.1145/3328485.
- [Ye et al., 2001] Ye, Yiming et al. (2001). "Agents-supported adaptive group awareness: Smart distance and WWWaware". In: *IEEE Transactions on Systems, Man, and Cyber-*

- netics Part A: Systems and Humans.* 31.5, pp. 369–380. issn: 10834427. doi: 10.1109/3468.952712.
- [Yee-King, D’Inverno, and Noriega, 2014] Yee-King, M, M D’Inverno, and P Noriega (2014). “Social machines for education driven by feedback agents”. In: <https://digital.csic.es/bitstream/10261/132144/1/Ws.MFSC2014.pdf>.
- [Yenduri et al., 2024] Yenduri, Gokul et al. (2024). “GPT (Generative Pre-Trained Transformer) - A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions”. In: *IEEE Access* 12, pp. 54608–54649. issn: 21693536. doi: 10.1109/ACCESS.2024.3389497. arXiv: 2305.10435.
- [S. Zhang et al., 2024] Zhang, Shao et al. (2024). “Mutual Theory of Mind in Human-AI Collaboration: An Empirical Study with LLM-driven AI Agents in a Real-time Shared Workspace Task”. In: *Preprint*. doi: 10.48550/arXiv.2409.08811. arXiv: 2409.08811.
- [W. Zhang et al., 2024] Zhang, Wenxuan et al. (2024). “Sentiment Analysis in the Era of Large Language Models: A Reality Check”. In: *Findings of the Association for Computational Linguistics: NAACL 2024 - Findings*, pp. 3881–3906. doi: 10.18653/v1/2024.findings-naacl.246. arXiv: 2305.15005.
- [Y. Zhang et al., 2023] Zhang, Yu et al. (2023). *The Effect of Metadata on Scientific Literature Tagging: A Cross-Field Cross-Model Study*. Vol. 1. 1. Association for Computing Machinery, pp. 1626–1637. isbn: 9781450394161. doi: 10.1145/3543507.3583354. arXiv: 2302.03341v1.
- [Zhou et al., 2023] Zhou, Pei et al. (2023). “How FaR Are Large Language Models From Agents with Theory-of-Mind?” In: pp. 1–18. arXiv: 2310.03051.
- [D. M. Ziegler et al., 2019] Ziegler, Daniel M. et al. (2019). “Fine-Tuning Language Models from Human Preferences”. In: doi: 10.48550/arXiv.1909.08593. arXiv: 1909.08593. <https://openai.com/index/fine-tuning-gpt-2/> <https://perma.cc/366Y-FGDX>.
- [Zittrain, Albert, and Lessig, 2014] Zittrain, Jonathan, Kendra Albert, and Lawrence Lessig (2014). “Perma: Scoping and addressing the problem of link and reference rot in legal

citations". In: *Legal Information Management* 14.2, pp. 88–99. doi: [10.1017/S1472669614000255](https://doi.org/10.1017/S1472669614000255).

Appendices

Appendix A

Survey Appendix

A.1 Survey Questions

Survey questions are detailed in Supplement S2 [doi:10.21954/ou.rd.28045166].

A.2 Survey Results

A PDF containing full anonymised survey answers is available at: doi:10.21954/ou.rd.28045865.

A.2.1 Demographics

Survey respondent demographics are available in Supplement S3 [doi:10.21954/ou.rd.28045442].

Appendix B

Personas Appendix

B.1 Cluster Analysis

Supporting resources for the analysis are available at: doi:10.21954/ou.rd.28045886, and the following cluster data is also available at:

doi:10.21954/ou.rd.28044944 [path: /survey/clusters/QQ9-14_5CL_analysis_FINAL.xlsx]

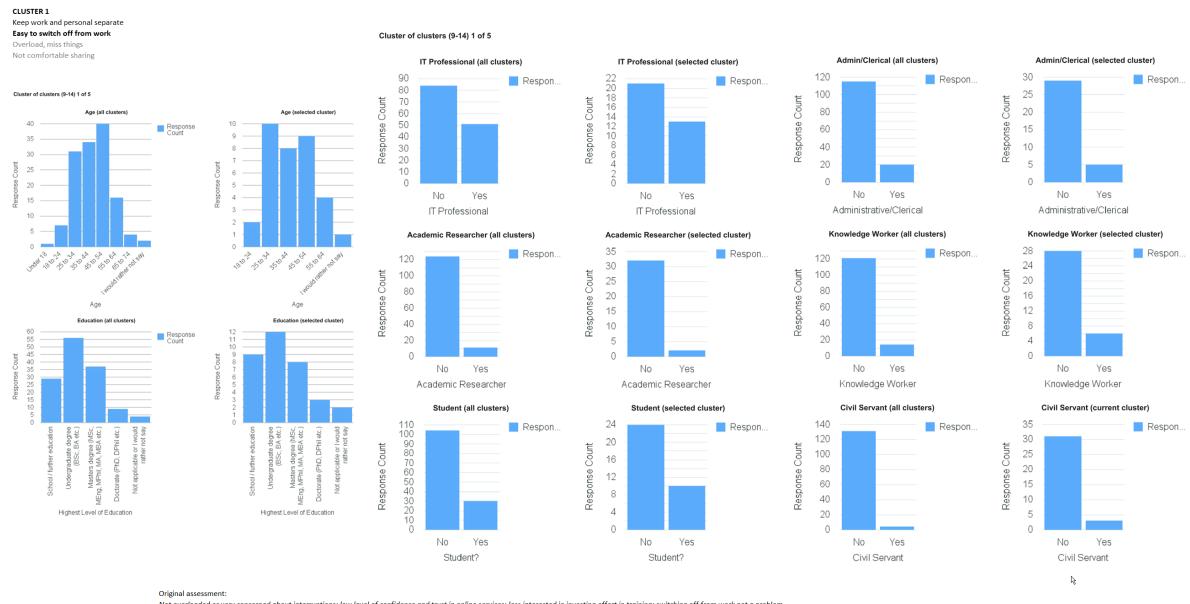


Figure B.1: Demographics for cluster 1

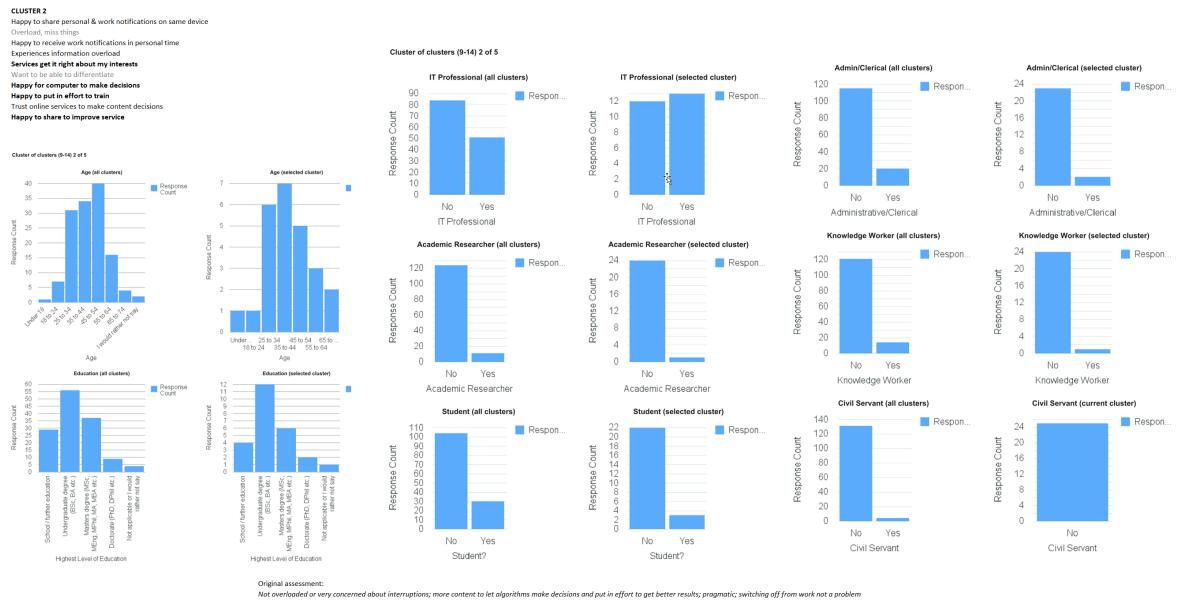


Figure B.2: Demographics for cluster 2

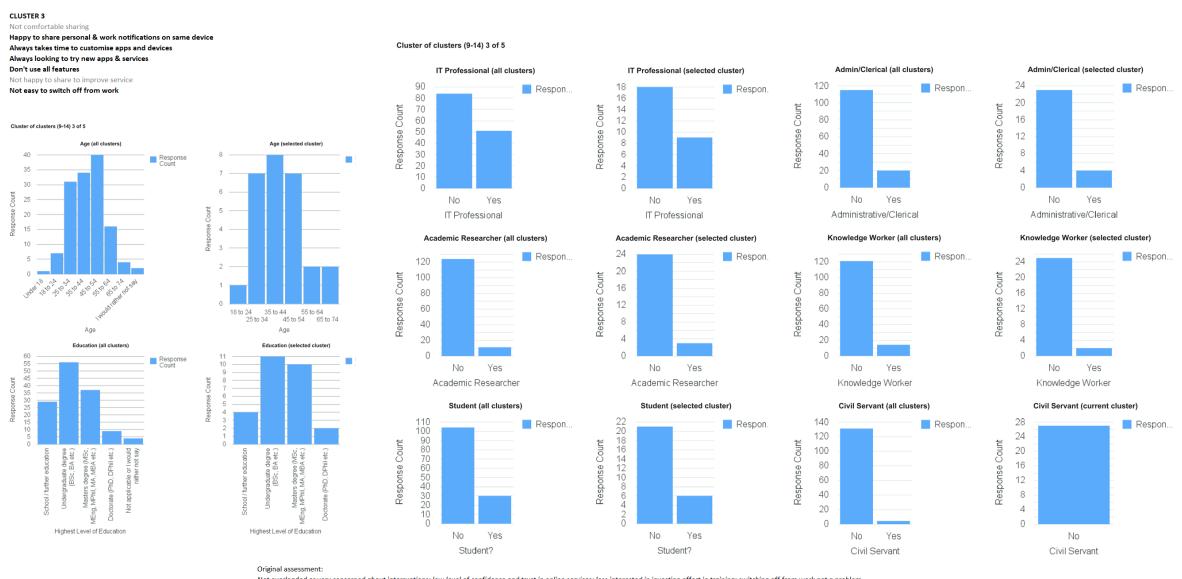


Figure B.3: Demographics for cluster 3

B.1. Cluster Analysis

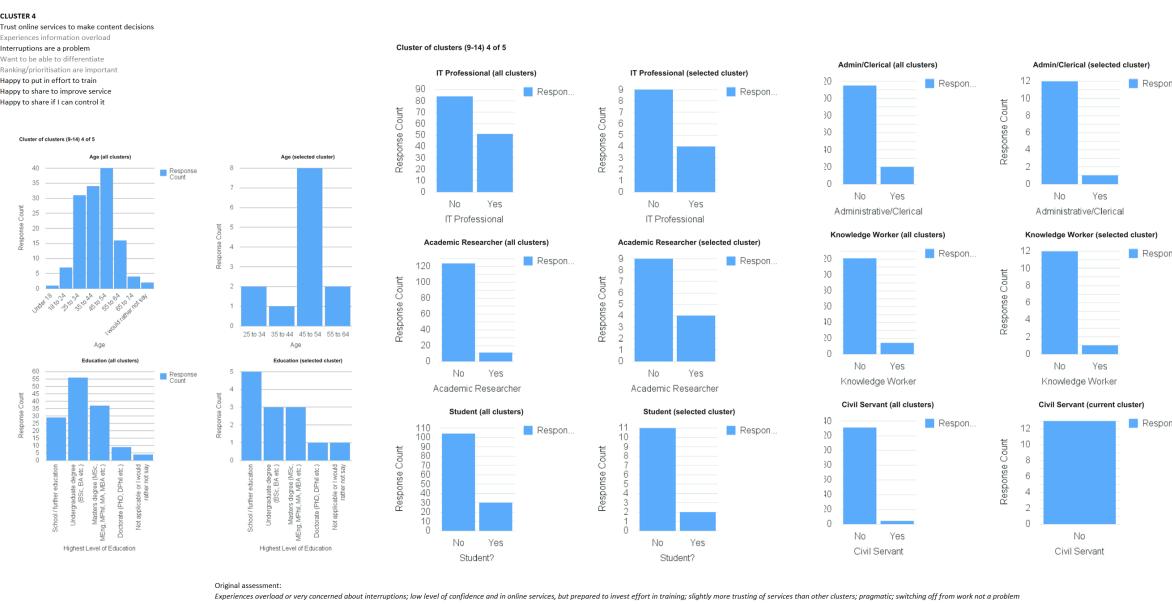


Figure B.4: Demographics for cluster 4

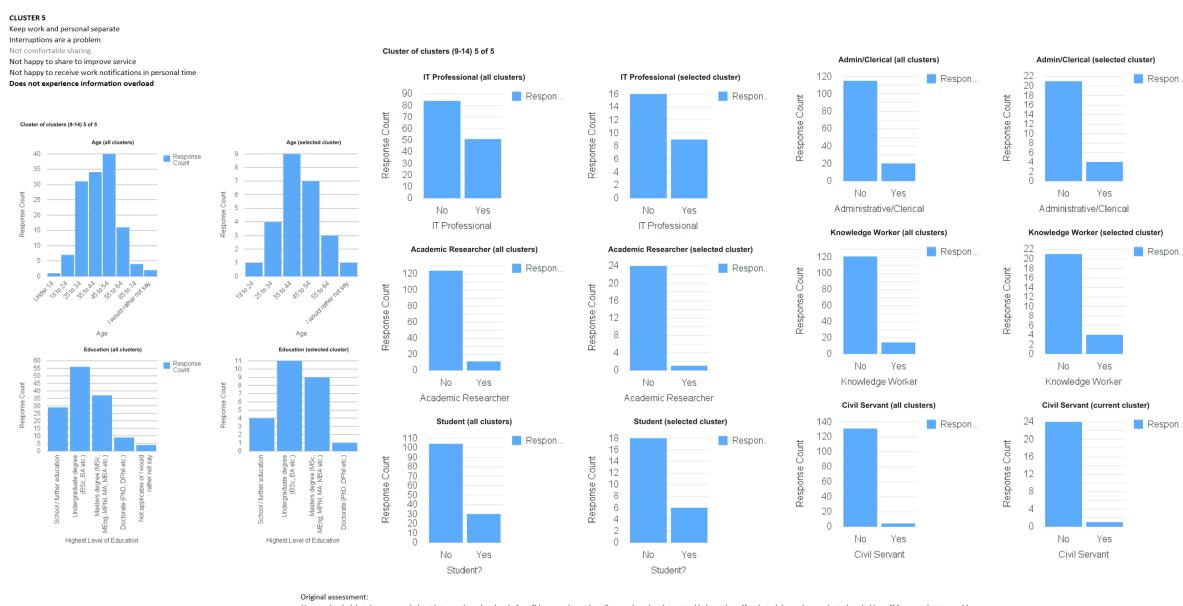


Figure B.5: Demographics for cluster 5

B.2 Final Personas

The following personas are also available in JSON format at:

[doi:10.21954/ou.rd.28044944](https://doi.org/10.21954/ou.rd.28044944) [path: /personas/personas_final.json]

They are also available in standalone format in Supplement S4 [[doi:10.21954/ou.rd.28045454](https://doi.org/10.21954/ou.rd.28045454)].

Susan	Age: 58 Gender: Female							
<p>Do</p> <p>I live in the Midlands in the UK and work as an administrator at the nearby university.</p> <p>I'm married and have two children – unlike me they both went to university. Since graduating one settled down quite close but the other lives and works in London.</p> <p>I have a few hobbies outside of work – baking, tennis and getting out into the local countryside.</p>	<p>Feel/Think/Believe</p> <p>While I enjoy my job, it's not the most important thing in my life; I have no trouble switching off at the end of the day, even when it's been very busy. When I socialise with work colleagues we rarely talk about work (otherwise I probably wouldn't socialise with them).</p> <p>I know my children lead busy lives now, but I miss seeing as much of them as I used to, particularly the youngest who is in London now. They do try and keep in touch but I don't always know what they're up to or how they are doing.</p> <p>I really enjoy my tennis and spend a lot of time helping organise club events.</p> <p>I'm not always entirely on top of things in my personal life because I don't check my email often enough.</p>							
<p>Technology Experience</p> <p>I use a desktop computer at work for admin, email and maybe a little web browsing. We have one at home too, but it's mostly my husband on that.</p> <p>I've had IT training at work and get along fine with computers – although I prefer to stay in my comfort zone.</p> <p>My son made me get one of those smartphones. I didn't really see the point at first, but it is actually quite useful for staying in touch and organising things. I think I mostly use Facebook and WhatsApp as well as things like the weather app.</p> <p>I admit I do use Facebook quite a lot, but there are a lot of stories about how much they know about you and what they do with that. If it wasn't so handy, I'd use it a lot less.</p>								
 <p>Problems → Needs → Existing Solutions</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33.33%;">Problems</th> <th style="width: 33.33%;">Needs</th> <th style="width: 33.33%;">Existing Solutions</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;"> Not as aware of her adult childrens' activities and day to day lives as she would like to be. Because of relatively low level of engagement with computers at home, Susan sometimes misses items of news or things to act on – particularly when they come in via email or get lost in Facebook feeds. </td> <td style="padding: 5px;"> Tools to help her track what is going on with friends, family and hobbies in social media without needing to log in all the time. A way to ensure that she does not miss important emails. </td> <td style="padding: 5px;"> Existing algorithms in social services that select content for users. Email filtering. Notification functionality in social media smartphone apps. </td> </tr> </tbody> </table>			Problems	Needs	Existing Solutions	Not as aware of her adult childrens' activities and day to day lives as she would like to be. Because of relatively low level of engagement with computers at home, Susan sometimes misses items of news or things to act on – particularly when they come in via email or get lost in Facebook feeds.	Tools to help her track what is going on with friends, family and hobbies in social media without needing to log in all the time. A way to ensure that she does not miss important emails.	Existing algorithms in social services that select content for users. Email filtering. Notification functionality in social media smartphone apps.
Problems	Needs	Existing Solutions						
Not as aware of her adult childrens' activities and day to day lives as she would like to be. Because of relatively low level of engagement with computers at home, Susan sometimes misses items of news or things to act on – particularly when they come in via email or get lost in Facebook feeds.	Tools to help her track what is going on with friends, family and hobbies in social media without needing to log in all the time. A way to ensure that she does not miss important emails.	Existing algorithms in social services that select content for users. Email filtering. Notification functionality in social media smartphone apps.						

Figure B.6: Final PATHY persona "Susan"

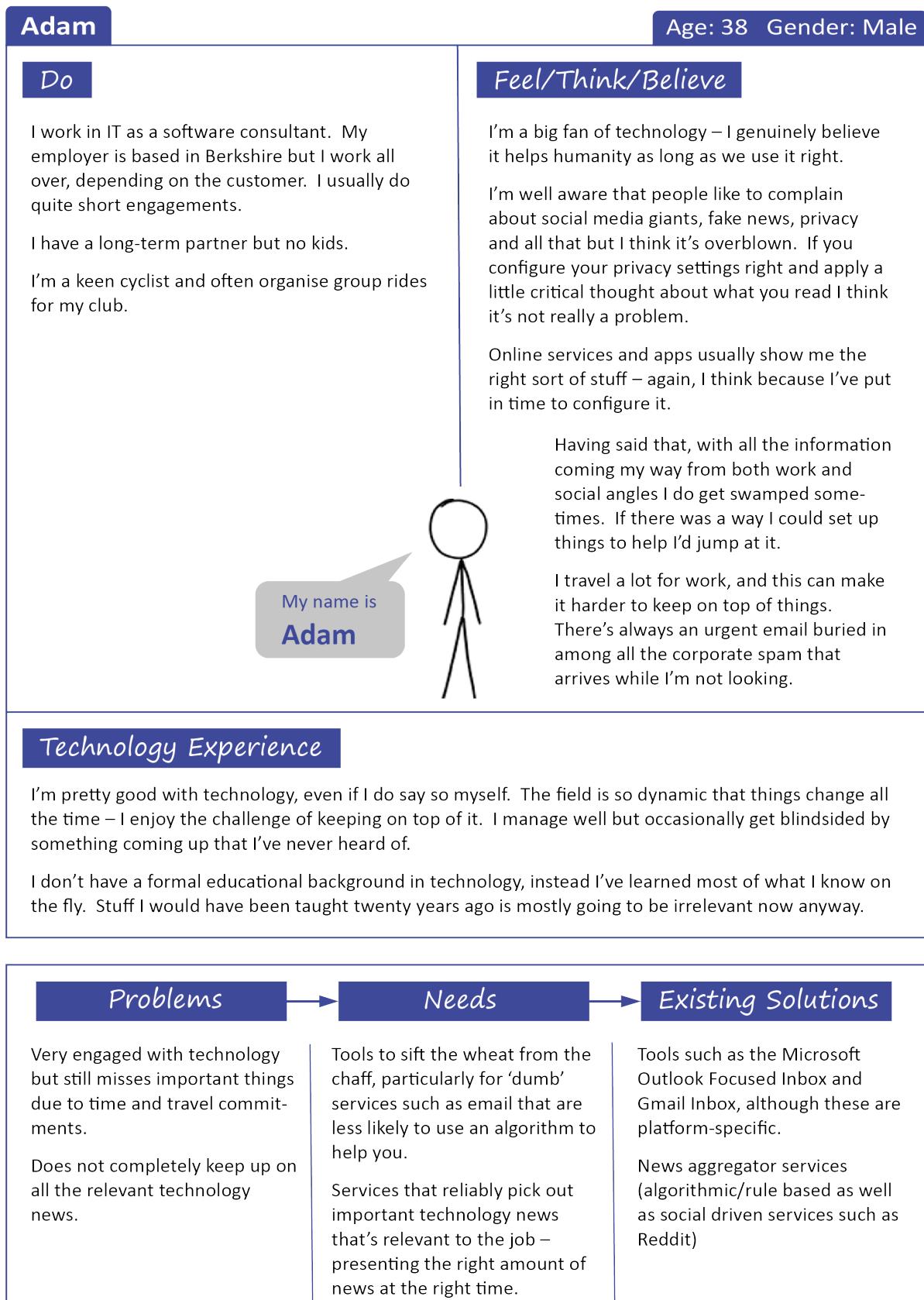


Figure B.7: Final PATHY persona "Adam"

Phoebe		Age: 23 Gender: Female						
Do	Feel/Think/Believe							
<p>I'm a recent graduate working for a accountancy firm in London. I'm on the management fast track, meaning that I move around many different parts of the business to learn how they work.</p> <p>I'm currently single, mainly because I have so little time to do things like organise and go on dates.</p> <p>My undergraduate degree was in Chemistry, but I quickly realised that I didn't want to be in a lab for a living.</p>	<p>I have a lot of university friends now scattered all around – I want to stay in touch via social media (and meet up when we can) but I don't always keep on top of what people are up to as many of them post a lot of rubbish on their feeds.</p> <p>I love my job but I feel my work consumes most of my time and I have trouble switching off. There's always a project going on.</p> <p>While I have many social apps that I use a lot, I'm not convinced that that have my best interests at heart and I sometimes wonder if I share too much about myself with them.</p>							
 <p>Technology Experience</p> <p>I guess I'm what people call a digital native. Technology isn't a big thing in its own right for me, but I suppose I depend on it for everything – work and social. While I happily admit that I don't use all the features on my phone (who does, really?), I am always keen to try out new apps and online services if they have something to offer me. If I am really into a particular app, I will take time to ensure that it works just so.</p> <p>I have a notebook computer for work, but I don't actually use it as much as I thought I would when I started – I mainly use it for typing things up and doing admin on the intranet (some of our admin apps don't work on mobile). I use my phone for a lot of my work, particularly for communication and collaborating. I only carry the one phone around with me for convenience, so it runs several work and personal apps.</p> <p>I mainly chat with people using an instant messenger. I pride myself on being able to select a meme for every occasion (although I'm not sure that counts as 'technology experience').</p>								
<p>Problems → Needs → Existing Solutions</p> <table border="1"> <thead> <tr> <th>Problems</th> <th>Needs</th> <th>Existing Solutions</th> </tr> </thead> <tbody> <tr> <td>Lack of time to stay on top of social media due to pressure of work. Finds it hard to switch off from work in personal time.</td> <td>Find ways of keeping up with friends on social media that don't consume a lot of time wading through minutiae. Help making a distinction between work and personal time when using a single device for both.</td> <td>Existing algorithms in social services that select content for users. Scheduled or manual 'do not disturb' features in apps used for work (i.e. Slack's Do Not Disturb hours).</td> </tr> </tbody> </table>			Problems	Needs	Existing Solutions	Lack of time to stay on top of social media due to pressure of work. Finds it hard to switch off from work in personal time.	Find ways of keeping up with friends on social media that don't consume a lot of time wading through minutiae. Help making a distinction between work and personal time when using a single device for both.	Existing algorithms in social services that select content for users. Scheduled or manual 'do not disturb' features in apps used for work (i.e. Slack's Do Not Disturb hours).
Problems	Needs	Existing Solutions						
Lack of time to stay on top of social media due to pressure of work. Finds it hard to switch off from work in personal time.	Find ways of keeping up with friends on social media that don't consume a lot of time wading through minutiae. Help making a distinction between work and personal time when using a single device for both.	Existing algorithms in social services that select content for users. Scheduled or manual 'do not disturb' features in apps used for work (i.e. Slack's Do Not Disturb hours).						

Figure B.8: Final PATHY persona "Phoebe"

Kenton		Age: 49 Gender: Male						
<p>Do</p> <p>I work as a client manager for a global consumer products company. You will have heard of us. My role is essentially to drive sales, but looking after existing customers and bringing new ones in. As many of my accounts are themselves global corporates, I end up doing a fair bit of international travel.</p> <p>I never went to university – it didn't interest me at the time – so I started my career at 18 just after A-levels.</p> <p>I'm divorced, with one son who lives with his mum but I see often.</p> <p>Outside work, I watch football and play golf. I used to do it more the other way around when I was younger.</p> <p style="text-align: right;">My name is Kenton</p>	<p>Feel/Think/Believe</p> <p>I have a lot to stay on top of at work. I sometimes miss things, but someone usually reminds me if it's urgent. I prefer not to miss things from important customers though, even if they are not urgent I like to give them a prompt reply. It can be difficult though, when you get things flying at you from all directions at all times.</p> <p>I'm on Facebook, Twitter and all that, like most people are. Not sure how I'd organise my social life without Facebook (how did I manage before?)</p> <p>I hear all the usual scare stories about social media giants, but they're a bit overblown by the media. I don't over-share, but I'm happy to put information out there – I think you get more out of it that way. Of course in an ideal world I'd be happy to have more control.</p>							
<p>Technology Experience</p> <p>I don't think I'm held back by not having a degree. A lot of graduates joining my company haven't got a clue about the real business world anyway. Experience counts for a lot and I've always done well for getting stuck in with new things.</p> <p>I really believe that you get out what you're prepared to put in with technology. I'll put in the time when I can to get things set up right, tell it my preferences or whatever helps.</p> <p>In mainly use office tools like Word and Excel and communications tools such as Webex day to day. We also have a new CRM system that I have to use, as well as the old CRM system that we haven't completely migrated off yet, and the other CRM system that we also use, for reasons I don't fully understand.</p> <p>I travel a lot, so I'm pretty good at managing with a notebook and mobile.</p>								
 <p>Problems → Needs → Existing Solutions</p> <table border="1"> <thead> <tr> <th>Problems</th> <th>Needs</th> <th>Existing Solutions</th> </tr> </thead> <tbody> <tr> <td>Has to use multiple different systems that are not integrated with each other. Receives many interruptions from different sources that need to be prioritised differently.</td> <td>Methods to handle and prioritise incoming messages.</td> <td>Email rules. Social media functionality. Features of bespoke applications that control notifications.</td> </tr> </tbody> </table>			Problems	Needs	Existing Solutions	Has to use multiple different systems that are not integrated with each other. Receives many interruptions from different sources that need to be prioritised differently.	Methods to handle and prioritise incoming messages.	Email rules. Social media functionality. Features of bespoke applications that control notifications.
Problems	Needs	Existing Solutions						
Has to use multiple different systems that are not integrated with each other. Receives many interruptions from different sources that need to be prioritised differently.	Methods to handle and prioritise incoming messages.	Email rules. Social media functionality. Features of bespoke applications that control notifications.						

Figure B.9: Final PATHY persona "Kenton"

Usha		Age: 43 Gender: Female
Do	Feel/Think/Believe	
<p>I'm a senior partner in a legal practice in a medium market town in England. My specialism is company law and I work mostly with small and medium enterprises in the local area.</p> <p>Outside of work, I enjoy horse riding and socialising.</p> <p>I'm married and have two school-age children.</p>	<p>My work is very busy, as I usually handle several cases at once. While I have a personal assistant who handles a lot of my calls and email, I'm frequently in direct contact with multiple clients over the course of a week, mostly by phone and email.</p> <p>While I'm pretty well organised and do manage to stay on top of all this, I do find it quite trying when I'm trying to concentrate on something to get a lot of interruptions, which can happen.</p> <p>I enjoy switching off at the end of the day and getting back to my family. I have a dedicated work mobile phone and that gets switched to do not disturb mode out of hours, so only emergencies get through to me.</p>	
 <p>Technology Experience</p> <p>I would say that I'm competent with technology, it's an essential tool to do my job and organise my life. It's also a useful way to stay in contact with extended family.</p> <p>I'm a moderate user of social media, both in a personal and professional capacity. Most of my family and friends use Facebook to some degree and a lot of things that I'm involved with are organised there, so I need to use it for that. I'm reticent to share too much information about my family and personal life online though. My practice maintains a Facebook page and Twitter account and I try to tweet reasonably regularly to maintain my professional profile. I also use LinkedIn, mainly as a way of making and maintaining business contacts.</p> <p>I depend on my smartphone and laptop when I am out of the office; I tend to use a desktop in the office. I use phone and email a lot for my work, but I also increasingly use conferencing tools such as Skype and Webex to interact with clients (I will use whatever the client has a preference for, if they have one).</p>		
Problems	Needs	Existing Solutions
<p>Being interrupted by incoming messages and notifications when trying to focus on something.</p> <p>Maintaining multiple social media channels (professional & personal) and staying on top of these.</p>	<p>Ways to manage/intercept interruptions coming from multiple sources.</p> <p>A more holistic approach to multiple social media sources, to manage both incoming and outgoing information.</p> <p>Ability to make a distinction between work and personal technology, particularly for interruptions.</p>	<p>Has a (human) personal assistant.</p> <p>Existing functionality provided by social media platforms.</p> <p>Uses different mobile phones for work and personal.</p> <p>'Do not disturb feature' on phone.</p>

Figure B.10: Final PATHY persona "Usha"

Appendix C

Awareness Agent Application Appendix

C.1 Additional Figures

Figure C.1 shows an alternative view of the Awareness Agent queue system.

Figure C.2 shows the JSON mapping documents used by the Allocate service.

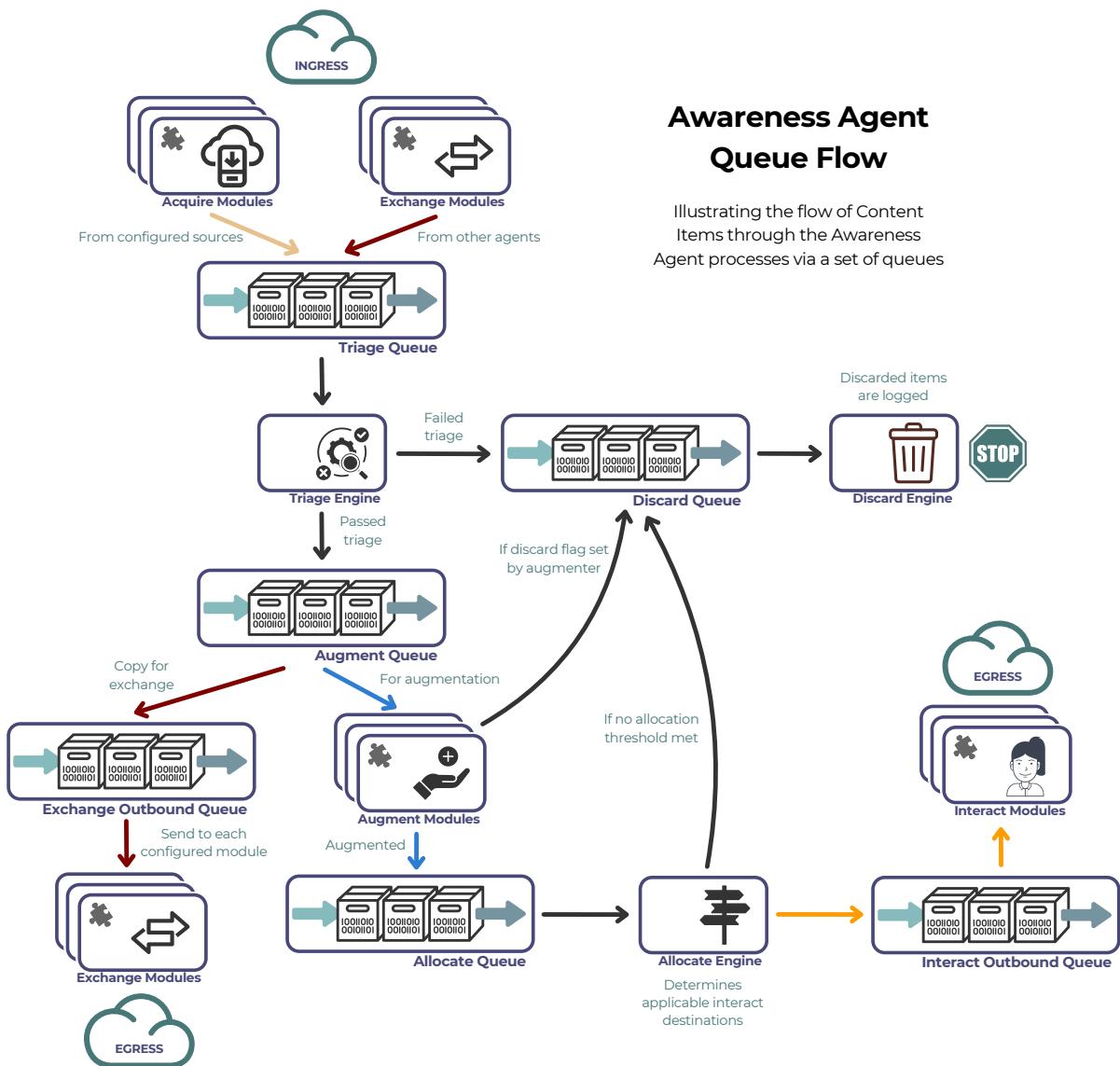


Figure C.1: CI Queue Flow in Awareness Agent

Augmentation Queue Mapping

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "AllocationAugmentationQueueMapping",
  "type": "object",
  "properties": {
    "augmentationName": {
      "type": "string",
      "description": "The name of this mapping augmentation"
    },
    "augmentationValue": {
      "type": "string",
      "description": "Augmentation value to be mapped"
    },
    "queueId": {
      "type": "string",
      "description": "Mapped Allocation Queue identifier"
    }
  },
  "required": [
    "augmentationName",
    "augmentationValue",
    "queueId"
  ],
  "uniqueItems": true
}
```

Context Queue Mapping

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "AllocationContextQueueMapping",
  "type": "object",
  "properties": {
    "contextId": {
      "type": "string",
      "description": "Context identifier"
    },
    "queueId": {
      "type": "string",
      "description": "Mapped Allocation Queue identifier"
    }
  },
  "required": [
    "contextId",
    "queueId"
  ],
  "uniqueItems": true
}
```

Queue Interact Mapping

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "AllocationQueueInteractMapping",
  "type": "object",
  "properties": {
    "queueId": {
      "type": "string",
      "description": "Allocation queue identifier"
    },
    "interactId": {
      "type": "string",
      "description": "Mapped identifier of an Interact destination to allocate to"
    }
  },
  "required": [
    "queueId",
    "interactId"
  ],
  "uniqueItems": true
}
```

Allocate Service Mapping Documents

Augmentation Queue Mapping is used to determine which internal queue(s) a CI should be sent to based on the classification augmentation value that the CI has. For a given Augmentation Name, each augmentation value is mapped to a Queue ID

Context Queue Mapping is used to determine which internal queue(s) should be locked or unlocked based on a Context Selection Action. The Context ID of the selected context is mapped to all Queue IDs that should be unlocked when that context is selected.

Queue Interact Mapping is used to determine to which Interact service content on a given queue should be sent to. Queue IDs are mapped to the applicable Interact Service IDs.

Figure C.2: Awareness Agent Allocate Mappings

C.2 Slack Application Manifests

The Awareness Agent Slack Apps¹ are deployed to Slack using App Manifests, as described at: <https://api.slack.com/concepts/manifests> [<https://perma.cc/Y6GK-WQAW>].

Manifest YAML² was created for the Acquire [6.7.4.1], Interact [6.7.5] and Simulate (synthetic content) [7.4.2] component, and is located at:

doi:10.21954/ou.rd.28044944 [path: /study/slack/app-manifest/personas]

¹<https://api.slack.com/docs/apps> [<https://perma.cc/HHN4-LQQD>]

²<https://yaml.org/spec/1.2.2/> [<https://perma.cc/2FWV-BBSK>]

Appendix D

Design and Development Log

Appendix

D.1 Overview

This appendix lists significant design decisions and development points during the course of the project, covering both the core Awareness Agent itself, and the ancillary services such as Data/ML Service and the Evaluation engine. This information is taken from our contemporaneous development log and is intended to show the path of design, testing and reflection that we took to arrive at our model, architecture and implementation of the Awareness Agent and associated evaluation framework.

Only entries that are directly referenced in the thesis body are included in this appendix; the full version of the Design & Development Log is contained in Supplement S1, located at: [doi:10.21954/ou.rd.28045163](https://doi.org/10.21954/ou.rd.28045163).

Entries are associated with the topics listed in Table D.1.

Table D.1: List of Design & Development Log Topics

Topic	Description	Reference
AwAg Platform	Supporting platform and application framework for the Awareness Agent	6.7
AwAg CI	Content Item design and use	6.7.1
AwAg Acquire	The Acquire Service	6.7.4.1
AwAg Augment	The Augment Service	6.7.4.2
AwAg Exchange	The Exchange Service	6.7.4.4
AwAg Interact	The Interact Service	6.7.4.5
AwAg UDML	User-Directed ML	6.7.6
AwAg Sim	Additional Awareness Agent components for simulation with synthetic data	7.3.2
AwAg Eval	Additional user interaction components for synthetic evaluation	8.1.2
AwAg UI	Additional Awareness Agent components for synthetic evaluation	6.8.5
ML Service	Ancillary Machine Learning Service (awagml)	6.8.2
Data Service	Ancillary Data Service (awagdata)	6.8.3

D.2 Abridged Design and Development Log

2019-07-19 Switch from JBot to jSlack

AwAg Platform Our original Slack test code used a relatively limited library called JBot^a, designed to get you up and running quickly with Slack bots. This was fine for initially proving some concepts, but the full range of our design depended on being able to fully access the Slack API, so the project was transitioned to the full-featured jSlack API^b.

^a<https://github.com/rampatra/jbot>

^bjSlack later transitioned to <https://github.com/slackapi/java-slack-sdk>, having been taken on with its original developer <https://github.com/seratch> by Slack itself

2019-07-26 Add Slack RTM support

AwAg To better support real-time monitoring of content in Slack, we added the now-legacy RTM API^a to the platform, enabling us to listen for content more effectively.

^a<https://api.slack.com/legacy/rtm>

2019-09-24 First implementation of a Content Item

AwAg Platform Initial testing used Slack content more or less natively, which was effective for the scope of the first prototype, but it became clear from our work that data abstraction and standardisation was needed to build a system that would be flexible and extensible enough to accommodate many diverse data sources. This point marked the start of developing the Content Item concept, in the first instance by applying a simple layer of abstraction over native Slack items.

2019-10-15 Add the concept of Augmentation Items

AwAg Augment We had tried several approaches so far to enhancing content to make it more easily actionable, with some issues arising: while it was certainly possible to summarise items or apply a classification, there was no organised framework within which this could sit, and the approach was not extensible. The previous successful introduction of the Content Item concept gave us an opportunity to address this issue in a structured way, by extending the CI structure to include Augmentations so that we could be both flexible about what enhancements we could add and also have a mechanism for abstracting this and managing/addressing the information.

2019-10-17 Add simple text based explanation mechanism for Augmentations

AwAg Augment First introduction of an explainability mechanism within the augmentation framework.

2019-10-19 Reorganise modules into Acquire, Augment, Control etc.

AwAg While this development change was little more than a code refactoring at this stage, this was a significant design development. It marked the introduction of the concept of high level components within the Awareness Agent system model, with the creation of the Acquire and Augment module types to handle those logical operations. This was a result of initial work to expand the scope of these aspects of functionality - early testing showed us that a consistent modular approach was needed to allow for the future expansion of scope that we were looking for.

2019-10-19 Introduce Triage service

AwAg Looking at the content flow through the prototype Awareness Agent, we realised that there was a need for a first point of contact assessment of incoming content - a triage process to borrow from medical terminology. We added the Triage service to act as the first port of call for all content coming from Acquire services, adding an ability to discard unwanted content based on simple heuristics before going on to Augmentation.

2019-10-21 Add Discard service

AwAg Before this point we had simply been dropping unwanted content by not passing it on to the next stage; however we recognised that there was utility in tracking discarded content - both from a scientific and application development perspective. By adding a service to which all unwanted items could be sent, we acquired the capability to easily record information about discarded items and also take any other actions appropriate at that point. This approach also had greater consistency with the overall CI queue-service flow model, with the Discard service fitting into a structured lifecycle for every CI.

2019-10-26 Add support for new Slack-specific metadata fields*AwAg CI*

We started formalising the concept of Extended Fields in the Content Item after testing showed us that we needed to be able to surface some source-specific information later in the CI lifecycle (for example at the point of augmentation or interaction). We recognised that this would be a generic need - sources other than Slack would need their own specific metadata - and we developed extended fields to fit this information within a standard CI structure in a manner consistent with the standard fields approach that we had already adopted.

2019-11-01 Transition CI data to using JSON-LD*AwAg CI*

As a consequence of the development of extended fields, we looked again at the basic data structure of the Content Item, which had hitherto been entirely bespoke JSON. We recognised that JSON-LD^a and the related concept of linked data notifications (LDN)^{b,c} provided us with an academically rigorous and supported mechanism for working with and sharing data from multiple sources, so we transitioned the internal CI structure to one compatible with JSON-LD.

^a<https://www.w3.org/TR/json-ld/>

^b<https://www.w3.org/TR/ldn/>

^c<https://csarven.ca/linked-data-notifications>

2019-11-09 Distinguish standard & extended fields with prefix

AwAg CI

As part of the transition to JSON-LD we looked again at how properties (fields) in the Content Item were named. We had three categories of property: native (existing in the source data item), standard (conforming to our standard fields definition [6.7.1.1]), and extended [6.7.1.2]. We decided to take an approach of using Compact IRIs (CURIEs)^a, with the namespace awag mapped to Awareness Agent schema <http://parse.net/awag/0.1/> in the @Context. We gave standard fields (where they existed as independent properties) a shorthand namespace of awag:std, and extended fields a namespace of awag:xtd. We made a decision to use unchanged names for native properties from the source system rather than renaming or adding an additional namespace. There had been a number of ways that we might approach this, and we retrospectively discuss this further in Section 6.10.3.

^a<https://www.w3.org/TR/2010/NOTE-curie-20101216/>

2020-09-28 Add LDN module type

AwAg

We added the basic framework for supporting LDN as an Acquire source via a client for an LDN inbox. We implemented our own local Solid server (nodeSolidServer^a running on FreeBSD via PM2^{bc}). We ultimately never progressed beyond the proof of concept stage for LDN Acquire support via Solid – although the technology was potentially useful, but did not have the level of actual content that we wanted to run a study. We discuss this more in Section 6.10.5.

^a<https://github.com/nodeSolidServer>

^b<https://www.npmjs.com/package/pm2>

^c<https://gist.github.com/PieterScheffers/457769f2090c6b69cd9d>

2020-10-08 Add Augment engine and structure

AwAg

Significant internal change to the organisation of the Augment process, adding a new engine to support it.

Augment

2020-11-08 Add basic outbound exchange functionality via MQTT

AwAg Exchange The initial implementation of the Exchange component uses MQTT; we implemented code to manage the round trip serialisation of Content Items to/from binary data to be sent and received from MQTT.

2020-11-11 Change MQTT serialisation to Kryo

AwAg Exchange We found in testing that serialising our complex Content Item structure to binary – and being able to successfully instantiate the returned deserialised item – is actually quite difficult. We tried here with the Kryo library^a instead of the original approach of using native Java bytestream functionality.

^a<https://github.com/EsotericSoftware/kryo>

2021-04-21 Rework MQTT serialisation again

AwAg Exchange We had continued to experience problems with the Content Item lifecycle even with the Kryo library, so changed approach once more. This (successful) iteration uses Apache Juneau for object-JSON serialisation, and Apache SerializationUtils^a for string-binary conversion. The Juneau approach had been a well tested feature of other elements of the agent so far, so we were able to get consistent object-string conversion using it, and the Apache serialisation tools worked reliably for the string-binary conversion.

^a<https://commons.apache.org/proper/commons-lang/apidocs/org/apache/commons/lang3/SerializationUtils.html>

2021-05-14 Add ‘Prominence’

AwAg CI & Interact Adding the concept of ‘prominence’ a categorical code that can be assigned to a Content Item having possible values LOW, NORMAL, HIGH, HIGHER. We developed this as a way of surfacing augmentations: the prominence is a property at the Content Item level, and is effectively another Standard Field, but the mechanism for setting it is left open; the assumption is that the Augment engine or indeed later services can adjust prominence as they process the CI based on augmentations or other factors. Prominence can then be used by Allocate or Interact services to change how the CI is processed or displayed. The design rationale for this was that it is an easy to check standard way of indicating how prominent a CI should be. We introduced this in response to difficulties we experienced using augmentations for this task – they have the information, but not accessible in a standard way. Having Prominence available allows components to make high level decisions about a CI without knowing the internal augmentation structure.

2021-05-15 Add permalink feature to Slack content item

AwAg CI Testing with user interface prototypes showed that the agent would benefit from being able to easily direct the user to the original item in context in the source system via a URL. In the case of Slack content, there was not such URL available directly, but we could construct one from the message content and current context within the Acquire service. We added this as a new extended Slack field (PERMALINK), but also mapped this as a Standard Field (URL) so that other types of CI can be used in the same way.

2021-06-08 Add Slack Service

AwAg This marked the start of adding two-way interactivity to the Slack Interact component, exposing functionality to receive events from Slack in response to user actions^a. The first iteration of this was to add and test basic functionality; we would then go on to expand this and integrate with the Awareness Agent framework based on testing results.

^aInitially using the now-legacy RTM API – <https://api.slack.com/legacy/rtm>

2021-06-20 Improve Slack interactivity

AwAg Initial testing of the Slack service was successful, so we expanded the types of interaction that were possible and genericised the process by adding a `SimpleSlackAction` object to the code. We used this to add a set of internal queues for Slack actions that were based on abstracted Slack request objects, with a separate request handler component to process these. We did this to take an approach consistent with the design ethos of the rest of the agent, which we had found in testing to be adaptive to change and different types of request.

2022-01-04 Initial creation of awagml

ML Service We added a new back-end service, referred to as “awagml”, to initially support generating classifications for items. The first instance was a simple classifier to test this functionality - initially testing with both `MultinomialNB` and `SGDClassifier` (SVM) classifiers using scikit-learn [Pedregosa et al., 2011]. While the Awareness Agent itself is a Java application, we decided to implement a separate ML service in Python for the following reasons: 1) availability and quality of ML and associated packages in Python, 2) support for rapid prototyping, and 3) consistency with design principle of keeping back-end ML commoditised and separate. Our implementation work on the ML Service is covered in more detail in Section 6.8.2.

2023-02-21 Add item summarisation based on OpenAI/ChatGPT

AwAg This was the first instance of being able to exploit the revolution in Large Language Models such as ChatGPT. The agent up to this point had used a very simple technique for producing a shortened summarisation of content items for display: it would truncate long items, as well as removing any HTML content. We were able to enhance the augmentation processing, swapping out the original Summarise augmenter with a new one that passed the CI data to the OpenAI API and asked for a summary, which it then included as the summary augmentation (in the case of a failure, we used the original technique as a backstop). This greatly improved the utility of the presented CIs but importantly it also vindicated the design approach we had taken: we demonstrated that the summary augmentation could be seamlessly swapped out, taking advantage of a powerful commodity ML service that had not been available to us at project inception.

2023-03-10 Add an applicationIdentifier/contextIdentifier

AwAg Platform While working on the process of setting up a study to test the Awareness Agent we realised that there were a number of technical points where there was a conflict if multiple agents shared the same Java JVM (i.e. deployed to the same application server). In particular the use of singletons for internal queues and other items. We implemented a named application identifier to be used internally to address this.

2023-03-27**Initial work to add RSS as an Acquire source***AwAg*

We added RSS as a second type of Acquire source, with a Client type acquire service configured to pull one or more RSS feeds on a scheduled basis^a. This was a validation of the approach that we had designed for Acquire Services and Content Items generally, as it showed that we were able to relatively seamlessly add in a different type of source that not only had different content but also a different mechanism for acquiring content. Iterative testing of this required a few minor changes to be made – mainly in relation to the particular content properties of RSS items – but importantly we found these to be non-breaking changes for the existing Slack Acquire implementation, and we were able to seamlessly use the Slack Interact implementation to output RSS-sourced items.

^aUsing Quartz Scheduler in cron-type mode: <https://www.quartz-scheduler.net/documentation/quartz-3.x/tutorial/crontriggers.html> [<https://perma.cc/EP6D-R2RZ>]

2023-04-14**Add data service for recording summaries***Data Service*

As part of our testing process we had identified that summary quality was inconsistent, with some items being better summarised by OpenAI than others. To gather more information on this, we decided to add the facility to record the input and output of the summarisation process, so that issues could be reproduced and re-tested after the fact. To do this, we added a new service to the back end, *awagdata*, initially supporting only this functionality, with data persisted in a local SQLite database. We would later go on to expand this data platform for many other uses. Our implementation work on the Data Service is covered in more detail in Section 6.8.3.

2023-04-16 Implement Summarisation Feedback

AwAg Testing with users showed that summarisation quality was not always perfect; in some cases – particularly where the source content was short – the ChatGPT model would summarise badly or even fabricate summary text^a. We added a feature to the UI to allow the user to give feedback on the quality of the summarisation; this would capture the item in question, its content and the user's rating of the quality of the summary text. The captured information was then stored using the awagdata service for later use.

^aSee also Emsley [2023]

2023-04-21 Add services to add and supply “dummy text”

Data Service Our first look at synthetic data was the enhancement to awagdata to allow what was initially referred to as “dummy text” to be stored & retrieved using services `add_dummy_text` and `get_dummy_text`. This text was simulated message content (to be generated elsewhere) that could be served sequentially one item at a time, grouped by a notional “topic”. The design idea was that a future simulation service could periodically call this and get a series of pre-canned simulated “messages” for a given topic, channel or discussion. By pre-generating the content and storing in the data service this way, we could ensure good performance and reliability, as well as the ability to pre-vet content.

2023-06-14 Attempt to coerce OpenAI to only output the requested JSON

AwAg Eval The OpenAI API proved slightly unreliable; our design for evaluation was to ask OpenAI to output its evaluation as a JSON document conforming to a supplied schema. It would usually do so, but sometimes would respond in plain text or using an incorrect JSON schema. We had several iterations working on OpenAI prompts to make this more reliable.

2023-06-15 Add evaluation data recording

Data Service In response to initial testing of the Evaluate functionality in the agent, we added the back-end facility to record the evaluation data via a new service, `record_evaluation_data`. This records the details of what was evaluated, how (OpenAI settings) and what the response was. The intent was to facilitate the recording of this information to support both debugging and the study.

2023-06-23 Add support for Perspectives

AwAg Eval We introduced a concept called Evaluation Perspective, which was an attempt to add another dimension to how we view and run evaluations. The concept was that a perspective was a second way of looking at the evaluation, by asking a slightly different question of OpenAI. For example “has this item been classified correctly?” is one perspective, but we could in theory ask slightly different questions. The new perspective data structure supported the automatic flow of this, with different perspectives being defined in the agent configuration and then automatically passed to OpenAI. However, we found success was limited, mainly because we were not able to devise a convincing alternative question to the default.

2023-06-25 Add Flow Monitor

Data Service Our service-queue design for the Awareness Agent meant that the flow of Content Items through the system should be clearly defined with known entry and exit points. We had found in testing that some items were in practice not appearing in the Interact UI when we had expected to see them. To help address these issues, and to provide general visibility on the internal workings of the agent, we added the Flow Monitor, recording each contact of a CI with a service or queue in `awagdata`. This meant that we could track the progress of individual items or aggregate flows and identify bottlenecks and other issues.

2023-06-26 Add Available Classifications to enable better evaluation

AwAg Eval Previously, our evaluation requests to OpenAI did not list the available options that awagml had chosen from when it had made its classification; to improve the quality of evaluation we added this information to the evaluation request. This was relatively easy to do technically, as our design for the Single Classification Augmentation Item already included this information.

2023-07-09 Add batch execution of OpenAI evaluation queries

AwAg Eval Our very first implementation had fired off a single evaluation request per Classification per Content Item per Perspective. This was quickly found to be very expensive in terms of token usage, as each request required significant prompting text, and adding more classifications to evaluate grew the demands substantially. We changed to a structure early on that would treat each CI as a single request, processing the evaluations for each classification. However, this was still not entirely efficient, so we added a facility for items to be combined and batched. That is, a single prompt request would include the common instructions and a certain number of combined CI's to evaluate. The idea was to reduce the amount of common text per item processed. However, this was at the cost of larger individual queries and higher complexity and we hit token limits with the GPT-3.5 models available to us at the time^a. We had mixed results with this, finding that combining too many items caused failures due to a number of factors - such as token per request limits being hit, and also a higher chance of the request receiving a malformed response. We found that 3 items per request was a reasonably safe number.

^aThe gpt-3.5-turbo-1106 model supported a maximum context window of 16,385 tokens, which a batched request of over 5 items could easily hit, while the more capable gpt-4 had not initially been available for API use

2023-07-11**Add evaluation failure recording***AwAg Eval*

We had added a specific mechanism for recording evaluation failures to awagdata, which we now incorporated into the Evaluation engine. Failures may be caused by an invalid response from OpenAI, network issues, running out of OpenAI quota for example. By testing for and recording which evaluations failed we gained the ability to re-run or debug them in a targetted manner.

2023-07-25**Add support for newer /chat/completions API for Evaluation***AwAg Eval*

At this time, OpenAI moved its current API from “Completions” to “Chat Completions”^a, a change made since we had started work on OpenAI evaluations. This was a significant change for us, and actually very helpful. One change introduced as part of it was to formally support JSON as a response format - we had previously requested a JSON response in the prompt text, with imperfect results. With the new API we could specify a defined response schema, which we could then parse. The changes made to the request side were also significant; we could move from constructing a single text prompt to using a more structured array of message JSON. This helped to a degree with our process of passing a mix of JSON and text in the evaluation request, allowing more structure. However, we noted it was still short of a full JSON request format where the API would be sent a schema-based request rather than being told this on each occasion. The change to an array of system/user messages in the OpenAI request also opened a design possibility to us, as it made it easier to make the request process more modular in terms of request construction and variation. We would go on to use this later.

^a<https://help.openai.com/en/articles/7042661-moving-from-completions-to-chat-completions-in-the-openai-api> [https://perma.cc/KBZ7-MGPP]

2023-09-07 Add support for tags and for recording/fetching raw evaluation items*Data Service*

We made a number of additions to awagdata to support new evaluation-related functionality in the agent platform, by allowing data items to be stored associated with text tags, and adding new services to store and retrieve raw evaluation items (the CI data required to perform an evaluation operation).

2023-09-07 Add support for recording tags (in evaluation etc.)*AwAg**Platform*

Tag support was a major study-focussed change that emerged from the first test study iterations. The main driver for this was data management and organisation of tests. We needed to support concepts such as "test run 1" or "evaluation run 1" so that we could identify which data (content items, evaluations etc.) were associated with each run. This would enable us to clearly define the start and end of different study phases, and also allow us to run multiple different evaluations against the same content and identify different runs. We did this by adding the concept of tags. Each write of an item to awagdata would be accompanied by one or more tags, plain text strings. Use source of the tags varied: we could set a "current tag" (or multiple current tags) in the Acquire service via the UI; each incoming CI would be tagged with these. Similarly for evaluations we added support for input tags (evaluate only those CIs that match the supplied tag) and output tags (i.e. store the evaluation output in awagdata with these tags attached). This allowed us a great deal of control and precision in the evaluation and general study administration process.

2023-09-12 Add on-demand processing for evaluations based on recorded evaluation items

AwAg Eval

Adding tag support had given us the ability to change our approach to evaluation processing. Our previous design for evaluation was to incorporate this step into the CI flow; however we had found this was not optimal. Sometimes we were not interested in evaluating during some types of tests for example (each evaluation execution has both a monetary and performance cost attached to it). We also wanted the ability to run evaluations in batch after changing something, such as the prompting text. To achieve this we added the ability to enable/disable the background Evaluate service as we had previously done for Simulate and Acquire, and also added a Slack command based facility to run evaluations in batch on demand. At the back end we added an awagdata facility to record all Content Items as they passed through identified by the currently active set of tags. We could then run on-demand evaluation batches against a set of CIs defined by the passed tag, and we were also able to identify and distinguish the output of these runs from the added output tags. We actually found that this was our preferred way of running evaluations, as it gave us in our role of study administrator much more control.

2023-10-02 Add awagUi implemented in Angular

AwAg UI

We had not been satisfied with the first attempt of evaluation exploration so we make the decision to design a dedicated Angular based UI for the study administrator to interact with evaluation results. We also added the ability for the user (study participant) to submit feedback data via this UI on the quality of the evaluations. This has awagdata as the back end, which required some additional services to be added for access to data for reporting and interaction in this way. We named this new web based interface “awagUi”.

2023-10-04 Add mode1/mode2 support to awagUi evaluation explorer*AwAg UI*

Initial testing of the evaluation feedback process with the pilot study participants showed some issues with the design of the UI; some concepts were not always clear and in some cases there was clutter in the UI that the user did not want to see. We approached this by designing two presentation modes, one with stripped down explanatory information and a differently organised layout. The user had the choice of switching between these.

2023-10-15 Add support for model description*AwAg Eval*

Testing of the quality of evaluations showed cases where the OpenAI model was making clear mistakes about the meaning of classifications in the model that it was asked to evaluate. We addressed this by passing a model description text along with the classification request to `awagml` and incorporating this in the prompting for the request - which produced noticeably better results. We then added support for this throughout the agent system, providing a mechanism for the user to add and maintain descriptions for models.

2023-11-22 Add recording of classification actions*AwAg Eval*

We had found during the pilot study phase that it would be helpful to maintain a record of the actions of the user to classify items – that is, where the user used one of the available UI mechanisms to perform a training event related to a content item (changing classification value or confirm correctness of a classification for example). We had not previously been capturing a log of these actions when they were performed, but realised that having this data would provide important experimental insight. A classification performed by the user is a de facto evaluation of the original classification by the human, mirroring the evaluation by the OpenAI-based evaluator. This would provide us with an additional mechanism for getting feedback data on the quality of the AI evaluation by comparing the agreement with the human and AI evaluations.

2023-12-08**OpenAI file handling***Data Service*

We had made a design decision based on our experiences so far to add the ability to train or fine-tune OpenAI models used for evaluation^a, so that we could compare the performance of trained vs untrained models. To support this we added a mechanism to generate, manage and upload training data files to OpenAI, and then to use these files to train models. To make the process manageable, we chose to handle the whole process within the awagdata layer. This includes generating training objects from recorded classification actions, storing these within our object store, using them to generate OpenAI files and managing the querying, execution and deletion of models on the OpenAI side. We took this approach because OpenAI provided only bare API support for these operations with no higher management ability; we had found that it was too easy to lose track of the various models and files on the OpenAI system, and not always easy to tell the provenance of these files. Our system tracks history, source and other information related to all artefacts that we create in OpenAI for easy and efficient management.

^a<https://platform.openai.com/docs/guides/fine-tuning> [<https://perma.cc/CKJ2-LVVK>]

2024-02-18 Add Evaluation support to awagdata*AwAg Eval*

Our original design for Evaluation was to have the processing done within the Java Awareness Agent application at time of CI generation; we later added the ability to batch this and run on-demand evaluations. During the pilot study we found that we had moved more and more towards on-demand tag based evaluation processing and rarely used the original mode. Because of this, we decided to replicate the evaluation functionality in the back end, so that it could be run entirely independently of the agent itself. This was consistent with the agent design concept, which does not include evaluation as a core function, and was also compatible with how we ran offline evaluations: all the data used to run these was already stored in the awagdata layer, and the outputs were also recorded there. Changing our evaluation engine to the awagdata platform meant that we could more easily run and manage evaluation jobs via simple web service calls. We also retained the code in the Java Awareness Agent, because the structured nature of the evaluation request object and the standardised prompt structure meant that all we needed to do was ensure that a few resources were kept synchronised between the two, such as schemas and prompt templates.

2024-02-18 Add Fixit route*Data Service*

We had discovered during the pilot study run that a minor technical error had prevented the proper recording of evaluation items during the course of the study – they should have been recorded to the Data Service as they came in, but this was not happening. This would have been catastrophic for the study as it would prevent any evaluation being run on the data, which could not easily be re-captured. However, the necessary data to construct evaluation items after was in fact being recorded elsewhere within awagdata, so we were able to write code to extract this information and retrospectively construct the necessary evaluation items.

2024-02-23 Add chat component to awagUI

AwAg UI To facilitate interactive testing, we added a simple chat UI so that the user^a could enter some text which would be executed against the OpenAI model; this is functionality already provided by the ChatGPT UI^b, but that UI did not provide access to trained models, only to vanilla ones. By adding our own UI we could run arbitrary requests against our own trained models, also utilising our own system request prompt elements. This allowed us to more efficiently test model and prompt changes in an iterative fashion.

^aIn this case the user is the researcher rather than a study participant

^b<https://chatgpt.com/>

2024-02-23 Add subsets support

Data Service We had observed during the pilot study that the volume of content was too great for the participant to process all of it – in itself this was not unexpected and not a problem. However, we found that we would get more useful study results if the user acted on the same subset of items in each phase (initial classification and evaluation feedback), but it was important also to get a good distribution of items and not just have the user process in date order. To support this we added support for subsets to awagdata. We did this by adding the ability to add a set of subsets for the data, where we could define random subsets^a of the recorded Content Items identified by tag and a subset percent label. So for example we might generate a subset of 25% of the items for tagA, which would be identified by the path /subset/tagA/25. The user-facing UIs could then be filtered to show only items in that pseudo-random subset.

^aUsing Python `random.sample()` to generate a pseudo-random sample

2024-03-14 Add statistics and improve reporting

Data Service To improve the efficiency of result processing during and after the studies, we added a stats service, that would generate a set of statistics based on the evaluations and other data for a given study/tag, outputting as a JSON document containing a stats “package”. Having such a package automatically generated would greatly improve the ability to generate updated statistics for each study instance.

2024-04-04 Support for text-likert mismatch via partial feedback

AwAg UI We had found in some cases that the text description generated by OpenAI for its feedback did not agree with the Likert value that it had assigned (typically in this case they would be opposite to each other). This was a significant enough problem that we added an element to the evaluation feedback UI to allow the user to flag this easily, allowing us to capture data on the number of affected items.

2024-05-02 Add agree/disagree exclusion

Data Service To improve the quality of data used to train models, we added a feature to allow the list of user classification actions to be used for training to be filtered to exclude those where the user agreed with the classification or those where the user had disagreed. We had found that the user was agreeing with the classification chosen by the trained models in the majority of cases (often a very large majority) so the training data sets had been very skewed towards entries where there was agreement. The training data would be more effective with a better balance of content [Dube and Verster, 2023], so that the model would be trained with a similar number of disagreements to agreements. Adding these filters meant that we could build up our evaluations in multiple steps, assembling a training set with a higher proportion of disagree actions than a single unfiltered pass would generate (i.e. oversampling ‘Disagree’).

2024-05-02 Add dataset merging

Data Service To support the process of building up training datasets with varied compositions and using these to fine-tune models, we added a feature to allow the administrator to merge datasets – so for example two datasets that had been generated with different ratios of agree/disagree could be combined into one and then used to train an OpenAI model. This allowed us more control over the composition of fine-tuning datasets.

2024-05-27 Excel stats pack export

Data Service We added the ability to automatically output a statistics package to Excel format for easy analysis, using pandas to output dataframes to Excel worksheets^a^b and combine to multi-sheet formatted workbooks.

^ahttps://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_excel.html

^b<https://perma.cc/AN99-MZMP>

Appendix E

Synthetic Content Appendix

E.1 Simulation Schemas

The schemas for synthetic content generation [7.4.0.1] are illustrated in Supplement S6.1 [doi:10.21954/ou.rd.28045469].

The schemas are also available at: doi:10.21954/ou.rd.28044944 [path: /sim/schemas].

The following schemas are defined:

- `simulation-messages-result` – passed to the OpenAI `chat_completions` API tools function¹.
- `simulation-dramatis-personae` – describes our Dramatis Personae document. This describes the JSON document that is passed with the simulation request providing a list of virtual individuals to include in the simulation.
- `simulation-entities` – describes our Entities document. This JSON document passed with the simulation request provides a list of virtual entities to include in the simulation.

¹https://cookbook.openai.com/examples/how_to_call_functions_with_chat_models [<https://perma.cc/FY7T-WUNA>]

E.2 Simulated Content Prompting

Supplement S6.2 [doi:10.21954/ou.rd.28045469] contains an abridged Python code snippet of the awagdata function for generating simulated content from OpenAI using the /chat/completions API and the prompt template used by this code.

E.3 Dramatis Personae Document

Supplement S6.3 [doi:10.21954/ou.rd.28045469] shows some abridged examples of *dramatis personae* JSON documents used in our work. Full documents can be found at:

doi:10.21954/ou.rd.28044944 [path: /sim/persona-data/dramatis-personae]

E.4 Entities Document

Supplement S6.4 [doi:10.21954/ou.rd.28045469] shows some abridged examples of entities JSON documents. Full content can be found at:

doi:10.21954/ou.rd.28044944 [path: /sim/persona-data/entities]

E.5 Simulated Content Topic Examples

This section contains an abridged list of simulated content topics for personas that are referenced in Chapter 7. The full list of topics is documented in Supplement S7 [doi:10.21954/ou.rd.28045490], and is also available in JSON format at:

doi:10.21954/ou.rd.28044944 [path: /sim/persona-data/topics]

adam-work-company-announce

Corporate messenger application messages for the workplace of our fictional protagonist, Adam Macy. The messages are exclusively internal corporate announcements coming from the executive and senior leadership team of company that Adam works for, Borchester Software. Generate messages that come

from all of the executives in the provided list. Borchester Software is an international company with approximately 500 employees who are based mainly in the UK, USA, Canada, Japan and Germany. Announcements can include (but are not limited to) industry topics, company-wide meetings, reminders about business conduct rules, customer contracts and acquisitions, new internal IT systems, and senior personnel changes. Please ensure that each batch of generated content contains messages that cover the whole range of topics. While a list of some clients is provided, you should also generate fictional client names to reference in those messages that relate to clients. You should also generate fictional internal team names to mention where necessary. There must be variation in message topic and content; avoid re-using the same phrases. The tone of the messages should be relatively formal business English with British spellings.

adam-work-company-general

Corporate messenger application chat for the workplace of our fictional protagonist, Adam Macy. The chat is general workplace discussion within company that Adam works for, Borchester Software (sometimes referred to by its initials), which is an international company with approximately 500 employees who are based mainly in the UK, USA, Canada, Japan and Germany. Topics should include discussion about company announcements and products, upcoming events, business climate, general news (where appropriate for corporate discussion) and other items people might choose to share with a company of this size. This group chat is intended for discussions aimed at the whole company - it is not the correct place for team discussions, technical support requests, local office matters etc. Message content should be unique and not duplicated. While a list of some clients is provided, you should also generate fictional client names to reference in messages. The tone of the messages should be informal/business English with British spellings.

adam-work-team-manager

Corporate messenger application chat between our fictional protagonist, Adam

Macy and his manager Charlotte Walker. Adam works for Borchester Software (sometimes referred to by its initials) and is in a team of 10 people called the Client Technology Group (CTG), headed by Charlotte. The team is spread all over. This chat is just between Adam and his manager, and topics should include client discussions, logistics, and usual topics that a manager and their direct report might need to discuss. In this scenario, Adam is a strong performer at work, but has a high workload with a lot of customer travel commitments. While a list of some clients is provided, you should also generate fictional client names to reference in messages. Some of the messages should convey some sort of sense of urgency or need for a prompt response. The tone of the messages should be informal British English with British spellings.

adam-work-team-client

Corporate messenger application chat for the work team of our fictional protagonist, Adam Macy. The chat is workplace discussion between members of the same IT consulting team that Adam is in. The company that Adam works for is called Borchester Software (sometimes referred to by its initials) and he is in a team of 10 people called the Client Technology Group (CTG). The team is spread all over. Other chats exist for more general topics, but this chat is specifically about issues, questions and logistics relating to client work. For example, the team might talk about new client engagements, seek help with client technical problems or arrange meetings related to the clients. Some of the messages should convey some sort of sense of urgency or need for a prompt response. While a list of some clients is provided, you should also generate fictional client names to reference in messages. The tone of the messages should be informal British English with British spellings.

adam-family-group-chat

General chat between members of a fictional British family, as seen by our protagonist, Adam Macy. The chat is a lighthearted exchange between family members on a variety of topics, with some messages relating to logistics and meeting

up. Other topics might include village gossip, family personal and work happenings and milestones, and requests for assistance or favours, Adam and Ian live in the county town of Borchester in Borsetshire, while family are dotted around local villages Penny Hassett, Loxley Barrett, Darrington, Hollerton, Edgeley, Waterley Cross and Lakey Green. The main local city is the cathedral city of Felpersham. The tone of the messages should be informal British English with British spellings.

adam-friends-chat

General chat between a group of friends of our protagonist, Adam Macy and his partner Ian Craig. The chat is a lighthearted exchange between friends on a variety of topics that a group of young to middle aged adult friends might discuss. This would include but is not limited to jokes, meeting arrangements, discussion about what people have been doing, world affairs and local gossip. Adam and Ian live in the county town of Borchester in Borsetshire, while friends are dotted around local villages Ambridge, Penny Hassett, Loxley Barrett, Darrington, Hollerton, Edgeley, Waterley Cross and Lakey Green. The main local city is the cathedral city of Felpersham. Chat participants should be limited to only Adam, Ian and the supplied list of friends (not any family or work colleagues), although messages may refer to any individual. The tone of the messages should be informal British English with some light banter using British spellings.

adam-cycling-club-general

General chat between members of a fictional British cycling club the ‘Borchester Wheelers’ (also called BWCC). Club members mostly come from the fictional town of Borchester in Borsetshire, UK, or nearby. Topics generally relate to all aspects of cycling but in particular club rides, cycle sport, cycle commuting, and maintenance. For your reference, the club does its big cycle rides on a Sunday morning at 9am (people meet at the car park of a cafe called ‘Spoke & Brew’ as the club does not have a physical clubhouse). Popular destinations for club

rides include the villages/towns of Ambridge, Penny Hassett, Loxley Barrett, Darlington, Hollerton, Edgeley, Waterley Cross and Lakey Green, and sometimes as far afield as the cathedral city of Felpersham. Club rides could be of the sporting, social or coffee shop variety, and there is a monthly new joiners ride. Some members commute to work by bike and occasionally meet to do their commutes together. Doing laps of Borchester Park is a popular weekend and after work activity, although people often complain about the car traffic there. Many members also cycle on the virtual platform Zwift, although there are different chat groups dedicated to this. People will often discuss any professional cycle races that are happening at the time, as well as occasionally other sports such as rugby or football. People will sometimes chide members to stay on topic if they stray too far into other sports. Sometimes people will send messages reminding members to look at some topic posted on the club's web forum. There are some general jokes and light banter. The tone of the messages should be informal British English with British spellings.

E.6 Simulated Content Output Examples

Listing E.1 shows an example of some generated messages for a simulated school parents chat messaging group.

```
[  
  { "category": "usha-school-parents-chat",  
    "userid": "alanf",  
    "username": "Alan Franks",  
    "text": "Has anyone else's child complained about the maths homework being too difficult this term?  
           I'm wondering if it's just mine or a wider issue... ??"  
  },  
  { "category": "usha-school-parents-chat",  
    "userid": "fallonr",  
    "username": "Fallon Rogers",  
    "text": "Anyone up for a coffee morning this Friday? Could use a break from all this school talk  
           and catch up on some adult conversation!"  
  },  
  { "category": "usha-school-parents-chat",  
    "userid": "shulahl",  
    "username": "Shula Hebden Lloyd",  
    "text": "Heads up, the school's annual sports day is scheduled for June 15th. Always a fun day out  
           for the kids (and competitive parents)."  
  },  
  { "category": "usha-school-parents-chat",  
    "userid": "lizp",  
    "username": "Elizabeth Pargetter",  
    "text": "That coffee morning sounds lovely, Fallon. Count me in! Need to vent about the latest  
           homework saga..."  
  },  
  { "category": "usha-school-parents-chat",  
    "id": 17,  
    "userid": "harrison",  
    "username": "Harrison Burns",  
    "text": "Can we organize a carpool for the sports day? I can take three extra kids from the north  
           side of Felpersham."  
  }]  
]
```

Listing E.1: Simulation Messages Example for category 'usha-school-parents-chat'



Appendix F

Synthetic Evaluation Appendix

Additional content for the topics covered in this appendix can be found in Supplement S8
[doi:10.21954/ou.rd.28045547].

F.1 Static Prompt Elements

This section contains the static text elements that make up evaluation prompts for modes 1, 2 & 3. The prompt text is made up of common and mode-specific structures which are assembled into System and User messages for OpenAI as described in Section 8.3.3.

F.1.1 System Message – Common

This **system** message element is used for all modes as the common system message for all Awareness Agent evaluation calls. It is worded in such a way to be compatible with the different modes, by avoiding mode-specific phrasing.

AWAG_SYSTEM_MESSAGE_COMMON

You are the AwAg Evaluator. Your job is to role play a persona, for the purpose of evaluating a software system.

The software system that you are evaluating is intended to manage a user's incoming information from multiple sources - including work and personal - so that

the user is not overwhelmed or distracted. Consider it a personal information triage service. Your evaluation should focus on how well this system classifies the content that it processes and you will be asked to indicate your agreement or disagreement with the decisions.

An AwAg Evaluation Request is presented as a structured JSON document, and its main elements are:

1. Persona - the definition or ID of the persona that you should adopt when evaluating the request;
2. Perspective - the way in which you should consider the items in your evaluation;
3. Items - the actual items that you should evaluate.

The exact structure of the AwAg Evaluation Request may vary, but you will be given guidance on how to interpret it. You will also be told how to respond for each request. Possible response types include binary agree/disagree, or using a likert scale from 1 (completely disagree) to 5 (completely agree) as well as text describing your evaluation.

You will be asked to respond with your evaluation(s) of item(s) in a structured way. Your evaluation result should include your own evaluation, and also the value of the evaluated classification value or selection.

F.1.2 System Message – Extra for Mode 1

This **system** message is added to Mode 1 requests and is used to reference the Evaluation Request Schema [F.2.3] that Mode 1 requests conform to.

AWAG_SYSTEM_MESSAGE_EXTRA_MODE1

Information about the persona that you should adopt and the items that you must evaluate is provided as a JSON document - called the AwAg Evaluation Request JSON Schema (this schema has ID '<https://parse.net/awag/0.1/classification-evaluation-request.schema.json>'). Sometimes you may be given only a subset

of this document to evaluate or asked to respond in a different way. Evaluate every item and classification combination, returning a result for each one.

F.1.3 System Message – Extra for Modes 2 & 3

This **system** message is added to Mode 2 and Mode 3 requests and is used to describe the request structure in a narrative textual rather than schema-based manner. While it is named ‘AWAG_SYSTEM_MESSAGE_EXTRA_MODE2’, the same text is used for both modes 2 and 3 requests.

AWAG_SYSTEM_MESSAGE_EXTRA_MODE2

Information about the persona that you should adopt is provided as a JSON document. The ‘definition’ part of the document tells you about the age and gender of the persona, as well as other information about them. The ‘does’ property tells you what they do, both for work and social/personal activities. The ‘feelThinkBelieve’ parameter tells you what the persona feels, thinks and believes, telling you about their motivations and opinions. The ‘technologyExperience’ property tells you about the experience this persona has with technology. The ‘problems’ property tells you what sort of problems the persona might encounter, specifically in relation to managing information overload. The ‘needs’ and ‘existingSolutions’ properties tell you about the solution needs this persona has, and what existing solutions they have to try and address these.

Use the ‘perspective’ element of the request to tell you what approach to take to evaluating the items. For example, the perspective might ask you to determine if an item has been correctly classified.

The items to evaluate are provided to you as an array in the ‘items’ property of the request. Each item contains the following important information: ‘id’ contains the identifiers needed to uniquely identify this item, which you should use in your response for identification only. The ‘content’ property contains the actual textual content of the item that you should evaluate. The ‘classification’ property contains a JSON object that describes how the system that you are eval-

uating has classified the item - this contains the description of the classification, the available options that the system had to choose from (fromAvailableClassifications), and the option that it selected (classifiedAs). When evaluating an item, you should consider how well the selected classification matches the content of the item, and whether one of the alternative available options would have been a better fit.

F.1.4 User Message – Common Likert

This **user** message is added to all evaluation calls that require a Likert type response: specifically Mode 1 and Mode 2.

EVALUATION_USER_MESSAGE_COMMON_LIKERT

In this case, your evaluation should take the form of a Likert scale with a value range of 1-5. This describes how much your persona agrees with the classification that you are evaluating. If you strongly agree with how the system classified the item, you should use a likert value of 5. If you strongly disagree with the system's classification, you should use a likert value of 1. You should use values between this when you can't be as sure; for example a likert value of 4 indicates that you mostly agree with the classification chosen by the system you are evaluating. On the other hand, a value of 2 indicates that you would probably choose a different classification. If you don't have enough information to decide either way or are otherwise unsure, you should choose a value of 3. Your evaluation must also include explanatory text written as your persona. It is important that you use the Likert scale consistently and use the full range as needed, and that the scale value you choose is consistent with the text of your evaluation. You must only consider whether the selected classification is the most appropriate of the listed available classifications - do not suggest any alternative classifications that are not listed.

F.1.5 User Message Prefix – Mode 1

This is the first **user** message added to all Mode 1 evaluation calls. Because Mode 1 uses Likert responses, it is followed by EVALUATION_USER_MESSAGE_COMMON_LIKERT.

EVALUATION_USER_MESSAGE_MODE1

Please evaluate the following AwAg Evaluation Request. You should evaluate each item from each specified perspective, and return the results in the required result schema. For each item classification, consider the content, persona and perspectives. Be sure to evaluate the item for every classification and perspective combination.

F.1.6 User Message Prefix – Mode 2

This is the first **user** message added to all Mode 2 evaluation calls. Because Mode 2 uses Likert responses, it is followed by EVALUATION_USER_MESSAGE_COMMON_LIKERT.

EVALUATION_USER_MESSAGE_MODE2

Please evaluate the following AwAg Evaluation Request. You should evaluate each item for the specified persona and perspective, and return the results in the required result schema. Consider how item has been classified in the context of the persona and perspective.

F.1.7 User Message Prefix – Mode 3

This is the first **user** message added to all Mode 3 evaluation calls (which do not expect a Likert response).

EVALUATION_USER_MESSAGE_MODE3

Please evaluate the following AwAg Evaluation Request. You should evaluate each item for the specified persona and perspective, and return the results in the

required result schema. Consider how item has been classified in the context of the persona and perspective.

In this case, rate whether you as your persona AGREE or DISAGREE with the selected evaluation based on the list of available classifications, returning this in the evaluationAgreement property.

F.1.8 User Message Example – Mode 2

The example **user** message gives examples of Mode 2 requests and responses to guide the LLM how to interpret these and construct responses. Only modes 2 and 3 have this type of example provided, with Mode 1 relying on a more schema-based approach. The content can be found in Supplement S8.3.8.

F.1.9 User Message Example – Mode 3

The example **user** message gives examples of Mode 3 requests and responses to guide the LLM how to interpret and respond. The Mode 3 example text is identical to the Mode 2 version, except that it replaces evaluationLikert with evaluationAgreement and has accordingly different evaluationText responses that do not mention a likert value. The content can be found in Supplement S8.3.9.

F.2 Evaluation Schemas

These schemas are used for evaluation actions - i.e. requests to the evaluation LLM and the responses back.

F.2.1 Persona Schema

The Persona Schema can be found in Supplement S8.1.1. This schema is derived from our work on Persona development, as discussed in Section 5.3.

F.2.2 Perspective Schema

The Perspective Schema, which is used as part of Mode 1 evaluation requests, can be found in Supplement S8.1.2.

F.2.3 Evaluation Request Schema - Mode 1

The Classification Evaluation Request Schema can be found in Supplement S8.1.3; this schema is only used with **Mode 1** evaluation requests as discussed in Section 8.2.3.

F.2.4 Evaluation Request Schema - Modes 2 & 3

The Alternative Classification Evaluation Request Schema can be found in Supplement S8.1.4. While **Mode 2** and **Mode 3** evaluation requests conform to this schema, it is never used directly in the evaluation process.

F.2.5 Evaluation Result Schema - Mode 1

The Classification Evaluation Result Schema for Mode 1 evaluations can be found in Supplement S8.1.5. This defines the `get_evaluations` function passed to the OpenAI chat completions call using the Function Calling technique¹.

Note that `classificationName` in the schema is actually what we refer to as the Classification ID.

F.2.6 Evaluation Result Schema - Mode 2

The Classification Evaluation Result Schema for Mode 2 evaluations can be found in Supplement S8.1.6. This schema defines the Mode 2 version of the `get_evaluations` function².

¹https://cookbook.openai.com/examples/how_to_call_functions_with_chat_models [<https://perma.cc/FY7T-WUNA>]

²With hindsight, we should probably have differentiated the return function names for different modes to improve understandability

The Mode 2 result reflects the simpler nature of the Mode 2 and 3 request types compared to Mode 1. While Mode 1 has nested result arrays for the multiple Perspectives and Classifications passed in the request, Modes 2 & 3 results are a flatter array of items only, reflecting that only one Perspective & Classification are evaluated per item in those modes.

F.2.7 Evaluation Result Schema - Mode 3

The Classification Evaluation Result Schema for Mode 3 evaluations can be found in Supplement S8.1.7, and defines the Mode 3 version of the `get_evaluations` function.

Mode 3 differs from mode 2 only in that it contains a binary `evaluationAgreement` value rather than the `evaluationLikert` rating returned by modes 1 & 2.

F.3 Evaluation Schemas - Data Service

These schemas are used for communication with awagdata - for example to store/retrieve evaluation results.

F.3.1 awagdata Record Evaluation Data

The Record Evaluation Data Schema can be found in Supplement S8.2.1. This schema describes the content of the POST request to awagdata [6.8.3] (/data/eval) used to store the results of an evaluation query.

F.3.2 awagdata Record Evaluation Failure

The Record Evaluation Failure Schema can be found in Supplement S8.2.2. This describes the content of the POST request to awagdata [6.8.3] (/data/eval) used to store the results of a **failed** evaluation query. This contains the response in raw, unparsed text format, in case invalid JSON was returned from the API, and the IDs of Content Items in the evaluation query (bearing in mind that multiple CI's could be in any one query).

F.4 Evaluation JSON Examples

Example JSON requests and responses for modes 1, 2 & 3 can be found in Supplement S8.4.

F.5 Evaluation Processor

F.5.1 Evaluation Processor on Python

Supplement S8.5 contains an abridged illustration of Python code of the awagdata implementation of the Evaluation Processor. This code, which is intended to be run within a Flask environment, runs evaluations individually or in batch using evaluation items stored in awagdata at runtime by the Awareness Agent.

See the following related development log [Appendix D] entries:

- 2023-09-07: *Add support for tags and for recording/fetching raw evaluation items*
- 2023-09-12: *Add on-demand processing for evaluations based on recorded items*
- 2024-02-18: *Add Evaluation support to awagdata*

F.6 Fine-Tuning

F.6.1 Training Items

Examples of training items used in our fine tuning process can be found in Supplement S9 [doi:10.21954/ou.rd.28045562].

Appendix G

Study Appendix

G.1 Study Protocol

We produced a Study Protocol document to direct our study process. This was also provided to participants to inform them of the nature and process of the study. It is available at: doi:10.21954/ou.rd.28044944 [path: /study/docs/202312 Awareness_agent_study_protocol_v1.pdf]

Note that we referred to phases 1-7 in this study protocol, ranging from prep & planning to writing up. However, when handling data (i.e. tagging), we use the term 'phase' in a different way, with Execution Phase 1 being Data Collection & Model Training and Execution Phase 2 being Synthetic Evaluation and Participant Feedback. With hindsight, we should have used different terms in the protocol document to avoid confusion.

G.2 Persona to Agent ID Mappings

Each persona [5.3] studied was assigned a generated UUID, the Agent ID (Table G.1). This primary identifier of an Awareness Agent instance uniquely identifies it in study data [6.6.7].

Table G.1: Mapping of Persona Name to Agent ID

Persona Name	Agent ID
Susan	c93f166f-566c-4ec2-a2fc-5cccd450a624
Adam	45321a47-5fdd-4eda-b8c8-51bb719ce824
Phoebe	27fd8525-d4dc-4acb-ae0f-3d93c3f4344f
Kenton	97cb95a6-6878-47e6-9b88-1772a0b2607d
Usha	b04a80d1-4149-44ee-af36-989acf959222

G.3 Persona Configurations Spreadsheets

For each study instance, we set up a spreadsheet in Google Sheets to contain instance-specific configuration information such as UD-ML models, synthetic content topics and schedules, RSS data feeds and other information such as Awareness Agent commands. These are documented in Supplement S10.2 [[doi:10.21954/ou.rd.28045577](https://doi.org/10.21954/ou.rd.28045577)].

G.4 OpenAI Models Used

Instances of the study were run in two tranches [9.3.2], with some variation in OpenAI models used to reflect changes in model availability and data focus between tranches.

G.4.1 Tranche 1

Used for personas *Susan*, *Adam* and *Kenton*.

- **Synthetic Content:** gpt-3.5-turbo-1106
- **Synthetic Evaluation:**
 - **Vanilla 3.5:** gpt-3.5-turbo-1106
 - **Vanilla 4:** gpt-4-turbo
 - **Vanilla 4o:** gpt-4o-2024-11-20
 - **Fine-Tuning:** gpt-3.5-turbo-1106

G.4.2 Tranche 2

Used for personas *Phoebe* and *Usha*.

- **Synthetic Content:** gpt-4o-2024-11-20
- **Synthetic Evaluation:**
 - **Vanilla 3.5:** n/a
 - **Vanilla 4:** gpt-4-turbo
 - **Vanilla 4o:** gpt-4o-2024-11-20
 - **Fine-Tuning:** n/a

G.5 Evaluation Output Tags

The application of output tags [8.2.6] was designed to be consistent across all persona instances [G.4] to support cross-instance data analysis. Each tag is associated with an evaluation run using a specific mode [8.2.3] and OpenAI model [8.2.8]. The tags used are listed in Table G.2, which lists the tag strings, OpenAI model used and tranches [G.4] the tag was used for.

Note that in the evaluation application and source data, the tags are prefixed with “phase2-” (i.e. ‘phase2-vanilla-mode1-01’), indicating that evaluations are performed for phase2 input tags [8.2.6.1]. However, this text is redundant, as all evaluations used phase2 inputs, so this is omitted for brevity here and in most study analysis (i.e. ‘vanilla-mode1-01’ is the same as ‘phase2-vanilla-mode1-01’).

Table G.2: Evaluation Output Tags

Tag	Model	Mode	Usage	Note
vanilla-mode1-01	Vanilla 3.5	Mode 1	T1	Original version of Vanilla Mode 1 evaluation, superseded due to schema issue [9.6.2]
vanilla-mode1-02	Vanilla 3.5	Mode 1	T1	Replacement Vanilla Mode 1 evaluation with corrected schema
vanilla-mode2-01	Vanilla 3.5	Mode 2	T1	
vanilla-mode3-01	Vanilla 3.5	Mode 3	T1	
base-mode3-01	FT Base 3.5	Mode 3	T1	Base fine-tuned model based on gpt-3.5 [8.2.8.2.1]
ext-mode3-01	FT Ext 3.5	Mode 3	T1	Extended fine-tuned model based on FT Base 3.5 [8.2.8.2.2]
vanilla4-mode2-01	Vanilla 4	Mode 2	T1, T2	
vanilla4-mode3-01	Vanilla 4	Mode 3	T1, T2	
vanilla4o-mode2-01	Vanilla 4o	Mode 2	T1, T2	Retrospectively run for T1 alongside T2
vanilla4o-mode3-01	Vanilla 4o	Mode 3	T1, T2	As above

G.6 Instance Preparation

The following steps were used to initialise each study instance:

- Assign Agent ID [G.2]
- Create instance of each Agent in Linux test environment (deploy Java application to Tomcat) [6.8.1]
- Set up DNS and routing
- Set up study-specific email and browser profile accounts
- Create Slack application instances [C.2]:
 - Interact
 - Acquire
 - Simulate
- Create Slack workspaces and add Slack apps:
 - Awareness Agent (main)
 - * App: Interact
 - Awareness Agent (sim)
 - * App: Simulate
 - App: Acquire
- Set up spreadsheet for study configuration [G.3]
- Prepare Technical Details Sheet [G.7.4]

G.7 Information Provided to Participants

G.7.1 Participant Information Sheet

The primary information resource for participants was the Participant Information Sheet, which is available at:

doi:10.21954/ou.rd.28044944 [path: /study/docs/202401_study_participant_information_sheet.pdf]

This document explained the purpose of the study, information on what participation entails, and details of the expected stages to be undertaken. It also covered consent and privacy information, emphasising the voluntary nature of participation, data usage and retention, use of personally identifiable data and AI systems.

G.7.2 Participant Consent Form

The form used to obtain participant consent is available at:

doi:10.21954/ou.rd.28044944 [path: /study/docs/202401_awareness_agent_study_consent_form.pdf]

G.7.3 Persona Details

We provided each participant a copy of the relevant PATHY persona for the study instance – see Appendix B.2.

G.7.4 Technical Details Sheet

A Technical Details Sheet was generated for each study instance, containing instance-specific information such as URLs, IDs and authentication. This was used both to initialise the study and inform the participant, and can be found in Supplement S10.1

[doi:10.21954/ou.rd.28045577].

G.7.5 Introductory Presentation

We produced a Powerpoint/PDF presentation to give some general project motivation and background, which is available at:

doi:10.21954/ou.rd.28044944 [path: /study/docs/202401_awareness_agent_study_intro.pdf]

G.7.6 Study Protocol

We provided a copy of the Study Protocol [G.1] to participants as background information for those interested in knowing more.

G.8 Evaluation Token Usage

The statistics for OpenAI token usage in evaluations can be found in Supplement S11.3 [doi:10.21954/ou.rd.28045580]. This data was recorded by the evaluation processor components for all evaluations carried out during the research for the listed tags. This will broadly correspond to the evaluations used in the studies, but in the case where some evaluation was run multiple times with a matching tag, that data will be included – which will not necessarily form part of the study dataset.

Each time an evaluation was run, token usage for Prompting and Completions as reported by the OpenAI API was recorded against the relevant agent/tag combination. Items Processed in this context refers to a single Content Item evaluation for each mode.

We can see that although Mode 1 uses a more complex prompting structure in terms of the JSON pushed to the LLM, it uses about half of the prompt tokens that Mode 2 and Mode 3 do. This is due to the other information that is included with the prompt requests for those modes – specifically, modes 2 and 3 provide example evaluation requests and results with each request, adding considerably to the input token count. However, output token count is lower for modes 2 & 3, reflecting their simpler structure.

Appendix H

Supplementary Materials Appendix

The resources of for this project are available as a collection at: doi:10.21954/ou.rd.c.7588025.

A consolidated document containing all of the supplements listed below is available at:
doi:10.21954/ou.rd.29293850.

S0 Index

doi:10.21954/ou.rd.28045100 Standalone index of supplementary materials documents.

S1 Design and Development Log

doi:10.21954/ou.rd.28045163 Unabridged version of the Design and Development Log,
providing supporting material for the RtD process.

S2 Survey Questions

doi:10.21954/ou.rd.28045166 List of all questions from the survey in Chapter 5.

S3 Survey Demographics

doi:10.21954/ou.rd.28045442 Information on key demographics of respondents to the
Chapter 5 survey.

S4 Final Personas

doi:10.21954/ou.rd.28045454 PATHY Personas that were derived using the process covered in Chapter 5.

S5 Persona Scenarios

doi:10.21954/ou.rd.28045460 Persona Scenarios that have been written for the personas developed in Chapter 5.

S6 Synthetic Content Materials

doi:10.21954/ou.rd.28045469 Information relating to Chapter 7 Synthetic Content, including schemas and example data for content requests and simplified Python code example.

S7 Synthetic Content Topics

doi:10.21954/ou.rd.28045490 Full list of simulated content topics for personas as referenced in Chapter 7.

S8 Synthetic Evaluation Schemas & Prompting

doi:10.21954/ou.rd.28045547 Details of prompting resources used in the Synthetic Evaluation process discussed in Chapter 8.

S9 Synthetic Evaluation Fine Tuning Examples

doi:10.21954/ou.rd.28045562 Examples relating to fine tuning within the Synthetic Evaluation process.

S10 Study Configuration & Information

doi:10.21954/ou.rd.28045577 Study configuration and technical details sheet provided to study participants.

S11 Study Detailed Results

doi:10.21954/ou.rd.28045580 Detailed results data for the Chapter 9 study.

S12 Source Code & Additional Materials

doi:10.21954/ou.rd.28045598 Information on source code for software used in this research, and additional materials.