Agents United: An Open Platform for Multi-Agent Conversational Systems

Tessa Beinema[1,2,*], Daniel Davison[1,*], Dennis Reidsma[1,*], Oresti Banos[1,3], Merijn Bruijnes[1,4], Brice Donval[5], Álvaro Fides Valero[6], Dirk Heylen[1], Dennis Hofs[2], Gerwin Huizing[1], Reshamashree B. Kantharaju[5], Randy Klaassen[1], Jan Kolkmeier[1], Kostas Konsolakis[1], Alison Pease[7], Catherine Pelachaud[8], Donatella Simonetti[1,5], Mark Snaith[7,9], Vicente Traver[6], Jorien van Loon[1], Jacky Visser[7], Marcel Weusthof[1], Fajrian Yunus[5], Hermie Hermens[1,2,†], and Harm op den Akker[1,2,10,†]

[1] University of Twente, Enschede, the Netherlands
[2] Roessingh Research and Development, Enschede, the Netherlands
[3] University of Granada, Granada, Spain
[4] Delft University of Technology, Delft, the Netherlands
[5] Sorbonne Université, Paris, France
[6] Universitat Politècnica de València, València, Spain
[7] University of Dundee, Dundee, United Kingdom
[8] CNRS-ISIR, Paris, France
[9] Robert Gordon University, Aberdeen, United Kingdom
[10] Innovation Sprint, Brussels, Belgium

Figure 1: A group of ASAP (1, 3, 5) and GRETA (2, 4) agents in the same Unity environment. A user can participate in a multi-party group conversation by selecting a response from the menu on the right (in the same interface or on a separate device).

ABSTRACT

The development of applications with intelligent virtual agents (IVA) often comes with integration of multiple complex components. In this article we present the Agents United Platform: an open source platform that researchers and developers can use as a starting point to setup their own multi-IVA applications.

The new platform provides developers with a set of integrated components in a *sense-remember-think-act* architecture. Integrated components are a sensor framework, memory component, Topic Selection Engine, interaction manager (Flipper), two dialogue execution engines, and two behaviour realisers (ASAP and GRETA) of which the agents can seamlessly interact with each other.

This article discusses the platform and its individual components. It also highlights some of the novelties that arise from the integration of components and elaborates on directions for future work.

CCS CONCEPTS

• **Human-centered computing** → **Natural language interfaces**; **Systems and tools for interaction design**; User interface toolkits; • **Information systems** → Open source software.

KEYWORDS

Open source platform, multi-agent system, social conversational agents, coaching

IVA '21, September 14–17, 2021, Virtual Event, Japan

ACM ISBN 978-1-4503-8619-7/21/09.
https://doi.org/10.1145/3472306.3478352

*Shared first authors & corresponding authors:
{t.c.beinema, d.p.davison, d.reidsma}@utwente.nl
†Shared last authors

ACM Reference Format:
Tessa Beinema, Daniel Davison, Dennis Reidsma, Oresti Banos, Merijn Bruijnes, Brice Donval, Álvaro Fides Valero, Dirk Heylen, Dennis Hofs, Gerwin Huizing, Reshmashree B. Kantharaju, Randy Klaassen, Jan Kolkmeier, Kostas Konsolakis, Alison Pease, Catherine Pelachaud, Donatella Simonetti, Mark Snaith, Vicente Traver, Jorien van Loon, Jacky Visser, Marcel Weusthof, Fajrian Yunus, Hermie Hermens and Harm op den Akker. 2021. Agents United: An Open Platform for Multi-Agent Conversational Systems. In *21st ACM International Conference on Intelligent Virtual Agents (IVA '21), September 14–17, 2021, Virtual Event, Japan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3472306.3478352

1 INTRODUCTION

Developing intelligent virtual agent (IVA) systems can require complex integration of many state-of-the-art technical components, especially when developing for long-term social interactions with multiple virtual agents. This article presents the Agents United Platform[1], a novel open source platform that provides researchers and developers with an integrated setup for the creation of long-term multi-IVA applications.

The platform was the result of technical development during the Council of Coaches project [19]. In this project, a health coaching application was built that offers coaching by a group of virtual agents. Users could engage in multi-party group discussions with these agents, receiving advice and information on various aspects of healthy living (also see Figure 1). Each agent in this council had their own personality, backstory, and expertise (e.g., physical activity or nutrition). Throughout this paper, we will provide examples in the context of this health coaching use case to illustrate the discussed concepts and technical explanations[2]. The application was built on a number of state-of-the-art components from the field of intelligent virtual humans and dialog systems and turned into a versatile, reusable and standalone platform. To support and coordinate the activities with respect to the platform, such as maintenance, future development, community building and documentation, the Agents United Alliance has been formed.

This paper presents the platform. We begin this article with an overview of its global architecture, followed by a discussion of its components and the advances made on individual components. Next, we discuss contributions through the integration of components, and conclude with expected future developments and directions for the platform.

2 THE AGENTS UNITED ARCHITECTURE

The architecture for the Agents United Platform follows the well-known *sense*, *remember*, *think* and *act* paradigm (see Figure 2) for intelligent systems that has been used for decades. The *sense* layer contains sensing and interpretation of human behaviour and health related parameters, the outcomes of which is stored in a central Shared Knowledge Base (the *remember* layer). The *think* layer contains the interaction management, such as management of coaching topics, content, and the social dialogue used to deliver the content. The *act* layer contains the virtual embodiment of the agents as well as the user GUI (for presenting response options to the user and for presentation of additional information) and the Wizard of Oz GUI (for presenting the controls with which the Wizard can select specific agent moves in experiments). As indicated in Figure 2, components in the architecture are or can be distributed over various types of devices. Communication with components on servers is performed through RESTful interaction, while communication between components on the experimenter's device communicate over an ActiveMQ middleware layer[3].

While individual components are presented in detail in their respective publications, the following subsections provide a summary of each component, and its novelty and role in the platform.

2.1 Holistic Behaviour Analysis Framework (HBAF): From Sensing to Interpreting Users' Behaviour

A virtual agent system should take users' personal situation into account during interactions, for example, by using sensor data collected in the users' daily life. For a coaching application as developed in the Council of Coaches project, a valuable starting point for the conversations is the users' behaviour. Ideally, for a health coaching application an holistic profile would be available. That is, a broad picture of the user as a whole, that includes behavioural aspects on multiple domains. In our use case, domains for which behaviours were measured included physical activity and social interaction. This information was gathered using a smartphone, wearable, a weight scale and a blood pressure measuring device.

The Holistic Behaviour Analysis Framework (HBAF) focuses on sensing and profiling user's context by combining smart multimodal sensing technologies in order to model users' behaviour in a comprehensive way [11]. More specifically, the framework serves as the principal generator of knowledge on a user's behaviour over different periods of time through modelling, inferring and combining human behaviour in a holistic way, including physical, social, emotional and cognitive domains [12].

The HBAF deals with heterogeneous data from multi-modal devices, such as smartphones, activity trackers and UniversAAL-enabled devices [31] – a standard that was originally co-developed by one of the Alliance partners and previously applied by them as input paradigm in healthy living scenarios [20]. In the COUCH demonstrator, UniversAAL connects a weight scale and a blood pressure measuring device, but it also supports a variety of smart home devices. This data is securely gathered, after which selected views and summaries are generated. Only this 'derived content' is accessible for the rest of the platform, ensuring that users' sensitive data are exclusively stored in the HBAF. The HBAF detects the user's physical (e.g., walking), social (e.g., having a conversation), emotional (e.g., feeling happy), and cognitive (e.g., reading a book) behaviours. These classes of information have been selected for their potential to ground opportunities for behaviour change through coaching conversations, although the HBAF's architecture is not necessarily limited to those classes. Furthermore, behavioural data is analysed at different time scales, affording the detection of short-term behaviours (e.g., sitting on a chair), long-term behaviours (e.g., being sedentary) and changes in these behaviours over time.

[1]Agents United Platform: www.agents-united.org
[2]A video showing the use case and interaction can be found at: www.youtube.com/watch?v=lOOEs1OWLkM
[3]ActiveMQ: https://activemq.apache.org/

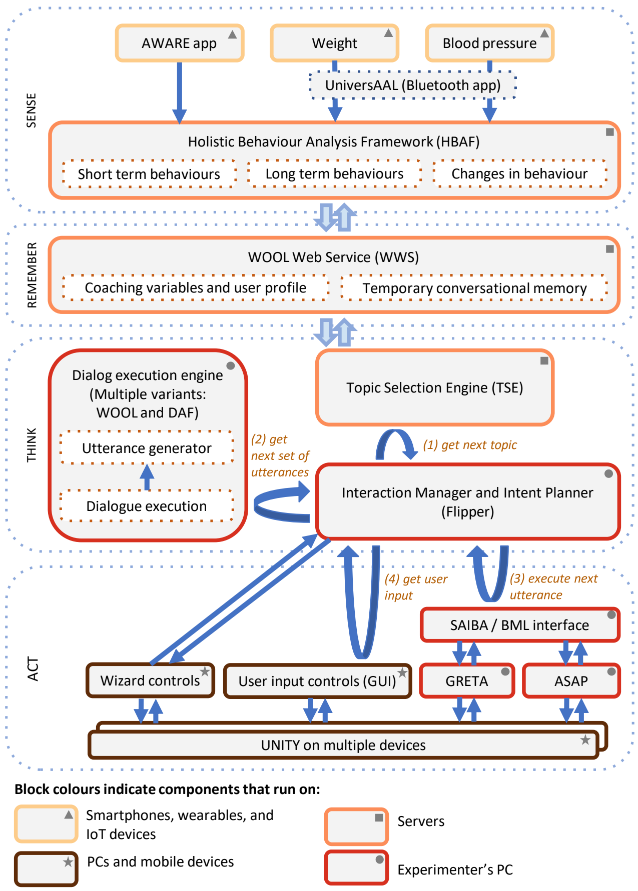

Figure 2: The Agents United architecture with its *sense*, *remember*, *think* and *act* layers. Note that the Holistic Behaviour Analysis Framework (see Section 2.1) and all *think* components can access the WOOL Web Service as it is the Shared Knowledge Base (see Section 2.2), but we have minimised those arrows for legibility. We elaborate on the numbered steps (1-4) in Section 2.4 (Flipper).

2.2 WOOL Web Service: Shared Knowledge Base With Access Authorisation

The WOOL Web Service[4] fulfils the role of the Shared Knowledge base (SKB) in the Agents United Platform. As the central 'memory' of the system it stores all necessary information and makes it accessible to all other modules in the platform. This does not only involve immediate memory (at the level of the ongoing conversation), but also longer term memory of sensor data, personal characteristics, coaching goals, and a conversational history per user that spans across recurring interactions.

In our use case, a naming scheme for variables is used to store information and distinguish between information that should be remembered between dialogues (e.g., dialogues that have been completed or a goal that has been set) and information that is stored and used during execution of a dialogue (e.g., which paths through the current dialogue have been completed). Variables in the last case will be overwritten when the dialogue is held again, whereas other data is retained longer term; also some variables are stored as histories or time series data whereas for other variables only the most recent value is retained.

Other examples of such a database exist, there are many more agent applications with some kind of memory storage (e.g., [9, 25]), but together with the middleware communication protocols employed between components in the Agents United Platform, the SKB is crucial for a fluent information flow between system components, which can be facilitated by a consistent naming scheme.

2.3 The Topic Selection Engine

The Topic Selection Engine (TSE) decides which topic is the subject of the conversation that is to be started, which allows for tailoring the conversational (coaching) strategy and content on a high level (following [1]). This way, the TSE ensures that the agents will discuss a relevant topic with the user. For example, in our use case, the coaches will introduce themselves to the user before they move on to coaching topics such as setting a new physical activity goal.

The engine uses topic trees (one for each conversational agent) that embody a taxonomy of the conversational topics. In our use case, each agent is the expert on one coaching domain (e.g., physical activity or nutrition) and the TSE tree for an agent includes topics that can be subject of a coaching dialogue for that agent's domain (e.g., introduction, goal-setting, or provide information). Topics in the tree may be more or less relevant during the interaction, based on available information such as users' current status (e.g., from sensor data), interaction history (e.g., previously addressed topics and conversation outcomes), and preferred coaching strategy (e.g. from the set of strategies in [2]). Examples of rules to compute the relevance of a topic are conversational (e.g., 'start with an introduction') or task related (e.g., 'discuss goal-setting' when no goal has been set yet or 'increase relevance for inform topics' when the strategy of the agent is health education). Selection of the most relevant topic uses the weighted average of selection parameters that are added to each topic. These parameters represent partially predetermined information ('how relevant is discussion of the topic') and their values are adapted dynamically based on available information (all retrieved from the SKB).

Upon request by the interaction manager (see Section 2.4) the TSE selects the most salient topic from the tree (as shown in Figure 3). It returns a topic message which contains an indication of the general topic (e.g., 'goal-setting'), information about the speakers involved and parameters specifying content (e.g., 'a preferred goal would be about a new daily step count target'). This information is used by other modules in the *think* layer to generate actual dialogue.

This TSE is one of the first mechanisms that allows automated multi-topic coaching, where the advice and delivery can be made dynamic, personalised, and sensor based, which are particularly relevant aspects for our use case of health coaching. State-of-the-art work [3, 5, 26] tends either to focus on how to present coaching dialogues (with a set order of topics), or to be more about the notification-style delivery of pre-defined coaching content (through text messages or smartphone notifications) [8, 18].

[4]The Web Service is part of the WOOL Dialogue Platform: www.woolplatform.eu

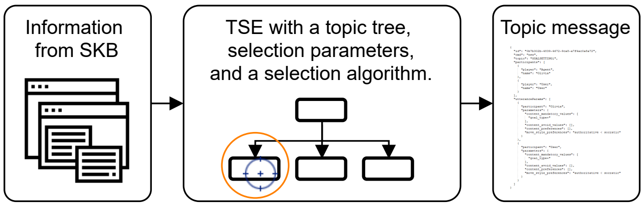

Figure 3: A schematic representation of the topic selection process. The TSE uses information stored in the WWS to select a relevant topic and return a topic message.

2.4 Flipper: Interaction Management and Conversational Intent Planning

The Interaction Manager is the conversationally-aware glue between various components, providing the 'beat' to which the other components in the *think* layer run. The Interaction Manager has been built using the Flipper system [32, 33], an information state based conversation engine that choreographs other modules involved in producing the conversation. Addressing a single topic can involve a complex dialogue of moves, and a single 'move' utterance might require multiple social conversational moves (including turn-taking and back-channelling), to be delivered properly. Flipper is used to orchestrate this process and, while doing so, keeps track of when moves have been delivered, whether they were interrupted, or not performed. In our use case, for example, when setting a physical activity goal, we want to know when certain coaching information is delivered and when the user has agreed on a certain goal (e.g. '7,500 steps'). This process of negotiating a suitable goal requires choreographed conversational moves between an activity coach and a social coach. The latter may respond to the activity coach with statements such as "Group activities are a great way to motivate you to reach your physical activity goal."

As interaction manager, Flipper communicates with most other modules as illustrated in Figure 2: Flipper (1) requests, from the TSE, new topics to be addressed; (2) controls the dialogue engine to successively determine possible next conversational moves and generate corresponding multi-modal utterances using information from the SKB; (3) requests the SAIBA behaviour realisers to produce selected utterances on the multi-agent embodiments; (4) receives user responses via the GUI, which it communicates back to the dialogue engine. This process allows for a separate definition of structural (dialogue) content and social conversation turn-taking management, which is a novelty of the integrated platform on which we will elaborate in Section 3.1.

2.5 WOOL & DAF: Dialogue Modelling and Execution Engines

The Agents United Platform features two variants of dialogue execution engines: WOOL, which models dialogues as plain transition graphs, and the state-of-the-art Dialogue and Argumentation Framework (DAF), which models dialogues as structured argumentation. In our use case, scripted linear dialogues (such as introductions or casual conversations) were modelled with WOOL, whereas dynamic dialogues (such as setting and reflecting on an activity goal) made use of argumentation and reasoning offered by the DAF.

The engines model the structure of dialogues that cover a topic (as selected by the TSE); and for every step in the dialogue they generate an appropriate description of a next utterance that one of the agents can deliver and a set of possible responses for the user. Determination of a next dialogue step and the content of utterances may be influenced by information in the SKB. A dialogue progresses to a next utterance whenever the interaction manager requests the next dialogue step (e.g., because it received a user response). The execution engines can also return instructions to the interaction manager to store information in the SKB upon completion of a specific move (e.g., 'this dialogue has been completed' or 'such-and-such goal has been set').

An overall novelty of these integrated execution engines is that their dialogues are implicitly modelled as being part of a larger topic structure that is navigated by the TSE, and the interaction manager can switch between dialogues (and engines) during the overall conversation.

2.5.1 WOOL: Labelled transition graphs as dialogue content. WOOL[5] is built as an adaptation of the Yarn language[6] and models dialogues as transition graphs that link agent statements and user replies. This dialogue engine and language provide a lightweight way to, for example, include an initial version of dialogues for testing, or to include single use dialogues (such as the scripted first introductions in our use case). The editor supports the dialogue development process through an intuitive GUI for authoring and testing dialogues. Once a dialogue has been started with the user, the user's response to a statement by an agent determines what the next agent-statement should be. This process is repeated until the end of the dialogue has been reached.

2.5.2 Dialogue and Argumentation Framework: Argument driven dialogues. The Dialogue and Argumentation Framework (DAF) (an implementation of the Platform of Argument and Dialogue [30]) has been built on top of the Dialogue Game Execution Platform (DGEP) [4] and the Dialogue Utterance Generator (DUG) [27]. DGEP is used to manage the dialogue through *dialogue games* that define and regulate on an abstract level how a dialogue can unfold. These dialogue games are specified in the Dialogue Game Description Language (DGDL) [14, 38] and describe in terms of dialogue acts which types of move are permitted and their allowed order (e.g., a response to a 'question' can be an 'answer' or another 'question'). In the Agents United Platform, DGEP has been initialised with models of the abstract structure of coaching (sub)dialogues (for an example see the 'goal-setting' dialogue game discussed in [29]). These dialogue games model dialogues in terms of move types that can follow each other while discussing a specific type of topic. Move types can be, for instance, question, answer, argument, counterargument, propose, accept and elaboration. Dialogue games may be different for (sub)topics as we learned from analysing conversations between multiple health professionals and patient-actors (the Patient Consultation Corpus [28]). Upon request by the interaction manager, DGEP will initiate one of these dialogue games, and successively

[5]WOOL Dialogue Platform: www.woolplatform.eu
[6]Yarn Spinner: https://yarnspinner.dev/

```
"set_{{currentGoal}}Goal(X)": [
  {
    "text": "Should we start with a {{currentGoal}} goal of $X
    steps?",
    "properties": []
  },
  {
    "text": "I would like to suggest a {{currentGoal}} goal of
    $X steps per day. Is that OK?",
    "properties": [
      "previousmove:rejectgoaltoohigh"
    ]
  },
```

Figure 4: An example of two dictionary entries in the DUG. Note that the terms between double curly brackets are filled by content from the SKB (in this case 'short-term' or 'long-term') and that '$X' is replaced by a value assigned in the logical content (e.g., 10,000 steps).

offer a list of all potential 'next moves in the game' for each of the agents in the conversation as well as for the user response options.

The abstract moves returned by DGEP need to be filled with content. For example, an abstract 'propose a new goal' move requires knowledge of what type of goal is to be proposed, and the words of the utterances (i.e., there are many ways to phrase a proposal for a goal of '1000 more steps per day'). Filling in the moves is done by the Dialogue Utterance Generator (DUG). The DUG instantiates the moves with content and filters them based on strategic and other factors. It has two components: argument models (defined using ASPIC+ [21]) and a dictionary. The argument models use inference rules to deduce possible move content (using information from the SKB), and the dictionary then maps these logical representations to utterances. For example, in our use case, the expression $set\_step\_goal(10,000)$ can be presented to the user as 'How about we set you a step goal of 10,000 steps?' or 'A step goal of 10,000 steps would be really good for you - what do you think?'. Utterances in the dictionary are marked-up with various annotations that can be used to select a suitable phrasing (e.g., to represent delivery styles such as 'friendly' or 'authoritative'; or content such as 'proposing a goal' or 'about step goals'; see Figure 4). The DUG also fills in slots in the utterances with user-specific SKB data (e.g., the user's name, or the last amount of steps that was measured).

2.6 GRETA & ASAP: Multi-Modal Behaviour Realisers

The setup of the Agents United Platform ensures that conversational content can be delivered via any SAIBA-compliant multi-modal behaviour realiser [13]. Currently, two embodied virtual agent platforms are fully integrated: GRETA [17] and ASAP [37]. In our use case, various agents from both platforms (as also previously shown in Figure 1) delivered the coaching content in an integrated virtual environment, where agents held conversations between each other and the user.

2.6.1 Virtual Embodiments via the GRETA Platform. Of the two virtual agent platforms, GRETA [17] is the most advanced with respect to the autonomous nonverbal behaviour generation that accompanies agents' utterances. The GRETA agents' behaviours are driven at three different levels: first, by indicating its communicative intentions using the Function Markup Language FML-APML [16]; second, from a semantic analysis of the agent's utterance [22]; and third, from the prosody [41]. The interaction manager (Flipper) outputs the communicative intentions. The other two methods then compute automatically where to place a communicative gesture. GRETA relies on the representation of image schemas [7] to compute the shape of a gesture. Such a representation captures the conceptual meaning of a text in terms of objects and actions [7] which are mapped to a physical gesture shape [23]. Having computed the different multi-modal behaviours, GRETA relies on its behaviour scheduler to organise them sequentially [15]. Finally, it uses the notion of Ideational Units [6] to co-articulate gestures within each other [40].

We adapted the open source GRETA platform in two ways, (1) allowing multiple agents in one environment, and (2) importing into Unity, which also allows for integration with ASAP (see Section 3.2).

The inclusion of multiple agents in one environment required agents to be added as independent entities. Therefore, we added a hierarchical structure to GRETA. At the highest level, we added an environment manager that knows about the objects and their location. It can contain various objects, in particular agents. Each agent is controlled by its own 'CharacterManager' module that contains its information: the 3D aspect, voice, behaviour lexicons and baseline [15]. Feedback mechanisms ensure the information flow between the components. They allow an agent to know where another agent is located and what it is doing.

Furthermore, we have integrated GRETA within Unity[7]. The integration was done using a TCP-IP message controller where Unity acts as the server and GRETA as the client. As soon as an animation has been computed within GRETA, it sends it frame by frame to Unity. Such an integration allows for the use of all graphic functionalities of Unity and the behaviour controller of GRETA.

2.6.2 Virtual Embodiments via the ASAP Platform. The other virtual agent platform that we incorporated, ASAP, is a behaviour realiser with a strength for choreographing and realising multi-modal behaviours of multiple agents [34, 35, 37]. It supports visualisation in the Unity game engine [10] and has been developed specifically for adaptive timing and mutual coordination of behaviour between multiple agents and between agent and user [24, 36].

We extended upon ASAP in two ways. Firstly, we integrated the feedback channels from GRETA into standard BML patterns which allows ASAP agents to seamlessly respond to GRETA agents. This is done in almost equally flexible and adaptive ways as was already present for ASAP [36]. Secondly, integration with Unity was extended in a way that allows not only multiple ASAP agents on one device (as was already possible), but even a mix of agents (ASAP and GRETA) embodied on multiple devices. Since this is a novel integration between multiple components in the platform we discuss this in further detail in Section 3.2.

3 NOVELTY IN INTEGRATED PLATFORM

The Agents United Platform seamlessly integrates the previously discussed components. In the following subsections, we highlight

[7]Unity3D: www.unity.com

some of the innovations that result from and are supported by this integration.

3.1 Mixing Structural and Social Models of Conversational Dialog

A first advance of the state-of-the-art by the platform lies in combining expertise, models, and insights from the fields of argumentation and social conversation. In the Agents United Platform, each component in the *think* layer expands a selected topic into more detailed steps. Ultimately the execution of a single utterance is orchestrated by the interaction manager, which may involve multiple social conversational moves that include turn-taking, back-channelling, and detection of completion or interruption.

The novelty of this integration between components in the system is the way that they work together to move from models of coaching topics and content, to structural models of dialogue to determine the content of successive moves in the conversation, via models of social conversational intents that are needed to deliver a single piece of content, to actual delivery of the content in the form of agent behaviour. By giving the interaction manager the task of choosing among all possible next moves from multiple agents as determined by the dialogue model, the social conversational turn-taking management can be defined by a separate set of rules – that is, separate from the content modelled by the dialogue engines and Topic Selection Engine. This not only allows for specialists to define those respective models and rules (e.g., by letting them focus on dialogue design or social behaviour), but also means that any additional device can be added to serve as an input for user responses or Wizard of Oz commands (e.g., to move the dialogue forward).

3.2 Multi-Platform Multi-Engine Multi-Device Multi-Agent Setups

A second advance of the state-of-the-art by the platform is that it allows for integrated use of both GRETA and ASAP, two platforms for realising agent behaviour. Agents from both platforms can be included in one Unity scene, but they can also be potentially spread out over any number of devices (as long as these devices support Unity) while remaining part of the same interaction.

All agents in an interaction are driven by one single instance of the multi-agent ASAP planner and realiser, which makes cross-device coordination between agents, user reply interfaces, and Wizard of Oz interfaces possible (see Figure 5). Thanks to the integrated intent planning and the fact that agents have access to information in each other's feedback channel, the agents of the two engines are aware of each other's presence in the conversation and are responding to each other; not just verbally, but also in their autonomous social conversational nonverbal cues (e.g., gaze).

3.3 Conversations With the Coach-as-a-Sensor

A third novel aspect of the integrated system is the potential to perform measurements and behaviour analysis using the 'Coach-as-a-Sensor' concept. Some types of behaviour (such as emotional experiences) are difficult to accurately measure with sensors – especially unobtrusively in an daily life context. The setup of the integrated platform is ideally suited for in-dialogue measurements. While similar to the experience sampling methodology (where users receive notifications for questions), embedding questions in a natural conversation with an agent would make these measurements less obtrusive. The collected information is then stored in the SKB and can be retrieved by the HBAF (e.g., for use in behaviour analysis as in Wohlfahrt-Laymann et al. [39]). Dialogues can be designed specifically with this goal in mind, making use of questions for psycho-social assessment to support more complex analysis of the social and emotional state of users based on the combination of sensor data and self-reports through the Coach-as-a-Sensor.

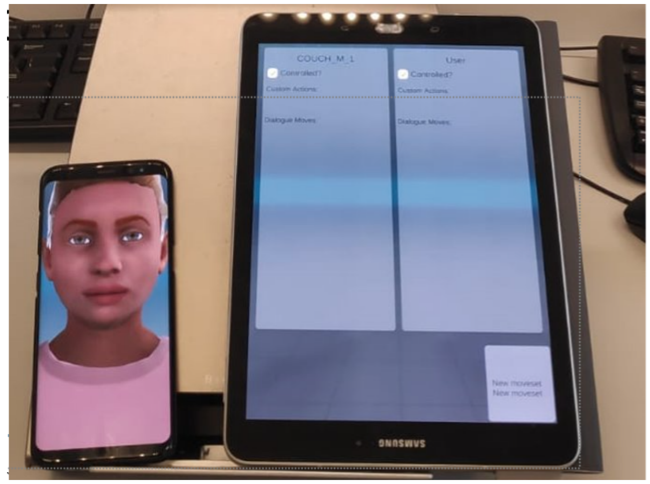

Figure 5: An agent and Wizard of Oz interface deployed across multiple devices.

While conversational systems typically have ways to retrieve and store information about the user as it arises through the interaction, the setup described here offers infrastructure that is geared towards interpreting such data on the sensing layer, closely embedding it in HBAF models and driving new insights.

3.4 Tailoring Interaction Strategies & Styles

A fourth aspect that is well supported by the integrated platform is the potential to tailor not just conversational content, but also the underlying interaction strategies and styles. For example, in our health coaching use case, discussing the relevant topics and providing tailored information from a content perspective are important. But the manner in which this content is discussed is also an important factor in keeping the user engaged and motivated to work on their behaviour. Adjusting the underlying coaching strategy (e.g., goal-setting or health education) and style (e.g., authoritative or socratic) is well supported by the integrated platform.

Whether these strategy and style choices are made implicitly (deduced by the system) or explicitly (chosen by the user or discussed between user and coach), the architecture can directly adapt the resulting interaction accordingly. Information on strategy and style can be stored in the SKB and shared between components. The Topic Selection Engine can use this information in its topic selection process and include parameters indicating delivery style preferences in its topic messages. This allows for the dialogue engines to either play the suitable scripts (WOOL) or to take these preferences into account during execution of the abstract dialogue games and utterance selection (DAF). Finally, the behaviour realisers are able to generate matching non-verbal behaviour.

4 FUTURE WORK

Given where the platform stands now, new questions arise. In this section we discuss a few directions for future work. A first direction for future work is in the further development of *group-aware* (semi-)autonomous nonverbal agent behaviours. We already implemented rudimentary automatic conversational gaze behaviour for the agents in the integrated platform. This gaze behaviour takes into account which agent is speaking and whom they are addressing. However, this behaviour does not take into account dialogue content, nor group dynamics including cohesion, leadership, relationship between participants, etc.; it just looks at the activity in the behaviour realiser feedback channels. In future work we could now investigate and further develop the generation of such group social behaviours. This could be in terms of non-verbal behaviour of individual agents in a group context, but also improved turn-taking management between agents.

A second direction would be to improve functionality to support the addition of new agents and content. In the integrated platform, topics are mapped to dialogues and specific agents that fulfil certain roles in those dialogues. Automating this process of casting agents for these roles would make the system more flexible. Future work could therefore focus on the development of a 'casting director' component for this purpose. Furthermore, it could investigate 'import functionality' for new agents with their own expertise – for example, from other developers through a marketplace. Such imported agents, or updates to content in general do come with their own challenges. You would want the content to be coherent, especially for use over a longer period of time and want to avoid scenarios in which the agent, for example, suddenly gives diametrically opposite advice compared to previous advice.

A third direction for future development would be more fine-grained access authorisation. In the current setup of the platform, all components can access the Shared Knowledge Base. Even though the Holistic Behaviour Analysis Framework already provides a layer of data security by only sending on 'derived conclusions' instead of all raw data, not all stored information is relevant for all components. While currently all content and components are added by those developing an application, if a some point it becomes possible to import new content or agents this security will be important. Furthermore, this could also mean that a method for gathering and revising consent – for example, through conversations between the agents and the user – would have to be devised to support access to specific (sensor) data by specific components.

5 RELEASING THE PLATFORM AND FUTURE COMMUNITY BUILDING

The Agents United Platform has been released open source under a LGPL v3.0 license. The platform's website[8] provides an overview and showcases the platform; the platform's Git repository[9] hosts the latest release and a demonstrator that showcases a potential application. It also contains documentation for the platform as a whole and its components, and tutorials[10] that explain how to setup and develop your own application using the platform.

To support and coordinate the activities with respect to the platform, such as maintenance, future development, community building and documentation, the Agents United Alliance has been formed. While the website provides more details on the Alliance and the Platform, we will summarise some key points here.

The agenda for further development of the platform is set by members of the Alliance together with the community. Initially, members of the Alliance were the technical partners from the Council of Coaches project. However, since September 2020, two additional parties have already joined to actively contribute to the platform's development.

Members of the Alliance are committed to actively working on the documentation, tutorials and development of the platform, as well as keeping the platform up to date with new releases of its components. This work is coordinated by the Personalised eHealth Technology programme of the University of Twente and supported by ongoing projects that use the platform or its individual components.

6 CONCLUSION

In this article, we presented the Agents United Platform, an open source platform for multi-agent conversational systems. The platform integrates a set of state-of-the-art components, advancing the capabilities of existing components and introducing several new ones. This offers novel capabilities to the platform as a whole and presents interesting directions for future work.

Furthermore, the platform is not just a topic of research in itself, but it is also a vehicle for research. It facilitates easy setup of new agent applications, which in turn allows researchers and developers to focus their efforts on the areas of interest for their research. They can, for example, setup different kinds of user studies on how agents can influence and inform users. Instead of needing to integrate a multitude of components to get to an application, they can now focus on study specific content, such as sensing, user adaption, dialogues or agent behaviours.

The viability of the platform has already been shown by uptake not only in student teaching, but also in at least 4 funded research projects that employ the full setup (one on agents and touch, one on multi-modal output generation, one on robots for autism, and one that involves user speech recognition); and by the interest of other research groups to not only use the platform, but actively contribute to the initiative.

ACKNOWLEDGMENTS

The authors would like to thank all students and researchers who have worked with various versions of the platform and who have provided valuable feedback for its development.

This work was supported by the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 769553 (Council of Coaches).

The Agents United Alliance is supported by the Personalised eHealth Technology programme of the University of Twente.

[8] Agents United Platform: www.agents-united.org
[9] Agents United GitHub: www.github.com/AgentsUnited
[10] Platform tutorials: www.github.com/AgentsUnited/documentation/wiki

REFERENCES

[1] T. Beinema, H. op den Akker, and H. Hermens. 2018. Creating an artificial coaching engine for multi-domain conversational coaches in eHealth applications. In *AAMAS Workshop on Intelligent Conversation Agents in Home and Geriatric Care Applications (ICA-HoGeCa)*, E. André, T.W. Bickmore, S. Vrochidis, and L. Wanner (Eds.), Vol. 2338. CEUR Workshop Proceedings, Aachen, 35–39.

[2] T. Beinema, H. op den Akker, L. van Velsen, and H. Hermens. 2021. Tailoring coaching strategies to users' motivation in a multi-agent health coaching application. *Computers in Human Behavior* 121 (2021), 106787. https://doi.org/10.1016/j.chb.2021.106787

[3] R.J. Beun, S. Fitrianie, F. Griffioen-Both, S. Spruit, C. Horsch, J. Lancee, and W. P. Brinkman. 2017. Talk and tools: The best of both worlds in mobile user interfaces for e-coaching. *Personal and Ubiquitous Computing* 21, 4 (2017), 661–674. https://doi.org/10.1007/s00779-017-1021-5

[4] F. Bex, J. Lawrence, and C. Reed. 2014. Generalising argument dialogue with the Dialogue Game Execution Platform. In *5th International Conference on Computational Models of Argument (COMMA '14)*. IOS Press BV, Amsterdam, Netherlands, 141–152. https://doi.org/10.3233/978-1-61499-436-7-141

[5] A. Cafaro, M. Bruijnes, J. van Waterschoot, C. Pelachaud, M. Theune, and D. Heylen. 2017. Selecting and expressing communicative functions in a SAIBA-compliant agent framework. In *17th ACM International Conference on Intelligent Virtual Agents (IVA '17)*. Springer, Switzerland, 73–82.

[6] G. Calbris. 2011. *Elements of meaning in gesture*. Vol. 5. John Benjamins Publishing, Amsterdam, The Netherlands.

[7] A. Cienki. 2005. Image schemas and gesture. *From Perception to Meaning: Image Schemas in Cognitive Linguistics* 29 (2005), 421–442.

[8] R.A.J. de Vries, K.P. Truong, C. Zaga, J. Li, and V. Evers. 2017. A word of advice: How to tailor motivational text messages based on behavior change theory to personality and gender. *Personal and Ubiquitous Computing* 21, 4 (2017), 675–687. https://doi.org/10.1007/s00779-017-1025-1

[9] M. Elvir, A. Gonzalez, C. Walls, and B. Wilder. 2017. Remembering a Conversation – A Conversational Memory Architecture for Embodied Conversational Agents. *Journal of Intelligent Systems* 26, 1 (2017), 1–21. https://doi.org/10.1515/jisys-2015-0094

[10] J. Kolkmeier, M. Bruijnes, D. Reidsma, and D. Heylen. 2017. An ASAP Realizer-Unity3D bridge for virtual and mixed reality applications. In *17th ACM International Conference on Intelligent Virtual Agents (IVA '17)*, Vol. LNCS-10498. Springer, Berlin, Heidelberg, 227–230. https://doi.org/10.1007/978-3-319-67401-8_27

[11] K. Konsolakis, H. Hermens, and O. Banos. 2019. A novel framework for the holistic monitoring and analysis of human behaviour. In *13th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI '19)*, Vol. 31. MDPI, Basel, Switzerland, 43.

[12] K. Konsolakis, H. Hermens, C. Villalonga, M. Vollenbroek-Hutten, and O. Banos. 2018. Human behaviour analysis through smartphones. In *12th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI '18)*, Vol. 2. MDPI, Basel, Switzerland, 1243.

[13] S. Kopp, B. Krenn, S. Marsella, A.N. Marshall, C. Pelachaud, H. Pirker, K.R. Thórisson, and H. Vilhjálmsson. 2006. Towards a common framework for multimodal generation: The behavior markup language. In *6th ACM International Conference on Intelligent Virtual Agents (IVA '06)*, Vol. LNCS-4133. Springer, Berlin, Heidelberg, 205–217. https://doi.org/10.1007/11821830_17

[14] J. Lawrence, M. Snaith, B. Konat, K. Budzynska, and C. Reed. 2017. Debating technology for dialogical argument: Sensemaking, engagement and analytics. *ACM Transactions on Internet Technology* 17, 3 (2017), 24:1–24:23.

[15] M. Mancini and C. Pelachaud. 2008. Distinctiveness in multimodal behaviors. In *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '08)*. International Foundation for Autonomous Agents and Multiagent Systems, Liverpool, United Kingdom, 159–166.

[16] M. Mancini and C. Pelachaud. 2008. The FML-APML language. In *Proceedings of the Workshop on FML at AAMAS*, Vol. 8. International Foundation for Autonomous Agents and Multiagent Systems, Liverpool, United Kingdom, 1–5.

[17] R. Niewiadomski, E. Bevacqua, M. Mancini, and C. Pelachaud. 2009. Greta: An interactive expressive ECA system. *8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '09))* 2 (2009), 1427–1428.

[18] H. op den Akker, V.M. Jones, and H.J. Hermens. 2014. Tailoring real-time physical activity coaching systems: A literature survey and model. *User Modeling and User-Adapted Interaction* 24, 5 (2014), 351–392. https://doi.org/10.1007/s11257-014-9146-y

[19] H. op den Akker, R. op den Akker, T. Beinema, O. Banos, D. Heylen, A. Pease, C. Pelachaud, V. Traver Salcedo, S. Kyriazakos, and H. Hermens. 2018. Council of Coaches: A novel holistic behavior change coaching approach. In *4th International Conference on Information and Communication Technologies for Ageing Well and e-Health (ICT4AWE 2018)*. SciTePress, Setúbal, Portugal, 978–989. https://doi.org/10.5220/0006787702190226

[20] P. Pace, R. Gravina, G. Aloi, G. Fortino, A. Fides-Valero, G. Ibanez-Sanchez, V. Traver, C.E. Palau, and D.C. Yacchirema. 2017. IoT platforms interoperability for active and assisted living healthcare services support. In *2017 Global Internet of Things Summit (GIoTS)*. 1–6. https://doi.org/10.1109/GIOTS.2017.8016250

[21] H. Prakken. 2010. An abstract framework for argumentation with structured arguments. *Argument & Computation* 1, 2 (2010), 93–124.

[22] B. Ravenet, C. Clavel, and C. Pelachaud. 2018. Automatic nonverbal behavior generation from image schemas. In *17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '18)*. International Foundation for Autonomous Agents and Multiagent Systems, Liverpool, United Kingdom, 1667–1674.

[23] B. Ravenet, C. Pelachaud, C. Clavel, and S. Marsella. 2018. Automating the production of communicative gestures in embodied characters. *Frontiers in Psychology* 9 (2018), 1144.

[24] D. Reidsma and H. van Welbergen. 2013. AsapRealizer in practice - A modular and extensible architecture for a BML Realizer. *Entertainment Computing* 4, 3 (aug 2013), 157–169. https://doi.org/10.1016/j.entcom.2013.05.001

[25] D. Richards and K. Bransky. 2014. ForgetMeNot: What and how users expect intelligent virtual agents to recall and forget personal conversational content. *International Journal of Human-Computer Studies* 72, 5 (2014), 460–476.

[26] C.L. Sidner, T. Bickmore, B. Nooraie, C. Rich, L. Ring, M. Shayganfar, and L. Vardoulakis. 2018. Creating new technologies for companionable agents to support isolated older adults. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 8, 3 (2018), 1–27.

[27] M. Snaith. 2020. An argument-based framework for selecting dialogue move types and content. In *8th International Conference on Computational Models of Argument (COMMA '20)*. IOS Press BV, Amsterdam, Netherlands, 355–362.

[28] M. Snaith, N. Conway, T. Beinema, D. De Franco, A. Pease, R. Kantharaju, M. Janier, G. Huizing, C. Pelachaud, and H. op den Akker. 2021. A multimodal corpus of simulated consultations between a patient and multiple healthcare professionals. *Language Resources and Evaluation* (2021). https://doi.org/10.1007/s10579-020-09526-0

[29] M. Snaith, D. De Franco, T. Beinema, H. op den Akker, and A. Pease. 2018. A dialogue game for multi-party goal-setting in health coaching. In *7th International Conference on Computational Models of Argument (COMMA '18)*. IOS Press BV, Amsterdam, Netherlands, 337–344.

[30] M. Snaith, J. Lawrence, A. Pease, and C. Reed. 2020. A modular platform for argument and dialogue. *9th International Conference on Computational Models of Argument (COMMA '20)*, 473–474. https://doi.org/10.3233/FAIA200540

[31] M.R. Tazari, F. Furfari, A. Fides-Valero, S. Hanke, O. Höftberger, D. Kehagias, M. Mosmondor, R. Wichert, and P. Wolf. 2012. The universAAL Reference Model for AAL. In *Handbook of Ambient Assisted Living: Technology for Healthcare, Rehabilitation and Well-being*, J.C. Augusto, M. Huch, A. Kameas, J. Maitland, P. McCullagh, J. Roberts, A. Sixsmith, and R. Wichert (Eds.). IOS Press BV, Amsterdam, Netherlands, 610–625.

[32] M. ter Maat and D. Heylen. 2011. Flipper: An information state component for spoken dialogue systems. In *11th ACM International Workshop on Intelligent Virtual Agents (IVA '11)*. Springer, Berlin, Heidelberg, 470–472.

[33] J. van Waterschoot, M. Bruijnes, J. Flokstra, D. Reidsma, D. Davison, M. Theune, and D. Heylen. 2018. Flipper 2.0: A pragmatic dialogue engine for embodied conversational agents. In *18th ACM International Conference on Intelligent Virtual Agents (IVA '18)*. ACM, New York, NY, United States, 43–50.

[34] H. van Welbergen, D. Reidsma, and S. Kopp. 2012. An incremental multimodal realizer for behavior co-articulation and coordination. In *12th ACM International Conference on Intelligent Virtual Agents (IVA '12)*, Vol. LNCS-7502. Springer, Berlin, Heidelberg, 175–188. https://doi.org/10.1007/978-3-642-33197-8_18

[35] H. van Welbergen, D. Reidsma, Z.M. Ruttkay, and J. Zwiers. 2010. Elckerlyc - A BML Realizer for continuous, multimodal interaction with a Virtual Human. *Journal on Multimodal User Interfaces* 3, 4 (2010), 271–284. https://doi.org/10.1007/s12193-010-0051-3

[36] H. van Welbergen, D. Reidsma, and J. Zwiers. 2013. Multimodal plan representation for adaptable BML scheduling. *Autonomous Agents and Multi-Agent Systems* 27, 2 (2013), 305–327. https://doi.org/10.1007/s10458-012-9217-6

[37] H. van Welbergen, R. Yaghoubzadeh, and S. Kopp. 2014. AsapRealizer 2.0: The next steps in fluent behavior realization for ECAs. In *14th ACM International Conference on Intelligent Virtual Agents (IVA '14)*, Vol. LNCS-8637. Springer, Cham, 449–462. https://doi.org/10.1007/978-3-319-09767-1_56

[38] S. Wells and C.A. Reed. 2012. A domain specific language for describing diverse systems of dialogue. *Journal of Applied Logic* 10, 4 (2012), 309–329.

[39] J. Wohlfahrt-Laymann, H. Hermens, C. Villalonga, M. Vollenbroek-Hutten, and O. Banos. 2018. Enabling remote assessment of cognitive behaviour through mobile experience sampling. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, Athens, Greece, 794–799.

[40] Y. Xu, C. Pelachaud, and S. Marsella. 2014. Compound gesture generation: A model based on ideational units. In *14th ACM International Conference on Intelligent Virtual Agents (IVA '14)*, Vol. LNCS-8637. Springer, Switzerland, 477–491.

[41] F. Yunus, C. Clavel, and C. Pelachaud. 2021. Sequence-to-Sequence Predictive Model: From Prosody To Communicative Gestures. In *Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management. Human Body, Motion and Behavior (HCII '21)*. Springer, Berlin, Heidelberg, 1–14.