Model Driven White Force Automation

by

Simon Vajda

Submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy

Deakin University

January, 2023

DEAKIN UNIVERSITY
ACCESS TO THESIS - A

I am the author of the thesis entitled

Model Driven White Force Automation

submitted for the degree of Doctor of Philosophy

This thesis may be made available for consultation, loan and limited copying in accordance with the Copyright Act 1968.

'I certify that I am the student named below and that the information provided in the form is correct'

Full Name: SIMON RAOUL LESLIE VAJDA

Signed:

Date: 22/1/23

DEAKIN UNIVERSITY
CANDIDATE DECLARATION

I certify the following about the thesis entitled (10 word maximum)

Model Driven White Force Automation

submitted for the degree of Doctor of Philosophy

a. I am the creator of all or part of the whole work(s) (including content and layout) and that where reference is made to the work of others, due acknowledgment is given.

b. The work(s) are not in any way a violation or infringement of any copyright, trademark, patent, or other rights whatsoever of any person.

c. That if the work(s) have been commissioned, sponsored or supported by any organisation, I have fulfilled all of the obligations required by such contract or agreement.

d. That any material in the thesis which has been accepted for a degree or diploma by any university or institution is identified in the text.

e. All research integrity requirements have been complied with.

'I certify that I am the student named below and that the information provided in the form is correct'

Full Name: SIMON RAOUL LESLIE VAJDA
(Please Print)

Signed: ....

Date: 4/8/22

Acknowledgements

I want to thank my supervisors, Alfred Deakin Professor K. Mouzakis, Dr S. Parker, Dr L. Hoon, and Dr A. Giardina for their patience and support. Without which this journey would have never been completed.

I would like to thank my family for their dedication, endurance and sheer strength of will. The journey has not been easy.

I would also like to thank my friends and peers who listened, advised and supported me over the years. I would also like to thank my industry partners for the encouragement and kind words.

Simon Vajda, 2022

Dedicated to my wife and daughter. For a better tomorrow.

Abstract

This thesis presents the Describe, Collect, Analyse, Recommend and Report (DCARR) framework for the identification, communication, representation, classification and automation ranking of candidate tasks and processes / routines. DCARR aims to address a resource allocation constraint for White Force facilitated, Live, Virtual, and Constructive distributed synthetic training for the Royal Australian Air Force.

We propose that DCARR, is a model driven framework for task and process / routine identification, using methods and language from Human Factors research. DCARR makes use of modelling and UML for abstraction and communication to a wide audience of stakeholders, the model representation of the information processing steps that human decision makers perform. Using classification terminology and concepts from Cynefin, DCARR is able to generate a ranked and ordered list of candidate tasks and processes / routines for automation using Human Automation Interaction theory as a basis for selection criteria.

Furthermore, we present the method for the development and evolution, and the method for the testing and validation of DCARR, using the proposed criteria of Coverage, Alignment, Representation, Reproducibility, Dynamicism, Existing Theory and Effort, and associated metrics. As part of the method for the testing and validation, DCARR was validated in consultation with subject-matter experts in White Force tasking and research, and members of Human Factors Research, and Air Operation Simulation Centre at Defence Science and Technology Group, Fisherman's Bend.

The outcomes of this validation indicate that the proposed framework is a *"Structured and repeatable approach to collecting and integrating subjective and objective evidence to deliver a prioritised list of recommendations that target automation"*. It is a useful outcome and is the first step in achieving the initial desire of having a model of White Force tasks and processes. A constraint identified is the challenge of estimating time and cost for applying this framework, which invites future work.

Contents

Chapter 1

Introduction

Training is the process of teaching another person skills or knowledge[1]. Typically, training would be conducted multiple times in a repetitive act, to improve the recipients' retention of the skills learnt, confidence in one's performance of the action and technique and overall performance and capability [2].

Training is context dependant. For complex training tasks, such as military training exercises, a trainee may need to use specialised equipment or be exposed to specialised environments. These training exercises can be extremely costly and dangerous. Live ammunition is often used, as well as expensive and costly equipment such as vehicles and aircraft. These military training exercises, often require large groups of trainees to participate alongside facilitators.

An example of a military training exercise is Exercise Pitch Black[2][3]. Exercise Pitch Black is a large scale training event, held over three weeks involving up to 2,500 personnel and over 100 aircraft from participants from around the world. The purpose of this exercise, is to ensure that Australian Defence Force (ADF) and Royal Australian Air Force (RAAF) maintains a state of readiness to act when the Australian Government requires.

An approach to deal with the cost and dangers (exposure to hazardous environments or to simulate specialised equipment, as well as ammunition, fuel, staffing, vehicle maintainence, logistics) of military training is through the use of simulation. This allows trainees to experience conditions as close as possible to real encounters. A common term used to describe the levels of simulation is LVC (Live, Virtual and Constructive) [3,4]. Exercise Pitch Black makes

[1] https://dictionary.cambridge.org/dictionary/english/training
[2] https://www.airforce.gov.au/news-and-events/events/exercises/pitch-black
[3] https://www.airforce.gov.au/exercises/pitch-black

use of Live and Virtual entities (where virtual entities are simulated non-enemy targets) as part of the training exercise.

Live simulation involves real people using real systems. This includes live military training, which is regarded as simulation, because the training is not conducted against a live enemy.

Virtual simulation is real people using simulated systems. This focuses on training motor control skills (e.g. by allowing a trainee to fly a simulated jet or drive a simulated tank), decision-making skills (e.g. by allowing a trainee to perceive information and determining what actions to perform and when) and communication skills (e.g. by allowing a trainee to engage in a team training scenario and engaging with facilitators or experienced role players who are supporting the training scenario).

Constructive simulation involves real people engaging with simulated people and simulated systems. The trainees provide input into the simulation (such as orders and directions) and the simulated people determine the actions based on the orders.

For ADF and RAAF, LVC training exercises require two types of people to be involved - trainees and facilitators [5]. These facilitators are referred to as White Force.

1.1 LVC distributed synthetic training

In Australia, the Air Operations Simulation Centre (AOSC) at Defence Science and Technology Group (DST Group)[4], conduct research into simulation approaches to provide Modelling and Simulation (M&S) capability. This M&S capability supports the ADF goals and objectives, and extends to individual aviation platforms in the RAAF, Army and Navy and Command and Control systems for aerospace warfare, both land and air based (including autonomous systems). AOSC houses the Deployable Air Combat Simulator, Air Defence Ground and Air Environment Simulators, Royal Australian Air Force (RAAF)[5]/Australian Army[6]/Royal Australian Navy[7] Cockpits and Joint Terminal Attack Controller (JTAC) simulator[8]. To support RAAF in preparation for Exercise Pitch Black, AOSC supports the conduction of Exercise Black Skies, which is a simulated and virtual training event to prepare pilots for Exercise Pitch Black[9].

[4]https://www.dst.defence.gov.au/news/2019/11/20/improving-fidelity-our-virtual-worlds
[5]https://www.airforce.gov.au/
[6]https://www.army.gov.au/
[7]https://www.navy.gov.au/
[8]https://www.dst.defence.gov.au/capability/aerospace-systems-effectiveness
[9]http://www.defence.gov.au/annualreports/15-16/Features/15-ScienceTechnologyAirForce.asp

Plan Jericho [6][10] is the Royal Australian Air Force's Plan to develop an integrated and networked fifth generation[11] Air Force. As part of the work program for Plan Jericho, is a project to *"Enhance Air Force's Live, Virtual and Constructive (LVC) and Ranges Capability"*[12]. In response to Plan Jericho, DST Group have developed its Science and Technology (S&T) program to support RAAF and the Aerospace Domain from 2017-2027. This program [7] outlines the S&T that DST Group aims to provide for RAAF.

One of these programs is Force Experimentation and Preparation. The drivers of this program are: to help RAAF realise the capabilities of a fifth generation Air Force, focusing on *"new methods of training, experimenting and conducting test and evaluation will be required, placing a greater emphasis on the need for Live, Virtual and Constructive simulation capabilities"* [7]. The areas of research under this program include:

- Live, Virtual, Constructive Environments
- White Forces
- Adaptive Systems
- Synthetic Environments
- Human in the Loop
- Modelling and Simulation
- Agents and Artificial Intelligence
- Training Analysis and Optimisation.

1.2 On The Need For White Force Multipliers

Figure 1.1 is a visual representation of the comparison between White Force[13] and trainees. For Exercise Black Skies[14], it is estimated that there is an 2:1 ratio of trainers to trainees. Military training, such as that being conducted as Exercise Black Skies, consists of three phases; Event Preparation, Event Delivery and Post-Event Analysis [5]. White Force are expected to perform multiple varied roles. One of the quality features of DCARR that differentiates it from other frameworks is the enabling of partial completion of stages. Furthermore, another

[10] https://www.airforce.gov.au/our-mission/plan-jericho

[11] https://www.airforce.gov.au/our-mission/fifth-generation-air-force

[12] https://www.leonardodrs.com/sitrep/q4-2018-what-s-next-for-air-combat-maneuver-and-instrumentation-acmi/lvc-is-revolutionizing-the-way-air-forces-train/

[13] Blue Force/Roleplayers act in support of the trainees, while Red Force/Roleplayers act as opposing forces.

[14] In 2018, Exercise Black Skies was run as Exercise Virtual Pitch Black, as there was a decreased contingent of DST Group personnel onsite undertaking research, as the focus of the exercise was on preparation for Exercise Pitch Black

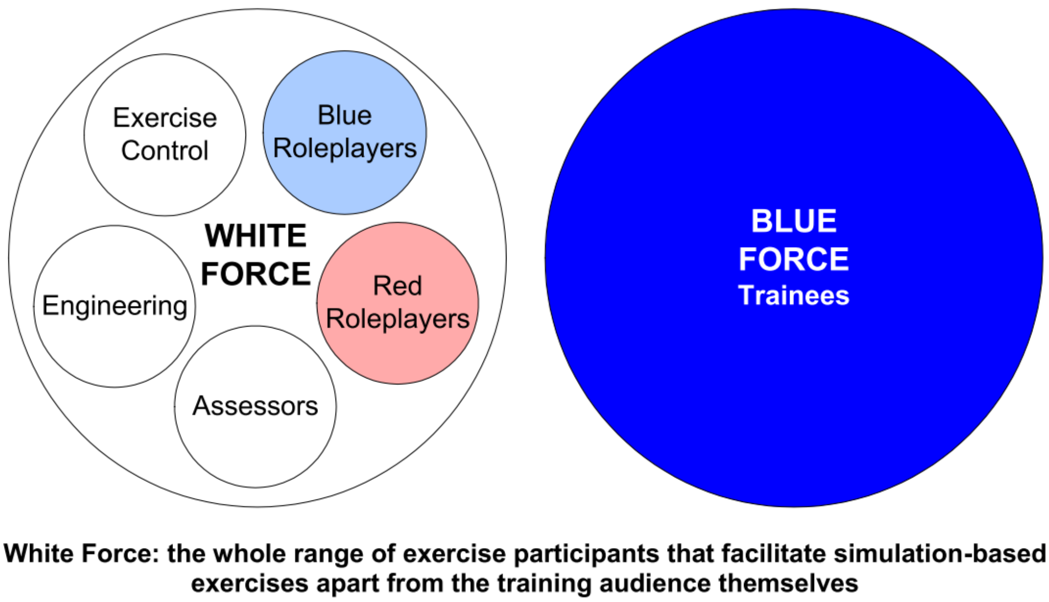


Figure 1.1: White Force as simulation facilitators [5]

quality feature, is the ability to refoucs to a different goal or repetition of exisiting process as goals shift from an unstable complex or chaotics environment, to a stable complicated or clear environment This will be expanded upon in the body of this thesis.

There is a minimum number (simulation engineers, EXCON, Blue and Red roleplayers) of fixed White Force roles that need to be filled in order to provide training. Maintaining such a large and skilled group is an expensive and time-consuming endeavour. Simpkin et al. [5] argue that there is a need to multiply the capabilities due to White Force being a *“limited resource who are critical for effective collective training."* [5]. The argument states that *“synthetic exercise best-practice advice, distilled from years of distributed training experience, advocates that exercise should be facilitated by a professional White Force comprised of experts in related military operational and technical domains."* [5].

Maintaining a dedicated White Force contingent that are subject-matter experts in military operational and technical domains is expensive and prohibitive. It is possible to involve contractors, who are often former RAAF pilots, who have experience to act as White Force[15]. It is also possible to use front-line operators to act as White Force[16], but during times of high operational tempo, it is not possible to extend White Force in this way. Therefore, having a balanced White Force that has the expertise, operational knowledge and training ability is problematic.

[15] One such example is Milskil, whose employees are *"experienced in high-end military and paramilitary operations"*

[16] Acting as Blue Force role-players who will communicate with trainees, control computer generated forces and assist exercise control to achieve training objectives

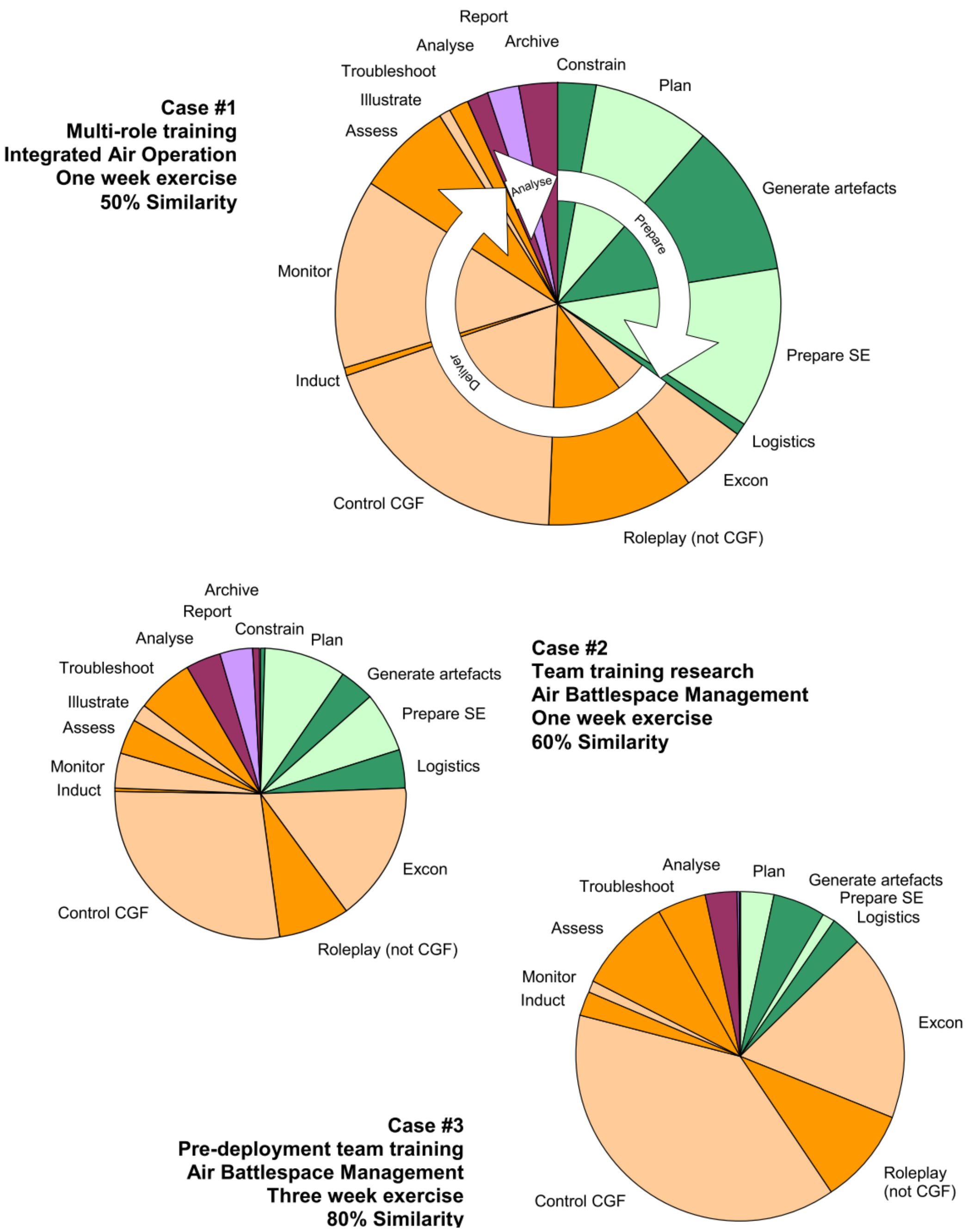


Figure 1.2: Relative White Force resource allocation based on similarity of exercises (Estimated) [5]

There is a need to multiply the capabilities of White Force. A "White Force Multiplier" refers to *“any technology or technique that leverages a given investment in White Force resources to provide improved White Force capability"* [5]. Figure 1.2 is the relative White Force resource allocation estimations based on the similarity of exercises. As the similarity of exercises increases, less effort is required in the Event Preparation and Event Delivery phase. But as simulation training becomes increasingly used, the requirement for diverse and disparate training objectives increases. This begins to add complexity to training requirements, impacting the expertise, knowledge and ability required by White Force personnel. As part of the Event Delivery phase, Blue Force roleplayers engage with trainees using voice and chat. In an effort to maintain realism and immersion, it is possible to become cognitively overloaded. This is true for all roles of White Force.

White Force facilitate training delivery to meet training outcomes. Training outcomes are focused on the trainees meeting the objectives of the training exercise. The training exercise is designed to meet an objective(s), which are created and defined by either the instructor/assessor or researcher. For Exercise Virtual Pitch Black, the training aim was to prepare trainees for Exercise Pitch Black 2018[17]. Therefore, White Force capabilities are based on the needs of training delivery, which in turn are based on the training objectives. While in one sense it is important to multiply the capabilities of White Force, it is also important to understand training delivery as a function of training objectives.

White Force capabilities are the tasks and events that are performed by human agents to facilitate distributed synthetic training. These capabilities are grouped into phases, where different tasks are performed by different roles (a human agent may perform one or more White Force roles).

Training objectives are those objectives that shape and define the training exercise. These objectives impact the requirements, capabilities and delivery of training to trainees. Hence, the objectives, training delivery and White Force capabilities are co-dependant. While a number of roles are required to facilitate training, such as Exercise Control and Assessors(1.1), the capabilities of these roles may vary as part of the training objectives.

In the context of this domain and problem, the use of the term multiply refers to the ability to boost output given a fixed input. While there is an understanding of the capabilities that White Force perform for training delivery, as per Simpkin et al [5], it is important to understand and measure these capabilities as a function of training objectives. Therefore, we can employ models and modelling to communicate concepts and processes to capture the capabilities of training delivery. These capabilities act as a function of training objectives and allow us to

[17] Appendix A - Exercise

understand White Force capabilities. Modelling and Model Driven approaches are the basis of Model Driven Engineering, a software development framework. By modelling the capabilities of White Force in a systematic approach, this will allow for the analysis and identification of candidate processes for multiplication through improvement, enhancement, optimisation or automation.

1.3 Need for a Model Driven Automation Framework

There are two potential methods to multiply capability. One approach is to optimise tasks and processes through the improvement, reengineering and reorganising of the steps involved and the resources used. Another approach is to identify and automate repeated tasks and processes to reduce or remove human involvement. Tasks and events can be measured to understand the current resource allocation and process reengineering can be performed to optimise it; or technologies can be employed to automate the tasks and events White Force perform. Optimisation relies upon process understanding and organisational support, which is often difficult to attain.

To multiply capability we can automate tasks and processes to reduce human involvement. Automation is dependent on a fundamental understanding of the inputs and requirements of each task and process performed by White Forces. Automation requires process understanding, but can leverage and make use of existing technologies available, such as distributed computing, machine learning and data analytics.

To create a White Force multiplier, we need to investigate and understand White Force tasks and processes and the relationship between them and the overall objective. Therefore, a need for a method/approach that is applicable to the problem domain of White Forces is necessary. This method/approach needs to be suitable for the examination, information extraction and communication of the parts and pieces required to perform White Force tasks and capabilities to a wide variety of stakeholders in the event that further automation work is done to achieve White Force capability multiplication. As part of this need to communicate to a wide variety of stakeholders, and as an extension of work by Simpkin et al. [5], the decision was made to continue using the visual communication methodology and approach of modelling and model driven visual communication, to achieve this end.

With the creation of a method/approach to investigate White Force tasks/processes, an understanding of the relationship between tasks/processes and objectives can be generated. This mapping of objective to goal to tasks/processes can further help with future capability multiplication through the identification of areas of impact, difficulty and complexity on the overall efficiency and delivery. Furthermore, through the use of a method/approach to identify "costly" tasks/processes, an effectiveness measure can be made to validate a White Force multiplier solution to demonstrate the cost reduction and resource usage improvements on overall delivery.

1.4 Organisation, research questions and contribution

To assist the reader with navigation, the following presents the structure of this thesis.

- Chapter 1 introduces the conceptual problem of the need for White Force multipliers and background context.
- Chapter 2 presents the background literature and related works.
- Chapter 3 presents the research hypothesis, questions and methodology used.
- Chapter 4 presents the step by step breakdown of the major contribution, and the development of the proposed framework given the proposed methodology.
- Chapter 5 presents the (mainly qualitative) testing and validation done due to limitations of COVID-19.
- Chapter 6 presents a discussion of the proposed work, including the major outcomes, a sample application of the outcomes and areas future work.
- Chapter 7 concludes the research by summarising the proposed research in relation to the research hypothesis and questions.

This research aims to address the following questions.

1. How can we systematically identify White Force tasks and processes/routines, including those that are hidden or ad-hoc, for further communication, representation, classification and investigation? This is addressed in Chapter 2.
2. What approach can be used to adequately communicate White Force tasks and processes/routines to a wide and varied audience of automation implementors, to facilitate discussion and review for further task representation? This is addressed in Chapters 2, 4 and 5.
3. What method can be used to suitably capture and represent, in a structured and standardised way, the variables (i.e. information and decision-making) of White Force tasks and processes/routines, to assist with classification of tasks into suitable categories of automation? This is addressed in Chapters 2, 3 and 4.
4. How can White Force tasks and processes/routines be acceptably classified into cognitively demanding tasks / tasks that resist automation due to instability and are chaotic/-complex in nature, versus those that are repetitive, structured and rule based? This is

addressed in Chapters 2, 3 and 4.

5. What selection criteria (e.g. repetitive, rule based, structured, consistent, and does not require a decision, interpretation or assessment of results) can be used to fairly rank and order candidates tasks for automation application? This is addressed in Chapters 2, 3 and 4.

The major contributions of this work are as follows.

- The DCARR model driven framework, which aims to identify, communicate, represent, classify and rank for automation candidate tasks and processes/routines found within large scale exercises/events/scenarios/etc.
- The Method for the Development and Evolution of the DCARR Framework
- The Method for the Testing and Validation of the DCARR Framework
- The proposed criteria + metrics used for the Development and Evolution / Testing and Validation of DCARR (Coverage, Alignment, Representation, Reproducibility, Dynamicism, Existing Theory and Effort).

1.5 Chapter Summary

This chapter introduced the importance of LVC Distributed Synthetic Training and Simulation in the context of the future work for the RAAF. This work also introduced White Force, as facilitators and subject-matter experts who are crucial to the overall success of Distributed Synthetic Training. There is an identified need, given the current resource usage and allocation, as well as future expectations, that for success on the world stage, White Force capabilities need to be multiplied through the use of automation.

Chapter 2

Literature Review

This chapter examines the background literature relating to White Force capability, Live, Virtual and Constructive (LVC) Distributed Synthetic technology, infrastructure, and usage. This review also examines the core concepts within Modelling and Model Driven Engineering (MDE), as a means of visual communication of capabilities, and Automation as a means of capability multiplication by enhancing the decision-making capabilities of the human operator within the task/process. After the introduction to core concepts, this chapter aims to examine the related work for knowledge and information representation, task and process definitions, method/approach generation, and description and communication to multiplication implementors.

Analysis of the problem space observes the following candidate sub-problems that warrant literature review.

- White Force tasks and processes have been examined, but as a retrospective approach to understand past exercises and scenarios. Therefore, a gap exists on a method to identify and understand current/future White Force tasks and processes. This is substantiated in Section 2.1 and Section 2.4.2.

- For identified tasks and processes, no method or approach exists for the classification and identification of tasks into nominal groups, such as tasks that are appropriate and suitable for multiplication. Therefore, a gap exists on the method for task/process classification. This is substantiated in Section 2.4.1.

- As a means of communication to multiplication implementors, no method or approach has been used to represent White Force tasks and processes. Therefore, a gap exists for the visual communication method, to a wide and varied audience, the components and decision-making required of White Force tasks and processes. This is substantiated in Section 2.4.3 and Section 2.4.5.

- White Force currently perform tasks and processes based on “best” practice standards from the simulation community[1], intuition and experience. Currently, these standards are focused on the interoperability and reuse of artefacts and infrastructure. Therefore, a gap exists for the translation of task and process representation into candidates for multiplication of capability through automation. This is substantiated in Section 2.4.4

To undertake the literature review process, sources are selected through key word analysis relating to domain and application, related works based on author / publication / area of interest, and citation chain and references.

2.1 White Force

For context, White Force, are defined as *"the whole range of exercise participants that facilitate simulation-based exercises apart from the training audience themselves"* [5]. The term White Force exists outside this definition and can vary depending on country of use, military service or domain context. Other such definitions include:

- a team of experts who facilitate training including exercise management, technical liaison and role players for both friendly and opposing forces [8]
- a team of experts who organise and run large-scale live training exercises [9]
- exercise management, and a team of operators who simulate other participating units [10]
- an umpiring force made up of programmers, controllers, and evaluators, who have access to a variety of truth data [11]

For this work, White Force is defined as exercise facilitators, in context of distributed synthetic training exercises delivered by Australian Defence Force (ADF), Royal Australian Air Force (RAAF) and Air Operations Simulation Centre (AOSC).

2.1.1 Exercise Virtual Pitch Black, Exercise Pitch Black and Exercise Black Skies

Exercise Virtual Pitch Black[2] is a virtual and constructive distributed synthetic training exercise in preparation for the live and virtual simulated threats of Exercise Pitch Black. Exercise Pitch

[1] https://www.sisostds.org/
[2] https://news.defence.gov.au/capability/seizing-opportunity-simulated-success

Black is an exercise, run over three weeks, that occurs every two years, "conducted from RAAF Base Darwin and RAAF Base Tindal"[3].

The purpose of the exercise is twofold; firstly it aims to provide RAAF trainees with the skills and opportunity to maintain a high level of readiness and preparedness in case the Australian Government requires it. Secondly, it is an opportunity to demonstrate to the large contingent of international participants, the high value Australia and RAAF place on regional security and fostering closer ties with regional, coalition and allied partners throughout the Asia Pacific region.

"Exercise Pitch Black features a range of Live and Virtual simulated threats, all of which can typically be found within a modern battle-space environment. The exercise provides an opportunity to test and improve force integration, using Bradshaw Field Training Area and Delamere Air Weapons Range, which is one of the largest training spaces available in the world. Up to 4000 personnel and 140 aircraft from the region and allies are in attendance, including Australia, Canada, France (New Caledonia), Germany, Indonesia, Netherlands, New Zealand, Singapore, Thailand, India, Malaysia and the United States"[4]. Therefore, due to the importance of the event, Exercise Black Skies / Exercise Virtual Pitch Black are conducted in the weeks prior to Exercise Pitch Black.

Using Exercise Virtual Pitch Black as an example, the tasks that White Force perform are split into three phases; Event Preparation, Event Delivery and Post-event Analysis, as per the observations and interviews conducted in the work by Simpkin et al [5]. In any given scenario or training exercise investigated, the same tasks appear in each identified phase.

Event Preparation
The tasks that have been identified within the Event Preparation phase are; the identification of constraints, development of exercise plans and scenarios, generation of scenario specific artefacts, preparation of the synthetic environment and arrangement of event logistics [5].

The roles White Force undertake within this phase include Instructors/Assessors, Researchers, Exercise managers, Simulation Engineers and Administrative Support [5]. The tasks for the Event Preparation phase occur before an exercise is scheduled to occur, and can typically range in duration from days to weeks.

Event Delivery
The tasks that have been identified within the Event Delivery [5] phase are the following.

[3] https://www.airforce.gov.au/news-and-events/events/exercises/pitch-black
[4] https://www.airforce.gov.au/exercises/pitch-black

Figure 2.1 shows the structural hierarchy of Exercise Black Skies 2016.

- Scenario control
- Role-play of unmanned scenario elements
- Control of computer generated forces
- Conducting familiarisation training
- Mission planning
- Briefing and debriefing support
- Troubleshooting and monitoring simulation system performance
- Monitoring trainee performance
- Recording observations
- Implementing dynamic scenario innovations
- Managing audio/visual systems

Figures 2.2, 2.3 and 2.4 are models showing the relationship between roles, goals, tasks and tooling used by Exercise Control, Blue and Red role-players for Exercise Black Skies 2016 (A Virtual and Constructive distributed synthetic training exercise, to prepare participants for the Live and Virtual Exercise Pitch Black 2016). The roles within this phase are researchers, exercise managers, simulation engineers, instructors/assessors, role players and administrative support [5].

During Exercise Virtual Pitch Black 2018 (Exercise Black Skies, was dropped in favour of Exercise Virtual Pitch Black, due to a reduced DST Group presence and research focus), participants who made up the roles of exercise managers (2.5), were contractors from MilSkill. Simulation engineer participants were made up of members from DST Group and MilSkil. Instructors/Assessors (Figure 2.6) were contractors from MilSkil and Role Players were a combination of members of RAAF and contractors of MilSkil (Figures 2.8, 2.9, 2.10, 2.7).

Post-Event Analysis

The tasks that have been identified within the Post-Event Analysis phase are; analysis of event records, recommendations based on lessons learnt and the storage of exercise artefacts for

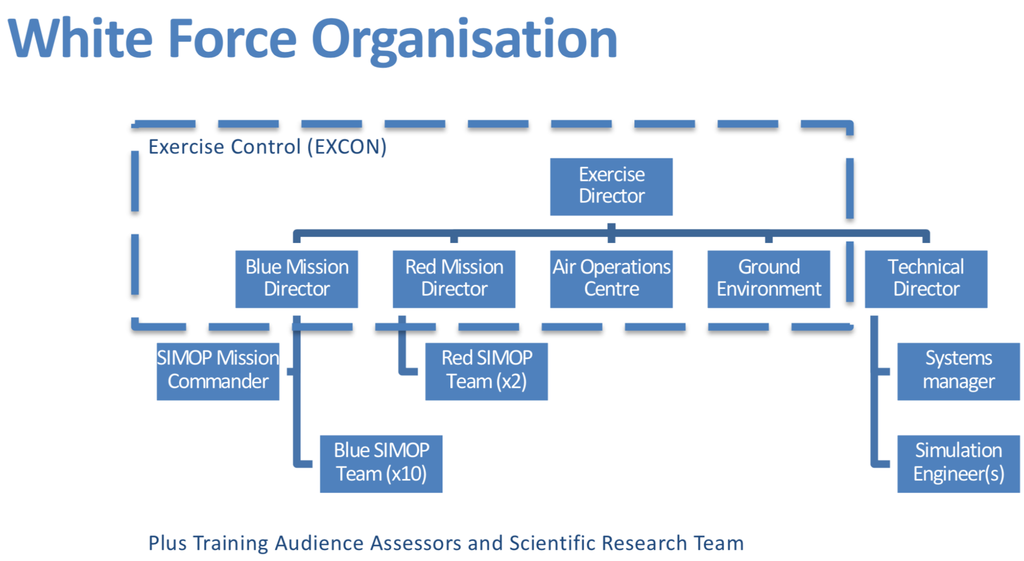


Figure 2.1: Overview of the roles and hierarchy for Black Skies 2016 [12]

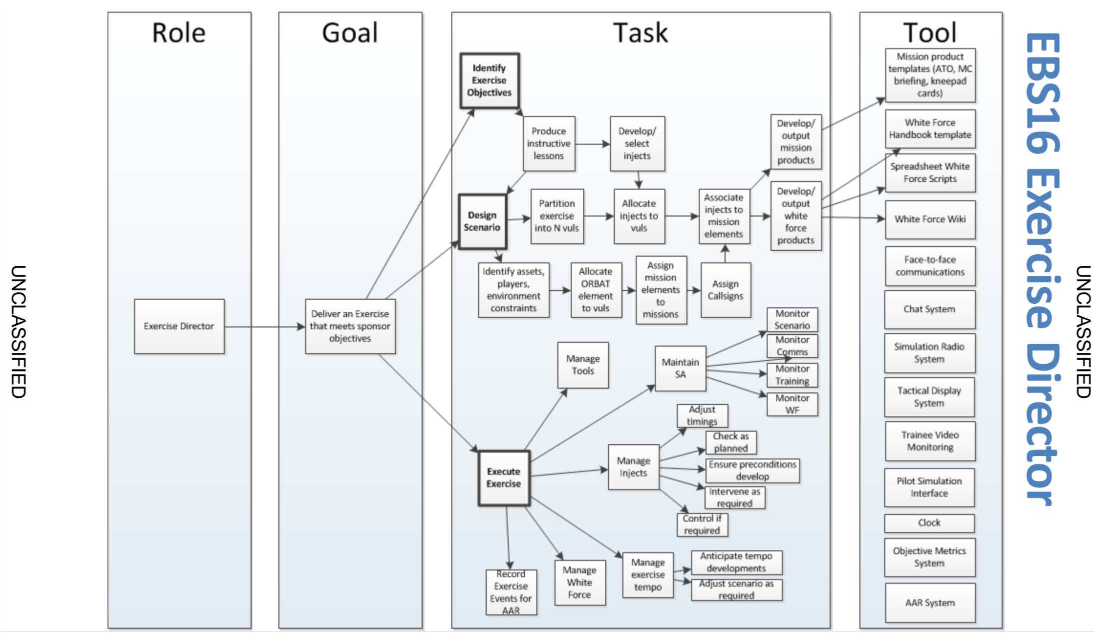


Figure 2.2: Overview of the tasks and tooling used by Exercise Control [12]

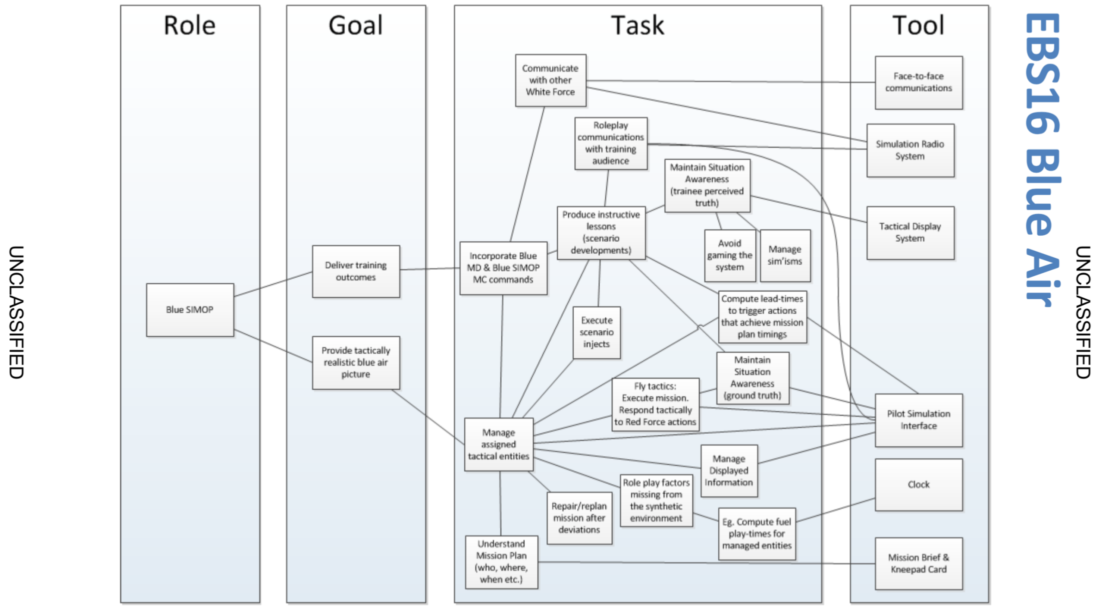


Figure 2.3: Overview of the tasks and tooling used by Blue Roleplayers [12]

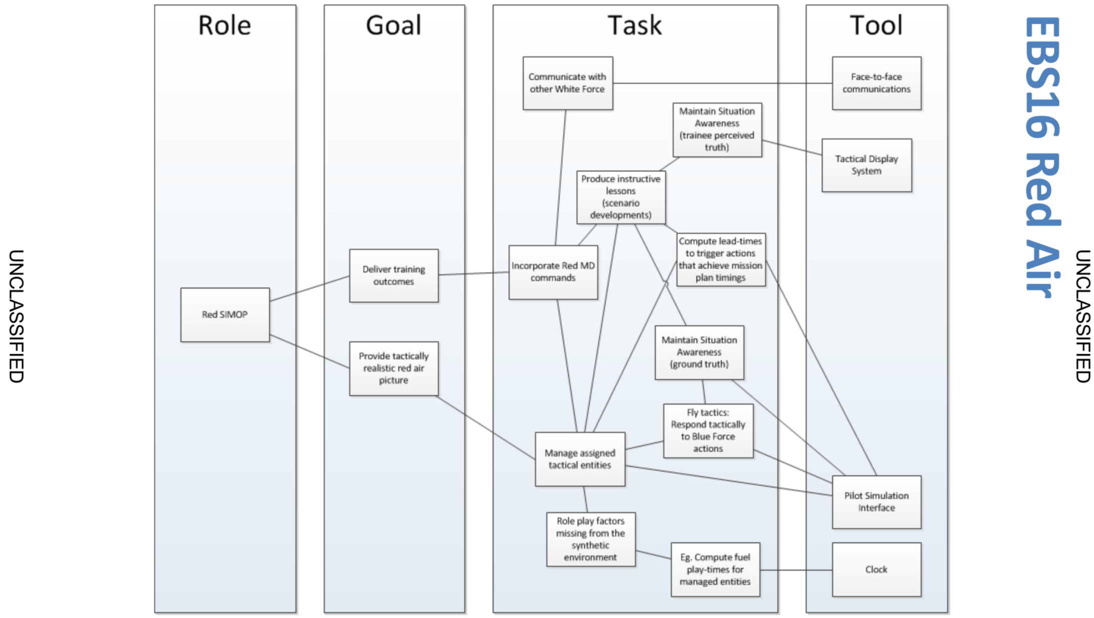


Figure 2.4: Overview of the tasks and tooling used by Red Roleplayers [12]

Figure 2.5: Exercise Virtual Pitch Black Exercise Manager

Source: http://images.airforce.gov.au/S20182494

Figure 2.6: Mission Briefing Conducted by MilSkil

Source: http://images.airforce.gov.au/S20182494

Figure 2.7: Red Roleplayers

Source: http://images.airforce.gov.au/S20182494

Figure 2.8: Blue Roleplayers

Source: http://images.airforce.gov.au/S20182494

Figure 2.9: Blue Roleplayers

Source: http://images.airforce.gov.au/S20182494

Figure 2.10: Blue Roleplayers

Source: http://images.airforce.gov.au/S20182494

reuse. The roles for the final phase are researchers, exercise managers, simulation engineers and administrative support. Often, depending on the duration of the exercise, a debrief session may occur at the end of each days' exercise delivery. Exercise Virtual Pitch Black was split into a briefing session in the morning for White Force personnel and trainees, with a short preparation phase before the planned daily operation would occur, generally for a number of hours.

After the planned daily operation would finish, lunch would occur before a debriefing session between Exercise Control, role players and trainees, where observations and recommendations would occur. This cycle of brief, daily operation and debrief would occur daily during the week-long exercise.

There are a number of highly complex processes and tasks that require a substantial amount of expertise, operational knowledge, training abilities and technical support. There are also a great deal of participants representing a variety of different organisations and stakeholders. For example, MilSkil are contracted to prepare and delivery training to RAAF, using existing tools, techniques and procedures which have been researched and developed over a number of years. Therefore, as a real world problem, this introduces another level of difficulty and complexity to deal with, in the hopes of generating a method/approach to assist with understanding White Force tasks and processes.

2.1.2 Distributed Simulation

Exercise Black Skies and Exercise Virtual Pitch Black are distributed synthetic training exercises/distributed simulations. This is due to the fact that as part of the large multinational presence at Exercise Pitch Black and other large scale exercises that occur, it is logistically not feasible to house each and every personnel member in the same localised environment. Hence, the exercises are conducted in different world locations and are by that nature, distributed. As part of the training in preparation for these exercises, it is as much an opportunity to train personnel (such as typical trainees), but it is also an opportunity to test the infrastructure and technology that underpins these exercises. Therefore, for Exercise Virtual Pitch Black/Black Skies, it is an opportunity to test the network backbone and underlying architecture and frameworks that the tools White Force use are built upon, in preparation for events such as Exercise Pitch Black.

There are two aspects to consider when talking about distributed simulations; the underlying architecture, and the software and tooling. There are a number of different architectures and protocols that exist and are used amongst the distributed simulations' community, the more common are: [13–18].

- Distributed Interactive Simulation (DIS)

- Aggregate Level Simulation Protocol (ALSP)
- High Level Architecture (HLA)
- Test and Training Enabling Architecture (TENA)
- Common Training Instrumentation Architecture (CITA)

The choice of underlying architecture has a great impact on the interoperability and compatibility between simulation endpoints. Ryan et al. [13] identified that within Australia, at least under a Defence umbrella, the most commonly used architectures are DIS and HLA.

Distributed Interactive Simulation (DIS) evolved from an early DARPA project, SIMNET. *"DIS is a network protocol standard, that operates by providing Protocol Data Units (PDUs) to create a synthetic environment."* [19] *"PDUs are data packets, that are broadcast over the simulation network"*, and are based upon a standard developed under the guidance of the Simulation Interoperability Standards Organisation (SISO) and formally approved as an IEEE standard (IEEE Std 1278TM — Distributed Interactive Simulation (DIS))[5] [20]. *"DIS uses a right-handed, Geocentric Cartesian coordinate system, called the World Coordinate System. The origin of the system, being the centroid of the earth and the units of measurement is based on the metric system, where 1 unit of length is 1 metre"* [13, 20].

"High Level Architecture (HLA) is a general purpose architecture, that does not specify the format of the data packets, instead HLA is considered an API, as opposed to DIS being a wire standard" [13], as HLA compliment simulators communicate with a Run-Time Infrastructure (RTI) following interface specifications. This means, that instead of streaming PDU/Packets over the wire (with the specifications defined within the PDU), HLA software communicates to RTI and follows predefined specifications. Therefore, HLA software can be reused with different RTI's from different manufacturers, but often there still can be software issues [13, 16, 17]. Both DIS and HLA are formally approved and recognised IEEE standard, developed under the guidance of SISO (IEEE Std 1516TM — High Level Architecture for M&S (HLA))

The observation of different Distributed Simulation architectures allows identification and understanding of the underlying architectures used by DST Group / AOSC White Force to conduct Synthetic training exercises. DST Group / AOSC use DIS (with some HLA - DIS gateways in place, see Appendix A - VPB18 Simulator Connections, for an overview), with plans to develop migration strategies for legacy simulators to HLA, as the technology matures over time [15], though these plans may no longer be the case. Knowing that DIS operates over the wire, and

[5] https://www.sisostds.org/ProductsPublications/Standards.aspx

is the protocol / architecture / framework favoured due to its maturity and interoperability, a small handful of software solutions exist that are used by White Force personnel.

Appendix A, (which references DIS), depicts Figures 2.2 and 2.9 and [21–26], which is one of the tools used by White Force to conduct Event Delivery, named ADGESIM[6][7][8]. ADGESIM contains a number of tools; DISVOX - Dis Voice Communications, TAARDIS - After action review suite and PSI - Pilot Simulator Interface. PSI is referenced in Figures 2.2, 2.3 and 2.4.

Figures 2.5, 2.7 and 2.10 show White Force at Exercise Virtual Pitch Black (2018) interfacing with a multi display setup that features a number of applications. Figures 2.2, 2.3 and 2.4 mention a number of the tools used during Exercise Black Skies (2016). These tools include ADGESIM (which contains PSI), Tactical Display System (TDS) and Simulation Radio and Chat comm systems.

Built into ADGESIM and mentioned in Appendix A - Technical, is Link-16 interoperability [21]. *"Link-16 is a high speed, jam-resistant digital data link which allows aircraft to share location information in near-real time and allows the transfer of text messages, images and two channels of digital voice (2.4 kbit/s and/or 16 kbit/s in any combination)"* [21]. Link-16 is defined in NATO's Standardisation Agreement STANAG 5516[9][10], while MIL-STD-6016 [27] is the related United States Department of Defence Link-16 MIL-STD. By implementing Link-16 MIL-STD/STANAG into ADGESIM, it allows simulators and real equipment to communicate, increasing fidelity and realism for trainees, enables distributed simulation with allied partners, such as the United States [28]. Inclusion of Link-16 raises a concern; by including the J series of communication protocols that are found with Link-16 and MIL-STD, it becomes **restricted** under ITAR[11], and requires a security clearance up to SECRET, making development difficult, complex and expensive [21].

2.2 Modelling

There are many factors to consider when discussing White Force and the role they play in synthetic training exercises. Some of these considerations are the amount of people, the components and tooling required for exercise objectives, the rules, restrictions and limitations of the scenario and exercise environment, and the overall complexity of the scenario, objectives, outcomes and research parameters. To adequately represent and communicate this information, we employ abstractions and visual communication techniques found within modelling

[6] http://ytek.com.au/products/adgesim/
[7] http://directory.simulationaustralasia.com/capability/adgesim/
[8] http://www.simulationaustralia.org.au/mysimaust/show_abstract/1096
[9] https://standards.globalspec.com/std/1307656/nato-stanag-5516
[10] https://infostore.saiglobal.com/en-au/standards/stanag-5516-ed-3-2006–457162/
[11] https://gov-relations.com/itar/

(typically the modelling approaches found within software development as a means of information communication). The following sections cover Models, Meta Models and Modelling Approaches that are of interest in addressing knowledge representation methods.

2.2.1 Models and Model Driven Engineering

In this work, we define model (or conceptual model) as an abstraction that aims capture information about a subject to help convey information or knowledge about what the model represents [29–31]. Models and modelling, are useful for visualising, specifying, constructing and documenting information from different views of a system [32].

Model Driven Engineering (MDE) is a software development methodology that aims to use models and the abstraction they provide, to create software [29–31,33,34]. MDE encompasses a number of disciplines including Model Driven Development (MDD), Model Driven Architecture (MDA) and Model Driven Testing (MDT). Model Driven approaches make use of models and metamodels [29–31,33,34].

Models, in the context of software engineering, represent an abstracted view of a software system. There are a number of different types of models that exist that present different views of a system. These can be grouped into two different types of views of a system: Structural/static, and Behavioural. These structural/static views, focus on the structure or architecture of a software system and include *“class diagrams, component diagrams, composite structure diagrams, deployment diagrams, object diagrams, package diagrams and profile diagrams”* [29–31, 33, 34]. Behavioural views focus on the interaction between entities and how a system functions and include *“activity diagrams, communication diagrams, interaction overview diagrams, sequence diagrams, state diagrams, timing diagrams and use case diagrams”* [29–31,33,34].

2.2.2 Meta-Models and Model Driven Architecture

Model Driven Architecture (MDA) is a part of MDE, which is championed by Object Management Group (OMG - an international technology standards consortium). MDA is a software design approach, where "architecture" does not refer to structural architecture of a system, but rather refers to a number of standards and model forms that act as the technological base [35].

MDA is related to a number of modelling standards and definitions including; Meta-Object Facility (MOF), Unified Modelling Language (UML), XML Metadata interchange (XMI), and Software Process Engineering Meta-model (SPEM) [35]. An important definition that stems from Model Driven Architecture is the Meta-Object Facility (MOF) [36].

The Meta-Object Facility is a model hierarchy and describes 4 levels - M0, M1, M2, M3. M0

Layer are instances/real world objects. M1 Models are models such as class diagrams and use cases, they describe abstracted views of M0 Layer. M2 Models are Meta-models and describe the notation and language used to describe M1 Models. A M2 Model / Meta-model proposed by OMG is Unified Modelling Language, or commonly referred to as, UML [33]. M3 Models are Meta-meta models, with the MOF being a M3 Model. M3 Models define M2 Models, such as UML, which can be used to create describe M1 models. In this way, an object that can be modelled (M0 to M1) and must conform to a Meta-model (M1 to M2) [37].

Meta-modelling is typically viewed as either a linguistic model or an ontological model [30]. Linguistic modelling is most representative of the hierarchical modelling found within MOF, with an emphasis on the "instance-of" relationship between objects, while M2 Models and M3 Models being language definition layers [30].

Linguistic modelling, therefore has a strong focus on the technical requirements but misses out on the ability to dynamically extend the set of domain types, which ontological meta-modelling is able to achieve. Ontological meta-modelling is able to capture the relationship between concepts [38]. Meta-modelling is a key component of MDE as it facilitates model transformations [35, 39].

2.2.3 Model Transformations and Model Transformation Languages

Model Transformations, an important component of MDE, aims to facilitate reproducibility by automating the building and modification of models where possible [40–42]. By defining models that conform to metamodels, transformations can be performed to generate code and software artefacts. A standard for Model Transformation Languages (MTL), proposed by OMG and based on the MOF is called QVT (Queries/Views/Transformations) [43].

Model Transformations and associated languages and approaches can be used to automate the generation and creation of artefacts that are complete, modular, extensible, reusable and reproducible [17]. Therefore, modelling, meta-modelling and model transformations are a viable approach to present information about the tasks and processes that White Force perform with the future aim of multiplying capabilities [17, 44–55].

2.2.4 Process Modelling

Models are abstracted views of real world objects. By modelling and creating a metamodel each instanced model must conform to its metamodel. There are languages and functions that can be applied to models, model transformations, and to automate the creation of artefacts. Different modelling definitions capture different abstracted views of a system. Different modelling techniques can be created that conform to the same family of existing modelling approaches. Process modelling approaches or the generation and creation of a new modelling approach, can be used to capture information from the people, components, tools, rules, restrictions and complexity of White Force operations and capabilities.

Different models have been used already to capture aspects of White Force capabilities. Figures 2.2, 2.3, 2.4 and Appendix A - VPB18 Simulator Connections are examples of this. These models either capture some relationship between the role and task (use case diagrams) or the role, goal, task and tool. Other modelling definitions/approaches exist that can be used to model and capture information. One such approach that exists for structured business processes, is the Business Process Model and Notation (BPMN) [56].

Business Process Model and Notation[12] [56] is a standard for the graphical representation and specification of business process modelling [57] [58] [59]. Business Process Modelling (BPM), focuses on the representation of processes of an enterprise, so that processes can be reviewed, improved, analysed and automated. Typically, BPM is conducted by business analysts who are skilled in the modelling approach and subject-matter experts (SME) who are experience with the processes being modelled. While there is difference between the typical tasks business analysts may face in a corporate environment compared to those found with a military context, BPM and BPMN is an applicable modelling approach, at least initially, to attempt to capture and identify the people, components, tools, rules, restrictions and complexity of White Force capabilities [60] [55].

BPMN is the standard for Business Process Diagrams (BPD). There are a number of limitations to BPMN, in that the scope is constrained to support only the concepts of modelling applicable business processes and not areas, such as; organisation structures, functional breakdowns and data models [58] [59].

BPMN describes a number of elements that are used to model business processes, such as; events, activities, gateways, connections and swinlanes. We can observe that BPMN offers a number of elements that can be useful in modelling White Force capabilities. The added benefit of BPMN is that there are approaches that can measure and identify the complexity of processes in BPD [61–63].

2.3 Automation

Automation, stemming from the word Automatic, *"includes the execution by a machine agent of a function that was previously carried out by a human"* [64]. Therefore, Automation can be described as the removal of the human from the process [64].

In this thesis, automation and automation theory forms the basis for the multiplier of White

[12] http://www.bpmn.org/

Force tasks and process. Automation is often referenced to, and synonymous, with technology, due to the fact that as technology improves so does automation solutions, technologies and techniques. Typically, automation has been associated with industrial automation.

2.3.1 Industrial automation

A clear example of automation to boost workforce capabilities can be seen with the Toyota Production System (TPS) / Just-in-time production [65]. Originally described as the Toyota Way, it is a set of 14 principles and behaviours that led Toyota's management and production approach. The main purpose of the TPS was to design out inconsistency, overburden and waste.

To achieve this, the underlying principles focused on continuous improvement, respect for people, a long-term philosophy over short term profits (i.e. short term profits are given a lower priority than long-term vision), right process for the right result, add value by developing people as partners and continuously solving problems which will drive organisational learning [65]. From this, other approaches stemmed, such as the philosophy of Lean Manufacturing.

Lean manufacturing (often referred to as "Lean"), which stemmed from the Toyota Way, focuses on; Improvement of Quality, Elimination of Waste, Reduction in time and Reduction in total costs [66,67]. A majority of the principles and concepts are similar to the Toyota Way and TPS, which has lead to Lean being considered a generalised approach of TPS. Lean aims to make obvious what adds value by reducing waste. It is in this way, it differentiates itself from TPS [66,67]. Another approach to consider from industrial automation is Six Sigma.

Six Sigma, differentiates itself from Lean, by focusing on the reduction of errors, causes of defects and minimising the variability in manufacturing [68]. The term originated from *"statistical modelling of manufacturing processes; that a manufacturing process can have a sigma rating, indicating its yield or percentage of defect free products. A six sigma defined manufacturing process is where 99.99966% of all opportunities to produce some features of a part are statistically expected to be free of defects"* [65,69]. Lean focuses on the addition of value and reduction of waste, while Six Sigma is focused on error reduction. A combined approach is called Lean Six Sigma.

Lean Six Sigma is a combined methodology *"to improve performance by removing waste and reducing variability and defects"* [70,71]. As part of the methodology, Lean Six Sigma has identified eight waste dimensions to eliminate; *"Defects, Over-Production, Waiting, Non-Utilised Talent, Transportation, Inventory, Motion and Extra-processing"* [65,69].

While the relevance and applicability of these approaches (TPS, Lean, Six Sigma and Lean Six Sigma) is targeted for manufacturing, there are important lessons to be learnt. These include

Level of automy	Description	Description
Level 1	Manual control	Computer offers no assistance
Level 2	Decision proposal stage	The computer offers some decisions to the operator. The operator is responsible to decide and execute.
Level 3	Human decision select stage	The human selects one decision and the computer executes.
Level 4	Computer decision select stage	The computer selects one decision and executes with human approval
Level 5	Computer execution and human information stage	The computer executes the selected decision and informs the human
Level 6	Computer execution and on call human information stage	The computer executes the selected decision and informs the human only if asked
Level 7	Computer execution and voluntarily information stage	The computer executes the selected decision and informs the human only if it decides to
Level 8	Autonomous control stage	The computer does everything without human notification, except if an error that is not into the specifications arrives. In that case the computer needs to inform the operator.

Figure 2.11: Generalised Level of Automation scale

how to define and create a successful automation approach that focuses on the reduction of errors, waste, variability and focuses on processes that add value, on skills and people, order and discipline.

2.3.2 Levels of Automation

Conceptually, automation is the removal of the human from a process, and implies that there is a varying degree of human involvement. This conceptually manifests as Levels of Automation (LoA), where automation can be applied at fixed levels and types (Figure 2.11) [64].

Levels of automation can be generalised into manual, decision support systems, human in the loop and autonomous systems. Manual level of automation, is the absence of automation technology and is instead the human in control of the entire process. Decision support systems (those systems that argument human decision-making, as per LoA literature [64]) are levels where the automation technology is able to assist the human operator with suggestions, options and supports the human to make decisions. At this level, the human is heavily involved in the process, but is supported by the system.

Human in the loop systems (those systems that involve the human operator as overseer / decision maker, as per LoA literature [64]) shift focus away from the human operator. These levels vary between the system involving the human operator to confirm and accept the systems decision, to the system deciding when to inform the human operator.

The final level is fully autonomous systems, where the human is no longer involved in the process at all. Autonomous systems are fully automated. Adaptive Automation, is where *"the level and type of automation might not be fixed but can change in real time."* [64, 72–81].

LoA are a defined scale that describe the relationship between system and human. LoA has been employed in a number of different fields and industries; avionics, teleoperations, remote control operations, manufacturing, air traffic control, piloting, unmanned aerial vehicles and self-driving cars [64].

The definition used for self-driving cars, is referred to as *"Level of Driving Automation (LoDA)* *defined by the Society of Automotive Engineers (SAE)"* [82, 83]. LoDA is an example of the generalisation of LoA and applicability to domains and contexts. Therefore, there is not one strict "hard and fast" way to describe the levels of automation for a particular domain, where each domain can define its own LoA.

Levels of Automation are the defined scale of human-automation interaction, the application of the scale can be done using Adjustable or Adaptable automation. Rather than a fixed level of automation being applied at any given time, adjustable/adaptive automation are two terms that imply a dynamic application of automation [64, 77, 84].

Adaptive automation is the dynamic application of automation, where the system or human can change the level of automation during runtime [80, 85–87]. Adjustable automation, is static, where only the human can change the level of automation. Adaptive automation is much more flexible and is the focus of a Human-Automation Interaction and the design of automation based on Human Performance Consequences [77, 88–92].

2.3.3 Human-Automation Interaction

Automation design needs to focus around the human operator. Human-Automation Interaction [77] is a field that focuses around the design considerations of automation solutions. These considerations relate to the concepts of function allocation, trust and reliability as an acceptance consideration, complexity of the automation solution, and the over reliance of automation/complacency of the human operator.

Another consideration of automation design is the differences between the mental models of the human operator and automation designer, interface design, usability and human computer interaction, and the Human Performance Consequences (complacency, skill degradation, mental workload and situational awareness) on the system design [93–97]. Therefore, rather than a universal approach, the field of Human-Automation Interaction and automation design is a nuanced approach that presents considerations for automation designers [79, 84, 98–100]. As mentioned, one of the important considerations for automation design is complexity, not just of the solution, but complexity of the workflow/process.

2.4 Related Work

With a background understanding of White Force tasks and processes, methods for knowledge and information representation (through the use of models) and a potential understanding of multiplication approaches, the following related works focus on the theories and methods. These include the methods for defining tasks and processes that White Force perform and are suitable for investigation and methods for knowledge and information extraction. These also include the approaches and theories of information representation, and the formatting and presentation of a suitable method/approach to highlight, define and represent suitable tasks for future multiplication implementations.

When discussing the problem of insufficient White Force resourcing, with White Force Multipliers as the postulated solution, the term complexity is used as a motivation. What White Force do is complex. As facilitators of LVC training, White Forces need to be skilled experts in their field of operations. White Forces act either as front line operators, able to provide realistic training opportunities, or skilled and technically adapt external contractors familiar with the use of operation of highly complicated tooling and equipment (ADGESIM, etc.). White Force staffing consists of former trained pilots, with the necessary skills, able to deliver on training objectives and outcomes.

To understand White Force task complexities, we need to consider the level and types of information required to undertake role-specific tasks and processes. This is presented as a mind map in Figure 2.12. This mind map was created during a discussion about White Force factors, conducted by members of AOSC at DST Group. These complexities and considerations are on top of the complexity required to operate and control entities / manage the simulation, as well as maintaining realistic voice communication to trainees (as a role player). These complexities and considerations are set to increase as the use of LVC training, scenario requirements and operational hardware requirements increases [101].

Subject-matter experts, such as those that comprise White Force, contain troves of tacit information required to fulfil tasks and duties [101]. The combination of tacit knowledge and task complexity, coupled with ad hoc tasks and processes (due a lack of formal representation and process definition) means that any White Force multiplier needs to resolve the distinction between knowable and unknowable factors, to further inform the investigation of suitable tasks and processes for multiplication.

Though initial study has been conducted to identify tasks and processes White Force perform, this is a retrospective approach to understand past exercises and scenarios. A gap exists on a method to identify and understand White Force tasks and processes, that is communicable with stakeholders and implementors of White Force multipliers. Once a method for identification is found, it is then possible to augment White Force tasks with automation multipliers.

2.4.1 Cynefin

Cynefin is a thought process framework to assist with definition and understanding of complexity, and it's impact [102]. Cynefin describes strategies for processing clear, complicated,

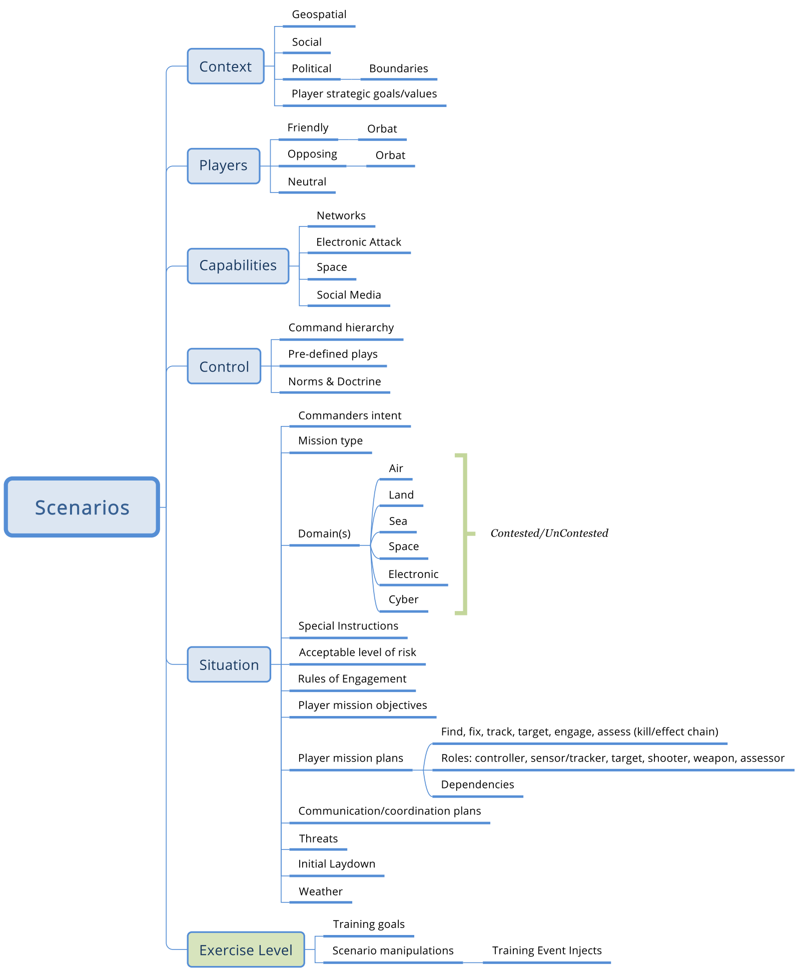


Figure 2.12: Mind map of some factors and information needing consideration during a scenario

Figure 2.13: Original Cynefin framework [104]

complex and chaotic problems [102–105] (Figure 2.13[13]).

Cynefin was originally proposed as a decision-making framework for leaders [102] and has seen use in multiple domains, such as healthcare [62, 105] and software engineering [104, 106] to manage complexity. It was initially comprised of 5 domains/realms, Simple, Complicated, Complex, Chaotic and Disorder[14].

Simple (later changed to Clear)[15] and Complicated are classified as belonging to an ordered universe and are regarded as deterministic in nature, while Complex and Chaotic are classified as belonging to an unordered universe and are regarded nondeterministic in nature. Each domain also contains a simple strategy for navigating through the domain, a solution definition, and classification. These are presented as follows.

- **Clear**: *sense-categorise-respond*: Best practice solutions. "Known knowns" [102]
- **Complicated**: *sense-analyse-respond*: Good practice solutions. "Known unknowns" [102]
- **Complex**: *probe-sense-respond*: Exaptive/Emergent behaviour. "Unknown unknowns" [102]
- **Chaotic**: *act-sense-respond*: Novel approaches. "Chaos" [102]

Using the Cynefin [102] model as our complexity definition, we can describe complexity as any element that is unknowable or non-deterministic. Therefore, to understand and communicate White Force tasks and processes, to multiplier implementors, it is necessary to identify areas

[13] https://thecynefin.co/cynefin-st-davids-day-2020-cynefin-framework/
[14] For this work, the realm of disorder is dropped
[15] https://thecynefin.co/cynefin-st-davids-day-2020-cynefin-framework/

of complexity and where possible decompose complex tasks, to classify and sort tasks into deterministic categories for further analysis.

Once resolved, the domain/complexity/complicatedness of the task could be mapped to the level and type of automation. Specifically, an adaptive, adjustable or fixed level of automation. In this way, Cynefin aligns and demonstrates synergy to Levels of Automation (Section 2.3.2). The deterministic nature of clear and complicated domains, the "known knowns" and "known unknowns", implies that understanding all aspects of the problems leads to a higher level of automation. Therefore, if a problem can be identified as clear, an automation technology that is autonomous is comparable.

Similarly, complicated problems match to human in the loop automation technologies. Complex problems, that is non-deterministic problems, where human input and decision-making is critical, align to decision support technologies. Finally, chaotic problems are those that are the hardest to solve and require novel solutions [102], are also the hardest to automate and involve human control in the process.

2.4.2 Human Factors and Event Analysis of Systemic Teamwork

Human Factors, *the science of people at work* [107], is loosely focused on the physical, cognitive and organisational ergonomics of people interacting with other people, systems and the cognition of doing so. Human Factors research, can help with understanding, measuring, representing and visualising the cognitive and behavioural steps involved in tasks and processes - including those that White Forces perform.

Previous work and studies have been conducted and applied to the domain of Command, Control, Communications, Computers and intelligence (C4i) [108, 109], as well as other network rich domains [107, 110]. These domains are representative of, and applicable to, the context of White Forces.

Event Analysis of Systemic Teamwork (EAST) is a Human Factors framework developed by Human Factors Integration Defence Technology Centre [109]. EAST is presented as a collection of methods (for collection, analysis and representation of data) and applied to civilian and defence domains [109, 110]. EAST provides a framework used to comprehensively describe, evaluate and analyse complex sociotechnical systems, such as understanding team working, communication usage and the linkage between agents [109, 110].

EAST [109, 110] is composed of multiple methods from Human Factors [107] to address different concerns - Hierarchical Task Analysis (HTA) for input, Coordination Demand Analysis (CDA) for team working, Communications Usage Diagram (CUD) for communications usage and Social Network Analysis (SNA) for linkage. The applicability of EAST, in relation to White Force multipliers, is the selection and presentation of methods/methodologies as a framework, which can be used for the generation and development of a method/approach to address the problem of White Force scarcity.

To address White Force scarcity, it is necessary to understand and define what the outcome of any proposed method/approach would be. In this case it would be the identification and presentation of tasks and processes of interest for investigation for White Force multiplication (i.e. deterministic and ordered tasks that would benefit from automation). To achieve this, we need to understand, identify, describe and define the capabilities of White Force (White Force tasks and processes).

HTA is *"a task description method and a variant of task analysis...of how a task is accomplished, including a detailed description of both manual and mental activities...etc"* [107,109–111]. HTA, used in conjunction with the structured approach defined by the EAST framework, can provide this research with a validated approach to exploring and identifying the tasks and processes of White Force. EAST describes a structured methodology of initial observation and data collection methods.

It is therefore possible to apply the collection and analysis methods of EAST to the domain of White Force, due to the similarity of the domains identified and examined by EAST [109,110]. The difference between EAST and a method/approach for White Force scarcity, is that EAST aims to examine, describe and evaluate complex sociotechnical systems. EAST achieves this, through the use of different Representation Methods.

Any proposed method/approach this research aims to produce, requires mechanisms to define and use different analysis methods to instead identify, evaluate and describe White Force multiplier opportunities. Hence, EAST can be combined with Cynefin as a classifier for deterministic/non-deterministic tasks. Together, they can be used to form an initial method-/approach to discover and classify, through data collection and data analysis methods, tasks and processes that exist, are performed by White Force, and are deterministic in nature.

2.4.3 OODA and Information Fusion

Cyenfin (Section 2.4.1) serves as a classification tool to identify deterministic and non-deterministic tasks. Hierarchical Task Analysis (Section 2.4.2) can be used as a method for task decomposition (to understand how a task is accomplished through both manual and mental activities). EAST (Section 2.4.2) can be used as a basis to generate a method/approach, combined with existing analysis and collection methods, to identify tasks that White Force perform and are suitable for analysis.

To further develop a method/approach, a task representation/analysis method needs to be developed/identified to communicate areas of interest and suitability to multiplier implementors (Section 2.1) [5]. One approach is to investigate similar methods and areas of research that deal with task representation and task analysis to achieve multiplication of human capability or reduction of human resource allocation.

Information Fusion (or Data fusion), is the process of combining multiple sources of information into a singular and accurate information source for a human operator [112, 113]. The core concepts of Information Fusion is the transformation of information from different sources to provide situational awareness to decision makers (i.e. humans) (Figure 2.14) [112, 113].

In the context of White Force, SME's who act in support of the training audiences (role players) [5] could be considered a fully manual form of information "fusers". White Force roleplayers need to take into account multiple sources of information (situation, context, players, capabilities, commanders intent, etc) to control entities with the primary objective of creating situations and executing scenarios for the purpose and benefit of the training audience [12].

Information Fusion is concerned with the automatic or semi-automatic approach of generating situational awareness for decision makers. White Force roleplayers need to manually maintain a high level of situational awareness to provide support for training audiences. Information Fusion can be considered the inverse of what any proposed method/approach to address White Force multipliers aims to do.

To alleviate White Force scarcity through multiplication of White Force capability, tasks and processes need to be identified and described to multiplier implementors. Information Fusion enables the fusion of multiple data sources to provide input to a single decision, while any method/approach needs to define and identify the breakdown of a decision for different personnel, actors, users, operators, systems into the set of data sources required. There is an argument that there is benefit to examining the domain of Information Fusion research to understand the approaches and methods of information representation.

Information Fusion, is composed of 2 areas of concern, High Level Information Fusion and Low Level Information Fusion (Figure 2.14) [112, 113]. Majority of work and research is focused around High Level Information Fusion [112, 113], and is driven by a series of Grand Challenges. These Grand Challenges were posed with the aim of unifying the different research approaches by partner nations [112, 113]. These challenges are [112, 113]:

- ***"Semantic Challenge****: What symbols should be used, how do those symbols acquire meaning?"* [112, 113]

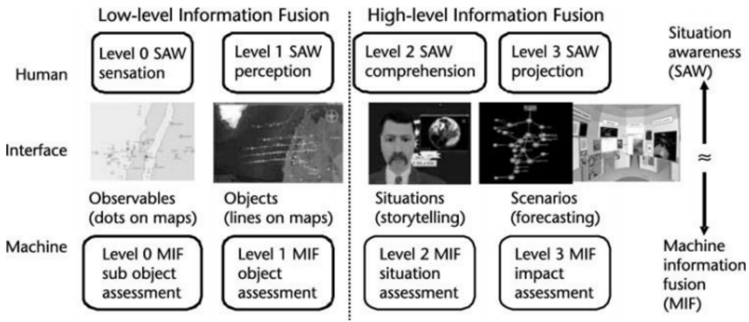


Figure 1.1 Relations between LLIF and HLIF as well as SAW and IF. [10].

Figure 2.14: Low Level to High Level Information Fusion

- *“**Epistemic Challenge**: What information should we represent, and how should it be represented and processed within the machine?”* [112, 113]
- *“**Paradigm Challenge**: How should the interdependency between the sensor fusion and information fusion paradigms be managed?”* [112, 113]
- *“**Interface Challenge**: How do we interface people to complex symbolic information stored within machines?”* [112, 113]
- *“**System Challenge**: How should we manage data fusion systems formed from combinations of people and machines?”* [112, 113]
- *“**Design Challenge**: How should we design information fusion systems formed from combinations of people and machines?”* [112, 113]
- *“**Evaluation Challenge**: How should we evaluate the effectiveness of information fusion systems?”* [112, 113]

Each participating research groups across these nations (Canada, Australia, and USA) have derived different approaches to address these Grand Challenges of Information Fusion. For knowledge representation (of interest to represent tasks and processes for White Force multipliers), this research is concerned with the Semantic and Epistemic Grand Challenges (what information is necessary and of interest, and how is it presented).

The Australian approach is **State Transition Data Fusion (STDF)**, based on the Mephisto Conceptual Framework.

- **Semantic Challenge**: **Mephisto Semantic Framework** [114] (The term Mephisto is based on the Metaphysical, Physical, Functional, Intentional and Social layers that form the result of the semantic challenge (i.e. the domains of interest to military and national security)). *“Axiomatic semantics in First Order Logics (FOLs) and Description Logics*

(DLs) covering various metaphysical, environmental, functional, cognitive and social concepts" [112].

- **Epistemic Challenge**: **Attitude and Attitude Too cognitive models** [114]. *"Cognitive agents with semantic, epistemic (declarative facts and rules) and episodic (procedural cognitive routines) long-term memories"* [112].

The American approach is **Information Fusion Situation Awareness (IFSA)**.

- **Semantic Challenge**: Development of **IFSA taxonomy**. *"Operational semantics of computational models to infer meaning over environmental, functional, cognitive and social concepts"* [112].

- **Epistemic Challenge**: **Information Management Model**. *"Agents for workflow and service based semantic, epistemic (facts and rules) and episodic (procedures) information processing"* [112].

The Canadian approach is Interpreted System (IS) and OODA Agents.

- **Semantic Challenge**: **Interpreted Systems**. *"Axiomatic semantics in Modal Logics covering various metaphysical, environmental, and functional concepts"* [112].

- **Epistemic Challenge**: **Cognitive Observe, Orient, Decide, Act Model** Interpreted Systems. *"User (agent) with semantic, epistemic (facts and rules), and episodic (procedural) interactive goals"* [112].

Of the different approaches, it is the effort by the Canadian research group [115, 116] that aligns with the objectives and goals of this work (representation of deterministic tasks and process that are suitable for investigation). The Epistemic challenge is addressed using the Cognitive and Modular Observe, Orient, Decide and Act models - referred to as C-OODA and M-OODA [115, 116]. Based on the observe–orient–decide–act or OODA loop (Figure 2.15), (developed by John Boyd [117, 118], which is still referenced and used in Air Force training), the M-OODA and C-OODA loops build upon and enhance the original concepts of OODA [119–121].

To address the Epistemic Grand Challenge of Information Fusion, M-OODA and C-OODA are advanced models that enhance OODA to address specific research aims. In order to represent tasks and processes, the OODA loop, as a simple 4 stage model of human information processing, is beneficial due to its synergy to the model of human information processing found within Human Automation Interaction (HAI) [84, 122].

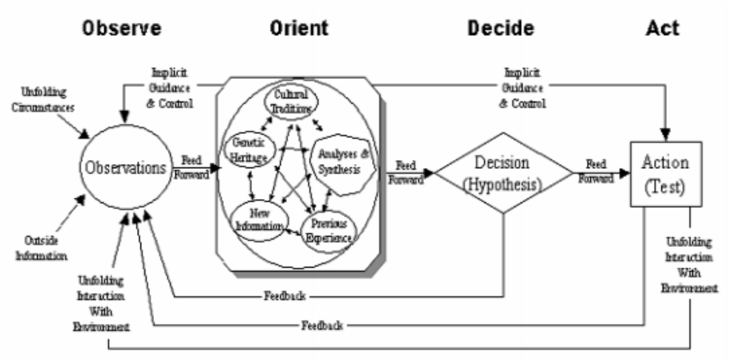


Figure 2.15: OODA loop

The Observe-Orient-Decide-Act or OODA loop (Figure 2.15) [117, 118] originally referred to the thought processing loop taught to pilots during training. It was taught that for success, a pilot would need to have a quicker loop than their opponent.

During combat, a pilot would need to observe information, (such as enemy movement, angle of attack, speed, environmental considerations, munitions and capabilities, etc.). This would be fed forward through an orientation loop and filtered, (such as training, experience, intuition, analysis, new information, objectives, etc.), to provide a decision or hypothesis. This decision, would result in either another round of information gathering and processing (observation and orientation) or an action implementation. After an action is implemented, the loop would continue until resolution or until success conditions were satisfied.

For the purpose of this research, White Forces, (who are comprised of subject-matter experts and are often made up of former RAAF pilots and contractors), perform tasks and processes aimed to emulate those of pilots. OODA could logically be used to compose, represent, describe and communicate the deterministic capabilities of White Forces. OODA can represent:

- The **observation** states of required information
- The **orientation** filters that are queried and referenced against
- The **decision** made and processing that occurs
- The **action** outcomes that are produced

2.4.4 Automation Levels and Types

Building upon Section 2.3.2, OODA [116, 118, 123, 124] as a simple four stage model of information processing aligns with the concepts within *A Model for Types and Levels of Human Interaction with Automation* by Parasuraman et al. (Figure 2.16) [125]. Prior work [125] posits the initial conceptual ideas of Levels of Automation and combines them with a simple model

TABLE I
LEVELS OF AUTOMATION OF DECISION
AND ACTION SELECTION

HIGH	10. The computer decides everything, acts autonomously, ignoring the human.
	9. informs the human only if it, the computer, decides to
	8. informs the human only if asked, or
	7. executes automatically, then necessarily informs the human, and
	6. allows the human a restricted time to veto before automatic execution, or
	5. executes that suggestion if the human approves, or
	4. suggests one alternative
	3. narrows the selection down to a few, or
	2. The computer offers a complete set of decision/action alternatives, or
LOW	1. The computer offers no assistance: human must take all decisions and actions.

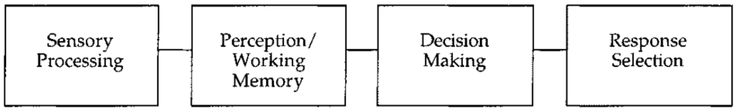


Figure 2.16: Level of Automation with a simple four stage model of information processing

of information processing (likened to OODA [116, 118, 123, 124]) to propose a method for automation design and implementation (Figure 2.17 [125]).

There is a benefit to this direct alignment between OODA for information processing and Level of Automation with a simple four stage model of information processing (Figure 2.16). This is due to the foundational work done by Parasuraman et al. in outlining Types and Levels of automation and Human Performance Consequences. OODA and Figures 2.17 and 2.16 contain four stages; an observation stage, a previous knowledge comparison stage, a decision stage and an action implementation stage. Concepts found within HAI, that deal with cognition and situational awareness, with the focus on the human operator as decision maker [116, 118, 122–125], are therefore applicable to OODA. Hence, there is strong alignment, applicability and suitability.

For White Force multipliers, this means that we can analyse tasks and processes of interest through the lens of HAI to provide automation implementors with a ranked outcome of tasks and processes worthy of further investigation. HAI focuses on human interaction (both impact and benefit) with automation, with Figure 2.18 [86,93] presenting a model of human interaction as overseer of automative tasking, and takes into account situational awareness and workload.

White Force get overloaded during distributed training exercises [101]. This is due to the workload (volume of tasks and processes they have to perform), combined with the complexity of

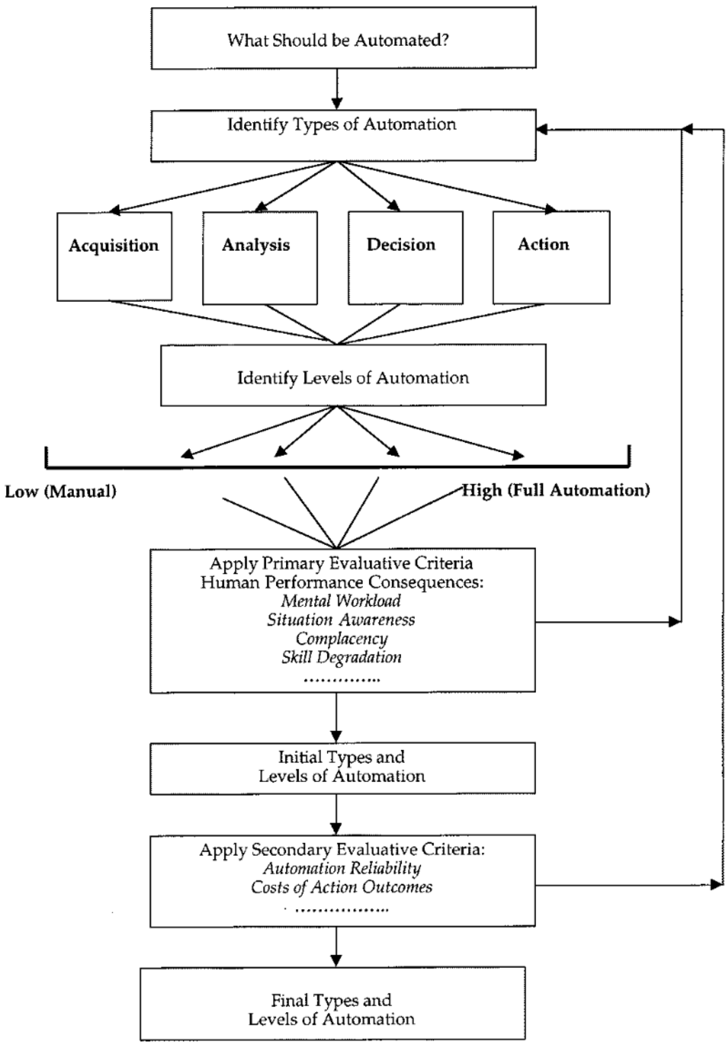


Figure 2.17: Model of Type and Level of Automation with primary and secondary performance consequences

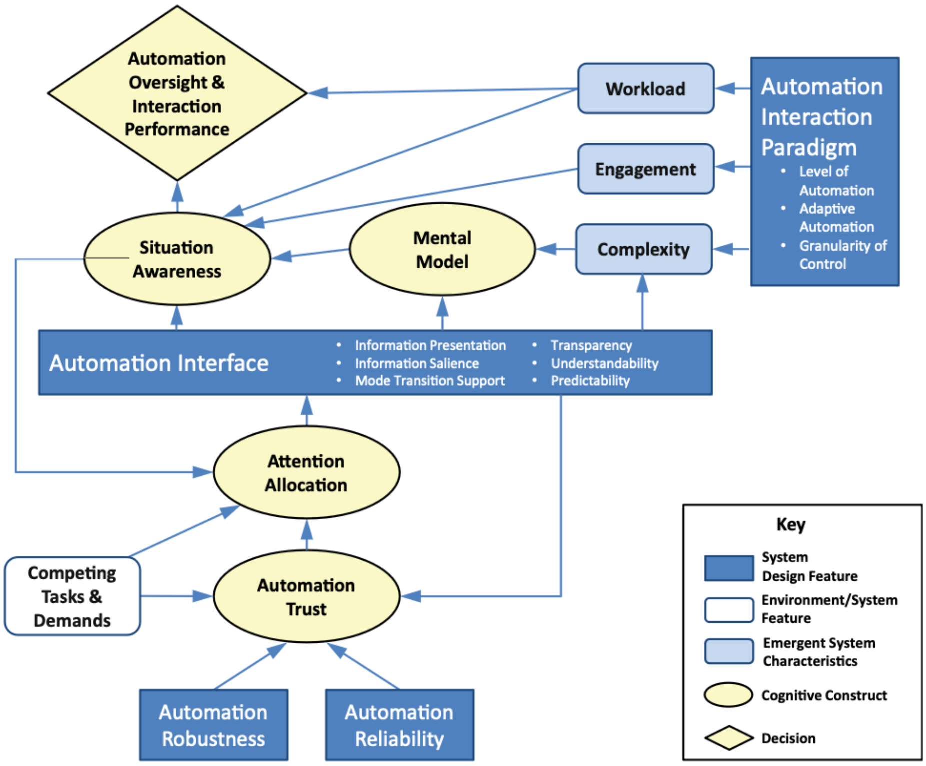


Figure 2.18: Human–autonomy system oversight model (HASO)

information required to maintain situational awareness. To reduce the burden / alleviate the workload we need to communicate candidate tasks and processes deemed suitable for automation. This substantiates the case to understand and consider of human performance consequences and HAI principles and practices. These principles and practices can inform / improve / augment / enhance / the human understanding of autonomous systems.

The general primary and secondary human performance consequences focused on a number of core concepts [84].

- *Mental workload*: Is the solution adding to the overhead of the human operator?
- *Situational Awareness*: Is the human operator able to make the correct decision?
- *Complacency*: Is the human operator disengaged with the automated task?
- *Skill degradation*: Are the skills of the human operator needed to be maintained/honed?
- *Automation Reliability / Trust*: Is the human operator able to trust that the automation solution is behaving "correctly"
- *Cost*: Is the cost/benefit ratio worthwhile to the stakeholders?
- *Interface design*: Is the interface alienating the human operator?

The Design of Human-Autonomy Systems conforms to a set of guidelines that were built upon the core concepts of human performance consequences, and tuned though application. These guidelines are: [86, 93, 126].

- *"Automate only if necessary – avoid out-of-the-loop problems if possible"* [86, 93, 126]
- *"Use automated assistance for carrying out routine tasks rather than higher-level cognitive functions"* [86, 93, 126]
- *"Provide SA support rather than decisions"* [86, 93, 126]
- *"Keep the operator in control and in the loop"* [86, 93, 126]
- *"Avoid the proliferation of automated modes"* [86, 93, 126]
- *"Make modes and system states salient"* [86, 93, 126]
- *"Enforce automation consistency"* [86, 93, 126]
- *"Avoid advanced queuing of tasks"* [86, 93, 126]
- *"Avoid the use of information cuing"* [86, 93, 126]
- *"Use methods of decision support that create human/system symbiosis, such as contingency planning and critiquing systems"* [86, 93, 126]
- *"Provide automation transparency"* [86, 93, 126]
- *"Ensure logical consistency across features and modes"* [86, 93, 126]
- *"Minimise logic branches"* [86, 93, 126]
- *"Map system functions to the goals and mental models of users"* [86, 93, 126]
- *"Minimise task complexity"* [86, 93, 126]
- *"Integrate information to support comprehension of information"* [86, 93, 126]
- *"Provide assistance for SA projections"* [86, 93, 126]
- *"Use information filtering carefully"* [86, 93, 126]
- *"Support assessments of confidence in composite data"* [86, 93, 126]
- *"Support system reliability assessments"* [86, 93, 126]

The model of the *Type and Levels of Automation* [84] serve as a foundation to define how humans could interact with automation. HAI, at its core, dictates that automation needs to be considered an enhancement of the human operator [122]. The guidelines warrant consideration when automating human-machine interactions, human-autonomy systems, or designing for HAI

These HAI guidelines, built upon the initial performance consequences (where the need for performance consequences was due to the often over reliance or over use of automation as a blanket fix-all [122]), identify friction points for the successful adoption of any proposed automation solution. The focus for the success of any system, needs to be on the human operator as decision maker. The overall impact and benefit of automation needs to be on augmenting and enhancing the cognition and situational awareness of the human operator in human-machine system.

2.4.5 Modelling and Unified Modelling Language

The importance of models, is that they are an abstraction of a real system, concept or idea, used to reduce complexity and allow predictions or inferences to be made [127, 128]. Another case for modelling is the conceptual use of metamodels as a method for the meta representation of a task or process (i.e. the procedure) independent of the underlying instance or occurrence. This relates to the frequency and similarity of tasks and processes within similar excerises [101]. Employing this modular approach to task representation offers inherent benefits in alignment with automation which is impactful to informing and generating actionable insight for White Force multiplication implementors.

Models and modelling is prevalent in the areas of research discussed so far; White Force [101], Complexity [129], Information Fusion [112, 130, 131], OODA [115, 116, 123, 132], Human Factors [108, 110, 133], Hierarchical Task Analysis (HTA), Critical Decision Method (CDM) and Event Analysis of Systemic Teamwork (EAST)(a collection of methods) [107, 108, 133], Endsleys Situation Awareness model [134], and Automation and Human Automation Interaction [64, 122, 123, 125]. We argue that a feasible method / approach should employ models to reduce complexity, convey information and present findings in an accessible and communicable structured format.

2.5 The need for a systematic methodology or framework

White Force assisted training encompasses chaos and complexity. Complexity manifests as the training follows a mix of ad hoc processes and procedures, which are facilitated by a diverse cohort of individuals [5], to converge and deliver consistent outcomes [12,22,23,135]. This is rendered more chaotic when things do not go to plan and White Forces need to adjust for immediate unexpected occurrences and unplanned deviance. This dynamicism in the ecosystem resists the application of automation workflows due to its instability [73,76,79,90,93,96,98,136,137]. An overarching systematic methodology or framework does not exist and is therefore required to identify, represent, classify and automate the White Force assisted training domain.

As part of an overarching systematic methodology or framework, it needs to allow for the:

- Identification of performed/routine tasks and processes [138, 139]. This is the need to identify the task and process/routines that are performed by White Forces. This is due to the need for analysis of appropriate candidates for process selection for automation [140,141]. No sustained investigation and formalised approach has been undertaken to identify the breadth and depth of performed/routine tasks and processes. Without identification, it is hard to know the full range of tasks that exist and that could be appropriate candidates for automation [12,23]. A systematic approach for the identification and resolution of performed/routine ad hod tasks and processes, is required to understand the relationship between White Force capabilities and training outcomes/understand what is being done [108, 109, 142–145]. Work by Stanton et al. [146], while focused on C4i domains, is primiarily focused around the ergonomics and human factors of multiple agents and "systemic teamwork". This has manifested itself as the development of the EAST framework. *"C4i is defined as the management infrastructure needed for the execution of a common goal supported by multiple agents in multiple locations and technology. In order to extract data from complex and diverse C4i scenarios a descriptive methodology called Event Analysis for Systemic Teamwork (EAST) has been developed. With over 90 existing ergonomics methodologies already available, the approach taken was to integrate a hierarchical task analysis, a coordination demand analysis, a communications usage diagram, a social network analysis, and the critical decision method"* [146]. This work, is excellent for the need for identification, but does not address communication.

- Communication of performed/routine tasks and processes [138, 139]. This is the need to communicate White Force tasks and process/routines, and the need to represent components of identified tasks and process/routines to a wide audience. Once identification of performed/routine tasks and processes is complete, it is necessary to communicate in a standardised way. This is to facilitate communication to automation/multiplication implementors, and for validation, testing and verification of the finalised capability multiplication solution. The need for communication is also part of process standardisation, [138] to identify rule based, structured, routine tasks, that can mimic human behaviour [138, 140, 147–151]. Currently, White Forces comprise SME's, (Pilots, trained C2, etc.) that contain a depth of information that has been learnt through training and experience over time. This information is used to conduct and execute the successful application of the required tasks and routines/processes for successful training outcomes [5, 12, 22, 23]. This indicates a need to understand the information and decision-making required to conduct and perform White Force tasks [112,117,119,120,152,153]. Therefore, there is no approach currently known, to easily communicate what the identified tasks are, for communication in a standardised way. If we had a method for the modelling, communication and knowledge/information capture of the processes/routines that White Force perform, it would be possible to standardise these tasks into rule based, structured routine tasks or at least identify those tasks that are and follow the process selection criteria of repetitive, rule based, high manual effort, complicated/clear classification. [138, 147–151, 154]. Work by Blasch et al. [112], focuses on the fusion of low level multivariate sensor data into high level information for situational awareness for human operators. *"High-level information fusion is the ability of a fusion system to capture awareness and complex relations, reason over past and future events, utilize direct sensing exploitations and tacit reports, and discern the usefulness and intention of results to meet system-level goals"* [112]. This is an excellent presentation and guide to Information Fusion. The current thesis is working on the exact inverse of the aims of Information Fusion - to decompose and breakdown decision making and identify sources of information inputs. The method of Blasch et al. [112] to use OODA as a model of decision making is excellent for communication but does not address classification.

- Classification of performed/routine tasks and processes [138, 139]. This is the need to classify identified and represented tasks and processes/routines into sets of tasks that are suitable for automation versus those that aren't. As part of process selection during analysis of automation candidates, it is necessary to identify processes/routines that can relieve White Forces from performing non-value adding tasks, which would allow White Force to focus on cognitively more demanding tasks, including those that are chaotic in nature and do not conform and resist automation workflows due to instability [141, 154]. Currently, there is a recognition for the need for task classification to reduce the cognitive burden placed upon White Forces [5]. This is due to the fact that White Forces perform different roles and are able to fill all gaps due to being skilled decision makers. No reliable evidence exists for the classification of White Force tasks that require expertise, knowledge and are chaotic/complex in nature, versus those that are repetitive, procedural, or are easily reproduced [102, 104–106]. Therefore, it is necessary for a method for the identification and classification of tasks and processes/routines that are suitable for automation due to being deterministic in nature, rule based, structured and repetitive [138, 147–151, 154], to reduce and relieve the current cognitive strain placed upon White Forces. Work by Snowden et al. [129], presents Cynefin, *"a new perspective on leadership and decision making that's based on complexity science. The result is the Cynefin framework, which helps executives sort issues into five contexts: Clear, Complicated, Complex, Chaotic and Disorder"* [102]. This work is pivotal in guiding the development of this research. The defitions of deterministic and non-deterministic are fundamental for the developmemnt of these concepts used for classification.

- Selection of candidate process routines that are suitable for automation implementation [138, 139]. This is the need for the ranking of identified, represented, communicated, and classified tasks into an ordered set of suitable candidates for automation. As part of the analysis stage, before design, production, testing and operation, processes need to be selected that are (preferably): repetitive, rule based, require high manual effort, observations and action outcomes are structured and consistent, the process involves limited human intervention, is not cognitively demanding, and does not require a decision, interpretation or assessment of results [138, 140, 141, 147–149, 154]. Currently, tasks exist and have been identified that meet these criteria, but little information other than observational data exists to support these claims [5]. White Force tools, processes and procedures have been built around the existence and continued existence of White Forces [12,23]. Therefore, there is not a clear method for identification of automation opportunities in White Force tasks and processes/routines [76,77,81,85,86,122,155–161]. To resolve this, it is required there be a method for the ranking and evaluation of tasks and processes/routines that support and are suitable for automation and capability multiplication [138, 140, 141, 147–151, 154, 162]. Automation has not been addressed by previously identified areas of concern, therefore the work by Sheridan and Parasuraman [84, 122] has been vital in identify the methods, techniques and best practice for automation application [84, 122].

Based on the prior work we have reviewed in this chapter, we have begun to formulate a consolidated method / approach to enable White Force capability multiplication. This proposed method/approach spans Cynefin, Information Fusion, HAI, LoA, EAST and Human Factors methodologies, and Modelling, Meta-modelling and abstractions.

To consolidate these theories and areas of research, we need a unified approach in conjunction with a consistent language for the representation and presentation of ideas and knowledge to multiplication implementors. This approach needs to address the presentation of theories in a consistent and structured way (i.e a framework) and to communicate the outcomes and deliverables of the methods of the method/approach to a wide and varied audience (i.e. model driven approach as a means of visual communication of concepts and ideas).

Cyenfin will be used as a deliminator in identifying ordered and deterministic task and process and understanding complexity [129]. Human Factors research that has investigated C4i training case studies [110], team and cognitive ergonomics, and associated methodologies, will be used for knowledge capture and presentation [107, 108, 110].

Information Fusion approaches that aim to address the Epistemic Grand Challenges, will be used for information representation for cognition and decision making [130]. Human Automation Interaction design and theory will act as the function for ranking and selecting tasks and process for automation implementors.

The related work explored in the preceding sections aims to consolidate established approaches and areas of research to generate a novel method/approach, for a niche domain/problem that has no clear current alternative. Other methods/methodologies may be appropriate. Given the end-to-end nature of White Force (automation capability multipliers for Distributed LVC Synthetic Training), and the fact that this area of research and training approach is an ongoing research problem, (where development, evolution and improvements are being made constantly), it is necessary to generate a method/approach to begin to identify potential solutions. Therefore:

- Cynefin will be used as a model to define deterministic vs nondeterministic/emergent behaviour, through a use of domains and strategies. This definition will be used as multiplication criterion selector - emergent / non-deterministic behaviour is unable to be defined/captured, because by doing so, it removes the emergent behaviour and attributes. Therefore, a task could potentially be automated, only if its states, actions, operations and decisions can be defined, modelled and represented.

- OODA is a teaching tool used by Air Force pilots in combat situations, with an emphasis on having a faster loop than your opponent. Approaches from Information Fusion, to address the Epistemic Grand Challenge, build upon OODA, in the forms of C-OODA and M-OODA as a tool to address how information could be presented in High Level Information Fusion systems. OODA will be used as a cognitive task visualisation model.

- EAST is a framework that combines a series of different Human Factors and ergonomic methods to measure the relationships between entities in C4i domains. The application is relevant and battle tested, having been applied to C4i training, and a wide variety of other situations (cycling, police, train operations, flight control, etc.). The phases of EAST (collection, analysis and representational methods), will serve as a template to structure any proposed method/approach on.

- Levels of Automation forms a foundation of Human Automation Interaction. Automation is the application of systems to enhance human performance. Using complexity definitions as scope limiters, OODA as task and process models and EAST as data collection and analysis methods, it is possible to identify where/what processes are suitable for application of Human Automation Interaction design and solutions.

- UML Models and Meta-Modelling (in the form of task models) will be used convey information in an easy and approachable format. Modelling will be used to abstract complexity and represent information in the form of in diagrams, metamodels, task models and charts.

2.6 Chapter Summary

This chapter first explored the background literature relating to White Forces, LVC Distributed Synthetic Training and Modelling and Model Driven Engineering as means to comprehend the tasks and roles White Force perform. Following this, we examined the prior work related to task classification (Cynefin), task and process definition and extraction (EAST, HTA and Human Factors methodologies), task model representation (Information Fusion and OODA), modelling, and the core principles of Human Automation Interaction, as a set of guidelines for White Force multiplication through automation.

The literature has surfaced key considerations for automation such as identification, representation/communication, classification, assessment for automation. These need to be sequentially investigated in order to ascertain if automation will be applied to chaotic and unstable environments. In the next section we will discuss our Research Questions that have been framed to address these considerations.

Chapter 3

Research Goal, Hypothesis, Questions and Methodology

This chapter presents the research hypothesis and questions of this thesis, based on the background introduction of White Force as facilitators to LVC Distributed Synthetic Training. To support the hypothesis and questions, this chapter also introduces the research method (including criteria and metrics) for method / approach construction, development and evolution, as well as the method for testing and evaluation (including scenarios used and the method for the final validation).

3.1 Research Goal and Hypothesis

Automation is purported to offer the benefit of consistency, alleviation of mundane activities and cognitive load on staff, and accelerate throughput. Automation is a common approach to scale up and facilitate capability multiplication. But it is not, a "one size fits all" solution or magic bullet. Automation requires stability (rule-based nature), rigidity (structured data), consistency (routine tasks) and frequency (repetitiveness), amongst other attributes.

To effectively apply automation, White Force tasks and processes/routines require a level of stability. This work intends to demonstrate, that transformation from nondeterministic to deterministic, is possible to achieve stability for automation. This, if successful, will be applied to various processes.

Therefore, our hypothesis, as an overarching goal, aims to address the following:

How can we multiply White Force capabilities with automation?

White Force exists in an inherently complex and chaotic environment of distributed synthetic training (i.e. the domain). This is due to the skills, knowledge and expertise required; the often inconsistent and fault intolerant execution; and the need to inject complex/chaotic elements to create unknown and unexpected scenarios that the training audience need to adapt and overcome (in war, anything goes).

It is necessary to understand if we can **identify** the tasks and processes/routines that White Force perform through the identification of current White Force activities (including the hidden and ad hoc tasks and processes/routines). From this *identification*, we need to adequately understand if we can **model** and represent (to communicate) the decision-making and required variables (observations, action outcomes, prior knowledge, expertise, experience, etc.). From this *model* of task functionality, we need to ascertain if we can **classify** tasks into nominal groups of difficulty or complexity, and separate tasks into those that are suitable for automation vs those that required human decision-making and judgement. Once *classified*, we need to assess if we can apply **automation** best practice/principles (human automation interaction guidelines, levels of automation, robotic process automation, etc.) to the selected tasks to generate a ranked and ordered list of candidates of automation for capability multiplication.

3.2 Research Questions

To address our goal and requirements, it is necessary to create a framework to identify, represent, communicate, classify and rank White Force tasks and processes/routines for automation suitability. Therefore, we decompose our effort into the following areas of concern; the identification, communication, representation, classification and automation ranking of White Force distributed synthetic training tasks and processes/routines. To do this, the following need to be addressed.

1. **How can we systematically identify White Force tasks and processes/routines, including those that are hidden or ad-hoc, for further communication, representation, classification and investigation?** By systematically identifying and ensuring coverage of tasks (including ad-hoc and hidden), we aim to reduce and minimise miscommunication and/or misrepresentation.

2. **What approach can be used to adequately communicate White Force tasks and processes/routines to a wide and varied audience of automation implementors, to facilitate discussion and review for further task representation?** Communication of tasks is important for the need to share and communicate knowledge in a digestible format by the varied audience (including shareholders, stakeholders, and capability multipliers). Information needs to be understood by an intelligent but potentially non-expert

/ technical audience.

3. **What method can be used to suitably capture and represent, in a structured and standardised way, the variables (i.e. information and decision-making) of White Force tasks and processes/routines, to assist with classification of tasks into suitable categories of automation?** The need for representation, is a precursor to function / task automation design and implementation, with identified observations, orientations, decisions and actions, acting as input/output parameters and variables, algorithms, procedures and functions, and abstracted knowledge.

4. **How can White Force tasks and processes/routines be acceptably classified into cognitively demanding tasks / tasks that resist automation due to instability and are chaotic/complex in nature, versus those that are repetitive, structured and rule based?** The onus of task classification is due to the context of how automation will be implemented. Differing organisations may have different resourcing and requirements available for automation multiplication, and may be subject to external or internal influences regarding the desired classification application

5. **What selection criteria (e.g. repetitive, rule based, structured, consistent, and does not require a decision, interpretation or assessment of results) can be used to fairly rank and order candidates tasks for automation application?** The selection criteria can be considered arbitrary. This is based on the needs and requirements of who is assessing it, the desired outcomes of application (reduction of resource usage or multiplication of capability, etc.), and internal and external factors (resource allocation, cost, time, shareholder buy in, etc.). This is because there is no "one size fits all" solution or magic bullet for automation.

By addressing the above, we can stabilise and introduce structure into the domain of White Force tasks and processes/routines. This would manifest as a framework, that serves as a road map, to navigate the application of automation and introduce stability into dynamic and chaotic environments that resist automation.

3.3 Research Methodology

This research aims to address and answer the Research Hypothesis (Section 3.1) and Research Questions (Section 3.2) proposed above. To do so, it will be necessary to develop and validate a method / approach to address each Research Question. The methodology for this research can be split into two major components.

- **Methodology for method / approach construction**

- **Methodology for testing and evaluation**

The methods for method / approach construction, and testing and evaluation work together in a *Qualitative top down cyclic mechanic* and follow an *Exploratory sequential mixed methods* approach. This means that an initial attempt is made at method / approach construction to address qualitative measures (Section 3.3.4), then is tested and evaluated using sample scenarios and exercises (Section 3.3.6). This approach is analogous to the iterative and incremental method of software development (Figure 3.1). Due to domain requiring discovery, this approach was selected to allow adoption to new information. Automation typically requires software to manifest, therefore the choice of a software development lifecycle - specifically the Iterative and Incremental Software Development method, aligned with the iterative method of discovery. Once a valid method / approach that aims to address the proposed Research Questions (Section 3.2) is constructed, tested and evaluated internally, the final validation step is to communicate the proposed method / approach externally, to members of White Force and members of AOSC (See Chapter 5).

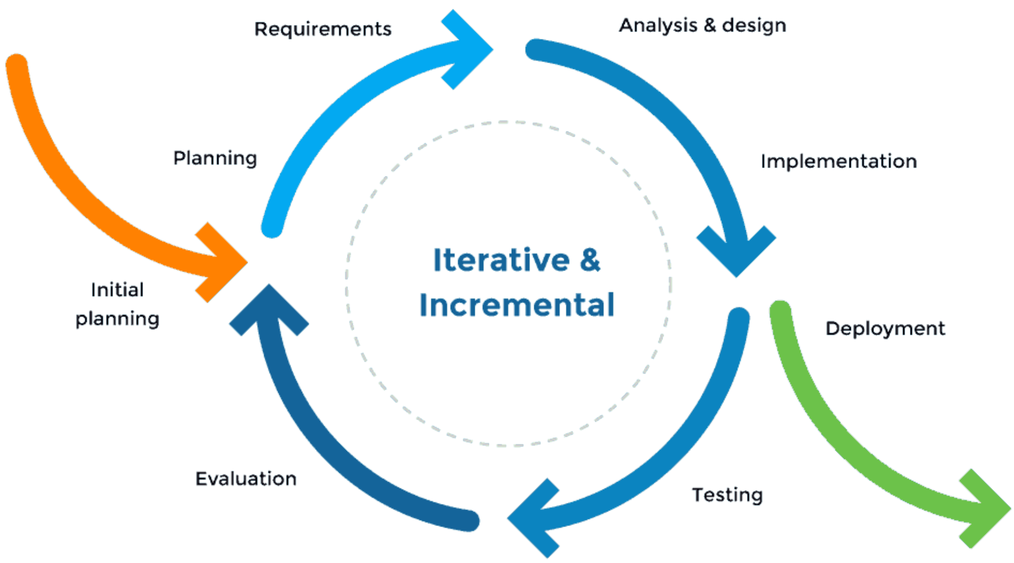


Figure 3.1: Iterative and Incremental Method. Source:[1]

To address RQ1: **How can we systematically identify White Force tasks and processes/routines, including those that are hidden or ad-hoc, for further communication, representation, classification and investigation?** It will be necessary to investigate current methodolo-

[1]https://www.plutora.com/blog/software-development-life-cycle-making-sense-of-the-different-methodologies

gies for task discovery and the applicability to the White Forces domain. These identification methods need to be suitable for the range, or breadth and depth of tasks and processes/routines, that exist within White Forces, and are an appropriate and practical approach. Preferably, these methods need to exist, founded in existing theory and research, and have been applied to similar domains.

To address RQ2: **What approach can be used to adequately communicate White Force tasks and processes/routines to a wide and varied audience of automation implementors, to facilitate discussion and review for further task representation?** It will be necessary, that once a set of currently performed, hidden and ad hoc tasks and processes/routines are identified (RQ1), an approach for the communication to a wide and varied audience is required. This approach needs to account for complexities in the underlying system, therefore, it is preferable for an approach to present information in an abstract way, using easily communicated visual elements that are approachable for different audiences.

To address RQ3: **What method can be used to suitably capture and represent, in a structured and standardised way, the variables (i.e. information and decision-making) of White Force tasks and processes/routines, to assist with classification of tasks into suitable categories of automation?** It will be necessary, after having a set of identified and communicated tasks and processes/routines, to represent these variables and components. Using a similar approach for communication (RQ2), it is preferable that this method leverages abstraction and visual communication, to convey meaning and understanding to a wide audience, and to assist with further investigation and classification. This method needs to be applicable to the tasks and processes/routines White Force perform, and be grounded in existing theory and practice (preferable). This representation method needs to be able to capture and represent the observational input data, the prior knowledge and filtering required, the decision-making and understanding, and the required outcomes or transformation/actions. With information represented in this way, classification of task components and elements are made easier due to the application of structure and stability that White Force currently resists.

To address RQ4: **How can White Force tasks and processes/routines be acceptably classified into cognitively demanding tasks / tasks that resist automation due to instability and are chaotic/complex in nature, versus those that are repetitive, structured and rule based?** It will be necessary to assess and classify the identified (RQ1), communicated (RQ2) and represented (RQ3) tasks and processes/routines into categories that are suitable for automation and deterministic in nature. These tasks are often repeatable, predictable, deterministic, structured, rule-based and stable. This classification will also be used to identify tasks and processes / routines that are unstable, chaotic and complex in nature, non-deterministic, require decision / sense making and requires human involvement. These complex tasks would benefit from more human resource allocation to improve overall fidelity vs the classified and appropriate candidate tasks for automation.

To address RQ5: **What selection criteria (e.g. repetitive, rule based, structured, consistent, and does not require a decision, interpretation or assessment of results) can be used to fairly rank and order candidates tasks for automation application?** It will be necessary, once identified (RQ1), communicated (RQ2), represented (RQ3) and classified (RQ4), the final set of candidate tasks and processes/routines be evaluated for automation suitability. This requires selection criteria, that candidate tasks will be evaluated against to generate task ranking and ordering based on desired outcomes for automation implementation. This evaluation needs to consider existing automation theory and best practices, including task / cost ratios, task frequency, task impact, task effort and projected lifetime of the task. This evaluation and selection criteria need to also factor in automation and human automation interaction best practices, distilled from existing theory and industry implementations.

3.3.1 Methodology for method / approach construction

Any method / approach needs to identify a set of tasks and process (RQ1), from which to create a subtree of tasks and processes that are suitable for communication to a wide audience (RQ2) and representation of the variables and decision-making required (RQ3). From this we can classify and organise tasks into categories of candidate tasks and processes/routines that resist automation due to instability vs those that are structured (RQ4). From this classification, selection criteria will be used to generate a ranking of tasks (RQ5), where future work can be undertaken to understand how best to apply resources, to create better/more efficient exercises using limited resource availability.

Due to unforeseen circumstances[2], method / approach construction needed to occur asynchronously from SME's in AOSC, DST Group. Due to this, a set of qualitative criteria and measures were developed in conjunction with the needs and requirements of SME's in AOSC, and from the research questions (Section 3.2). This allowed for internal testing and evaluation without the need for external validation at each iterative stage.

These criteria were used to build, generate, and iteratively develop each step of the method / approach. In conjunction with the criteria, the method used to drive development of the method / approach was a qualitative approach, by using a theoretical lens to view the problem of White Force. A literature review of related work (Section 2.4) was conducted to identify similar domains and applications, as well as identification of existing frameworks. Once identified, existing frameworks were augmented and enhanced to address gaps identified by the developed criteria. In this way, through literature review of existing work, application and development

[2] https://en.wikipedia.org/wiki/COVID-19_pandemic_in_Australia

of existing methods to address criteria requirements, and testing and evaluation against criteria measures, a framework was gradually built up and developed to address the lack of existing method / approach.

3.3.2 Establishing suitable criteria

Capability multiplication is fundamentally difficult due to complexity. This complexity can take the form of task, variable data input, situational and security concerns, and domain knowledge. It is necessary to establish, for the development, testing, and evaluation of any proposed method / approach, a set of criteria for measurement for requirements validation. These criteria need to be defined in collaboration with the SME's in AOSC. By doing so, any threats to external validity are negated, due to the researcher using the same measure for outcomes (i.e. criteria), that SMEs will be using, even though the researcher is not an SME in the domain of White Forces. This also ensures any method / approach is generic (i.e. given a set of measures (i.e. criteria), and the mechanism for method / approach development and generation (iterative and incremental), another individual, given a context, can and would produce something). The following criteria are customised to the research questions, and validated in collaboration with the requirements of SME's in AOSC. Any further criteria can be appended and integrated in future iterations of the method / approach as required. It is important to note that while criteria is established in collaboration with SME's in AOSC, as part of the development and evolution of the method / approach, the application of the criteria is conducted internally. This allows for rapid testing and evaluation against the metrics and measures of the suitable criteria. A final validation step is required, with members of AOSC, to demonstrate the application of these criteria, and allows for an opportunity for SME's to conduct their own testing and validation to ensure it meets specification

3.3.3 Research questions as practical criteria initiators

From Section 3.2, it is possible to derive and extract a set of criteria, that any proposed method / approach will need to address, in order to be appropriate to meet the requirements of SMEs in AOSC, for generation, development, testing, evaluation, validation and usage.

1. **How can we systematically identify White Force tasks and processes/routines, including those that are hidden or ad-hoc, for further communication, representation, classification and investigation?** When aiming to identify the full range of tasks and processes/routines of an event, scenario or exercise, including the hidden and ad-hoc tasks and processes/routines, we are aiming to search and identify the full *breadth* and *depth* of tasks and processes that make up the full *coverage* of components required.

2. **What approach can be used to adequately communicate White Force tasks and processes/routines to a wide and varied audience of automation implementors, to**

facilitate discussion and review for further task representation? Having a method to identify tasks and process/routines is important, as is the requirement to communicate and share knowledge and information. Therefore, to communicate to a wide audience, it is important to use an *existing theory* that is tried and tested to communicate this information. This method of communication is able to present and highlight the *alignment* and relationship between tasks and processes/routines, to allow decision makers to visualise and see the linkages and chains between the full *breadth* and *depth* of components.

3. **What method can be used to suitably capture and represent, in a structured and standardised way, the variables (i.e. information and decision-making) of White Force tasks and processes/routines, to assist with classification of tasks into suitable categories of automation?** By having an overview and *coverage* of tasks and processes, represented and communicated using *existing theory*, to capture the *alignment* and relationship between tasks and processes/routines, it is therefore necessary to use appropriate *representation* to adequately capture components in an abstracted and approachable way. This needs to be grounded in *existing theory*, to allow for further investigation and classification. Whatever methods are being used, need to be flexible enough to allow for *reproducibility* for different types of tasks and allow for *dynamicism* for differing overarching exercises/scenarios/case studies/domains of applicability.

4. **How can White Force tasks and processes/routines be acceptably classified into cognitively demanding tasks / tasks that resist automation due to instability and are chaotic/complex in nature, versus those that are repetitive, structured and rule based?** In the same way, that the *coverage* of tasks need *representation* to communicate *alignment* and be applicable to a variety of domains (*dynamicism* and *reproducibility*), they also need to be grounded in *existing theory*. This is especially true for the classification of tasks into domains of instability vs stability of automation application.

5. **What selection criteria (e.g. repetitive, rule based, structured, consistent, and does not require a decision, interpretation or assessment of results) can be used to fairly rank and order candidates tasks for automation application?** The selection criteria for automation ranking will need to be based on *existing theory*. It will need to factor different types of tasks, which have been given structure and rigidity due to the identification, communication, representation and classification of all existing tasks and processes/routines.

3.3.4 Practical criteria for generation, development, testing and evaluation

Based on the breakdown of research questions to criteria, it is possible to extract key elements of a proposed practical criteria that any method / approach needs to address for generation, development, testing, evaluation, validation and usage to meet the requirements of SME's in AOSC. While not fully formed/examined, this is an initial attempt at the generation of a set of criteria to achieve the development of a method / approach. This, in itself, is a contribution extending from this research.

1. Coverage (Figure 3.2) - are the range of tasks and processes that are required for White Force to perform during a given research or training exercise.

    (a) Breadth - deals with the tasks required and which are suitable for investigation for multiplication. This is extended through the use of existing theory of complexity from Cynefin (Section 2.4.1) (i.e. determinism vs non-determinism).

    (b) Depth - deals with the macro, meso, micro representations of tasks. Hierarchical Task Analysis from Human Factors methodology (Section 2.4.2), can be utilised to determine how much detail is required for an appropriate task representation

2. Alignment (Figure 3.3) - is the mapping between objective, goal and deterministic tasks required.

3. Representation (Figure 3.4) - is the visual presentation (Section 2.4.5) and communication of alignment and coverage. Information Fusion (Section 2.4.3) knowledge representation follows an inverse of the approach required here (multiple data sources fused into a singular decision aid vs a decision being made and the decomposition of tasks required to represent the information required to make said decision)

4. Reproducibility (Figure 3.5) - is the defined and reproducible procedure that is communicable to stakeholders and multiplier implementors. An example is any such step by step framework, such as EAST (Section 2.4.2).

5. Dynamicism (Figure 3.6) - is the side effect of a reproducible method. It can work given multiple similar problem structures. An example is the EAST framework, that is applicable to multiple domains of networked communication and teamwork (Section 2.4.2).

6. Existing theory (Figure 3.7) - is the requirement that any method / approach be generated and developed, using existing theories and frameworks. This will lead to increased validity and robustness (Section 2.4).

7. Effort (Figure 3.8) - is the final and most important criteria. Not addressed in Section 3.3.3, due to its prevalence in each RQ. Given an infinite amount of time, anything can be multiplied. But given a finite amount of resources, what is the least amount of effort/resources/cost required to implement any method / approach step. Therefore, this

can be used as a cost metric and can influence the other criteria listed. The problem then becomes to discern what is most suitable for multiplication in the least amount effort without having to conceptualise entire systems

As part of these criteria, Section 3.3.6 deals with the associated metrics and measures for testing and evaluation. To assist the reader with representation of the proposed criteria, and the desired end state for success as part of testing and evaluation, a simple set of flow charts follow, that highlight the relationship between criteria and associated measures and metrics.

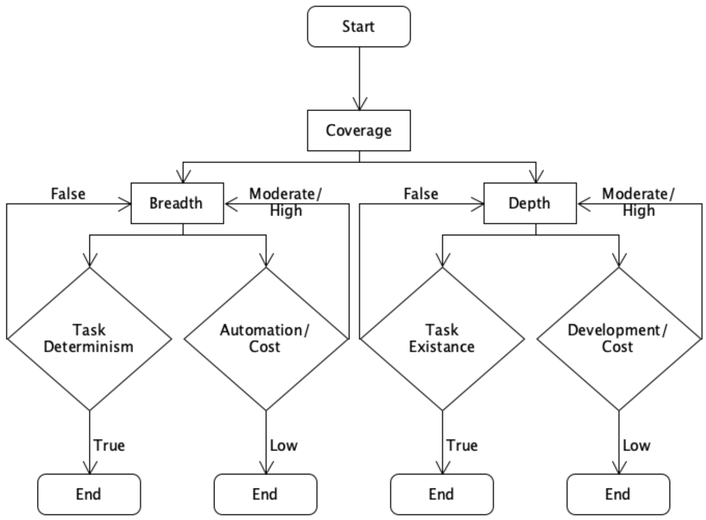


Figure 3.2: Coverage is split into Breadth and Depth. Breadth is concerned with the deterministic and automatable tasks and processes and the method of identification. Any method needs to achieve this, therefore the goal is True. Breadth is also concerned with effort required in terms of automation effort or cost effort, therefore the goal is Low effort. Depth is concerned with the discovery of known, hidden and ad hoc tasks and processes and the method of identification. Any method needs to achieve this, therefore the goal is True. Depth is also concerned with effort required in terms of development effort or cost, therefore the goal is Low effort.

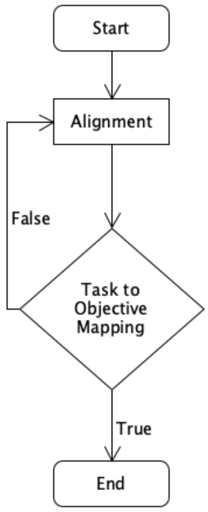


Figure 3.3: Alignment is concerned with the method for mapping tasks, goals and objective. This resolves into a binary value with the goal being True.

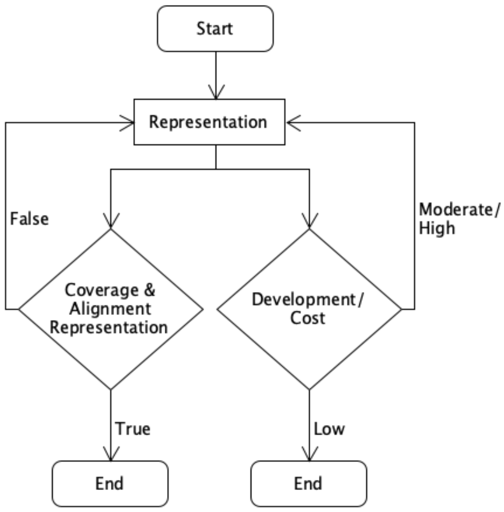


Figure 3.4: Representation is concerned with the method for communicating Coverage (both breadth and depth) and Alignment. If methods exist for both of these criteria, and a method also exists for the representation and communication of these criteria, then the goal end state is True. Representation is also concerned with the effort required to do this representation in the form of development effort and cost, therefore the goal is Low effort.

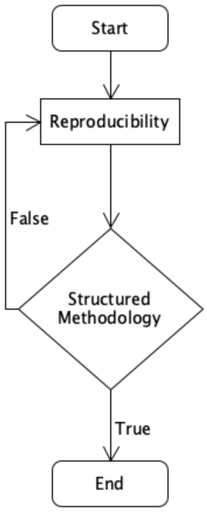


Figure 3.5: Reproducibility is concerned with the method / approach and the ability to replicate and reproduce the steps for task/process identification and representation. Therefore, having a structured and reproducible methodology is the desired end state, therefore the goal is True.

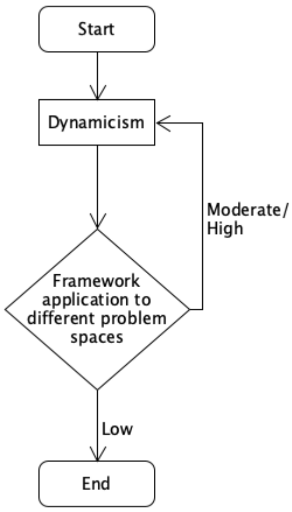


Figure 3.6: Dynamicism is concerned with the ability of any proposed method/process to be applied to different domain problem spaces, exercises and processes. Therefore, the proposed method / approach needs to require a Low amount of effort to be applicable over multiple areas of concern.

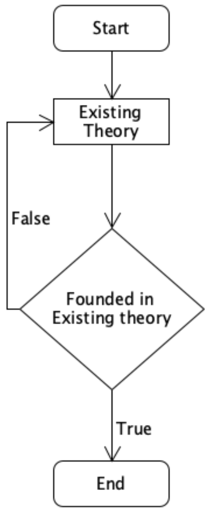


Figure 3.7: Existing Theory is concerned with building upon existing knowledge and research, to generate a method / approach which is grounded and established. Therefore, by using and incorporating existing theories, methods and frameworks, the goal state is True.

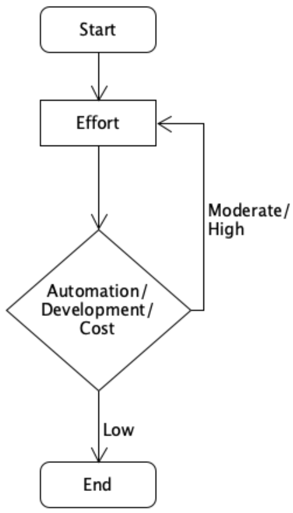


Figure 3.8: Effort is an overarching or meta criteria, in that it is impactful upon other criteria. Effort is classified as automation, cost, development, time, etc., and is representative of factoring in real world limitations into the consideration of the development and evolution of this proposed method / approach. At any stage, the desired goal is to be the lowest amount of effort required to achieve anything.

3.3.5 Methodology for testing and evaluation

For testing and evaluation of the generation and evolution of any method / approach, and following the qualitative top down cyclic mechanic of iterative and incremental development, it is necessary to test and evaluate the method / approach against scenarios and exercises that are a representative subset of the type of tasks and processes performed by White Forces. Scenarios used in the testing and evaluation are either provided by SME's in AOSC, provided by external SMEs who are experienced RAAF pilots (MilSkil[3]) or created by the researcher (Section 3.3.7). By using curated heuristics, to provide a qualitative measure against the proposed criteria (Section 3.3.4 - Practical criteria for generation, development, testing and evaluation), the researcher is able to bypass the need for external human participants and instead is able to attempt worked scenarios with the internal supervisory team.

3.3.6 Measures for testing and evaluating practical criteria

For each criterion, it is necessary to have measures to check the validity of the method / approach throughout each stage of development and evolution. These measures will assist the researcher to determine how best to proceed and at what point the method / approach has addressed all the research questions and associated criteria. The measures presented are an initial representation of some of the factors that may need to be considered. For some measures, it is possible to qualify, as determined by the implanting organisation, a "Low, Moderate, High" scale, for a quick qualitative assessment. These criteria measures (Table 3.1) (e.g. task existence is a binary true/false value, with the task existing or not) are not entirely accurate or exhaustive, but serve the purpose of providing a quick measure to determine if a task/series of tasks are worth assessing for suitability for multiplication.

Table 3.1: Criteria metrics

	True or False	**High, Moderate or Minimal Effort**
Coverage — Breadth	Is the Task deterministic?	Automation, Cost
Coverage — Depth	Does the Task exist?	Development, Cost
Alignment	Is there a mapping between Task and Objective?	—
Representation	Is Coverage and Alignment represented?	Cost, Development
Effort	—	Automation, Cost, Development,
Reproducibility	Is there a structured methodology?	—
Existing theory	Is there a foundation in existing theories?	—
Dynamicism	—	Framework application to different problem space

1. Coverage - **Effort (Low/Medium/High), Task Existence can be determined (True/-False), Deterministic in definition (True/False)** - is affected by effort (given cost, development or automatability effort, how do we proceed with coverage with the limited available resources), is in a definable binary state (it is a necessary task/process that is required and is identified during coverage), and deterministic in nature (the task identified can be defined and structured in a meaningful way for further examination). Coverage can be described as a top down qualitative cyclic mechanic for the rapid evaluation of tasks for the potential for automation. Using a card game analogy such as poker, high

[3]https://www.milskil.com/

level tasks are defined using breadth coverage to ascertain quickly if the task is worth effort (initial cards are played and during the first round of betting, it is possible to fold/raise/call). If the task is suitable, then depth coverage is used to identify the deterministic and binary state of the task for further evaluation.

(a) Breadth - **Effort (Low/Medium/High), Deterministic in definition (True/False)** - is a subset of coverage, it can be thought of as breadth-first search for traversing tree data structures, in that coverage and alignment form a tree structure of necessary tasks required to achieve the end state of the overall objective (i.e. the root node). Breadth coverage can be likened to playing a hand of poker - given a set of tasks (hand of cards), the user will evaluate each set of tasks, against a priority list of effort and determine if further examination into the automatability is plausible (a player will evaluate the hand of cards dealt and determine during the first round if to fold/raise/call).

(b) Depth - **Effort (Low/Medium/High), Task Existence can be determined (True/-False)** - is a subset of coverage and is similar to depth-first search. Like breadth, when addressing depth coverage, an identified task will be evaluated, until a meaningful deterministic structure is defined for further examination. Using a poker analogy - depth first relates to the series of betting rounds played until the pot is won, or the hand is lost.

2. Alignment - **Linking of Objective and Tasks (True/False)**- is a binary state, in that it needs to exist. Overarching objectives of any scenario under review will form the basis of the root node, from which all coverage, representation and effort will stem. The measure is quantitative 1 or 0 (yes or no)

3. Representation - **Effort (Low/Medium/High), Coverage and Alignment (True/False)** - is the visual representation of coverage and alignment. Once coverage has identified tasks for further examination in a binary and deterministic way, that align to the overarching objective, it is necessary to represent these structures in a visually communicative way to convey knowledge and information. The measure for this is quantitative because, even though it is heavily influenced by effort, it is still binary for the identified tasks. ((Coverage + Alignment) / Effort)

4. Reproducibility - **Structured Steps (True/False)** - is a byproduct of a structured approach to identifying coverage, alignment and representation. It is binary in that the end product of any proposed method / approach will need to be reproduced, therefore having a structured and written/communicated method / approach with steps achieves this measure.

5. Dynamicism - **Problem Structure Similarity(Low/Medium/High)** - is a byproduct of the method / approach. While designed for a specific use case, the potential for future extensions to make it applicable to series' of tasks is available. A qualitative measure is the likelihood and/or success of application to similar domains. This measure could be rated as low chance, medium chance or high chance. Another measure, could be the amount of effort required to apply the structured method / approach to another domain - high effort, medium effort or low effort. This scale matches the same scale used for other criteria and will make communication during the development and evolution easier.

6. Existing theory - **Literature to Support Design Decisions (True/False)** - is a binary measure, in that, the method / approach is built upon theories and research from a wide variety of fields and disciplines.

7. Effort - **Cost, Development, Automation Effort (Low/Medium/High)** - given enough time and resources, anything is possible. For automation, given enough time and resources, each and every edge case and scenario can be handled, and a perfect system designed. Therefore, the last criterion to be measured is Effort. Effort is the least amount required for MVP status. Efforts can be "cost" related (resource availability, cost of the resource or value of the resource to the organisation). Efforts can be "development" related (time taken of the task to manually achieve, time taken to develop a solution to automate the task, and the overall projected lifetime of the task going forward). Efforts can also be "automatability" related (the instinctive factors to consider for automation; data, information and knowledge, technology requirements, and automation impact i.e. benefit for the organisation). For example, a cost effort can be the prohibitive availability of SME's as a resource, skewing the criteria to prioritise anything that would multiply the effectiveness of this limited resource. An example of a development effort would be the predicted lifetime of the task under review, such as controlling entities in a large scale constructive environment — for the foreseeable future, this is the core mechanic of LVC distributed synthetic training exercises. Finally, an example of an automatability effort is the impact of technology on the domain — as machine learning techniques improve and agents become "smarter", a technology that would allow SME's to manage and control larger synthetic entities, would greatly improve the resource allocation i.e. multiplying limited resources. A caveat to this information is that none of these are perfect measures, but instead are listed as an introduction to the types of effort that need to be considered when applying/measuring other practical criteria. Therefore, these suggestions of types and measures of effort are provided as a rule of thumb for estimation, allowing for a quick and cheap qualitative measure to be applied.

3.3.7 Scenarios used for testing and evaluation

As part of the testing and evaluation of the generated method / approach, it is necessary to evaluate the proposed criteria (Section 3.3.4) against a set of heuristics and measures (Section 3.3.6). To do this, the researcher applied the method / approach against a defined scenario or exercise which is a representative subset of the type of tasks and processes performed by White Forces.

As part of the complexity and complicatedness of achieving a method / approach for the identification of White Force multipliers, so too is the classification of scenarios used for testing and evaluation. The following scenarios range from complex to clear, from large scale distributed synthetic training exercises, to cooking and cleaning exercises. The benefit of including a variety of scenarios of different complexity, is that it allows for greater testing and evaluation using simple data and observations. This testing and evaluation allows confirmation that development is progressing and able to handle complex scenarios, with obscured and hidden tasks and processed.

The following are the proposed scenarios.

Scenario 1 - White Force training exercise

Based on the information presented by Simpkin et al. [5], training exercises follow a 3 phase delivery model of Event Preparation, Event Delivery and Post Event Analysis. This scenario is used initially to gain an understanding of the scale of the data under review.

Some types of tasks that are consistent amongst exercises are:

1. Defining training and research objectives
2. Re-use previous exercise assets/artefacts
3. Develop exercise plans and scenarios, and identify constraints
4. Prepare synthetic environment and configure simulations
5. Manage PR and maintain schedule
6. Control scenario and monitor real time scenario
7. Monitor trainee performance and register observations into data logs
8. Manage audio/visual systems and monitor simulation system performance
9. Familiarisation training, plan, brief, execute and debrief mission, control CGFs
10. Record observations, analyse event records and make recommendations

11. Store exercise assets/artefacts and maintain event records.

This exercise overview is based upon real world experience, expertise and observations. This is the starting exercise that any proposed method / approach needs to be based upon.

Scenario 2 - High Yo-Yo

In development with industry partner Coherics[4] (who are comprised of former military SMEs, and specialise in delivering tailored training outcomes for Defence and Defence partners), a scenario was developed that is representative of the type of task trainees (and by extension White Force acting as Red Air) perform. The following is a typical Basic Fighter Manoeuvres (BFM) and is regarded as a subset of manoeuvres trainees are expected to perform.

This task, as a complex subset of Scenario 1, allows for fine-tuning of the method and approach to capture all identified tasks and processes in a complex but clearly defined standard operating procedure.

The High Yo-Yo manoeuvre (Figure 3.9 [163]) *"is an offensive, Lag Pursuit manoeuvre designed to prevent an overshoot by controlling excessive closure while safeguarding range."* [163, 164]. It

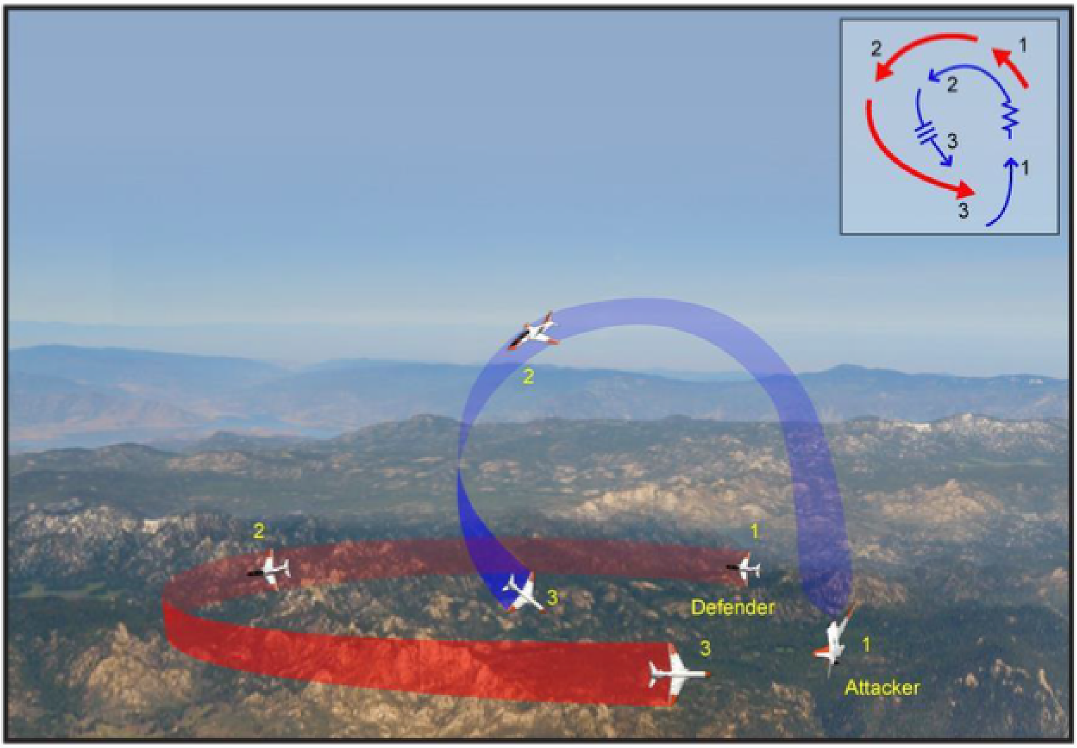


Figure 3.9: High Yo-Yo [163]

[4] https://www.coherics.com.au/

The student exercise is as follows. *"The manoeuvres will begin with a canned scenario setup to maximise training efficiency. The initial setup is as detailed below in Figure 3.10 [163]. Once the student takes the Fox 2 missile shot at a range of 1NM, the defender calls "Flares" and breaks into the attacker. When flown correctly the setup will create an overshoot situation for the attacker"* [163, 164]. This student exercise aims to present the trainee and White Force Qualified Flight Instructor as Red Air, an opportunity to align aircraft to conduct a High Yo-yo manoeuvre.

The Qualified Flight Instructor as Red Air guide is as follows. *11Maintain predictable and stable platform until the student has successfully captured WEZ. Watch RAC, AA and perch while scanning for accurate G, KIAS and altitude"* [163, 164]. The White Force Qualified Flight Instructor as Red Air is tasked with maintaining a stable and consistent platform, to provide the trainee the opportunity to practice the BFM.

"When the student has successfully captured the WEZ, and student's QFI is ready to move on, create further range problems for the student to solve. Initially extend once the student rolls OOP by releasing G and increase KIAS. This will increase the range. Once the student reverses in the perch reapply best turn rate parameters and turn hard into the student. Maintain altitude" [163, 164]. Once the trainee is familiar with both the canned scenario, and entry into the high yo-yo BFM, further instructions are provided to the White Force Qualified Flight Instructor as Red Air to extend the BFM and present new challenges to the trainee to overcome. The initial set of instructions are to create range problems.

"Only introduce vertical problems once the student is deemed competent at high yo-yo recognition, mechanics and able to solve simple range issues from the perch. To introduce vertical problems, once the attacker commits nose low to regain WEZ pull hard up into the plane of the attacker to generate a vertical overshoot. The result should be the attacker losing the offensive by crossing through the defender's altitude" [163, 164]. Once the trainee is comfortable with range problems, the White Force Qualified Flight Instructor as Red Air is given instructions to introduce vertical problems.

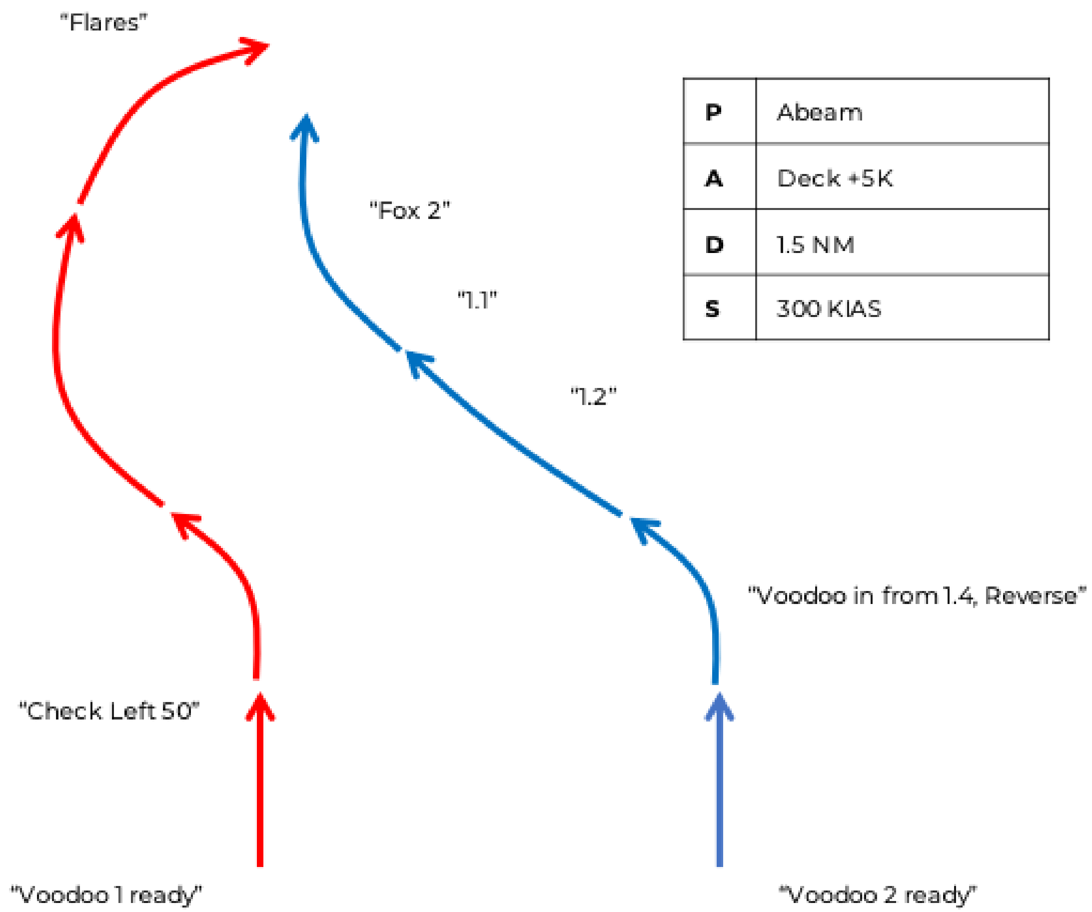


Figure 3.10: Canned Exercise for High Yo-Yo [163]

As can be seen with this scenario exercise, there are a series of training objectives and outcomes contained within the provided exercises and instructions. Therefore, as a testing exercise, the High Yo-yo BFM is a subset of the tasks and processes that White Force need to perform as part of their duties.

Scenario 3 - Simple scenario 1 - Cleaning

To provide users with a base upon which to ground the theories and concepts of any proposed method / approach, this scenario was created as a simple exercise. The purpose of this scenario is to provide users with a familiar example that is universal in experience, and to provide the researcher with a means to test and evaluate steps within the method / approach, by not requiring SME's or extensive data collection experimentation. During the development of this simple exercise, the researcher took into account the types of activities that occurred daily within the researchers' environment.

As a thought exercise, the scenario objective is *House is clean and safe for baby under 1 year* *of age*. This task needed to occur daily, due to the mobility and activity of the scenario subject (i.e. the baby). The definition of the house included the rooms; Kitchen, Bathroom, Bedroom, Living Room and Study. Some example activities within the scenario include;

1. Tidy toys
2. Vacuum floor
3. Clean High Chair
4. Wash Dishes
5. Put washing away
6. Wipe down benches and table tops

Scenario 4 - Simple scenario 2 - Cooking
Continuing from the motivation of the previous scenario, this scenario was created to further explore methodologies and approaches of any proposed method / approach. The aims and objectives of this scenario were to create a dinner for the researcher and their family.

As a thought exercise, the scenario objective is *Dinner is made*. This task needed to occur daily. For this use case, a simple two component dinner was used, of *Spaghetti Bolognese*. The components of this meal were split into Spaghetti and Bolognese sauce. For the task of *Cook Spaghetti*, some activities include;

1. Add water to pot
2. Turn on stove
3. Add pasta to pot
4. Drain pasta water

Scenario 5 - Simpilot controlling two aircraft
As part of early development and feedback, a simple worked example was provided to the researcher by a member of the team at AOSC. This was initially provided alongside Scenario 1, but due to the ambiguity and complexity of the scenario, including limited access to SME's expertise and experience, it was used once an established method / approach existed in the later stages of development and evolution. This allows for extensive testing and evaluation of the method / approach in its ability to address the proposed suitable criteria for validation. The exercise/scenario is as follows

WORKED EXAMPLE: A simpilot is controlling two aircraft. He/she receives a command over the radio from a trainee in the control agency (wedgetail): “aircraft 2, make your way to the fuel tanker, contact me when you are 20 nautical miles out” (this is paraphrased). From the perspective of the trainee, in a Large Force Engagement (LFE) exercise this is a ‘clear’ task, as those trainees already have proficiency. However, for new students undergoing initial employment training, especially for the first time, this task may be ‘complicated’.

From the perspective of the simpilot, he/she now has a goal to “fly to tanker and respond on radio”. Again, for a real pilot, or experienced simpilot this is ‘clear’. But for anyone else, I feel that it gets ‘complicated’, fast!

Tasks sequence in order to satisfy the above goal:

- Acknowledge receipt of the command over radio.
- Check location of fuel tanker (Note the tanker will be orbiting in a circle.)
- Select the entity named “aircraft 2”
- Adjust aircraft 2’s heading, so it will intercept the tanker.
- Configure distance measurement tool.
- Wait until distance measurement tool reads 20 nautical miles to the tanker.
- Contact control agency.

Discussion: Not all tasks above are completed instantaneously (we can call that synchronous). The ‘contact control agency’ task invokes a whole new task hierarchy for operating the radio (the channel may be jammed, the channel may already be in use, etc.). This is where HTA comes in handy.

Tasks may produce errors or exceptions $\rightarrow$ complexity. Maybe even chaos when the simpilot has not experienced the error before.

Simpilot may be instructed from white force to deliberately disobey or misinterpret the commands. View through Cynefin lens, this is intended to create complexity/(chaos?) for the trainee.

There are two aircraft in the worked example. We have not considered what the simpilot is doing with aircraft 1.

3.4 Chapter Summary

This chapter introduced the research hypothesis, research questions and method used, that drive the body of this thesis. Using an exploratory sequential mixed methods approach, the researcher will use two methods, (i) methodology for method / approach construction, and (ii) methodology for testing and evaluation. Once a method / approach has been developed, using the proposed criteria and metrics, and scenarios for testing and evaluation, the final validation of the method / approach will occur in consultation with subject-matter-experts from AOSC.

As presented within this chapter, a set of criteria was generated from the research questions that will drive the Method for Development and Evolution, and the Method for Testing and Validation. These criteria were developed in conjunction with members of AOSC, to allow for progression and forms the basis of the underlying requirements of the proposed method / approach. In conjunction with the Method for Development and Evolution, and the Method for Testing and Validation, this work presented the proposed criteria, and metrics derived from said criteria.

In the following chapter, the Method for Development and Evolution of the proposed approach, and the Method for Testing and Validation of the proposed approach, including proposed criteria + metrics, will be applied to generate the major contribution of this work - the proposed approach to address the research questions for identification, communication, representation, classification and automation ranking of candidate tasks and processes/routines to address the hypothesis of how we can multiply White Force capabilities using automation.

Chapter 4

DCARR: A Model Driven Framework for the multi hierarchical task decomposition and selection of candidate tasks and processes for automation applicability

The purpose of this chapter is to address the research hypothesis of; *"How can we multiply White Force capabilities with automation?"*, by addressing the Research Questions that aim to produce methods/techniques/selection criteria for the *identification, communication, representation, classification and automation ranking* of candidate tasks and processes/routines within the domain of White Force facilitated distributed synthetic training.

This chapter presents the outcome of the method for the construction and for the testing and evaluation of the proposed framework. This chapter contains a step by step evolution of the proposed framework which aims to address the identification, communication, representation, classification and automation ranking of White Force tasks and processes/routines. This framework is presented as an initial metamodel, as means of communication, and evolves and develops to a tested and evaluated framework, ready for validation to members of DST Group. This chapter also contains, a complete step by step breakdown of the framework, with included worked examples and demonstrations.

4.1 DCARR — Model Driven Framework

White Force exists in an inherently complex and chaotic environment, due to skills, knowledge and expertise required; the often inconsistent and fault intolerant execution created through experience and intuition; and the need to inject complex/chaotic elements to create unknown and unexpected scenarios that the training audience need to adapt and overcome. To effectively apply automation, White Force tasks and processes/routines require a level of stability which is currently lacking (Section 3.1).

The DCARR (Describe, Collect, Analyse, Recommend, and Report) Framework aims to provide White Forces a tool to leverage abstraction techniques and existing methodologies and concepts to create a systematic method for the identification, communication, representation, classification and automation ranking of candidate tasks and processes/routines within the domain of White Force facilitated distributed synthetic training. DCARR is the major contribution of this research, but other contributions include the proposed Method for Development and Evolution of DCARR, the proposed Method for Testing and Validation of DCARR, and the proposed criteria and metrics used during development, evolution, testing and validation, which were created in support of the requirements from DST Group. To do this, DCARR was developed using an exploratory sequential mixed methods approach (Section 3.3), where concepts were mined from literature (Section 2.4) and evaluated against a set of criteria (Section 3.3.4), using scenarios that range from simple to complex (Section 3.3.7), to address the proposed research questions (Section 3.2).

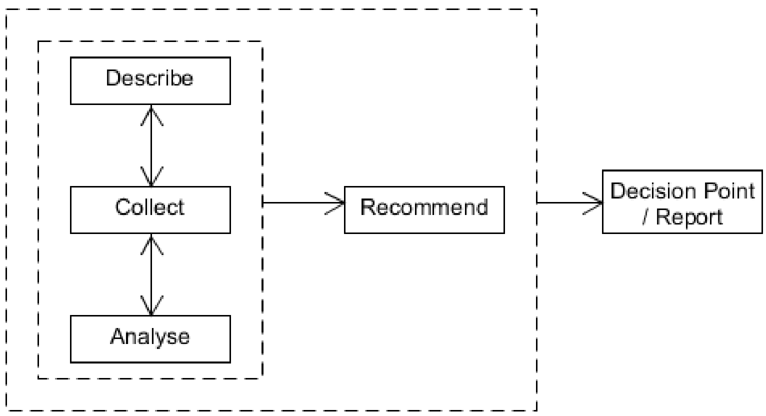


Figure 4.1: DCARR Phase Diagram

4.1.1 DCARR Overview

The DCARR Framework (Figure 4.2), is a framework used to guide a user with tools and methods to identify a ranked and ordered list of candidate tasks and processes/routines, which have been; identified, communicated, represented, classified and ranked, from an exercise/scenario/workflow/event/etc. Figure 4.1 presents the relationship between the phases of DCARR.

The first phase, *Describe*, is based around the initial probing and exploration of tasks to quickly establish the relationship between tasks and objectives. During this phase, users begin to describe and communicate tasks in a unified manner. Once a list has been established, models are drawn to visually represent tasks to allow further discussion amongst stakeholders. This phase aims to address methods for task identification, communication and representation.

The second phase, *Collect*, is used to plan, observe and collect quantifiable data to support the outputs from the *Describe* phase. This measurable data is used to support and validate the assumptions and initial outcomes from the *Describe* phase. This phase aims to address methods for representation and classification.

With data collected, and key decisions identified, the third phase, *Analyse*, is used to confirm the described tasks and processes/routines from *Describe* phase, along with any assumptions. To confirm these tasks, quantifiable data is used to generate measurable instances of candidate tasks and processes/routines. It is during this phase, that data is used to identify resource usage, task frequency, task difficult and overall impact. Figure 4.1 shows the relationship between the *Describe, Collect and Analyse* phases, in that it is possible to travel backwards and forwards as required. For example, if during task identification, certain tasks are missed or not described, it is possible to retroactively complete the required steps and generate instances of tasks as models, using collected data. This phase aims to address methods for classification, representation and automation ranking.

The fourth phase, *Recommend*, occurs once the *Describe, Collect and Analyse* phases are complete and analysis has been undertaken. This phase focuses on the evaluation of the analysis to identify tasks which most satisfy a desired automation outcome. This goal can be the overall reduction in limited available resources by multiplying existing capabilities. This phase is solely aimed at addressing selection criteria for automation ranking.

Once evaluated and recommendations for candidate task automation are communicated, the final phase, *Report*, is the presentation and reporting of findings to stakeholders. This step is crucial to the communication and success of the application of the framework, as without appropriate actionable items, there is no point in undertaking an investigation.

4.1.2 DCARR relationship to existing theories

Figure 4.3 is a visual representation of the steps of the framework and the relationship to related work within literature. Each of the major phases are guided and informed by work done within a number of parallel fields.

DCARR builds upon existing research conducted in the field of Human Factors, in particular work done around EAST (Section 2.4.2). EAST provides a structured framework approach as well as the usage of Human Factors methods for the deconstruction of tasks using Hierarchical Task Analysis. The use of this framework and area of research, addresses task and process/routine identification.

The applicable theory of Cynefin (Section 2.4.1), provides definitions and examples of deterministic and non-deterministic domains, and approaches to assess and navigate successfully. It is the definitions of complexity as a non-deterministic domain that greatly influence the selection of tasks as appropriate vs not appropriate. This theory addresses task classification.

Information Fusion (Section 2.4.3) is guided by a series of grand challenges, one of which relates to the presentation of information within Information Fusion systems. This challenge and one of the accompanying approaches, makes use of a simple cognitive model, originally based on OODA — Observe, Orientate, Decide, Act — loop. It is this representational approach that is used in conjunction with UML (Section 2.4.5) modelling techniques to visually communicate the identified tasks and processes. This area of research (Information Fusion) provides the method for task representation, while the language and techniques of UML address task communication.

With tasks identified, communicated, represented and classified, it is necessary to analyse the collected information to provide meaningful insights and actionable outcomes. This is where applicable theory about Automation Levels and Types (Section 2.4.4) is required. Using the proposed simple model of Human Information Processing, which aligns to OODA, it is possible to rank and order candidate tasks and processes/routines into quantifiable components for measurement and evaluation. This theory and area of research provides the selection criteria for automation ranking.

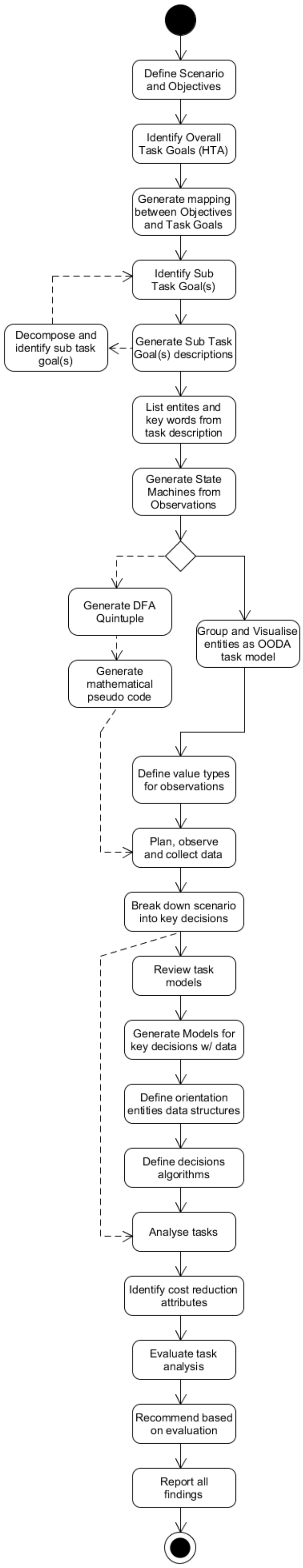


Figure 4.2: The DCARR Framework, detailing the flow of information in Describe, Collect, Analyse, Recommend and Report phases.

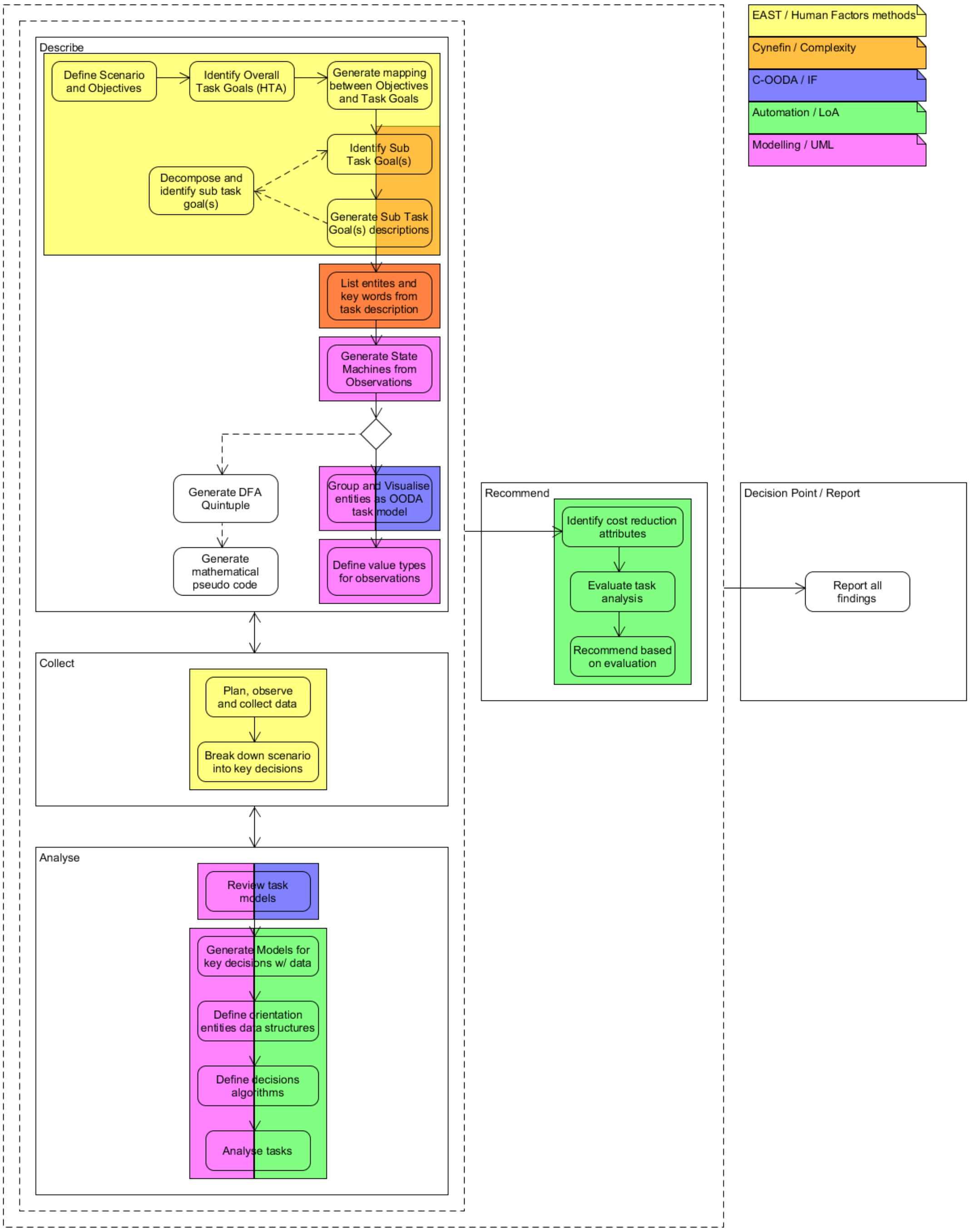


Figure 4.3: DCARR Framework + Phase Diagram. Colour coded to indicate sections influenced by related work

4.2 DCARR — Step by Step

With and overview and understanding of the five stages of DCARR (Figure 4.2), as well as an understanding of the areas of existing theory that have been used to shape and develop DCARR (Figure 4.3), this section presents a breakdown of the steps of DCARR.

As presented, DCARR follows a flow chart UML notation, where steps occur sequentially and follow on from each other. In practice, (and due to the limitations of the notation used), DCARR, as a concept, encapsulates 5 phases — each with their own purposes, methods and outcomes. The overarching aim of DCARR is the investigation and decomposition of a large scale complex/complicated series of concurrent and asynchronous tasks and processes. Where the tasks and processes performed may be hidden or follow ad hoc processes and procedures. Once identified, tasks are communicated/represented in a manner to elicit data collection for further investigation/classification into ranking and ordering of candidate tasks and processes/routines for automation.

Each stage of DCARR aims to work cohesively with other stages, as per Figure 4.1, but also independently depending on limitations and external factors that implementors may face (such as lack of data collection opportunities, lack of access to SME's and members with the correct experience and knowledge). Therefore, while presented as a flow chart, the reader is advised to keep in mind the potential for partial application of phases, steps and/or tasks.

The following is the step by step breakdown of each task of DCARR, as well as expected outcomes of each phase. Where possible an example has been provided to demonstrate what an outcome can look like.

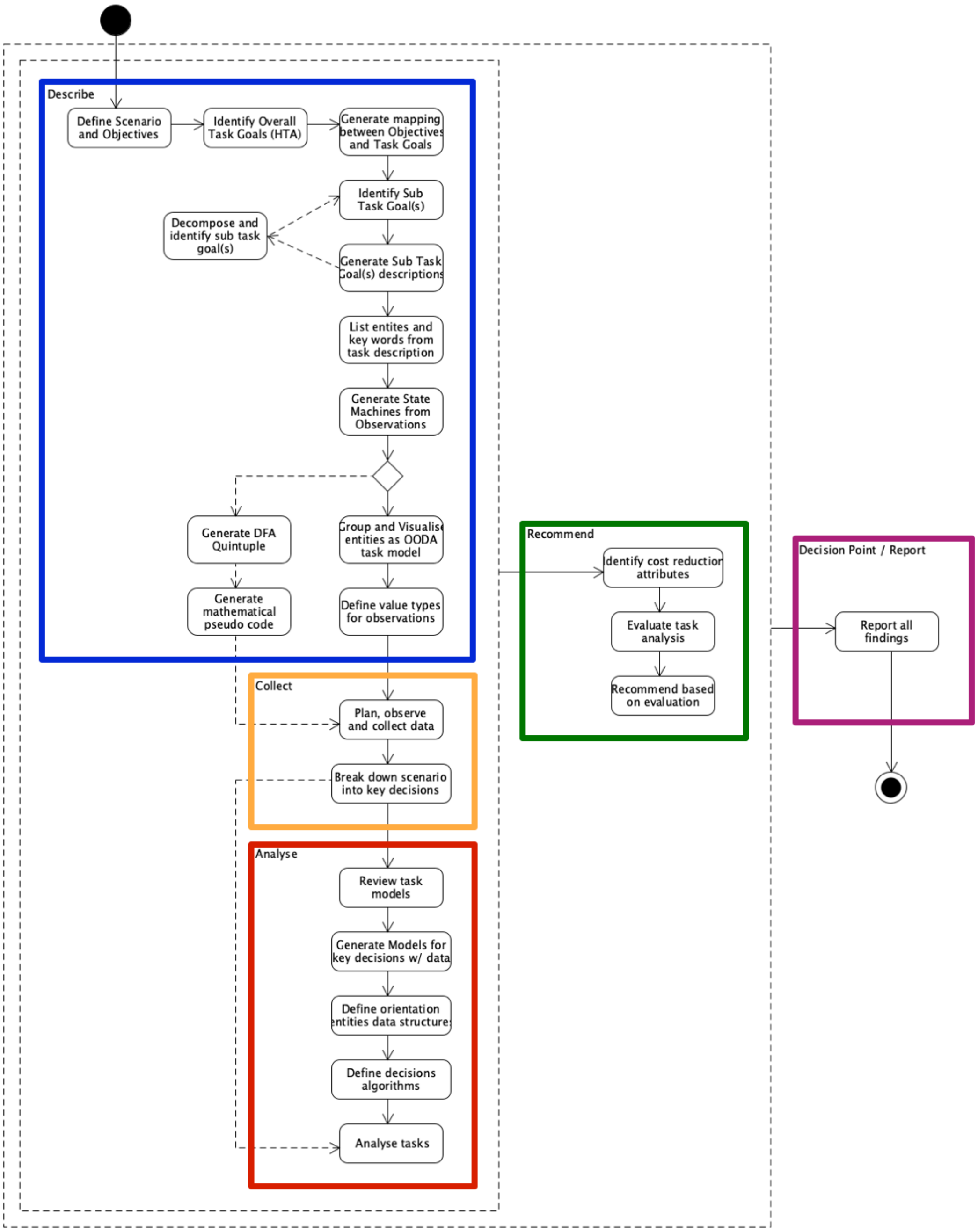


Figure 4.4: DCARR Framework split into phases

4.2.1 Describe

The Describe phase (Figure 4.4) of DCARR is the initial entry point into achieving an understanding of the scope of the exercise/tasks and processes at hand. It allows for stakeholders, implementors, supervisors, experts, contractors, etc., to engage and share knowledge to describe a mixture of standardised and ad hoc tasks and procedures, both hidden and obvious. This entry point focuses on the initial representation of objective/goal mapping, scope identification, tasks identification, classification and communication, as well as setting the stage for data collection activities should implementors choose to proceed with DCARR application. As stated, the purpose of this phase is to identify and generate the following:

- *Scope of activities under review (Scenario and Objectives)*
- *Decomposition of tasks (Enough to be describable)*
- *Quick identification and selection of deterministic/appropriate tasks (and identification of nondeterministic/out of scope tasks)*
- *Plain language descriptions (to identify object, action, output and context of tasks)*
- *Formatting of tasks for analysis and automation evaluation*
- *Different representations of tasks for insight and review (state machines, DFA Quintuple and pseudocode, UML task models)*

To achieve these outcomes, it is necessary to conduct discussions and interviews with experts and those individuals that contain the required information. Task and subtask goal descriptions need to be generated to understand the flow of data and information, including required input data, decision-making, expertise and prior knowledge/training and what the expected outcomes of the tasks performed are.

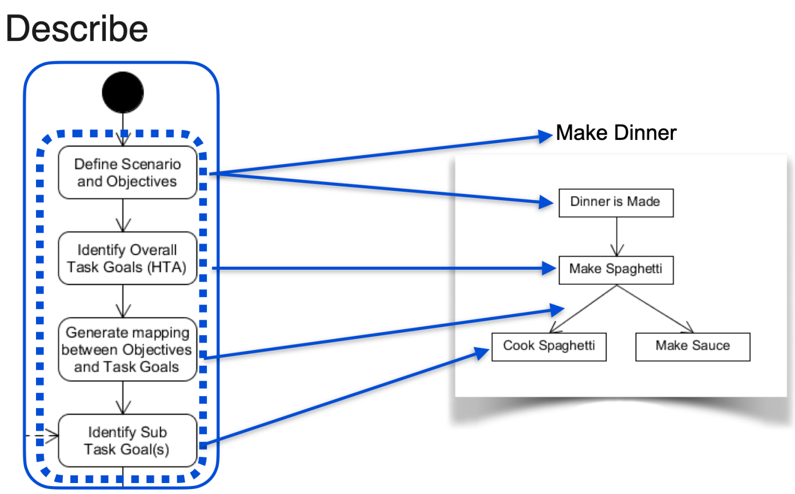


Figure 4.5: Define Scenario and Objectives, Identify Overall Task Goal, Generate mapping between objectives and task goal, and Identify subtask goals

Define Scenario and Objectives (Figure 4.5)

Step 1: Write down the scenario or process you want to investigate

Step 2: Write down the objectives that need to be met in order for the scenario or process to be completed.

Identify Overall Task Goal (Figure 4.5)

Step 3: Write down the goal or goals that need to be met to achieve the objectives.

Generate mapping between objectives and task goal (Figure 4.5)

Step 4: Map the goals to objectives. Each goal must map to at least one objective and each objective must have at least one goal.

Identify subtask goals (Figure 4.5)

Step 5: Write down the decomposed subtask goals that have to be met in order to achieve the overall task goal / goals.

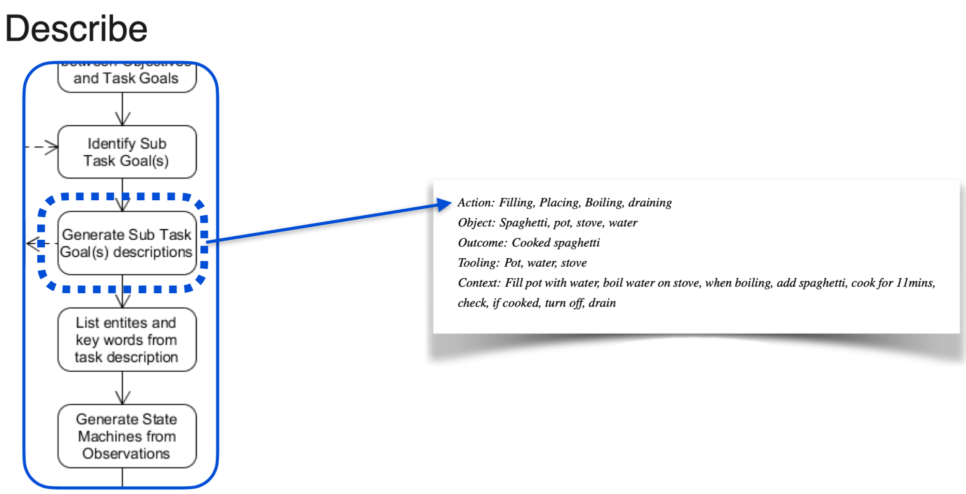


Figure 4.6: Generate subtask goal descriptions

Generate subtask goal descriptions (Figure 4.6)

Step 6: For each subtask goal that exists answer the following questions

- *What is the action performed? (verb)*
- *To whom, or what is the action performed on/to? (object of the verb)*
- *What is produced by the action? Or Why is it necessary? (what is the expected outcome/output)*
- *What are the processes, guides, equipment, tools, etc used?*
- *What is the context of this task in relation to the scenario? (temporality)*

Decompose and identify subtask goals

Step 5a: If the task cannot be described using Step 6, decompose the subtask goal into smaller parts and repeat Step 6

(Can't Describe → Decompose task → Can't Describe → Can't Decompose → Out of Scope)

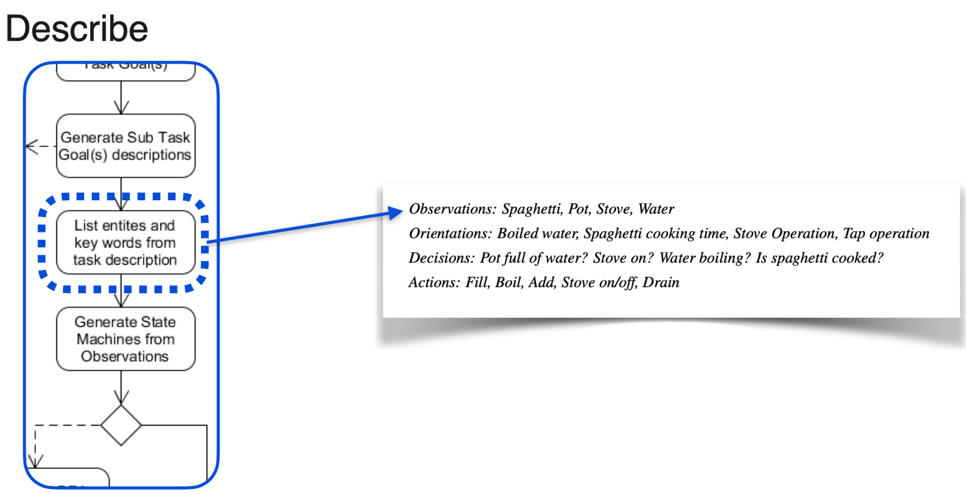


Figure 4.7: List entities and keywords from descriptions

List entities and keywords from descriptions (Figure 4.7)

Step 7: From the task description extract keywords and entities, such as the actions, objects, purpose/end state, tools, processes/guides and anything else important from context.

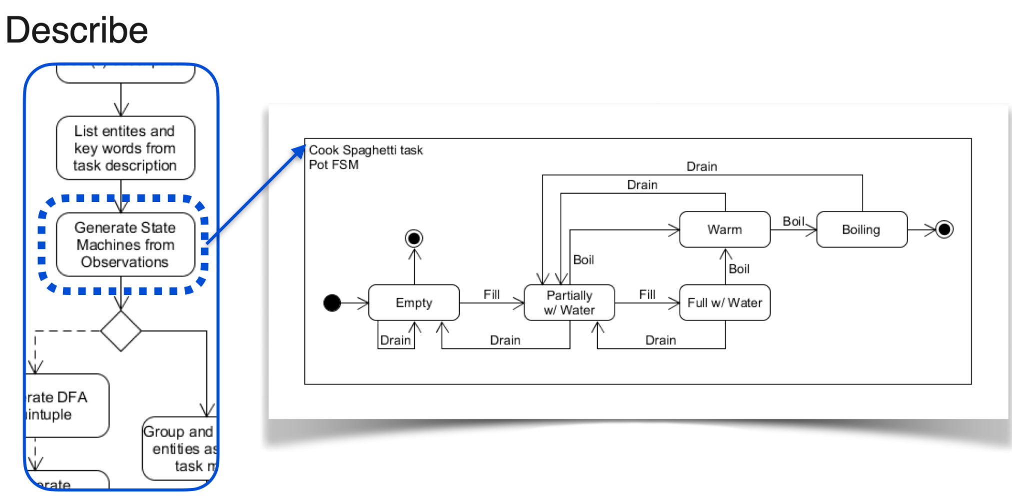
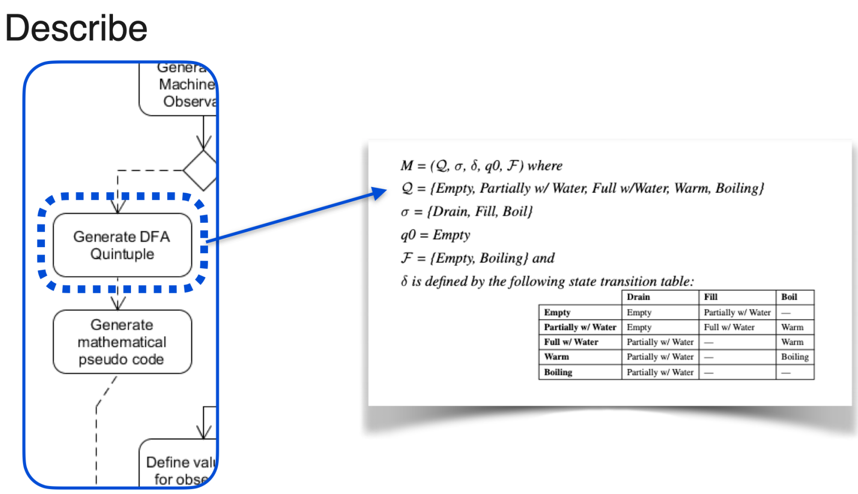


Figure 4.8: Generate state machines from observations

Generate state machines from observations (Figure 4.8)

Step 8: For all objects in all tasks, generate state machines to represent the different outcomes that can be generated by different actions



	Drain	**Fill**	**Boil**
Empty	Empty	Partially w/ Water	—
Partially w/ Water	Empty	Full w/ Water	Warm
Full w/ Water	Partially w/ Water	—	Warm
Warm	Partially w/ Water	—	Boiling
Boiling	Partially w/ Water	—	—

Figure 4.9: Generate DFA quintuple

Generate DFA quintuple (Figure 4.9)

Step 9: For each state machine, generate a *"deterministic finite automation quintuple ($\mathcal{Q}$, σ, δ, q0, $\mathcal{F}$), consisting of*

- *a finite set of states* $\mathcal{Q}$
- *a finite set of input symbols called the alphabet* σ
- *a transition function* $\delta : \mathcal{Q} \times \sigma \rightarrow \mathcal{Q}$
- *an initial or start state* $q_0 \in \mathcal{Q}$
- *a set of accept states* $\mathcal{F} \subseteq \mathcal{Q}$."[1]

[1] https://cs.stanford.edu/people/eroberts/courses/soco/projects/2004-05/automata-theory/basics.html

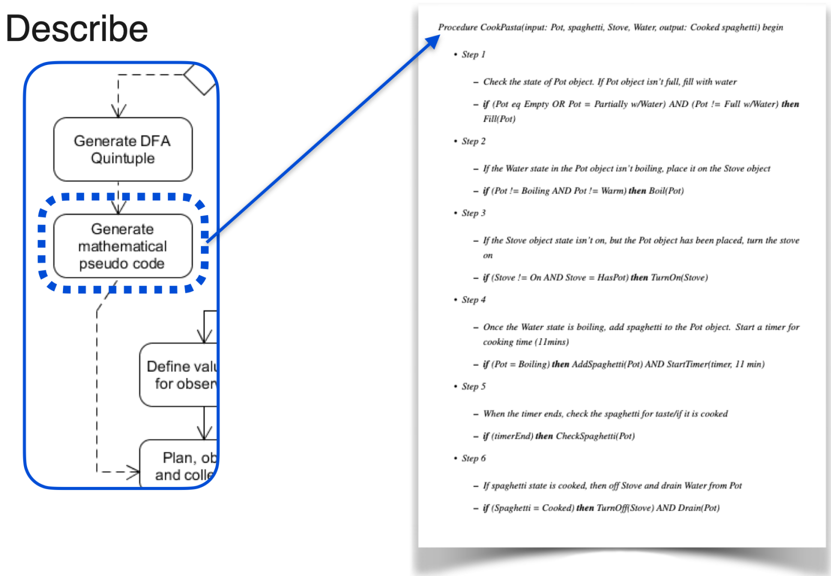


Figure 4.10: Generate mathematical pseudocode

Generate mathematical pseudocode (Figure 4.10)

Step 10: For each task generate pseudocode. Use a mixture of mathematical notation (set and matrix theory) and control structures from conventional programming languages. (*This pseudocode should be a reflection of the state machines / DFA quintuple*)

e.g. algorithm Cook Spaghetti is

Procedure CookPasta(input: Pot, spaghetti, Stove, Water, output: Cooked spaghetti)

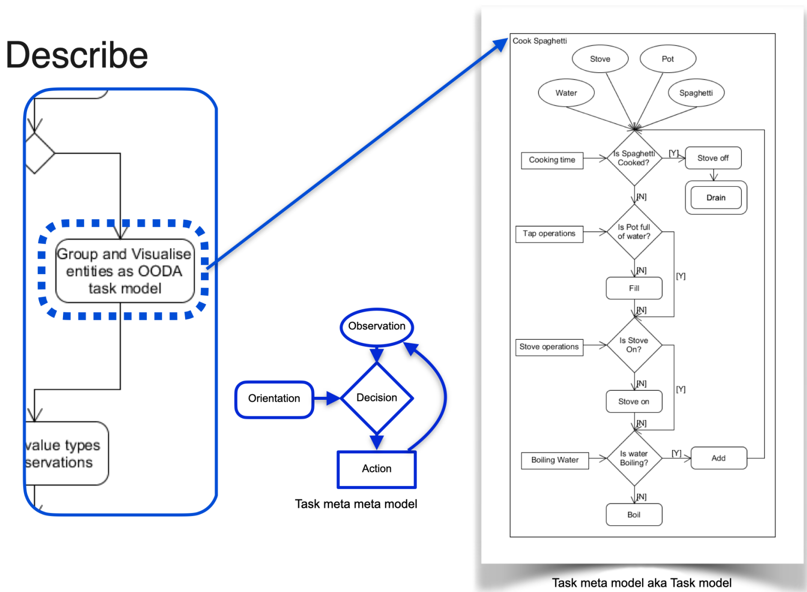


Figure 4.11: Groups and visualise entities as OODA task models

Groups and visualise entities as OODA task models (Figure 4.11)

Step 11: Use UML to arrange keywords and entities into a visual representation of the task

- *Observation — Input/Object to the task*
- *Orientation — Extra required knowledge*
- *Decision — Reasoning based on Observations and Orientations*
- *Action — Output of the task, as a result of the Decision*

Define value types for observations

Step 12: Attempt to identify the value types of observation entities.
(Observation entities can be sensors that denote states/state changes (open/closed), wet, dry, heat, triggered event, etc)

4.2.2 Collect

Collect phase (Figure 4.4) is all about collecting quantitative data to support the assumptions made from the outcomes of Describe phase. This phase, focuses on the collection of supporting information to validate identified tasks and processes from Describe phase. With an understanding of the expected and required inputs/outputs (observations and actions), it is possible to ascertain if hidden ad hoc tasks and processes occur that may have missed during Describe phase outcomes. This phase is solely focused with data collection for further analysis. Outcomes from Describe phase can direct implementors to capture required observations for validation of assumptions made. As stated, the purpose of this phase is to:

- *Observe tasks within the scenario under review*
- *Collect data to confirm outputs and identify gaps from* **Describe** *phase*
- *Collect real world data to test and validate future analysis and evaluation.*

While it logically follows on from outcomes of Describe phase, due to limitations of running and conducting exercises or data collection, Collect phase can be run independently of Describe outcomes, and a retrospective analysis can be conducted to reconstitute temporal data flow for examination and analysis. While not recommended, it is possible to do so.

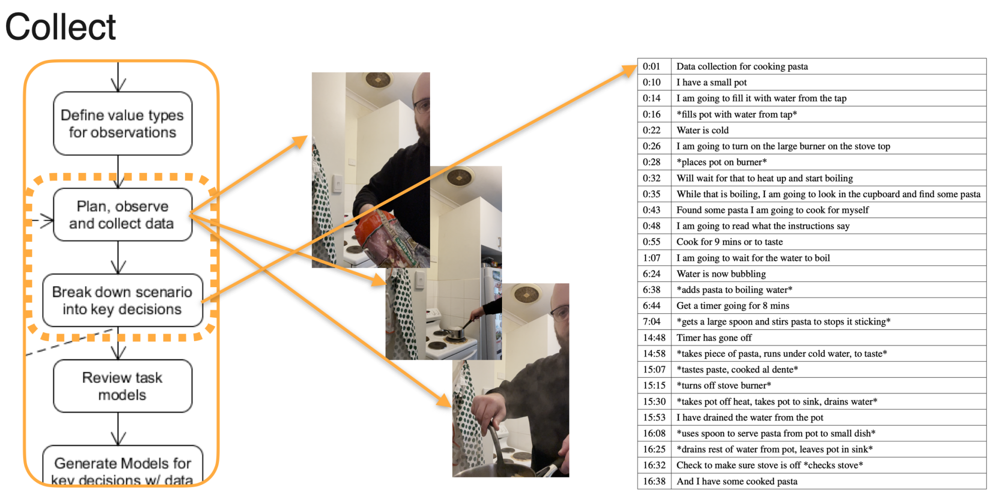


Figure 4.12: Plan, observe and collect data, and Break down scenario into key decisions

Plan, observe and collect data (Figure 4.12)

Step 13: Based on describe phase; plan, observe and collect data from an instance of the described scenario with definable objectives and goals. Use identified tasks and OODA task models and/or Objects, Actions, Outcomes, Tooling and Context from task descriptions to inform data collection. (Plan to record data, set up equipment as required, identify key subject-matter experts and participants. Gain stakeholder support to conduct interviews. Gain access to logs, records and transcripts.)

- *Extensive Data collection methods are applied in this step*
  - *Interviews*
  - *Observations*
  - *Video recordings*
  - *Screen recordings*
  - *Transcripts*
  - *Logs*
  - *Documents and records*
  - *Focus groups*

Break down scenario into key decisions (Figure 4.12)
Step 14: Using data collected from Plan, Observe and Collect data and outputs from **Describe** phase to generate a timeline of key tasks and decisions. Uses Critical Decision Method and interviews with Subject-Matter Experts.

- *"Select incident"* [110]
- *"Obtain unstructured incident account"* [110]
- *"Construct incident timeline"* [110]
- *"Decision point identification"* [110]
- *"Decision point probing"* [110]

4.2.3 Analyse

With an understanding of the scope and range of tasks and processes under investigation, from Describe phase, plus the quantitative data to confirm those assumptions, Analyse phase (Figure 4.4) is all about generating and measuring tasks and processes. During this phase, it is possible to retrospectively review outcomes from Describe and using data from Collect, generate hidden and unknown tasks and processes. This phase is entirely dependent on having some form of data to generate quantitative measures from. Outcomes from this phase are validated assumptions from Describe, based on data collection from Collect. These outcomes can be measures and assessed to determine areas of automation interest, such as resource usage, frequency, difficulty and impact. As stated, the purpose of this phase is to:

- *Take task descriptions/representations from the* **Describe** *phase and analyses them*
- *Use outputs from* **Collect** *phase to confirm descriptions are accurate based on real world data*
- *Analyse looks at* ***resource usage, frequency, difficulty and impact***
- *Generate representations of necessary Orientations and Decisions based on OODA principles for relevant tasks that require further analysis than DFA, State Machines and Pseudocode*
- *Generate a quantitative and qualitative analysis of tasks to be used for evaluation in the* **Recommend** *phase*.

As mentioned, the requirement of Analyse phase is to have data to analyse. Therefore, for full benefit and full effect data collection needs to occur. But without data to analyse, assumptions can be made, using estimations and extensive SME expertise and knowledge to generate further assumptions of resource usage, frequency, difficulty and impact to be analysed. These assumptions can be used as a potential data set for further evaluation to demonstrate and highlight the overall areas of estimated benefit DCARR can provide. In this way, partial application without data collection can occur as a means for benefit of estimation of full application of DCARR.

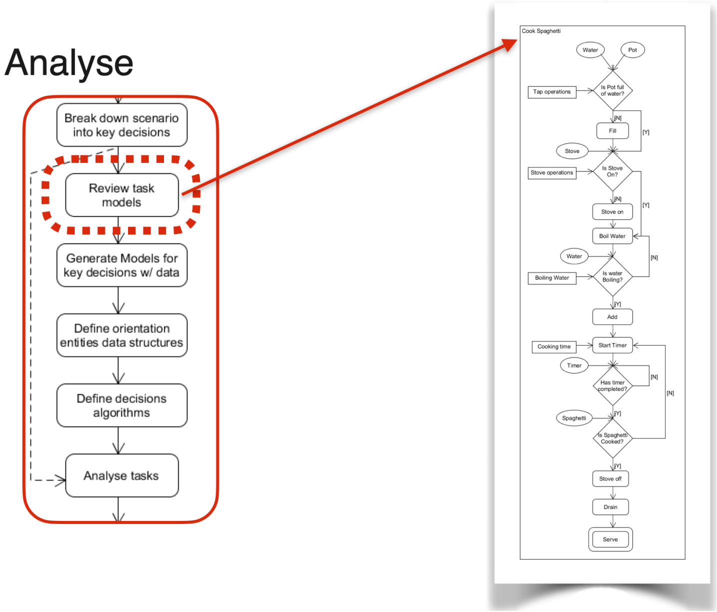


Figure 4.13: Review task models

Review task models (Figure 4.13)

Step 15: Using data collected and breakdown of key decisions from **Collect** phase, identify the information used for each task that required further analysis. Compare the initial task models from **Describe** phase against real world data from **Collect** phase and refine task models to accurately depict tasks.

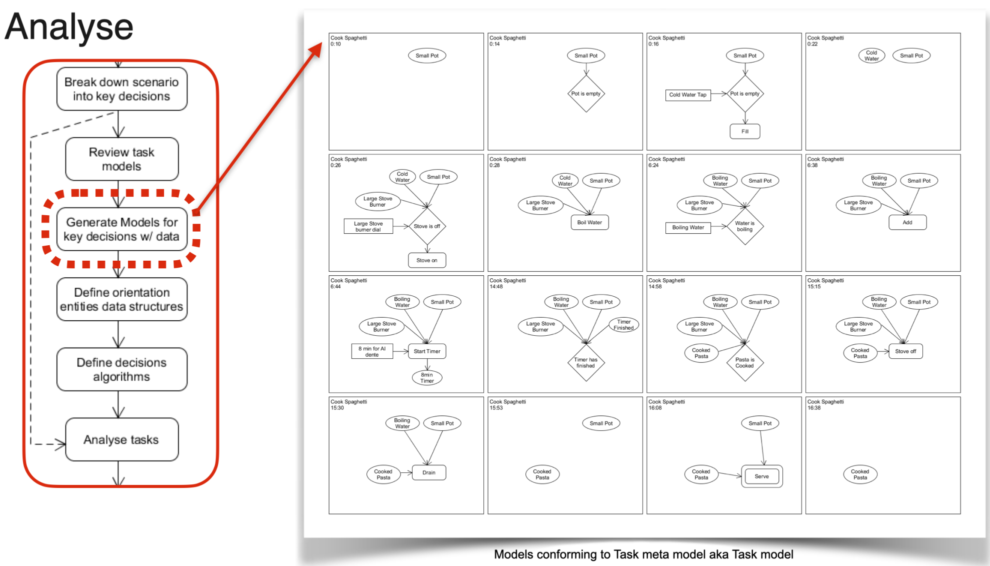


Figure 4.14: Generate Models for key decisions with data

Generate Models for key decisions with data (Figure 4.14)

Step 16: Using data collected and breakdown of key decisions from **Collect** phase, generate instances of tasks from task models. This outcome will generate a timeline of the visual representation of tasks.

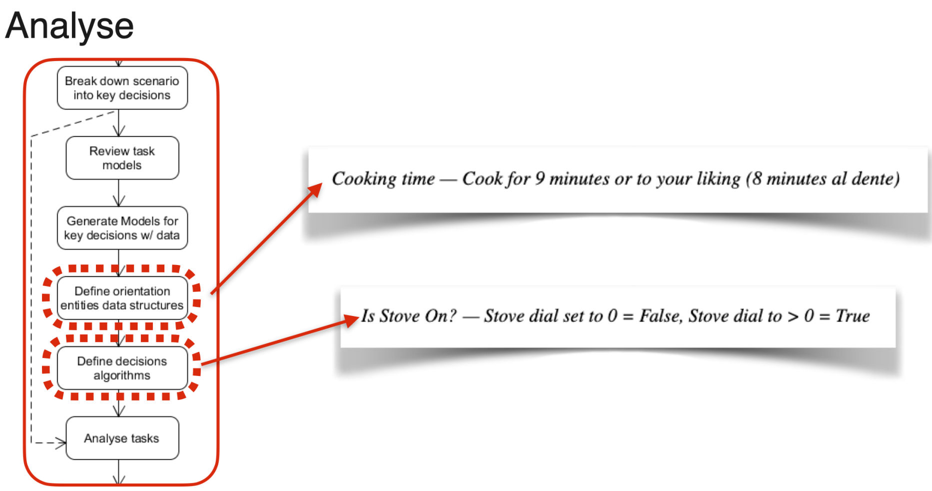


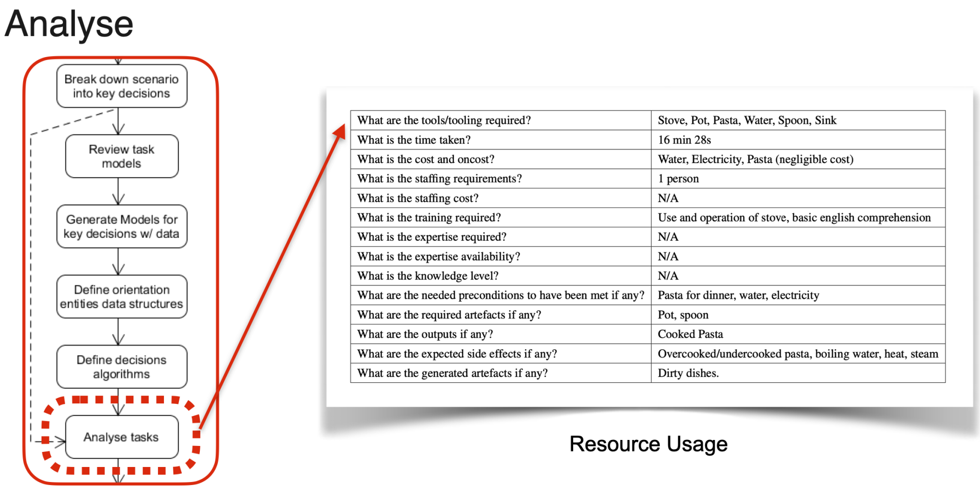
Figure 4.15: Define orientation entities data structures, and Define decisions algorithms

Define orientation entities data structures (Figure 4.15)

Step 17: Using data from **Collect** phase, define orientation entities as queryable data structures. This is the information required for more complicated tasks that human operators use to inform what decisions can be made. The step is an important knowledge capture process for any future work. *(Need to know commanders intent, capabilities, objectives to perform this task)*

Define decisions algorithms (Figure 4.15)

Step 18: Similar to orientation definitions, define the decisions made as algorithms. These are the aspects of observations and existing orientations, that human operators use to make decisions. *(Using a part of commanders intent, an aspect of capability and a certain objective, make this decision)*



What are the tools/tooling required?	Stove, Pot, Pasta, Water, Spoon, Sink
What is the time taken?	16 min 28s
What is the cost and oncost?	Water, Electricity, Pasta (negligible cost)
What is the staffing requirements?	1 person
What is the staffing cost?	N/A
What is the training required?	Use and operation of stove, basic english comprehension
What is the expertise required?	N/A
What is the expertise availability?	N/A
What is the knowledge level?	N/A
What are the needed preconditions to have been met if any?	Pasta for dinner, water, electricity
What are the required artefacts if any?	Pot, spoon
What are the outputs if any?	Cooked Pasta
What are the expected side effects if any?	Overcooked/undercooked pasta, boiling water, heat, steam
What are the generated artefacts if any?	Dirty dishes.

Figure 4.16: Analyse Resource Usage

Analyse tasks

Step 19: For each task identify the type of task

- *Functional Task — Given an Observable state, apply a rule set that dictates which actions to perform*
- *Procedural Task — Task that may invoke multiple tasks, doesn't always involve a state change, may require multiple orientations and decisions to determine which actions to invoke*

e.g. Cook Spaghetti is a Functional Task. Goal/Output is to obtain cooked pasta.

Step 20: For each task, identify the **resource usage** (Figure 4.16)

- *What are the tools/tooling required?*
- *What is the time taken?*
- *What is the cost and oncost?*
- *What are the staffing requirements?*
- *What is the staffing cost?*
- *What is the training required?*

- *What is the expertise required?*
- *What is the expertise availability?*
- *What is the knowledge level?*
- *What are the needed preconditions to have been met if any?*
- *What are the required artefacts if any?*
- *What are the outputs if any?*
- *What are the expected side effects if any?*
- *What are the generated artefacts if any?*

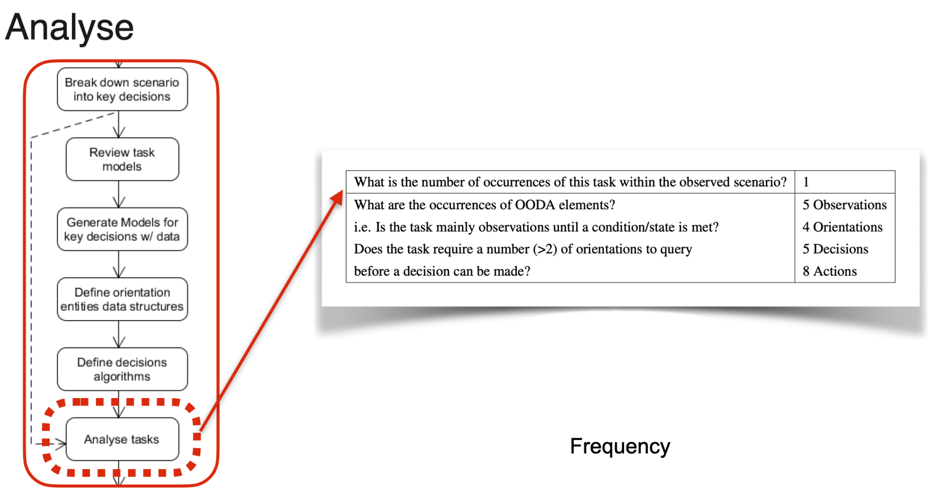


Figure 4.17: Analyse Frequency

Step 21: For each task, identify the **frequency** (Figure 4.17)

- *What is the number of occurrences of this task within the observed scenario?*
- *What are the occurrences of OODA elements? i.e. Is the task mainly observations until a condition/state is met? Does the task require a number (>2) of orientations to query before a decision can be made?*

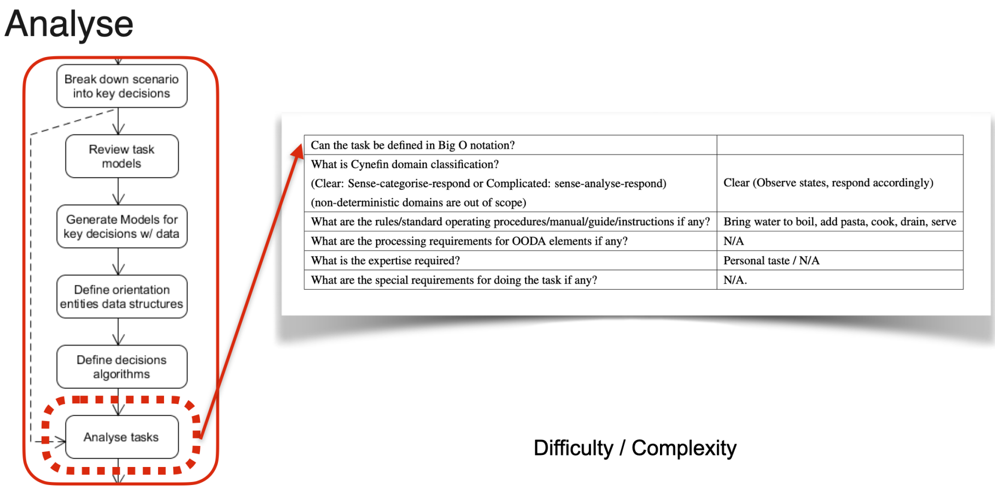


Figure 4.18: Analyse Difficulty/Complexity

Step 22: For each task, identify the **difficulty/complexity** (Figure 4.18)

- *Can the task be defined in Big O notation?*
- *What is Cynefin domain classification? (Clear:Sense-categorise-respond or Complicated:Sense-analyse-respond) (non-deterministic domains are out of scope)*
- *What are the rules/standard operating procedures/manual/guide/instructions if any?*
- *What are the processing requirements for OODA elements if any?*
- *What is the expertise required?*
- *What are the special requirements for doing the task if any?*

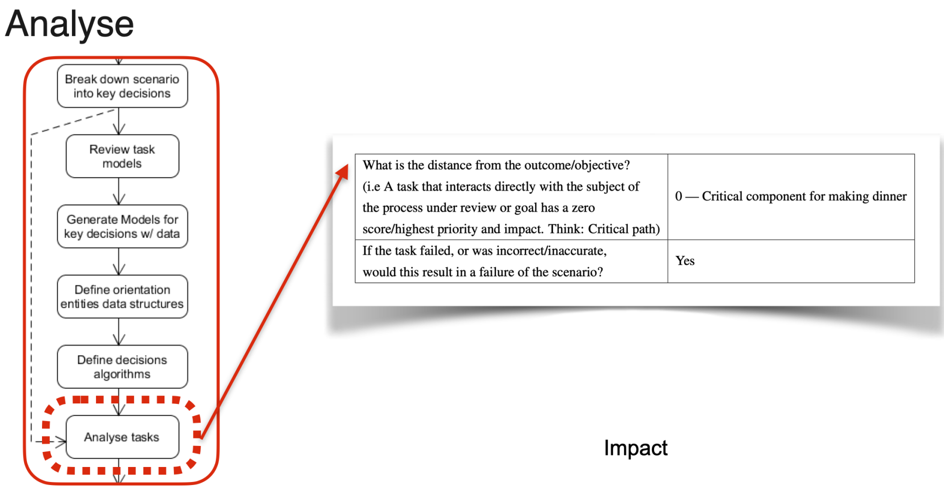


Figure 4.19: Analyse Impact

Step 23: For each task, identify the **impact** (Figure 4.19)

- *What is the distance from the outcome/objective? (i.e A task that interacts directly with the subject of the process under review or goal has a zero score/highest priority and impact. Think: Critical path)*
- *If the task failed, or was incorrect/inaccurate, would this result in a failure of the scenario?*

4.2.4 Recommend

Using the outcomes from Analyse phase, with quantitative and qualitative measures based on data collection or estimation, Recommend phase (Figure 4.4) evaluates this analysis to provide a ranked and order set of appropriate tasks and processes for automation application, given a desired cost reduction attribute (i.e. Focus limited resources on decision-making tasks and tasks of impact). This phase also is a chance to present all findings and outcomes to stakeholders and capability multipliers for assessment and further actioning. As stated, the purpose of this phase is to:

- *Evaluate outputs from* **Analyse** *phase against attributes that can identify cost*
- *Provide a list of suggestions for automation opportunities based on desired outcome from framework application.*

This phase is highly dependent on having analysis to evaluate, even if that analysis is based on intuition, experience or expertise. Assumptions can be made, with evaluations done to indicate a sensible starting place for capability multipliers to action, to determine if further application and investment is appropriate.

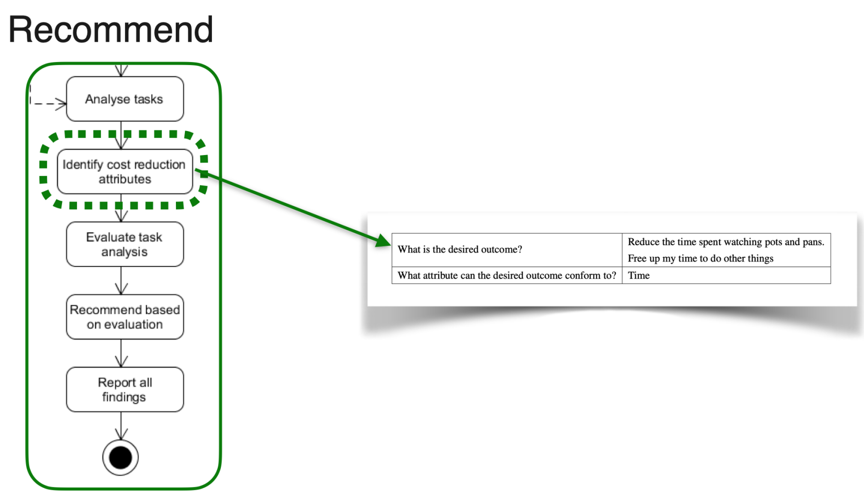


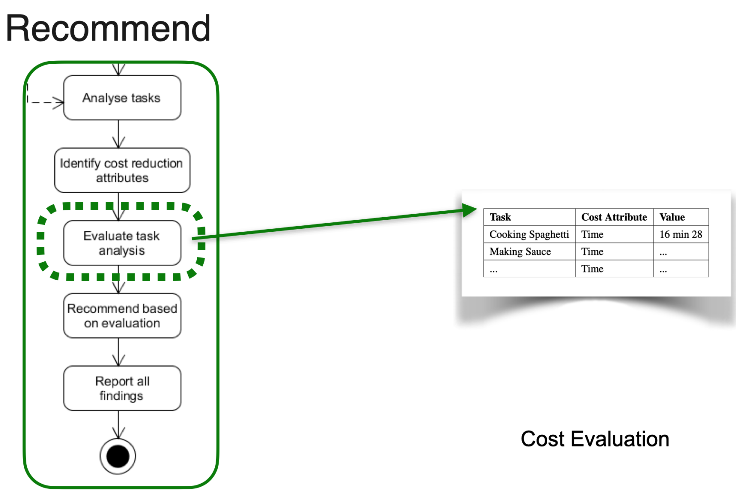
Figure 4.20: Explicitly identify desired outcome as a cost reduction attribute

Explicitly identify desired outcome as a cost reduction attribute (Figure 4.20)

Step 24: Answer the following:

- What is the desired outcome from Automation and this framework's application?

- What attribute can the desired outcome conform to?



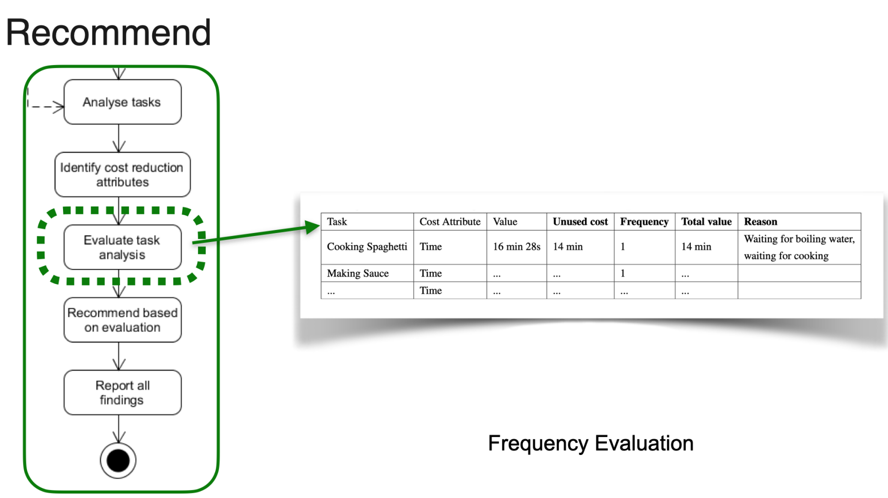
Task	Cost Attribute	Value
Cooking Spaghetti	Time	16 min 28
Making Sauce	Time	...
...	Time	...

Figure 4.21: Cost evaluation analysis

Evaluate task analysis against cost attributes

Step 25: Identify and rank tasks and aspects of tasks that impact on **cost**. Cost can be defined as monetary, resource, time or limited subject-matter expertise. Any goals of this task should initially focus on cost reduction.

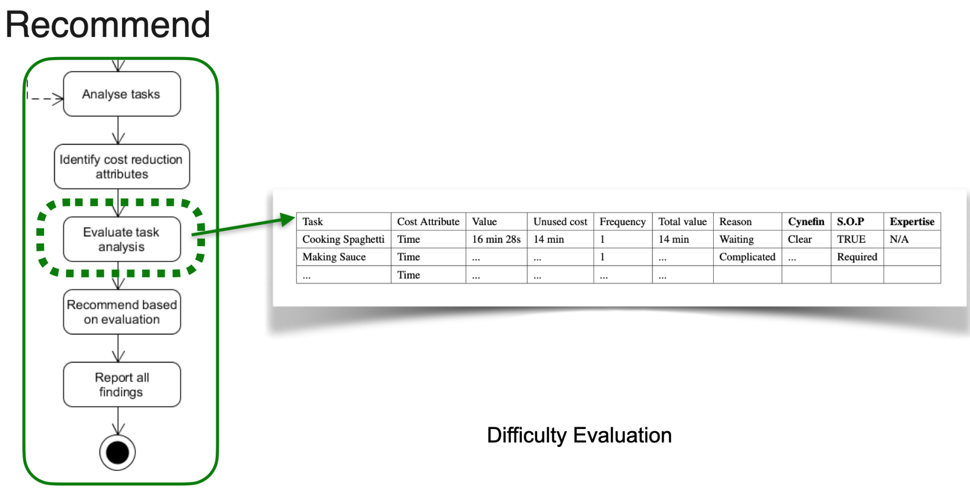
For each task, identify and order (high to low) the relative **cost** attribute (Figure 4.21)



Task	Cost Attribute	Value	Unused cost	Frequency	Total value	Reason
Cooking Spaghetti	Time	16 min 28s	14 min	1	14 min	Waiting for boiling water, waiting for cooking
Making Sauce	Time	...	...	1	...	
...	Time	...	...	...	...	

Figure 4.22: Frequency evaluation analysis

Step 26: Append and for each task, calculate (based on data from **Collect** phase and timeline from **Analyse** phase), any cost attribute that is wasted/not used, multiplied by **frequency** for a total value and reasons why (Figure 4.22). Order by total cost

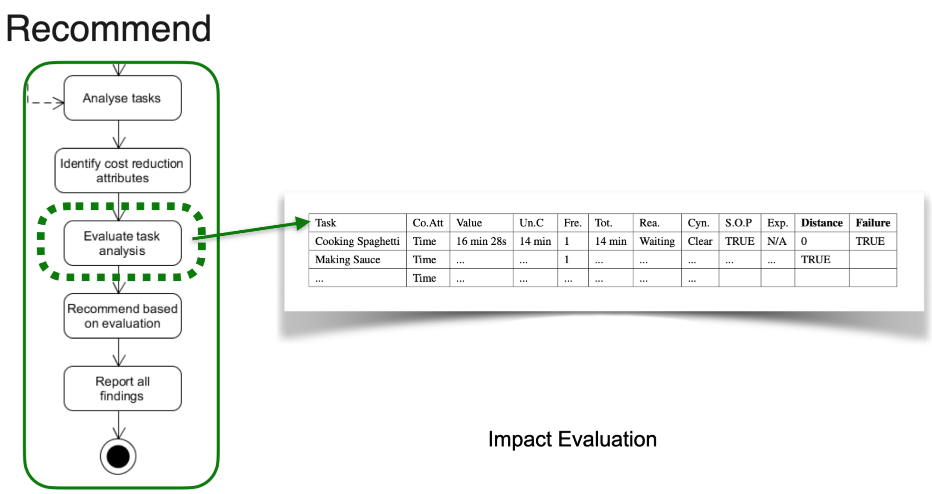


Task	Cost Attribute	Value	Unused cost	Frequency	Total value	Reason	**Cynefin**	**S.O.P**	**Expertise**
Cooking Spaghetti	Time	16 min 28s	14 min	1	14 min	Waiting	Clear	TRUE	N/A
Making Sauce	Time	...	...	1	...	Complicated	...	Required	
...	Time	...	...	...	...				

Figure 4.23: Difficulty evaluation analysis

Step 27: Append and for each task, assign **difficulty** attributes (Figure 4.23). (*Ideal world — Clear tasks, that follow a rigid rule set/standard operating procedures, that require no expertise*)

- *Cynefin domain — Clear, Complicated*
- *Rules/Standard Operating Procedure — True, False*
- *Expertise — Not required, Required*

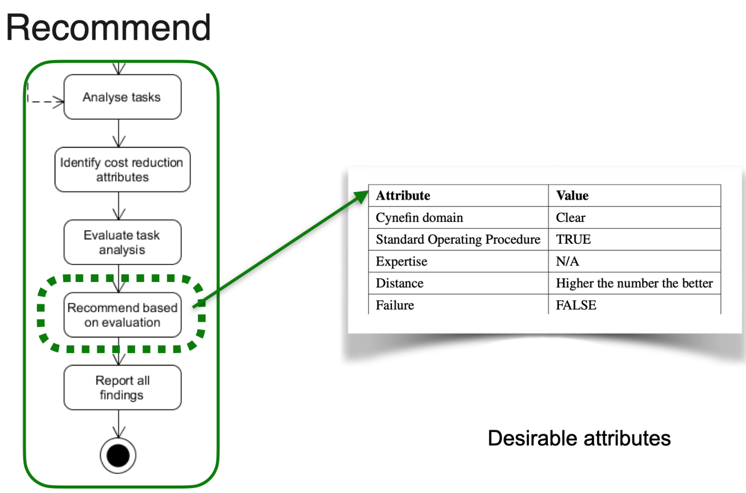


Task	Co.Att	Value	Un.C	Fre.	Tot.	Rea.	Cyn.	S.O.P	Exp.	**Distance**	**Failure**
Cooking Spaghetti	Time	16 min 28s	14 min	1	14 min	Waiting	Clear	TRUE	N/A	0	TRUE
Making Sauce	Time	...	...	1	...	...	...	...	...	TRUE	
...	Time	...	...	...	...	...	...				

Figure 4.24: Impact evaluation analysis

Step 28: Append and for each task, assign **impact** attributes (Figure 4.24) *(Ideal world — High distance and non-objective failing tasks)*

- *Distance — 0..N*
- *Failure — True, False*



Attribute	**Value**
Cynefin domain	Clear
Standard Operating Procedure	TRUE
Expertise	N/A
Distance	Higher the number the better
Failure	FALSE

Figure 4.25: Set of desirable attributes for ranking tasks for automation

Step 29: Previous steps have looked at the outcomes from tasks analysis and generating rankings based on different attributes. Using the table of attributed tasks, it is possible to identify different insights. Initially rank by the identified total cost attribute *(total cost = cost * frequency)*, then by either/or the difficulty and impact attributes. Tasks that follow Figure 4.25 are ideal for automation, based on desired outcomes:
e.g. Cooking Spaghetti has a total cost of 14 min of unused time. It has a clear procedure to follow, is a simple task that requires no expertise, but is critical to making dinner (Distance 0) and can cause dinner to be a failure.

4.2.5 Report

The final phase of DCARR is Report phase. This is an opportunity to present findings and outcomes in a coherent and concise format to communicate to interested parties. Generally the outcome of this phase would be the generation of an actionable report or document to highlight the findings and communicating the recommendations from the evaluation of the Recommend phase. Much like writing a thesis, the report stage is the culmination and final record of all the hard work done and achieved. As stated, the purpose of this phase is to:

- *Prepare outputs from* ***Describe, Collect, Analyse and Recommend*** *phases and generate a report*
- *Focus of the report should be on the:*
  - *Purpose of application of Framework*
  - *Scenario under review*
  - *Desired outcomes*
  - *List of outputs from* ***Recommend based on desired outcomes and evaluation rankings***
  - *Included evidence to support recommendations.*

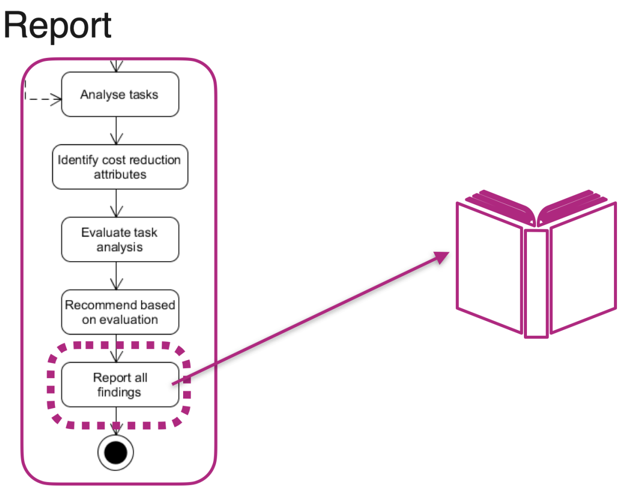


Figure 4.26: Report all findings

Report all findings (Figure 4.26)

Step 30: Write report

Step 31: Submit report

Step 32: Learn and improve framework based on application.

4.3 Development and Evolution of DCARR

As outlined in the research method (Section 3.3) which aims to address the research questions (Section 3.2), the development and evolution of DCARR followed two methodologies; Framework Construction (Section 3.3.1) and Framework Testing and Evaluation (Section 3.3.5). In support of these methods, a set of criteria was devised (Section 3.3.2), which aimed to address the research questions (Section 3.3.3). A breakdown of the criteria for development and evolution (Section 3.3.4) that were used in agreement to achieve the aims of SMEs is as follows. Table 4.1 is the breakdown of the metrics and measures for testing and evaluation (Section 3.3.6) for the assessment and validation of the proposed criteria.

- **Coverage - Breadth** — *The effort required and the method to identify tasks that are worth investigation (tasks that are deterministic in nature)*
- **Coverage - Depth** — *The effort required and the method to determine the necessary level of decomposition to accurately depict the task process*
- **Alignment** — *The mapping / linking between the overarching objective that drives the scenario and the tasks involved*
- **Representation** — *The effort required to communicate the identified coverage and alignment*
- **Reproducibility** — *The structured methodology*
- **Dynamicism** — *The applicability and effort required to examine similar structured/semi-structured/free form problems.*
- **Existing Theory** — *The use of existing theories that are applicable/have been applied*
- **Effort** — *Cost, Developmental, Automative effort required. Can be represented in terms of time allocation, price, technical debt, technical availability, process life, development resources, etc.*

To address the changes and development of DCARR, a simple table is used to highlight and demonstrate the criteria validations during each stage of the development and evolution, and accompanies each figure. Table 4.2 is the full completed table highlighting the figures that represent the changes and evolution of the DCAAR framework throughout development.

As part of testing and validation, a series of scenarios were used (Section 3.3.7). These scenarios were developed with assistance from SMEs from DST Group, SMEs from external contractors, literature review and the researchers own experience. The initial development approach, was to generate a metamodel (Figure 4.27 and 4.28) of an abstracted and theoretical LVC distributed scenario based on the work by Simpkin et al. [5].

Table 4.1: Criteria metrics. Desired true/false feature state is true. Desired minimal, moderate or high effort value is minimal. Minimal is interchangeable with Low

	True or False	**High, Moderate or Minimal Effort**
Coverage — Breadth	Is the Task deterministic?	Automation, Cost
Coverage — Depth	Does the Task exist?	Development, Cost
Alignment	Is there a mapping between Task and Objective?	—
Representation	Is Coverage and Alignment represented?	Cost, Development
Effort	—	Automation, Cost, Development,
Reproducibility	Is there a structured methodology?	—
Existing theory	Is there a foundation in existing theories?	—
Dynamicism	—	Framework application to different problem space

Table 4.2: Overview of the development and evolution of the DCARR Framework in relation to the proposed criteria metrics. For true/false features, the desired outcome is true. For low, moderate, high effort values, the desired effort value is low.

Criteria / **Figure**	**Breadth**	**Depth**	**Alignment**	**Representation**	**Reproducibility**	**Dynamicism**	**Existing Theory**	**Effort**
Figure 4.30	False	False	False	False	False	High	False	High
Figure 4.31	False	False	True	False	False	High	False	High
Figure 4.35	High	High	True	High	False	Moderate	True	High
Figure 4.39	High	Moderate	True	High	True	Moderate	True	High
Figure 4.44	Moderate	Moderate	True	Moderate	True	Moderate	True	Moderate
Figure 4.52	Moderate	Moderate	True	Moderate	True	Moderate	True	Moderate
Figure 4.59	Moderate	Moderate	True	Moderate	True	Moderate	True	Moderate
Figure 4.65	Low	Low	True	Low	True	Low	True	Moderate
Figure 4.72	Low	Low	True	Low	True	Low	True	Low
Figure 4.82	Low	Low	True	Low	True	Low	True	Low

Figure 4.27 is the UML representation of the Meta Object Facility or MOF, that defines how an object diagram conforms to a User Defined Model. This User Defined Model, conforms to a Metamodel, which is turn conforms to the Meta-metamodel notation. For the benefit of discussion, Figure 4.28 is a representation of the MOF in relation to DCARR.

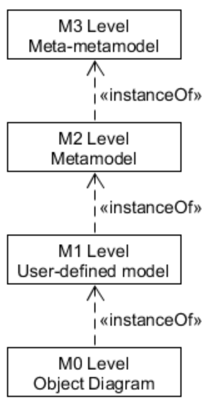


Figure 4.27: Simple representation of the Meta Object Facility or MOF

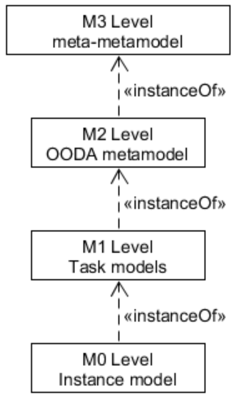


Figure 4.28: MOF for the proposed framework

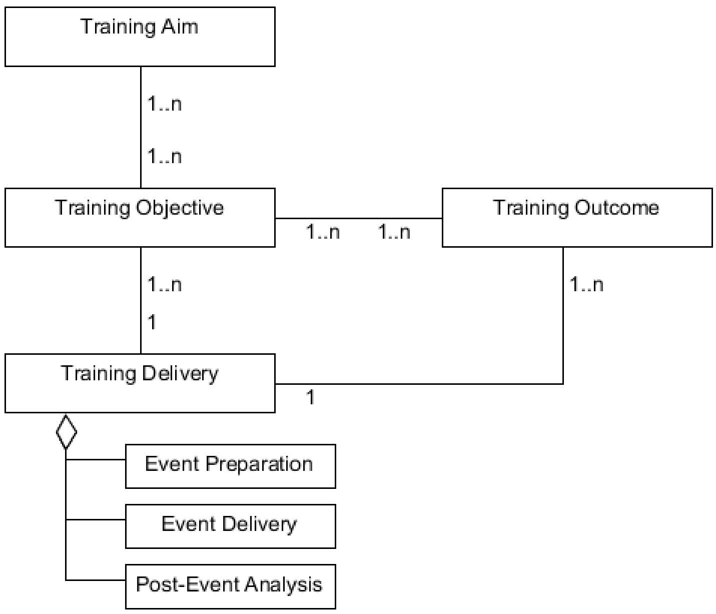


Figure 4.29: Initial Phase Diagram of LVC training, that aimed to align Delivery to Outcome, Objective and Aim

4.3.1 Building DCARR

DCARR was built to address the research questions and hypothesis that stemmed from the identified gaps in literature. To do this, DCARR was built and developed to address the proposed criteria (including metrics), agreed upon between the researcher and SME's at DST Group. Initial development on DCARR began as an attempt at meta-modelling a distributed synthetic training exercise, as described in the work by Simpkin et al. [5]. As criteria was addressed, it became apparent that a metamodel of an exercise was insufficient at addressing the proposed criteria. This led to the introduction of a semi structured framework, which over time developed and evolved into DCARR.

As per the research method (Section 3.3), the DCARR framework was built in an iterative approach. For an overview of the metamodel to framework version development refer to Appendix B. For all figures and diagrams related to the development, evolution, testing and validation of this research, refer to Appendix C.

Figure 4.29 is the initial model used to encapsulate and represent event preparation, event delivery and post-event analysis. This is derived from the work Simpkin et al. [5]. These outcomes can be reasoned to stem from training delivery, which is driven by the concept of training objectives and drives training outcomes. Overall, the objectives for training are driven by an overarching aim of the exercise. This is an initial attempt, with which a metamodel can be developed from.

Criteria / Figure	Breadth	Depth	Alignment	Representation	Reproducibility	Dynamicism	Existing Theory	Effort
Figure 4.30	False	False	False	False	False	High	False	High

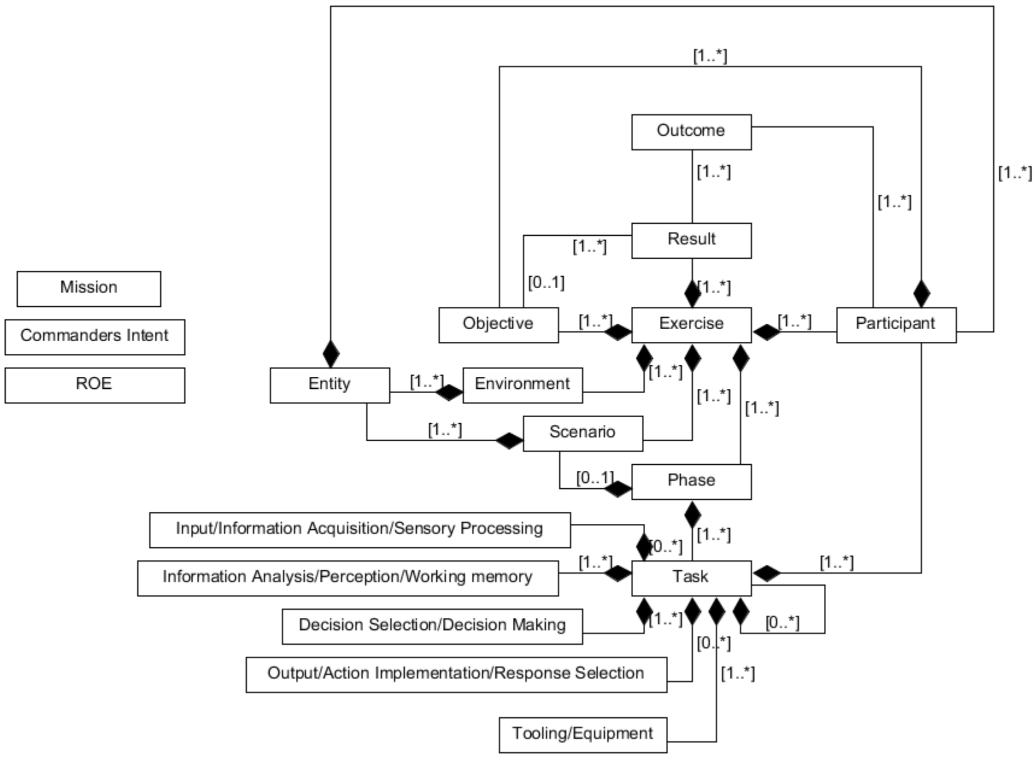


Figure 4.30: First attempt at meta-modelling training delivery

From the initial phase diagram (Figure 4.29), it was necessary to formalise and develop the concepts contained within to allow for evaluation against the criteria metrics. To do so, the researcher attempted to generate a metamodel of the concepts surrounding event delivery to capture the necessary information that would be pertinent to automatability discovery that addressed the criteria. Figure 4.30, is an initial attempt at meta-modelling training delivery. This was based on meta-modelling concepts and theories from UML methodology. As demonstrated, the initial metamodel includes the concepts of an exercise as a central pivotal event that occurs, which is based on the overarching objectives that drive any exercise. From the exercise are the concepts of participants, results, environments, entities, phases, outcomes and tasks. The concept of task is further divided into the concepts of input, analysis, decision-making and output (and other related terminology). It is this subdivision of a task into fundamental concepts that further drives the development and evolution due to the need to address Coverage - depth and breadth. This is also the first attempt at establishing Alignment, presenting a mapping between task and objective.

Figure \ Criteria	Breadth	Depth	Alignment	Representation	Reproducibility	Dynamicism	Existing Theory	Effort
Figure 4.31	False	False	True	False	False	High	False	High

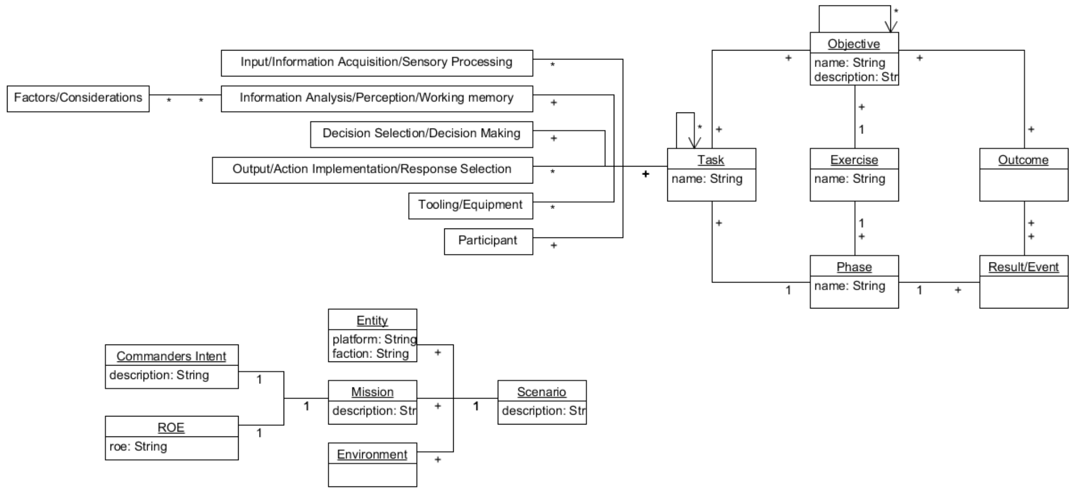


Figure 4.31: Refinement of event delivery using meta-modelling techniques

Figure 4.31 builds upon and refines the concepts and ideas of the initial metamodel of LVC synthetic training delivery. The focus of this metamodel is the simplification of the method for Alignment, by establishing clear relationships Exercise, Objective and Task. This metamodel aims to address Coverage - Breadth, by task decomposition into a simple model of information processing, and the notion that tasks can contain subset of tasks, which can contain a subset of tasks, etc. This model also begins to address Coverage - Depth, through the recursive task decomposition, but instinctively it can be ascertained that there exists no limiting or terminating factor for depth and decomposition. Other observations are the disconnect of Scenario from Exercise (as well as concept of Entity, Mission, Environment, Intent and Rules of Engagement).

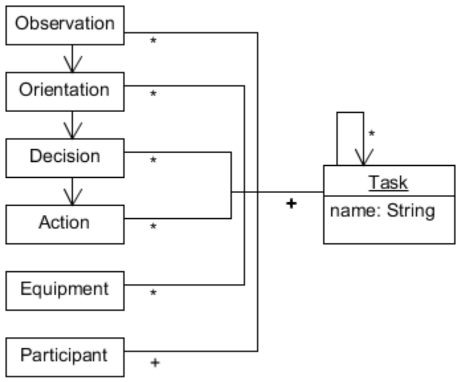


Figure 4.32: The relationship between elements of simple information processing is missing, therefore this step assigns the relationship / data flow from Observations, to Orientation, to Decisions and finally to Actions. This is also the introduction of OODA as a concept into the development of the future framework.

Figure 4.32 is the addition of relationship between elements of information processing for Task and introduction of terminology of OODA Loop.

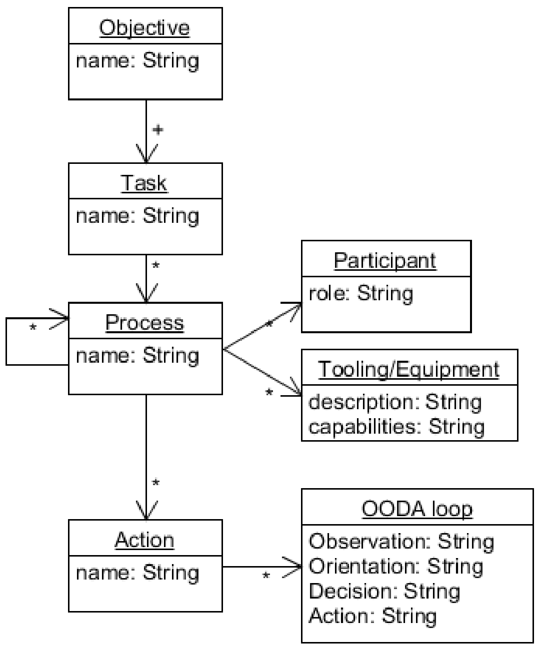


Figure 4.33: Breakdown and the segmentation of recursive "Task" entity into a hierarchical tier system of Task, Process and Action entities. The benefit of this is the ordering and representation of tasks as a tree structure, with Task acting as root nodes, while Action acts as leaf nodes. Task becomes the large scale divisible component that comprises processes, while action is the smallest unit of indivisible work that can occur.

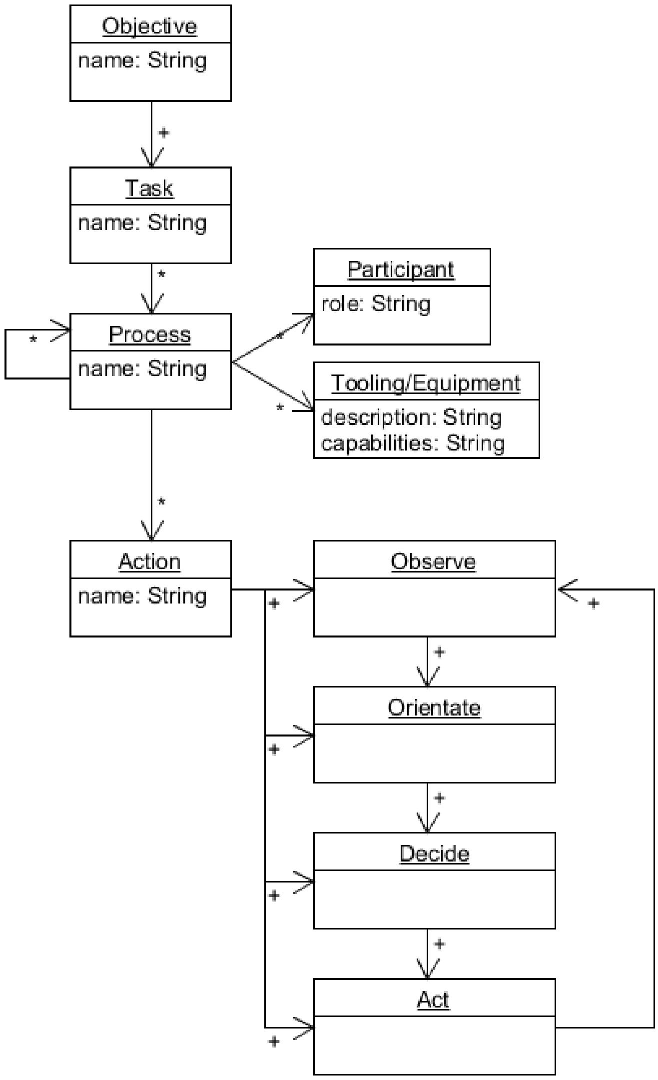


Figure 4.34: Maintaining task name and hierarchy changes from Figure 4.33, and reconstruction of individual OODA elements from Figure 4.32

Figure 4.33, is the decomposition of Task into Process and Action to address methods for Coverage. This saw models missing relevant relationship between data and actions within the concept of OODA Loop, therefore Figure 4.34 reintroduced individual entities for Observe, Orientate, Decide and Act. It also added a recursive component, linking Act into Observe - the implication being that every action changes the observable state and therefore there is a direct relationship between the two.

Criteria / Figure	Breadth	Depth	Alignment	Representation	Reproducibility	Dynamicism	Existing Theory	Effort
Figure 4.35	High	High	True	High	False	Moderate	True	High

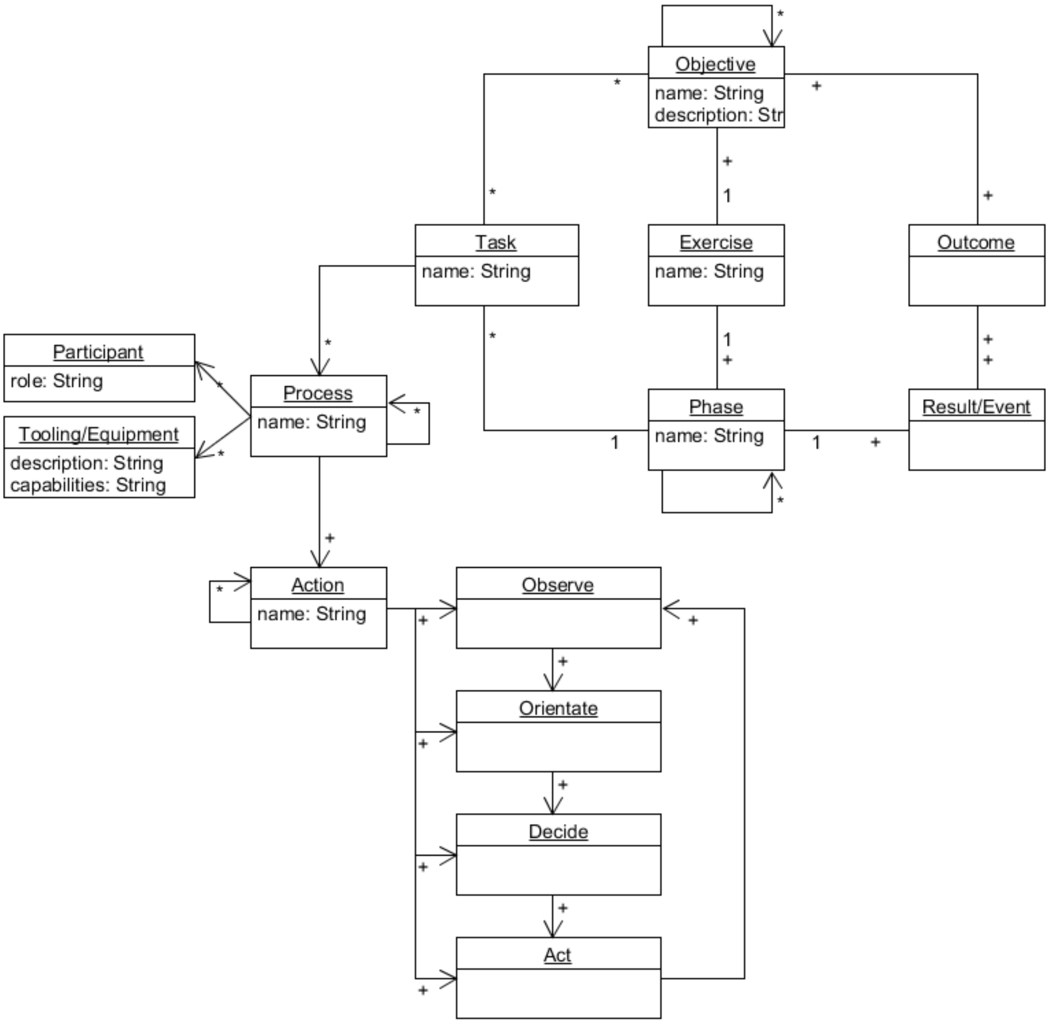


Figure 4.35: Separation of OODA loop into O-O-D-A elements. Redefinition of Macro, Meso and Micro to Task, Process and Action

Figure 4.35 focuses on Task decomposition and task description with the breakdown of individual definable OODA elements. The attempt being an approach to classify and quantify the elements of tasks that impact on training delivery. This is also an attempt at addressing Coverage - Depth, Representation, Alignment, Existing Theory and Dynamicisim criteria. At this stage, it is also possible to represent and communicate the elements of an OODA action as a flow charts using UML specification.

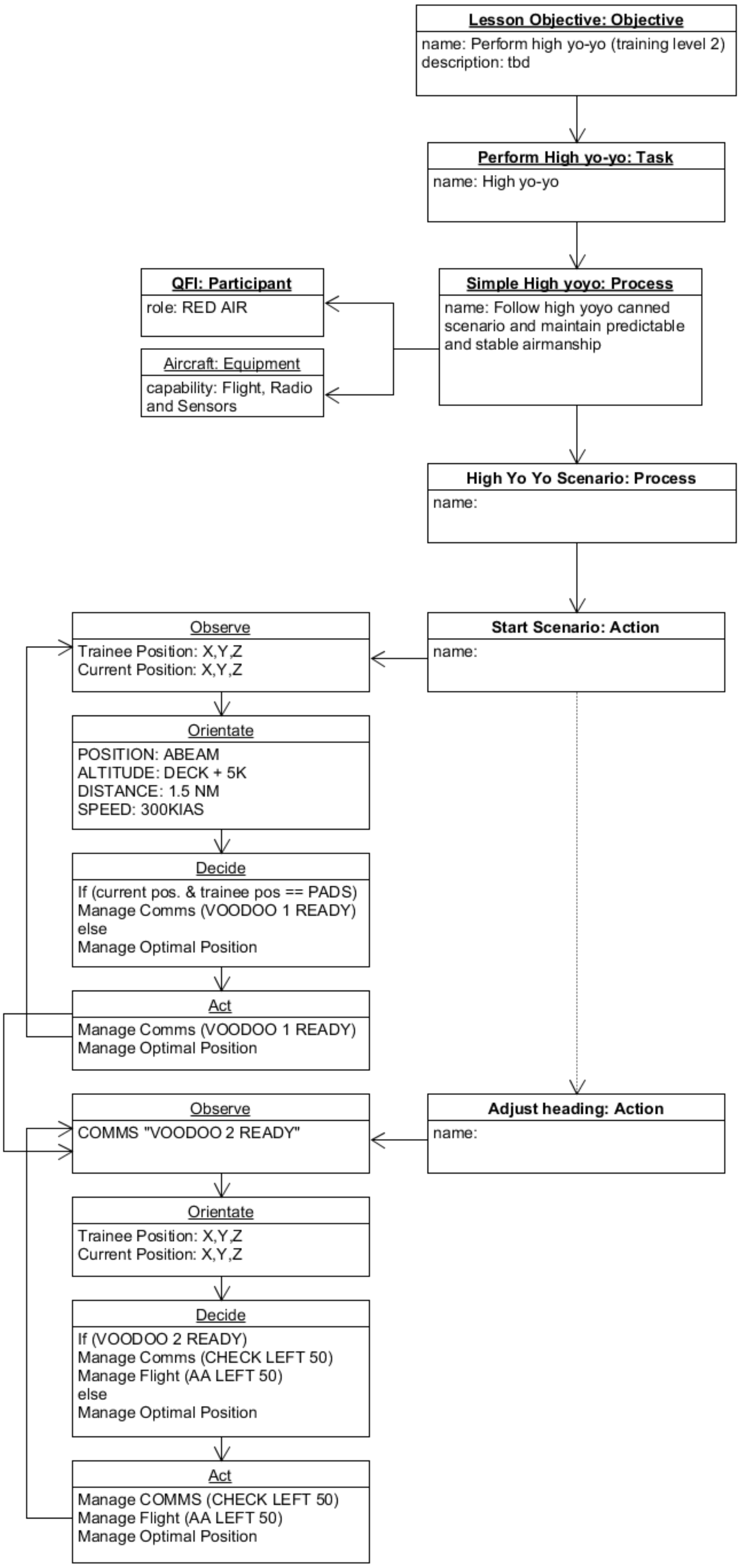


Figure 4.36: Worked example using Figure 4.35 and High Yoyo scenario, focused on Alignment, Coverage - Depth, Representation, Existing Theory and Dynamicisim.

Figure 4.36 is the worked example using the High Yoyo scenario and using Figure 4.35 as the metamodel used to represent the relationship between Objective, Task and the required information processing. A singular task was chosen (Perform High Yoyo), to ascertain Alignment, appropriate Coverage - Depth, Representation, Existing Theory and Dynamicism. As demonstrated, there is a relationship between the lowest indivisible component of action, (Start Scenario and Adjust Heading). Therefore, as shown in Figure 4.35, there is a change to the recursive aspect of the model - instead of processes being recursive, actions are instead. During testing and development, the size and complicatedness of the model became unwieldy and difficult to communicate, therefore changes were made, in the form for different representations.

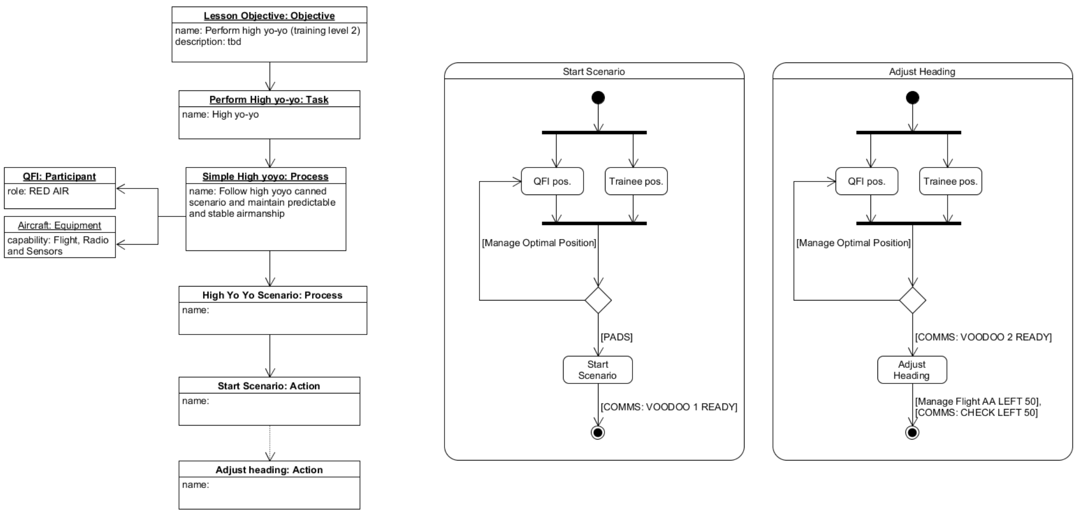


Figure 4.37: Worked example using Figure 4.35 and High Yoyo scenario. Major change to representation. Using Flow chart UML diagrams to represent Action task O-O-D-A components.

Figure 4.37, demonstrates the changes made to representation. From entities that flow into one another, UML flow chart diagramming was used to communicate an action could be represented as a singular state transition regardless of external factors. These changes (actions as flow charts/state changes, and the separation of alignment and coverage/representation into individual models), began to shape and form the beginnings of a framework from the initially presented metamodel concept.

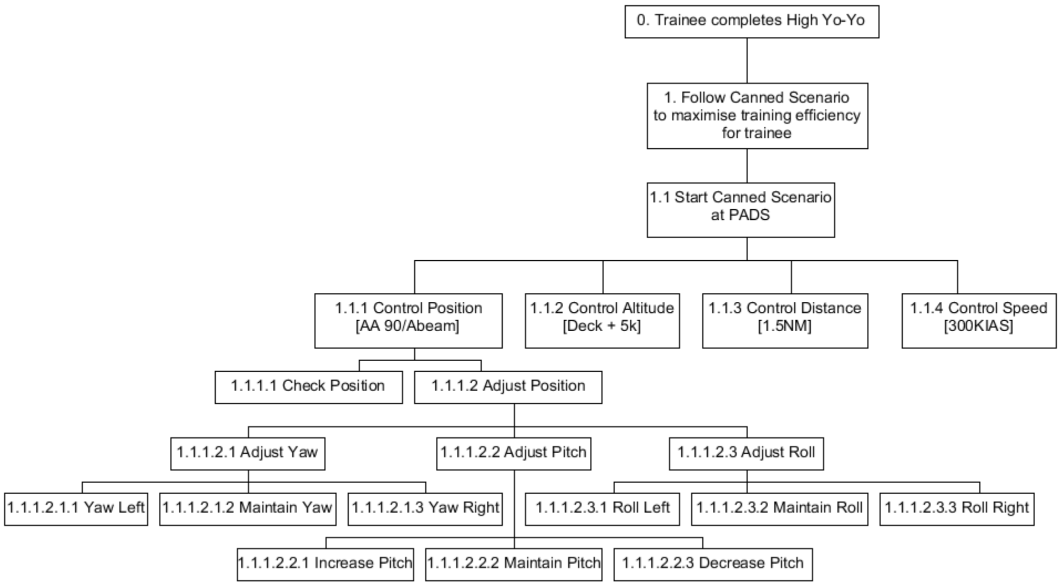


Figure 4.38: Worked example using Figure 4.35 and High Yoyo scenario. Major change to Coverage - Breadth and Depth. Using Hierarchical Task Analysis for the formal method for task decomposition and OODA element elicitation

By separating task representation from alignment and coverage, it became possible to further explore the initial concepts identified earlier, i.e. representation of alignment as a tree structure. Figure 4.38 demonstrates this change to Coverage through the use of an existing and well established method of Hierarchical Task Analysis from Human Factors methodology, for the formal method of task decomposition and representation.

Figure \ Criteria	Breadth	Depth	Alignment	Representation	Reproducibility	Dynamicism	Existing Theory	Effort
Figure 4.39	High	Moderate	True	High	True	Moderate	True	High

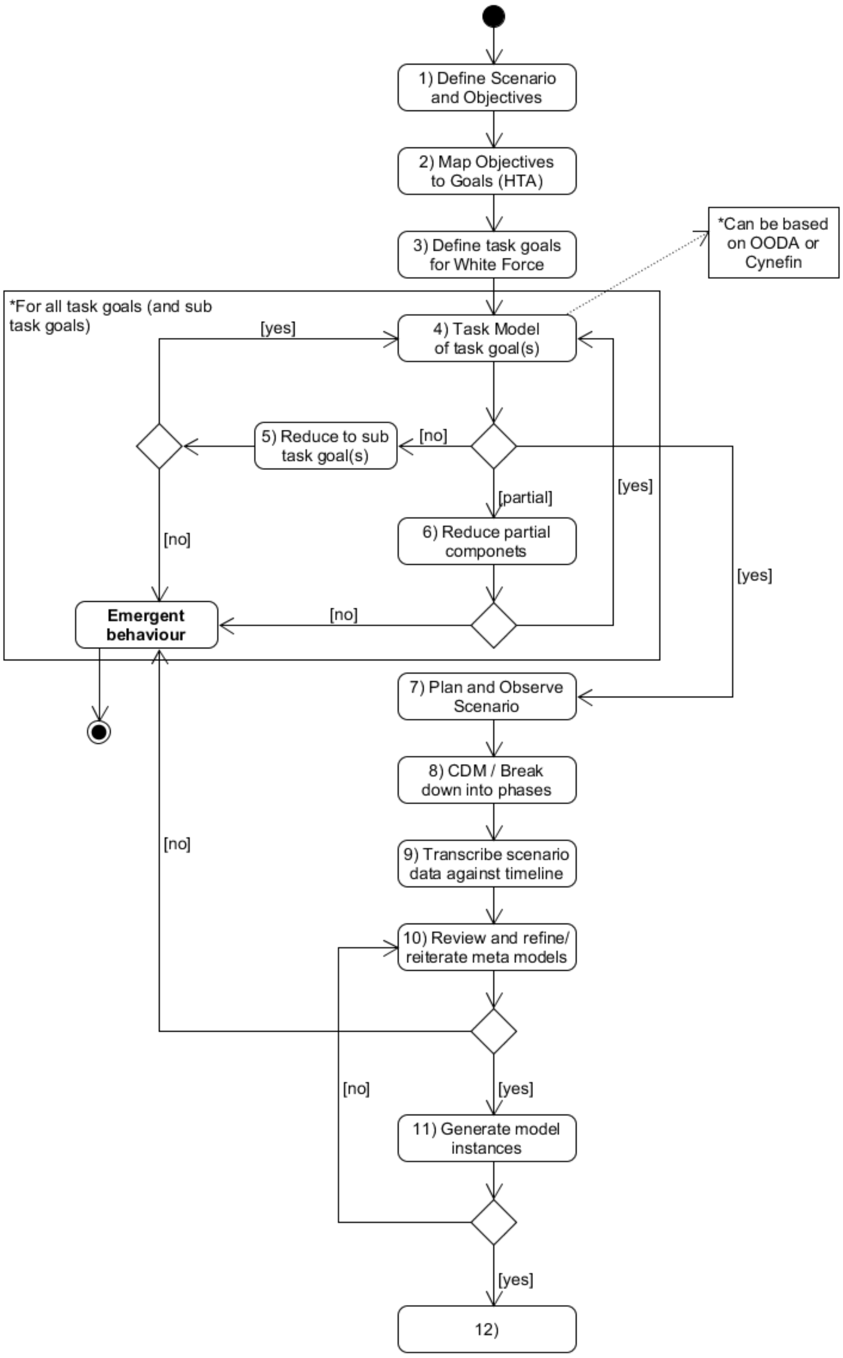


Figure 4.39: Introduction of EAST structured framework method. No longer a metamodel

Figure 4.39 is the culmination of a series of changes that transformed the initial metamodel into a semi structured method. This included the introduction of Hierarchical Task Analysis as a Human Factors method for appropriate task coverage depth and breadth, as well as the introduction of Event Analysis of Systemic Teamwork structured framework as a similar domain application and structured approach. With the use of EAST, this transforms the metamodel driven approach into a reproducible model driven Framework. This framework includes the start of structured step-by-step directions to follow. It clearly defines and forces alignment with the step of Objective overview. It also includes the introduction of Cynefin as the theory of emergent behaviour and complexity. This model aims to address coverage breadth and depth.

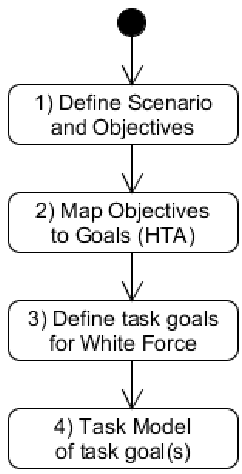


Figure 4.40: Focused on Objective definition and Goal mapping

Figure 4.40 focuses on the initial simple steps of the framework - the definition of the scenario under review and objectives of said scenario. From here, the next step is to use aspects of the HTA method to generate a mapping from Objectives to Goals. This enforces a strong relevance for the tasks under review and the overarching objectives. The relates back to Alignment. The next step is the definition of goals for White Force and finally the generation of task models of task goals. The recursive nature of the final step presented is highlighted and demonstrated in Figure 4.41.

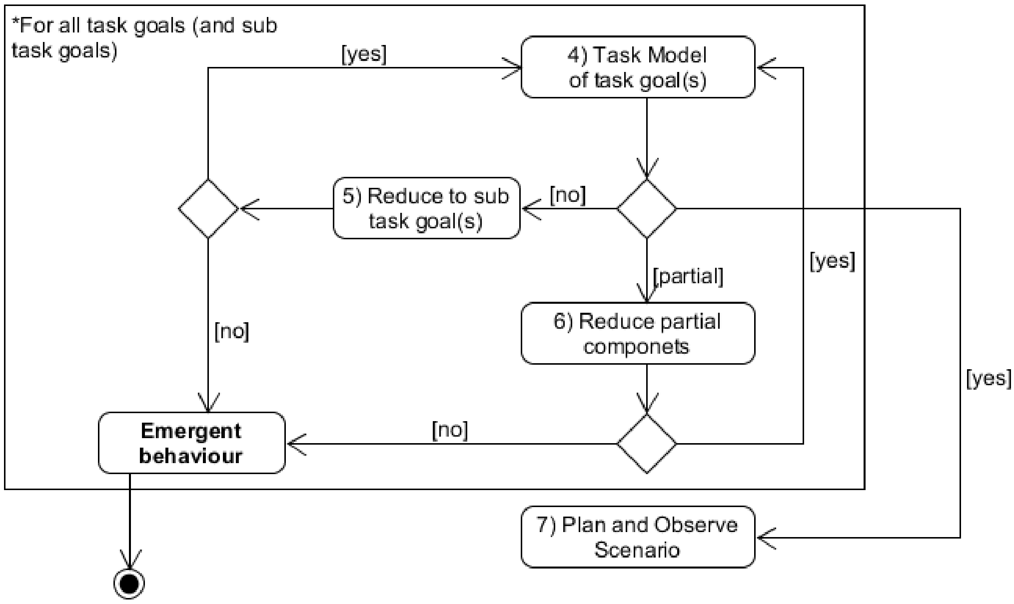


Figure 4.41: Recursive approach (as per HTA method) for the decomposition (Coverage depth / breadth) of tasks. Introduction of Cynefin definitions of emergence and complexity. Clear demarcation for Out of Scope Non-deterministic tasks.

Figure 4.41 highlights the recursive nature of Coverage - depth. For all high level task goals (and subsequent subtask goals as required - as per the methodology of HTA), a simple decision is made to identify and classify the appropriate Coverage - Depth and Coverage - Breadth. If a task model, at the highest "level" can be generated, meaning that the task is deterministic in nature and all OODA elements can be described, then the framework can move onto the data collection phase. Otherwise, if a task model is unable to be generated, then subtask goals (as per HTA) are to be defined. If further decomposition is unable to occur, then partial decomposition of OODA elements is required (perhaps as part of Collect and Analyse phases, when task models are reviewed against collected data, the undefinable aspects of tasks can be made clear). If partial reduction is unable to occur, then the task is deemed too complex/non-deterministic and regarded as emergent behaviour and therefore is out of scope / unable to be assessed. This is the first introduction of the concepts of Cynefin, complexity and determinism. It is also the introduction of termination points as part of the process and analysis of tasks.

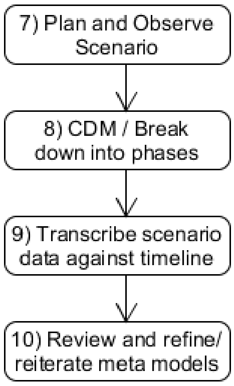


Figure 4.42: Data collection method and timeline transcription (as per EAST framework)

Figure 4.42 makes use of the methodology of EAST and Human Factors methods for the collection of quantitative data to support future recommendations. As part of the outcomes from Figure 4.41, areas of interest have been identified as observations as part of task models. Therefore, as part of data collection, it is necessary to collect data to support the assumptions made. Once data has been collected, a timeline of critical decisions and phases is generated with data included to support the described task models. This allows for the initial task models to be reviewed and refined as required.

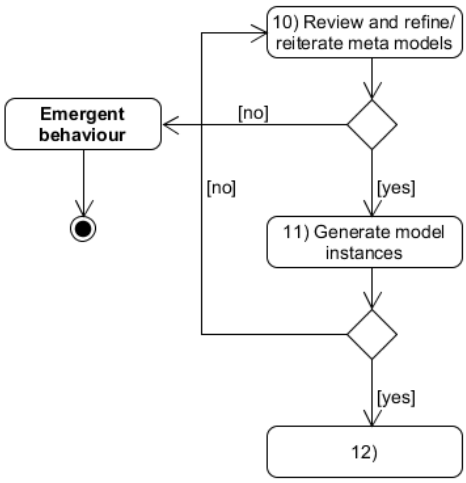


Figure 4.43: Using collected data to confirm theoretical concepts against reality and real data. Framework has no analysis or recommendation stage.

Figure 4.43 focuses on the tail end of the initially proposed framework. As previously stated, (Figure 4.41), during the review/refine task model step, it may or may not be possible to reconcile the partially decomposed task models with collected and observed data. Therefore, as with the non-deterministic and non-decomposable task models, anything that cannot be defined is regarded as emergent and regarded as out of scope. Once reviewed and refined as required, instances of task models with collected data are generated. These are used in further analysis. Unfortunately, this iteration of the framework does not include the steps or methods for further analysis. Therefore, it is necessary to further refine and develop the framework through testing and criteria validation.

Figure \ Criteria	Breadth	Depth	Alignment	Representation	Reproducibility	Dynamicism	Existing Theory	Effort
Figure 4.44	Moderate	Moderate	True	Moderate	True	Moderate	True	Moderate

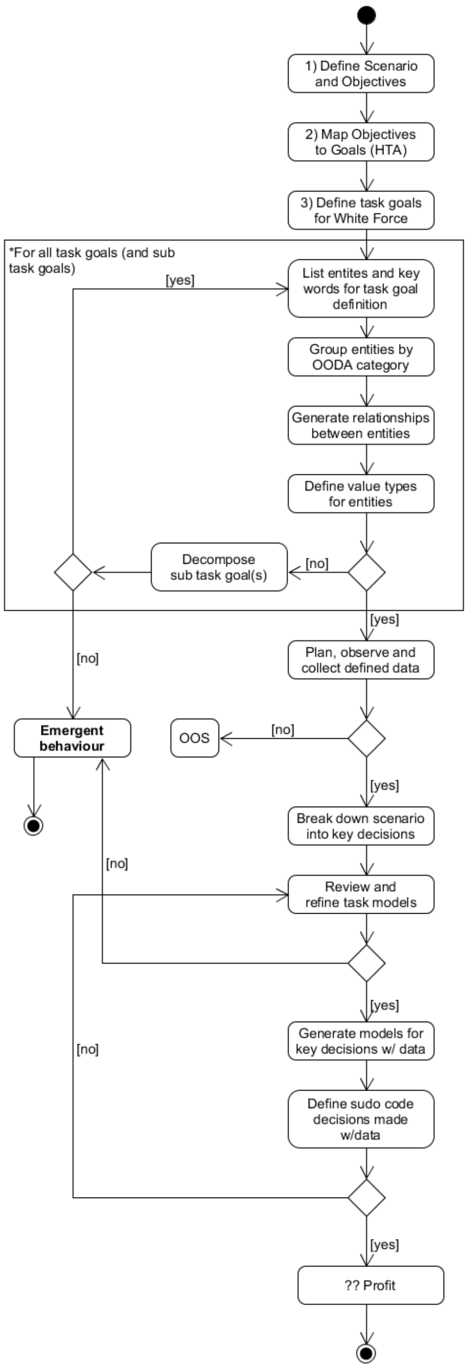


Figure 4.44: Relational diagram based on OODA elements. Attempt at fleshing out incremental steps and procedures

Figure 4.44 aims to expand upon the initial ideas and concepts behind the driving motivation of the framework - a method to examine and identify which tasks are suitable for multiplication within an overarching process. This iteration focuses on the Actions performed and presents a method for the visual representation of the components that make those tasks and processes. Task decomposition is focused on subtask goals until able to extract OODA elements, keywords and entities. This is also an attempt at representation. Maintains the alignment criteria from the previous iteration.

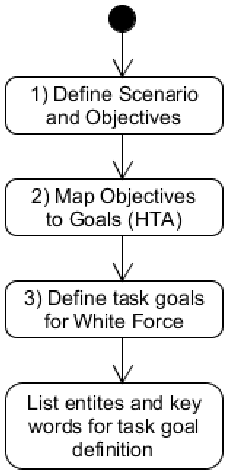


Figure 4.45: Same focus on Scenario and Objective mapping as previous iterations.

Figure 4.45 shows the similar steps for alignment as the previous iteration of the framework. The differences being rather than the generation of Task models for task goals, this iteration focuses on the extraction of entities and keywords from task goal definitions. The change was necessitated for the lack of a concrete method for task model generation. Instead, this iteration of the framework aims to provide a simpler step-by-step approach to guide users.

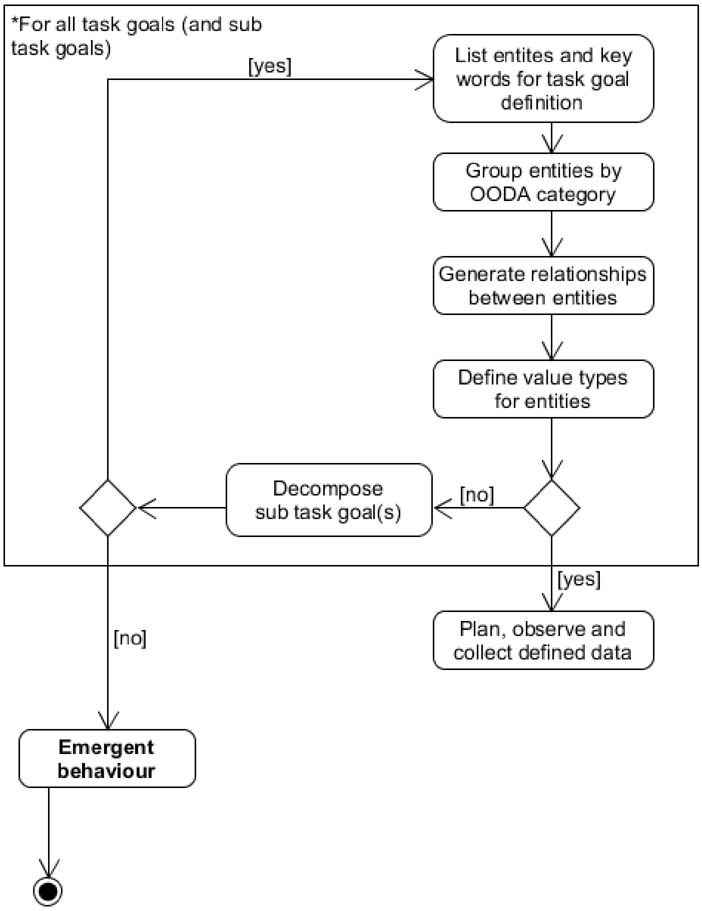


Figure 4.46: Major iteration in terms of task representation. Focused on the visual representation of action/micro tasks through the use of relational diagrams to represent O-O-D-A elements

Figure 4.46 is a similar recursive loop as the previous iteration. The attempt of this iteration of the framework is to focus on task representation through the clearly defined steps to be followed. Rather than broad attempts at task goal decomposition into subtask goal or partial subtask goal, instead the focus is on the extraction of OODA entities and the generation of the relationships between entities - similarly to the flow chart representation of OODA elements. The other differences here are the value typing for entities. This is to better equip the data collection phase and begin to provide benefit from smaller applications (earlier exit/termination points) of the framework (rather than the intensive and exhaustive 5 phase program). The same out of scope / emergent behaviour terminator exists, to address Coverage - breadth and Coverage - depth.

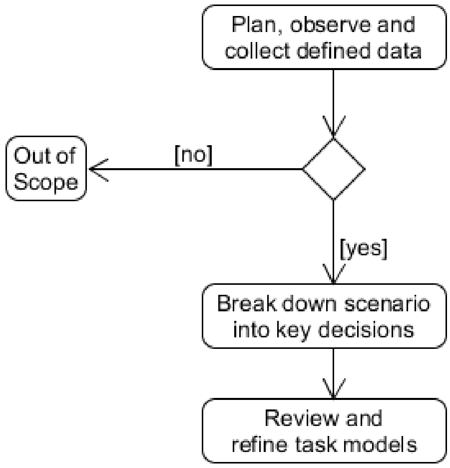


Figure 4.47: Data collection methods and Out of Scope terminators due to unavailable data collection.

Figure 4.47 is an improvement upon the previous iteration. Rather than a step by step replication of the method for data collection as proposed by EAST, instead, an "Out of Scope" point is introduced in case data is unable to be collected due to its nonexistence, security concerns or technological limitation. This allows for the frameworks' application to continue with the caveat that some information is allowed to be missing.

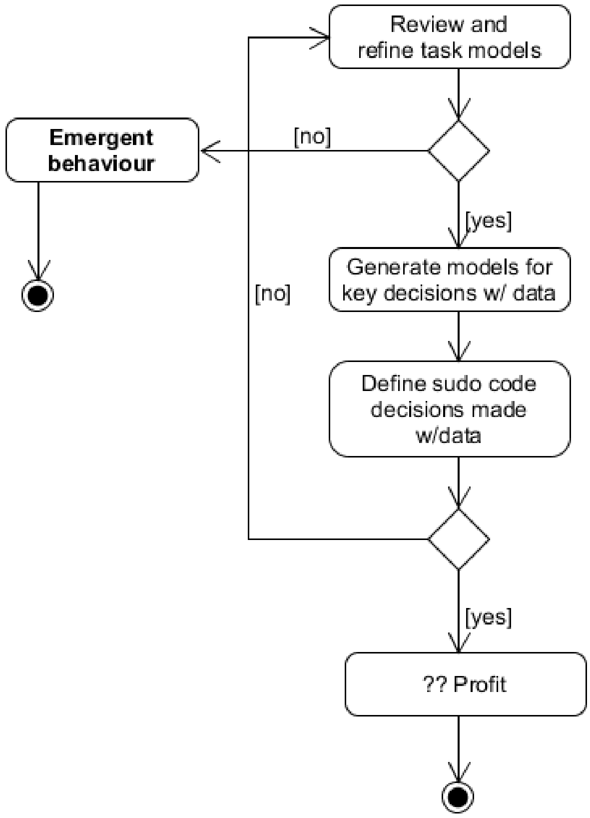


Figure 4.48: Revision of theoretical models. Termination point for unexpected unexplained information. Generation of pseudo code for decisions, attempt at data transformation for actionable outcomes. Still missing analysis and review section.

Figure 4.48 highlights what is becoming the analyse and recommend phases. Previous iteration provided the steps for the review and refinement of task models, with the inability to do so, resulting in the declaration of emergent behaviour and termination or with the step to generate models for analysis. This iteration continues to expand upon the analysis requirements by expanding on the notion of giving substance to OODA elements. Previously introduced the concept of value typing entities. These entities are those that are measurable and quantifiable i.e. observations / data. The approach here is to provide substance to decisions by generating simple pseudocode to classify and represent the decision-making for future recommendations. Sadly, this iteration still comes up short with further analysis and recommendations and therefore requires further refinement and development.

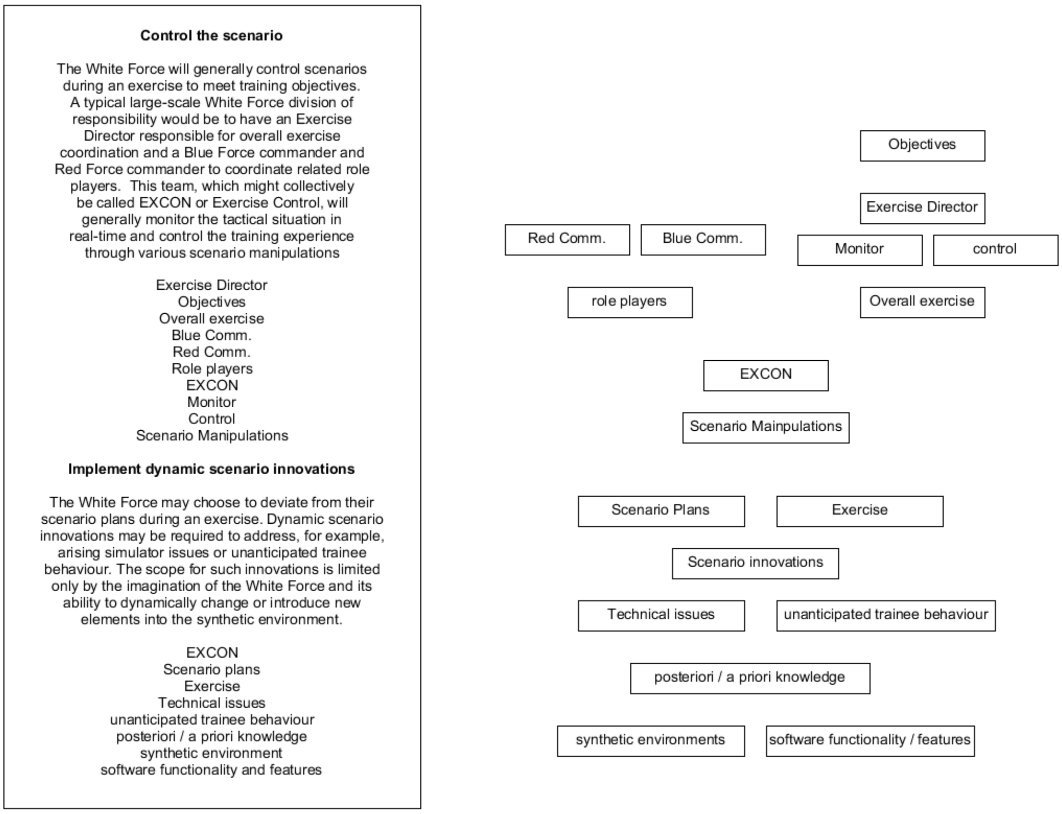


Figure 4.49: Worked example using High Yoyo scenario. Generated task descriptions and entity extraction

Figure 4.49 is a worked example using the High Yoyo scenario and the steps of the framework. As shown, the task of "Control the Scenario" is presented with a definition provided of what that entails. Below it are participants, entities and concepts that have been extracted. Below that, is the task "Implement dynamic scenario innovations", with a definition and entities also extracted. To the right of the figure, free form, are the entities - each entity exists within its own box.

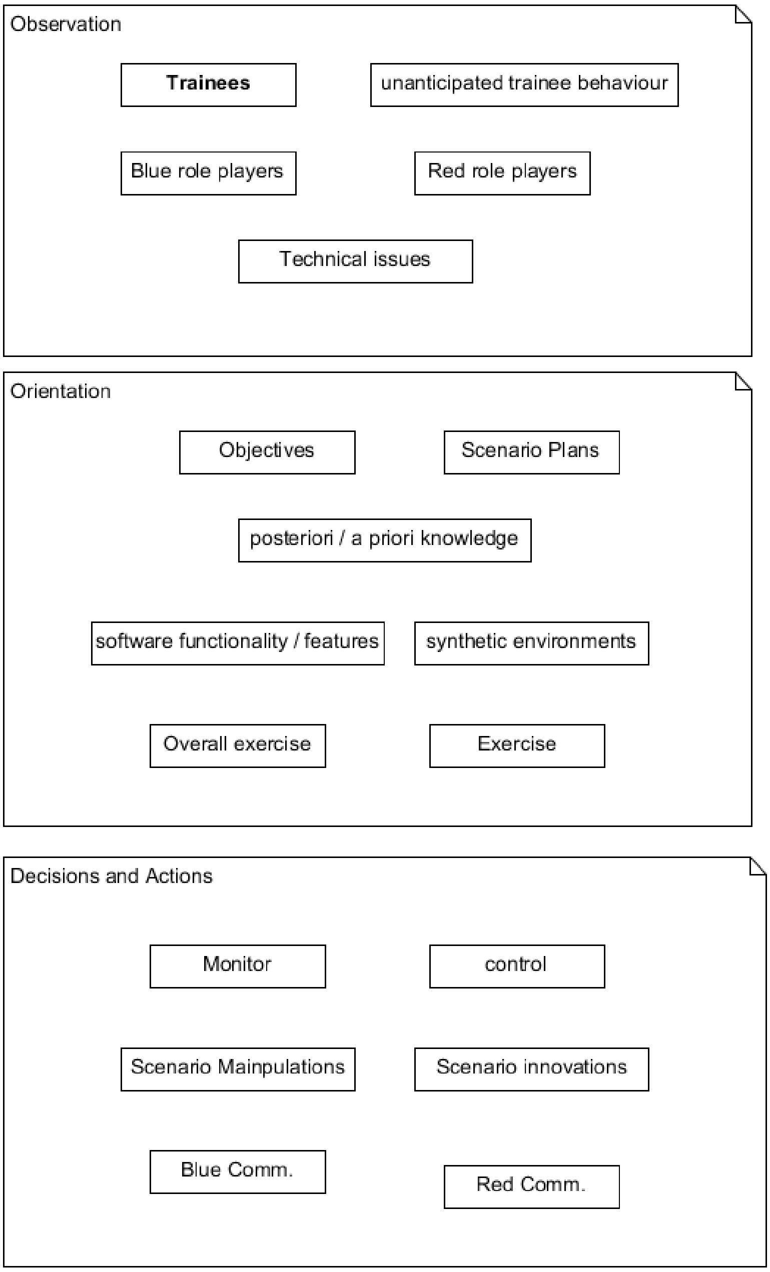


Figure 4.50: Worked example using High Yoyo scenario. Entity grouping into OODA elements

Figure 4.50, is the further worked example using the High Yoyo scenario. This figure presents the entities from Figure 4.49, grouped into OODA components. All observations and observable states are group. All orientations and knowledge bases (information that informs decision-making) are grouped. All decisions to be made and actions outcomes are also grouped together.

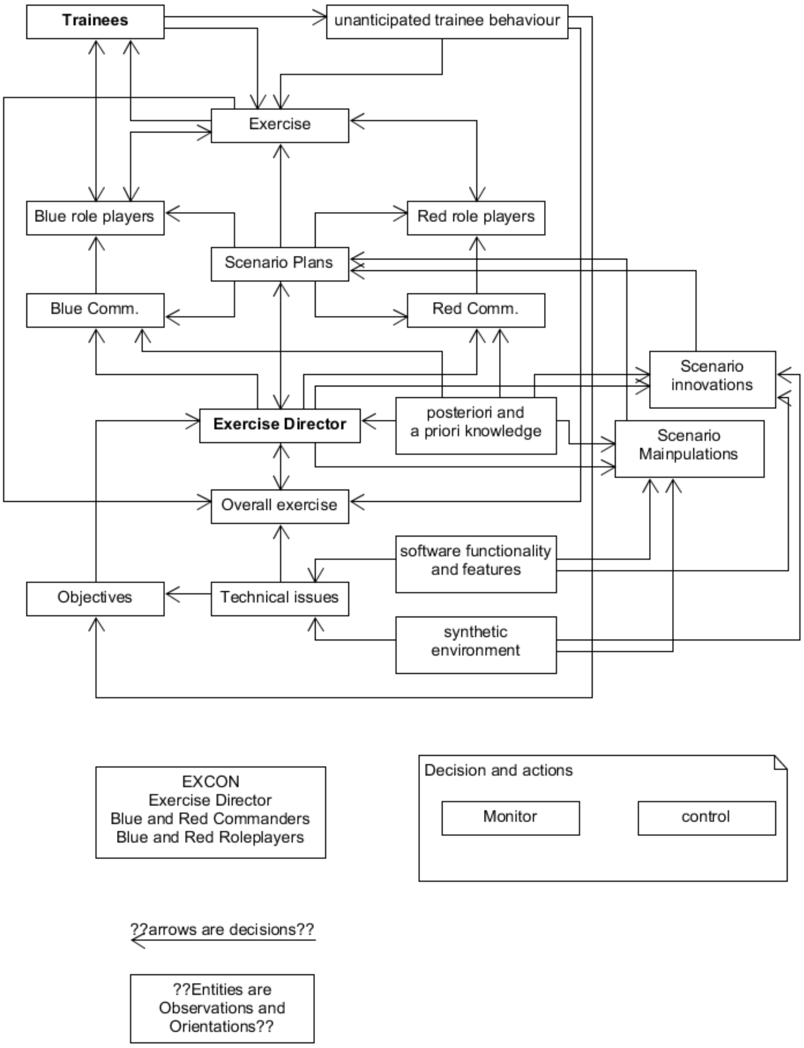


Figure 4.51: Worked example using Figure 4.44 and High Yoyo scenario. Relational diagram between OODA elements. No method defined for analysis

Figure 4.51, is the entity relational diagram generated using the entities extracted from Figure 4.49. This is a first pass attempt at generating a meta / task model of a series of tasks. No further details between relationships exist other than arrows. No method for further analysis exits. No value typing for data exists. This attempt was aimed at addressing representation using a semi structured framework and using an applied scenario for testing. For these reasons the model does not work, therefore further development and testing is required.

Figure \ Criteria	Breadth	Depth	Alignment	Representation	Reproducibility	Dynamicism	Existing Theory	Effort
Figure 4.52	Moderate	Moderate	True	Moderate	True	Moderate	True	Moderate

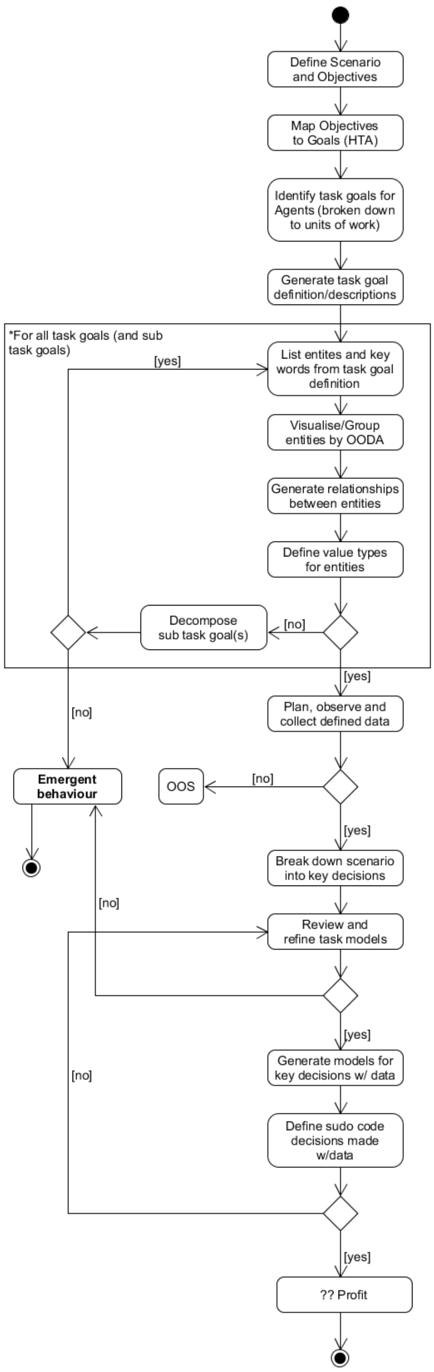


Figure 4.52: Task models based on UML visualisation of OODA elements

Figure 4.52 is the introduction and inclusion of Information Fusion knowledge representation as OODA UML Task models based on M-OODA. This iteration is focused on task descriptions and Coverage - depth terminators based on collected descriptions. Still missing structured outcomes and actionable items. First use case outside of High Yoyo example. Application to test and validate coverage, depth, breadth, alignment, representation, dynamicism, existing theory, reproducibility and effort.

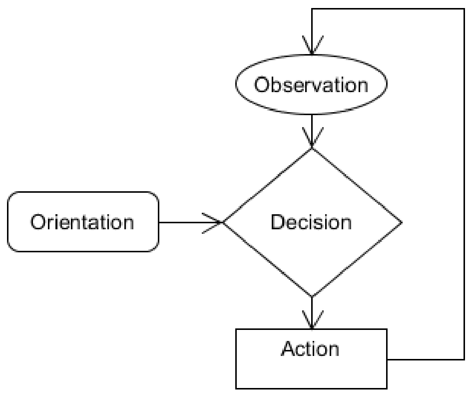


Figure 4.53: Simple Model to represent OODA based on M-OODA and Information Fusion research

Figure 4.53 is a simple UML model, based on the concepts of OODA and the approach used by members of the Canadian consortium within the Information Fusion community to represent OODA as an M-OODA representation. M-OODA aims to capture fine-grained and detailed information to support data flow and transitions, but for the purposes of this research, we use commonly found symbols with UML and Flow Chart UML meta modelling to represent Observations, Decisions, Orientations and Actions. Observations are represented as an oval, Decisions as the commonly used diamond, with branching to indicate decision flow, rounded edged square as orientation entities that feed into and support decisions and rectangles to denote actions.

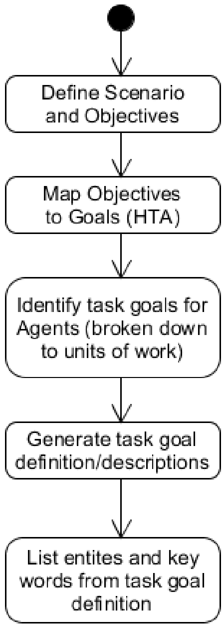


Figure 4.54: Maintaining same alignment criteria through the use of Scenario and Objective Definitions. Major iteration in the generation of task goal descriptions to aid in the extraction of OODA elements

Figure 4.54 contains the same method for Alignment. The differences being a finer controlled and articulated approach to the identification of task goals and the generation of task goal definitions / descriptions. This approach, of task description generation, is to aid with the identification and extraction of OODA entities for further analysis.

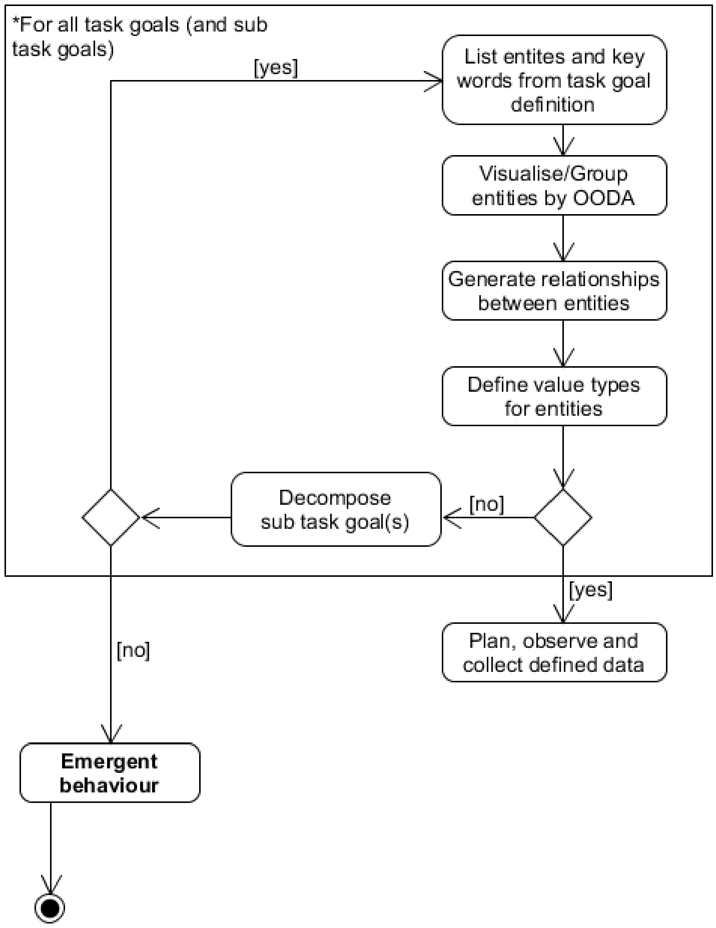


Figure 4.55: Iterative approach with clear termination. Task description leads to task models with relationship. Otherwise, decompose task. This addresses minimum coverage - depth required, without needing an exhaustive approach. Reduces effort required.

Figure 4.55 outlines and details the same approach as presented in the previous iteration of the framework. Therefore, the major changes in this iteration of the framework, are the inclusion of a defined task metamodel and the requirements for the generation of task descriptions.

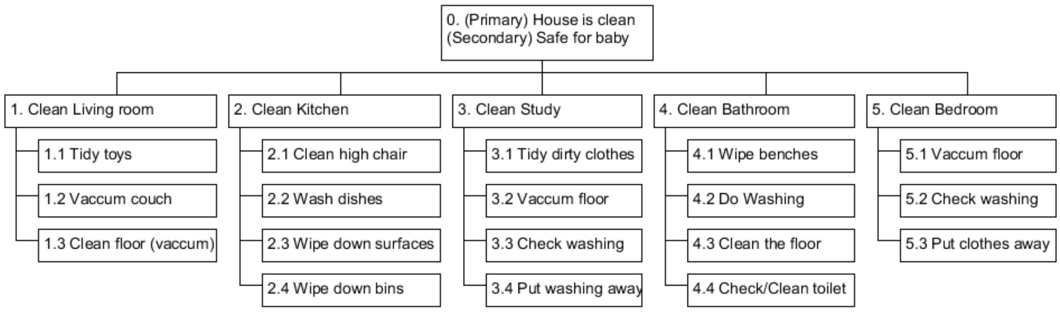


Figure 4.56: Worked example using Figure 4.52 and simple Activities of Daily Living task - Cleaning. Example demonstrates HTA task decomposition to address Coverage Depth and Breadth

Figure 4.56, is a worked example using the changes made, and is focused on the task goal generation and subtask goal identification using HTA methodology. The scenario under investigation has changed into a simpler use case (Clean House) that will allow for rapid development and testing and aims to be easily communicable to a wider variety of audiences as a tangible concept most would be familiar with. As shown within the top most entity (0. House is clean), is the overarching primary objective, with a set (1, 2, 3, 4, 5) as the task goals required to achieve this objective. Each task goal, then contains a set of subtask goals (e.g. 1.1, 1.2, 1.3) required to complete the task goal.

1.1 Tidy toys

Toys are kept in living room in playpen area on top of rug. Primary place of play. Free play is encouraged and toys are often thrown around and not always confined to playpen area

1.Vaccum couch

Play pen area does not exclude couch. Baby has access to couch and can climb up. Likes to sit with family and have stories read. Has snacks with family on couch (and drinks). Couch collects crumbs and dirt from baby. Feet rest of couch. Collects dirt from rest of house

1.3 Clean floor (vaccum)

The living room is where the family spends most of the time it consists of wooden floors and a a rug between the couch and tv Baby spends alot of time on the rug, and on the wooden floors roaming. The family wears slippers around the house and takes them off when entering the area of the rug.

The floor is vaccumed daily to stop the spread of dust and debris being within reach of baby and to stop her from eating it

Figure 4.57: Worked example using Figure 4.52 and simple Activities of Daily Living task - Cleaning. Task descriptions are used to analyse for OODA elements to generate UML task models

Figure 4.57, is a continuation of the worked example and the simpler scenario of Clean House. Presented in this figure are the relevant task descriptions for subtask goals 1.1, 1.2 and 1.3 from Figure 4.56.

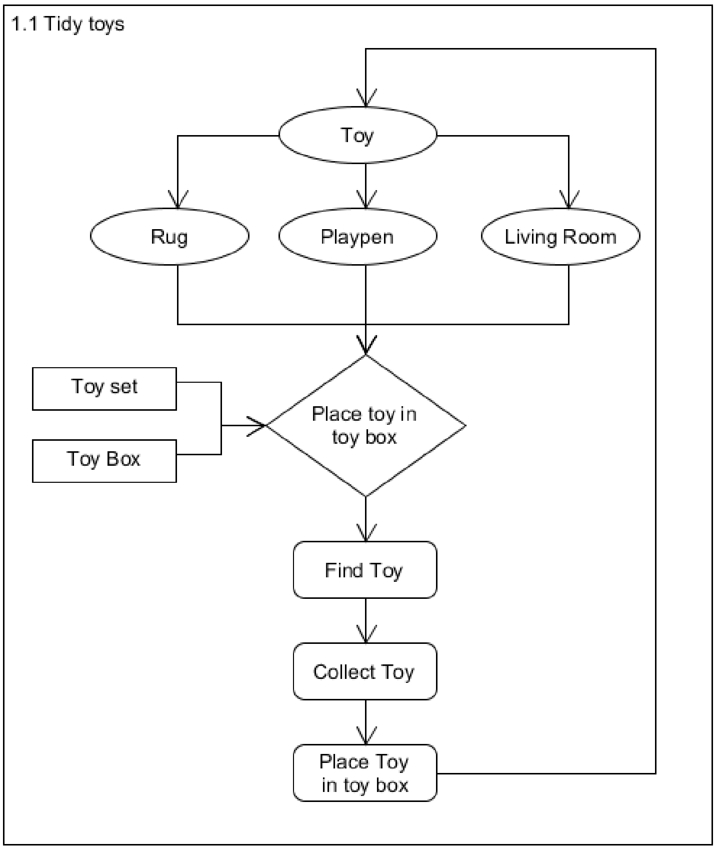


Figure 4.58: Worked example using Figure 4.52 and simple Activities of Daily Living task - Cleaning. Task model demonstrating a simple visualisation of a task.

Figure 4.58, is a continuation of the worked example using the simpler use case of Clean House and the meta task model representation used to generate a task model based on the description presented in Figure 4.57. This is the first major interaction and approach to address and aim to reduce the burden of communication for the criteria of Representation.

Figure 4.52 is still missing further approaches for analysis and recommendation and therefore further development and refinement is required.

Figure \ Criteria	Breadth	Depth	Alignment	Representation	Reproducibility	Dynamicism	Existing Theory	Effort
Figure 4.59	Moderate	Moderate	True	Moderate	True	Moderate	True	Moderate

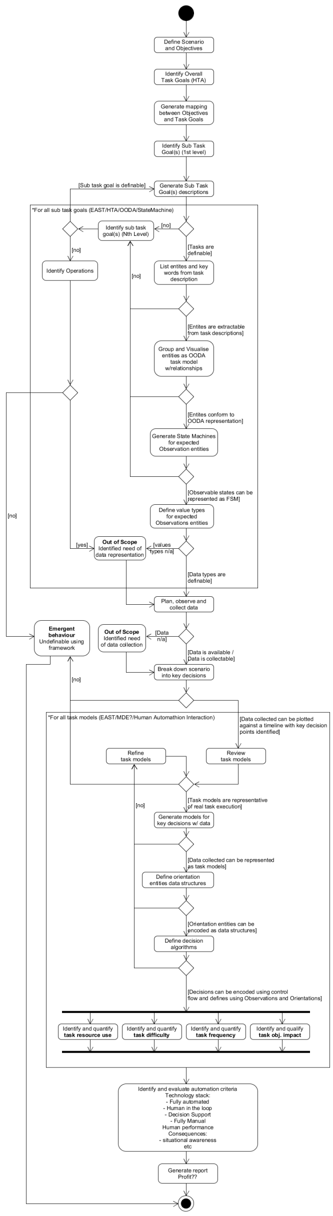


Figure 4.59: Addition of methods for the analysis and evaluation of task models. Clear articulation of data flow

Figure 4.59 is a major iteration that is focusing on analysis and evaluation methods of task models. Clearer definitions used to articulate the flow on information throughout the framework.

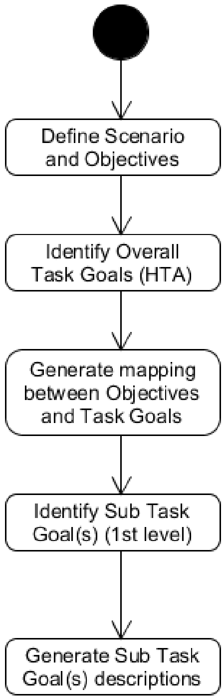


Figure 4.60: Same focus on Alignment, Task goals, objective mapping, and task description generation.

Figure 4.60 is focused on the same alignment criteria, aiming to provide a clear overview between objective and tasks. The differences between this iteration and previous version, is the iterative breakdown approach of definition, identification, mapping and further identification vs definition, mapping, identification. The changes are small, but as part of testing and evaluation the logical changes are required to assist with decomposition

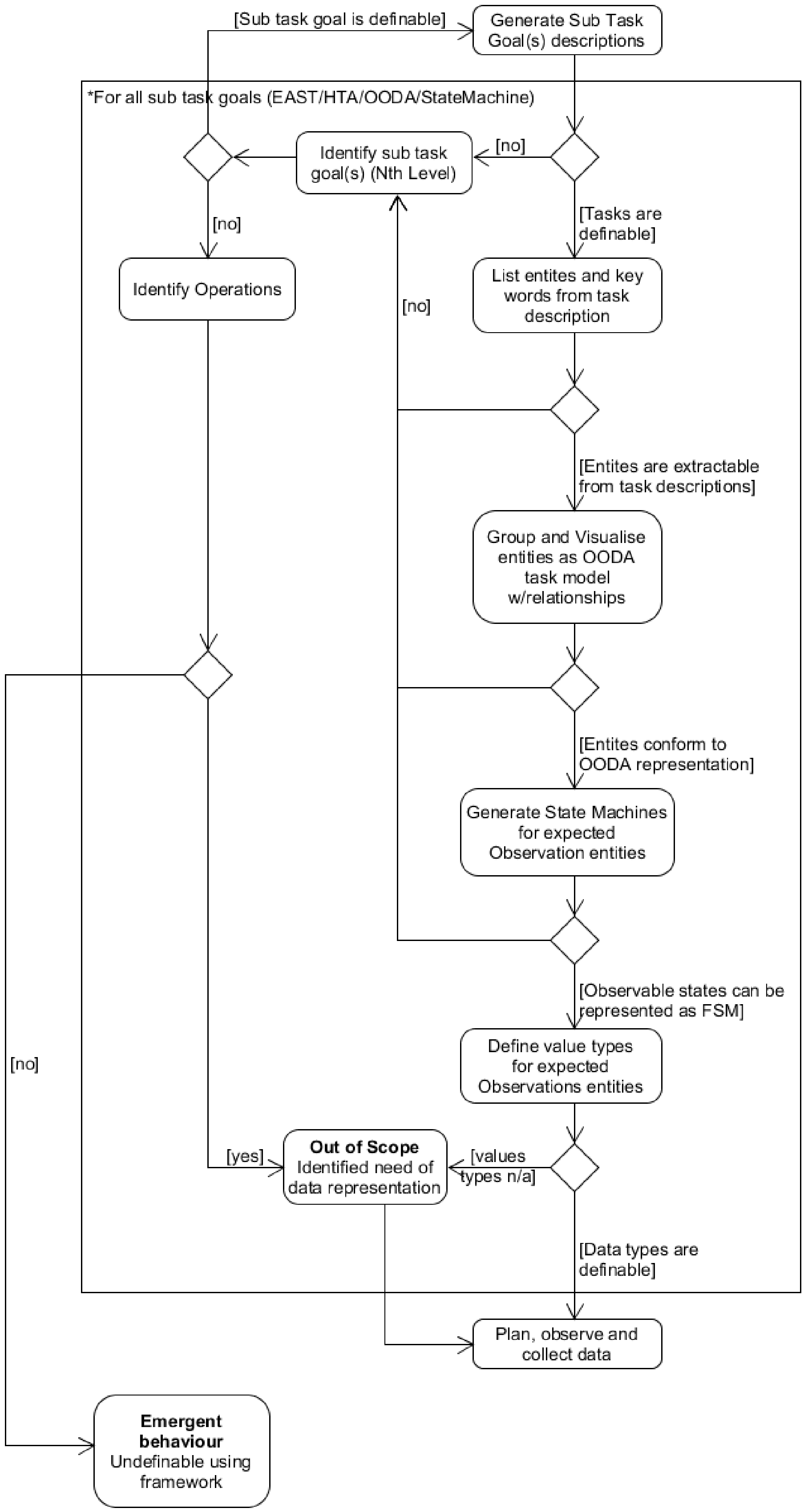


Figure 4.61: Articulated flow of information through framework. Inclusion of state machine generations based on Observable state entities. To be used in analysis and evaluation.

Figure 4.61 is focused on the major descriptive steps of the initial investigation. As opposed to previous iterations, where entities were extracted from definitions, visualised and grouped by OODA and relationships generated BEFORE any decision could be made about further decomposition, this iteration is different. At each step, a branching decision can be made to determine if descriptions can be made. If the task cannot be described, then decompose further. If entities cannot be extracted, then decompose further. If task models cannot be generated, then decompose further. As identified in earlier iterations, where data typing and pseudocode were being generated to help assist with the instantiation of intangible OODA elements, this iteration adds the generation of Finite State Machines for Observation states. This aims to assist with the representation criteria and allows for communication amongst stakeholders. The other changes in this iteration are the inclusion of out of scope terminators for non-definable value types for observation entities and a reclassification for non-decomposable non-deterministic tasks based on the identification of operations within the task.

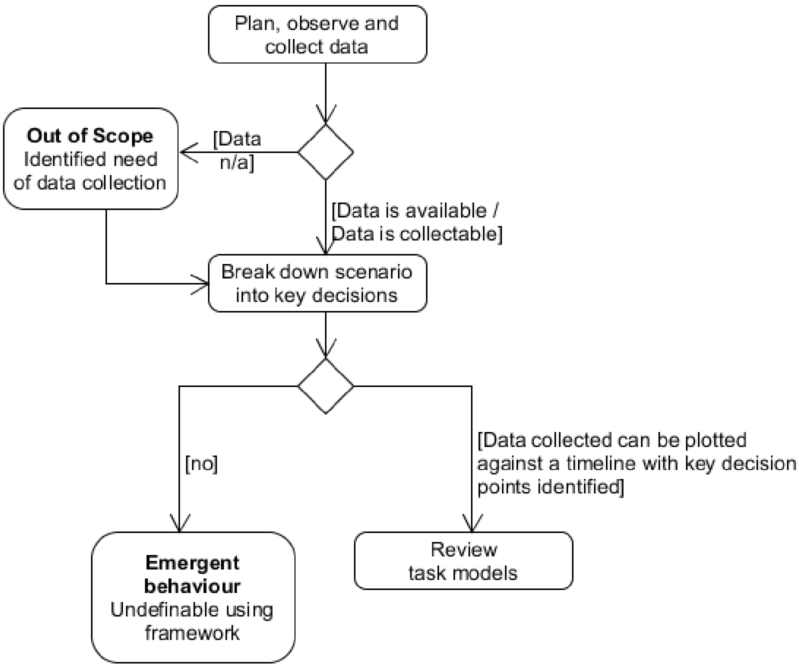


Figure 4.62: Data collection phase. Clearly articulates Out of Scope limitations based on lack of data collection. Clearly articulates Emergent behaviour limitations based on non-deterministic tasks

Figure 4.62 expands upon previous iterations of data collection through the inclusion of a step over, into breakdown of scenarios, from out of scope deliminator for unavailable data, as well as emergent behaviour terminators for scenarios that are unable to be broken down into key decisions. These changes aim to clearly articulate and enhance the step-by-step approach of the framework in its current format.

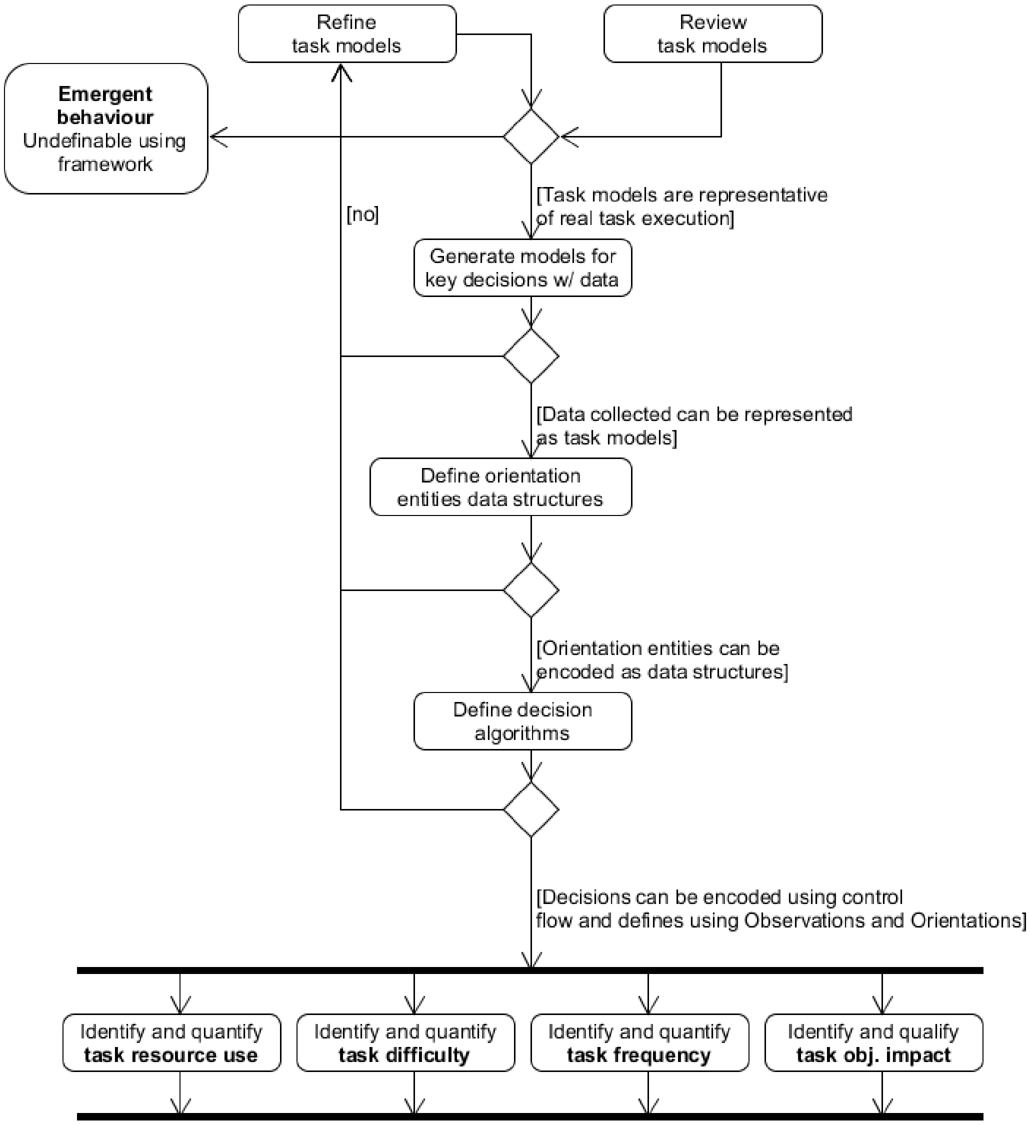


Figure 4.63: Analysis methods based on task models and data collection. Focus on data transformations of OODA elements (Observations as State Machines, Orientation as queryable data structures, Decisions as algorithms.)

Figure 4.63 builds upon the previous iteration and finally aims to address the missing analysis and recommendation components of the framework. Refinement and Review are split into two different components, where each of the following steps after Review, allow for a decision to refine task models based on data collection. Similarly to Figure 4.61, this iteration extends on the notion of instantiation of OODA elements. Observations become finite state machines, Decisions are defined as simple pseudocode algorithms, and finally Orientations are defined as queryable data structures or knowledge bases. The last major component of this figure is the inclusion of a series of analyse methods that focus on task resource use, task difficulty, task frequency and impact on the overarching objective. This initial approach aims to address the goal of the framework to determine where (i.e. what task) multiplication should be applied to.

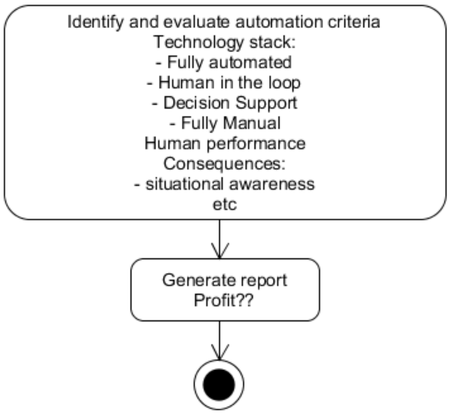


Figure 4.64: After analysis, proposed methods to generate actionable outcomes from framework application.

Figure 4.64 is the continuation of Figure 4.63, in that, where Figure 4.63 provides the method for review and refinement of task models, as well as the instantiation of components and initial attempt at identifying and classifying vital aspects of tasks to aid with multiplication recommended (i.e. analysis of tasks), this figure provides a series of next steps to aid in the recommendation of multiplication considerations. While useful, this initial attempts at both analysis and recommendation are lacking in completeness and through testing, further refinement is necessary.

Criteria / Figure	Breadth	Depth	Alignment	Representation	Reproducibility	Dynamicism	Existing Theory	Effort
Figure 4.65	Low	Low	True	Low	True	Low	True	Moderate

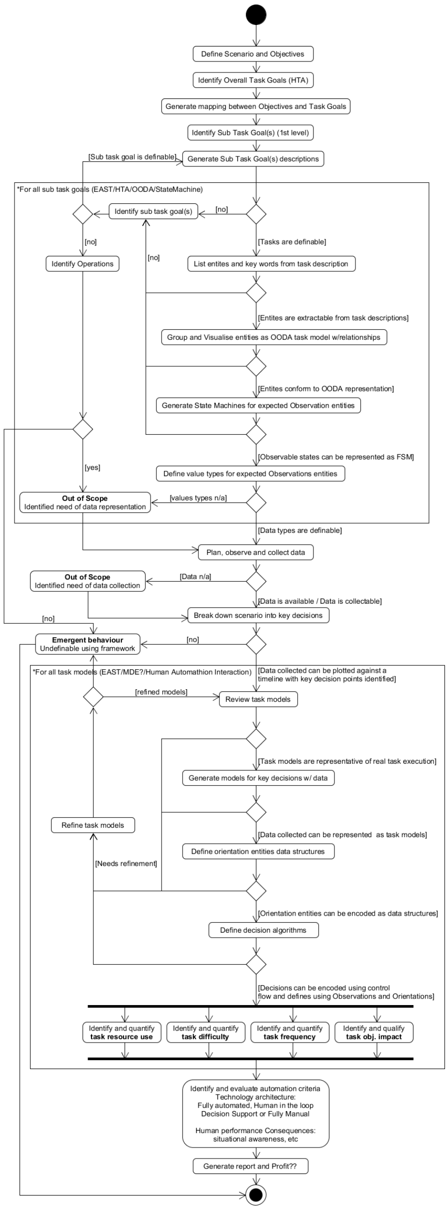


Figure 4.65: Incremental visual tweaks and minor adjustments from Figure 4.59. Slight adjustments to analysis and recommendation methods.

Figure 4.65 includes minor visual tweaks and adjustments. Inclusion of task description method used to generate task descriptions as part of Data collection. Clear segmentation into phases (Figure 4.70). This is a major iterative version of DCARR - the Describe Collect Analyse Recommend and Report Framework.

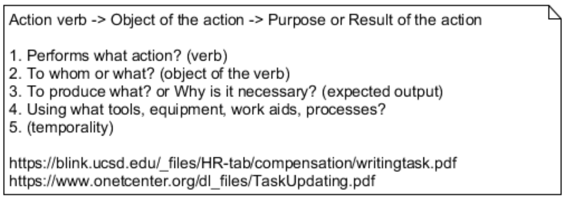


Figure 4.66: Method used for the generation of task descriptions. Simple questions to capture OODA elements.

Figure 4.66 is the simple set of questions to be used to generate task descriptions - and to decide if a task is non-deterministic and requires further partial decomposition, as per HTA methodology. The task description method itself aims to guide the user to identify and classify, in plain English, OODA components. For example, action verb equates directly to the Action entity of OODA. The object of the verb equates to the observation entity (i.e. data transformation / functional process - where data is changed or i.e. procedural process where observations trigger an event to occur on/in a different area/task). As part of the task description it is also necessary to capture the expected outcomes/outputs to confirm that tasks behave as expected, as well as temporality of the task to understand where in the timeline of tasks this expected task is likely to occur, or on which preceding task this task will occur after. Finally, the task description aims to capture tooling and equipment usage to understand Orientations (i.e. processes or training, usability and interfaces of software, affordances of objects, etc.).

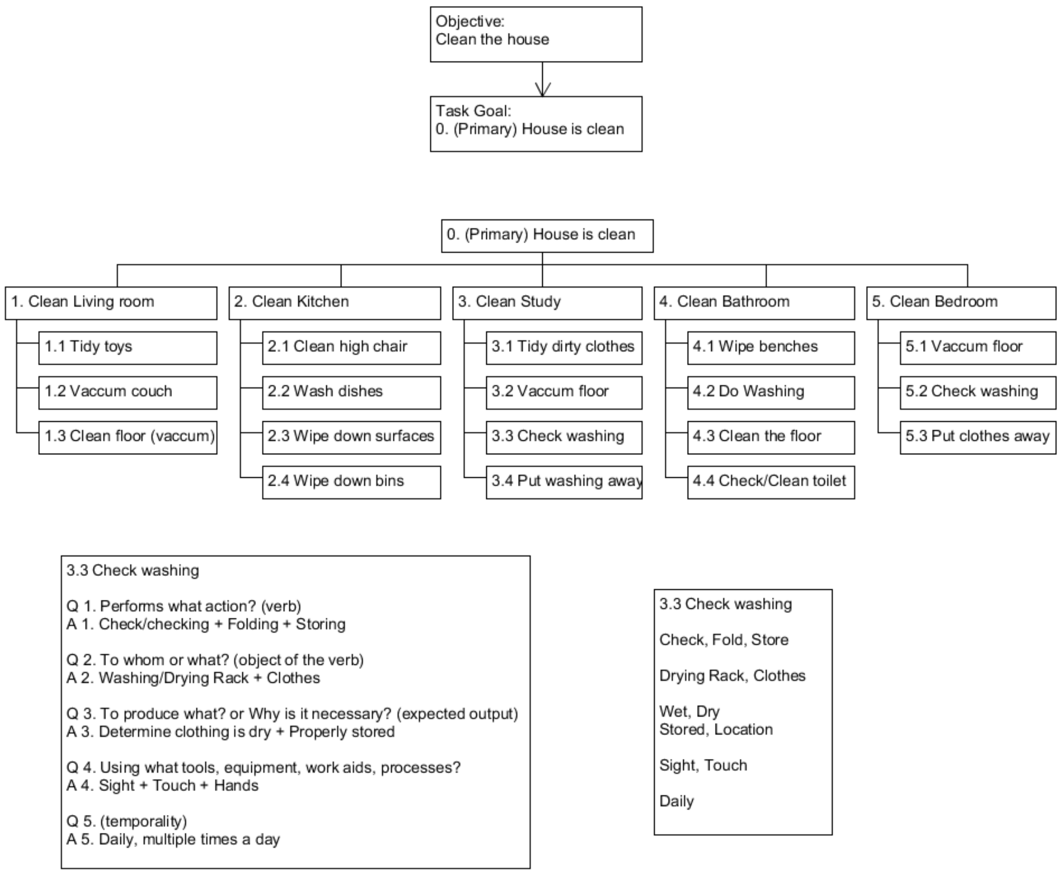


Figure 4.67: Worked example using Figure 4.65 and simple Activities of Daily Living task - Cooking. Mapping between objective and task goal. HTA breakdown to address Coverage - Breadth and identify appropriate tasks. Simple task description method and entity extraction.

Figure 4.67, is a worked example using the framework. This example is using the simpler use case of Clean House. The differences being the explicit mapping between Objective (Clean the house) and Task Goal (House is clean). The HTA breakdown of Task goal, into subtask goals is the same as presented earlier, except this time we also see the inclusion of the use of the task description method to generate a simple task description with included OODA elements extracted.

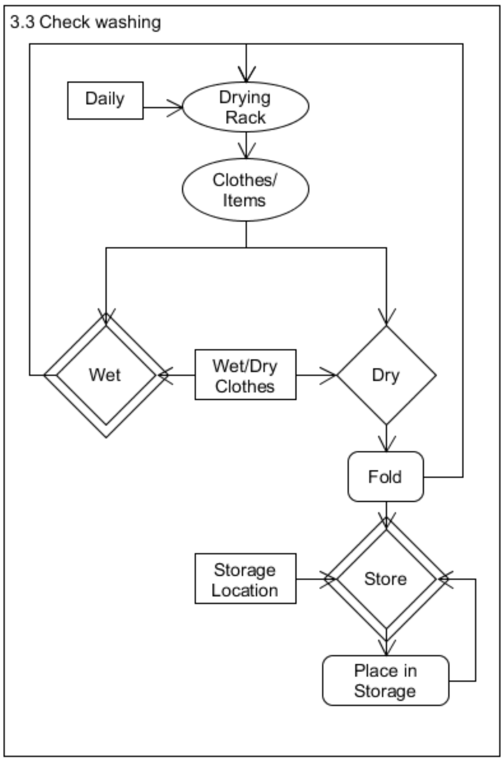


Figure 4.68: Worked example using Figure 4.65 and simple Activities of Daily Living task - Cooking. Task model representing extracted entities from task description (Figure 4.67)

Figure 4.68, is a worked example using the framework and the simple use case of Clean House - focused on the subtask goal of Check Washing. This figure uses the OODA task metamodel as a basis to generate a task model using OODA elements extracted from the task description. The double decision diamonds refer to termination points within the decisions.

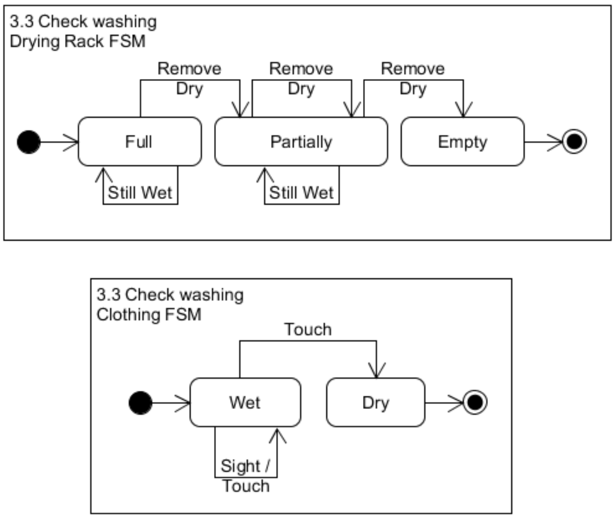


Figure 4.69: Worked example using Figure 4.65 and simple Activities of Daily Living task - Cooking. State machines of observable states extracted from task descriptions (Figure 4.67)

Figure 4.69, is a worked example using the framework and the simple use case of Clean House - focused on the generation of finite state machines to represent the observable states defined as part of the object of the verb from the task description method. The benefit of using FSM to represent observations is it allows the full deterministic gamut of use cases to be defined, in preparation for multiplication implementation. As part of the FSM we can also see the tooling, processes, equipment and work aids as defined in the task description used to interact with the observable state. These act as orientations, with which we can filter observations through, much like traditional OODA loop representations, to inform decisions and define action selection outcomes.

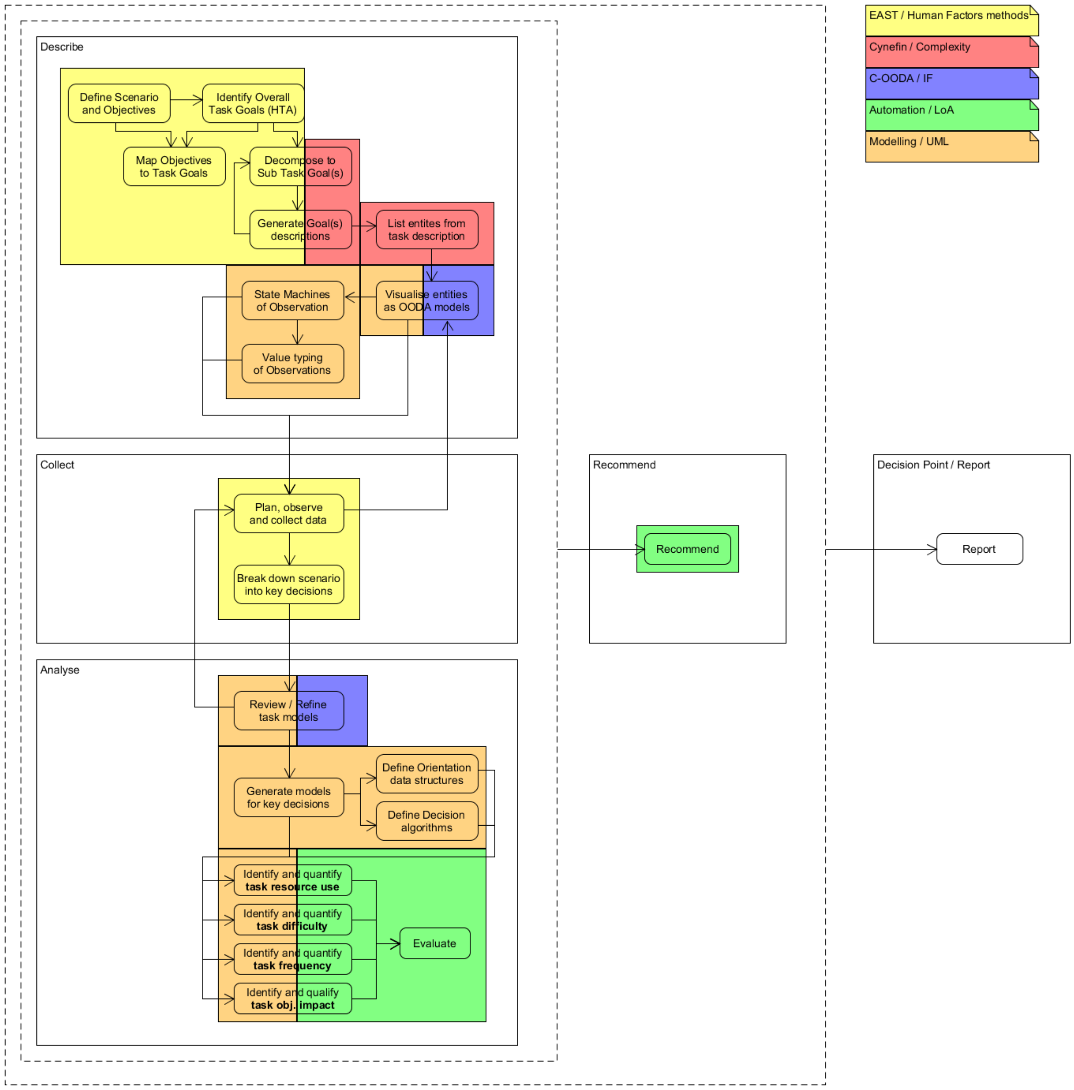


Figure 4.70: Decomposition of DCARR Framework into clear phases of Describe, Collect, Analyse, Recommend and Report. Relationship between phases allows for forwards and backwards traversal. Clearly defined research areas that have shaped and informed development and evolution.

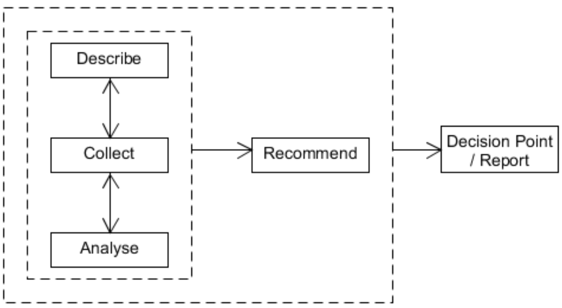


Figure 4.71: Simplified Phase Diagram of DCARR - Complementary to Figure 4.70

Due to the major iteration and development of the framework, in reaching a clearly defined and appropriate level of Description efficacy (To address Coverage, Alignment, Representation, Reproducibility, Dynamicism and Existing Theory), Figure 4.70, is the decomposition of the framework presented in Figure 4.65, into clearly designated and segmented components of work which form the 5 phases of DCARR - Describe phase, Collect phase, Analyse phase, Recommend phase and Report Phase. This figure is also colour coded to present and highlight the use of existing theories throughout the development and evolution of DCARR. Figure 4.71, is a continuation of the presentation of DCARR from Figure 4.70, through the high level overview of the 5 phases of DCARR and the relationship between them.

Figure \ Criteria	Breadth	Depth	Alignment	Representation	Reproducibility	Dynamicism	Existing Theory	Effort
Figure 4.72	Low	Low	True	Low	True	Low	True	Low

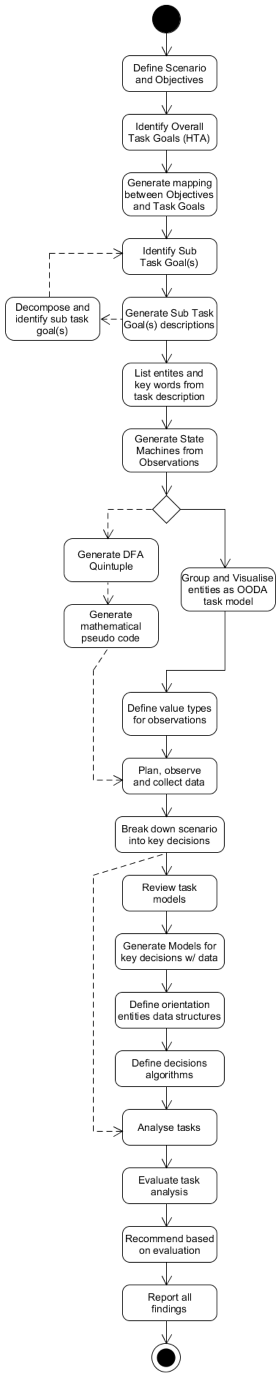


Figure 4.72: Major iteration aimed at refinement and presentation of DCARR, from Figure 4.65, into an easy-to-follow step by step method

Figure 4.72 is another major iteration aimed at refinement. Addresses all aspects of criteria. Coverage - Breadth and Depth is terminated by deterministic task descriptions and HTA method. Alignment is present with the mapping from Task goal and Objective and HTA method. Representation is low effort due to the simple 4 element diagram of knowledge representation in the form of OODA task models. Reproducibility is true due to structured breakdown and sequential steps. Dynamicism is low effort due to no reference or requirement for domain specific problems; can be applied to any process that contains human involvement and multiple tasks; is based upon multiple areas of research and theory from UML, Information Fusion to Human Factors and Human Automation Interaction. Effort required is low, due to the ability to terminate at different phases and extract meaningful insights to generate actionable outcomes. Allows for full application or partial application given cost/effort. Refer to Section 4.2 for a full step by step breakdown of the complete method and approach.

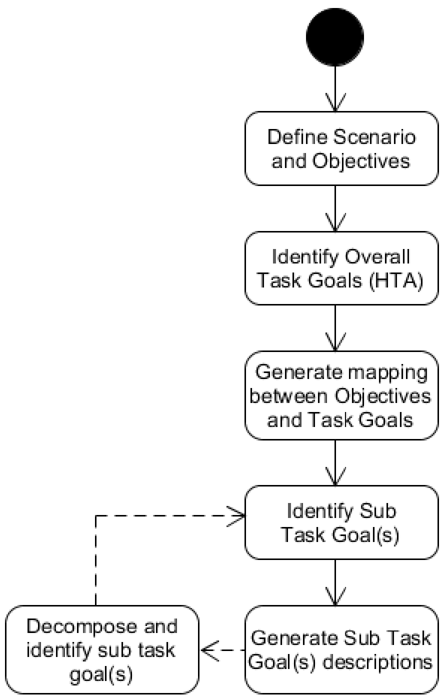


Figure 4.73: Maintains Alignment and Coverage - Breadth and Depth through the use of deterministic task descriptions and task decomposition from HTA

Figure 4.73 and includes a refined decomposition loop. This figure forms the tasks of the first part of *Describe* phase.

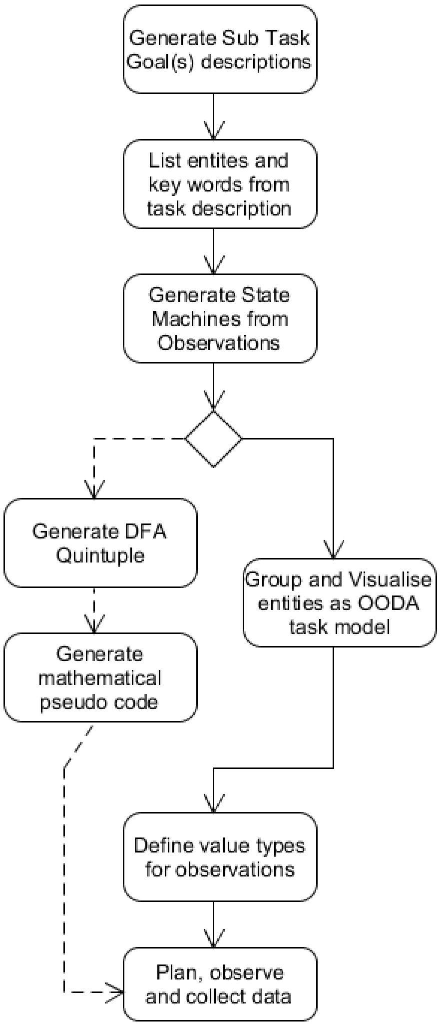


Figure 4.74: Maintains Representation as it separates alignment from task representation. UML Task Models, Deterministic Finite Automata, Pseudo Code and State machines are different representations of the same information to appeal to a broader audience to allow for discussion and communication.

Figure 4.74 drastically and dramatically refines and reduces the cognitive overhead of processing and understanding the steps of DCARR. Rather than, as per each step in the framework, identify sub-task goals / decide if further decomposition is required, the method presented, is laid out as a step by step, with a branching choice depending on audience communication requirements. While this streamlines presentation and communication of DCARR it also reduces the communication of termination points and the flexibility of DCARRs application. Forms the second part of *Describe* phase

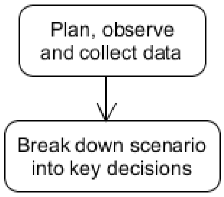


Figure 4.75: Same Collection phase methods as previous iterations. Once identified areas as part of Describe phase, plan for and gain access to collect data to support described models

Figure 4.75 is the same step-by-step approach, but reduced in communication complexity. While easier to comprehend and communicate, it is lacking in communication of out of scope and emergent behaviour termination points of previous iterations. Also, there is reduced task branching, which has been removed for brevity and clarity. This figure forms the tasks within the *Collect* phase.

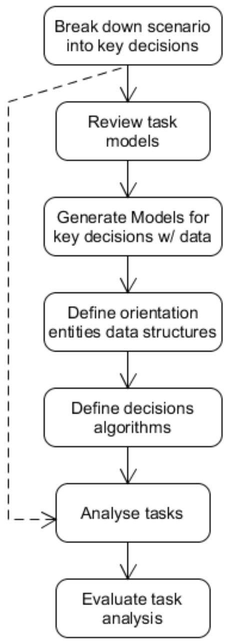


Figure 4.76: Analyse phase includes generation of artefacts using collected data, simple algorithms and data structures that can be queried to replicate or train future systems. Analysis and Evaluation of analysis steps provide full and complete step by step breakdowns based on goals and desired outcomes

Figure 4.76 is reduced for brevity and communication complexity, but lacks termination/out of scope deliminator and task branching. This figure forms the tasks of the *Analyse* phase.

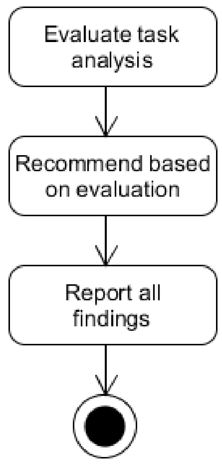


Figure 4.77: Recommend and Report phases. Based on desired outcomes / data collected and task descriptions. End to end termination point for full application of DCARR Framework

Figure 4.77 is reduced for brevity and communication complexity, but lacks termination/out of scope deliminator and task branching. This figure forms the tasks of the *Recommend* and *Report* phases.

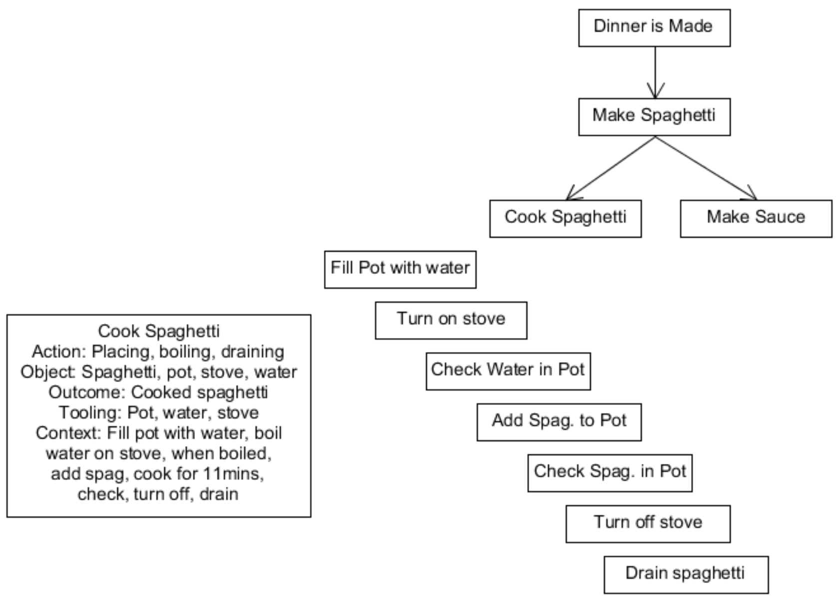


Figure 4.78: Worked example using Figure 4.72 and simple Activities of Daily Living task - Cooking. Mapping between Objective, Goal and Sub task goal. Breakdown of steps and descriptions

Figure 4.78, is a worked example using the framework and the scenario of Cooking. This figure presents the scenario (Make Spaghetti) and objective (Dinner is made) definition, task goal mapping and sub-task goal identification (Cook Spaghetti and Make Sauce). Also included is the Sub Task Goal description for Cook Spaghetti with defined Actions, objects, outcomes, tooling and context.

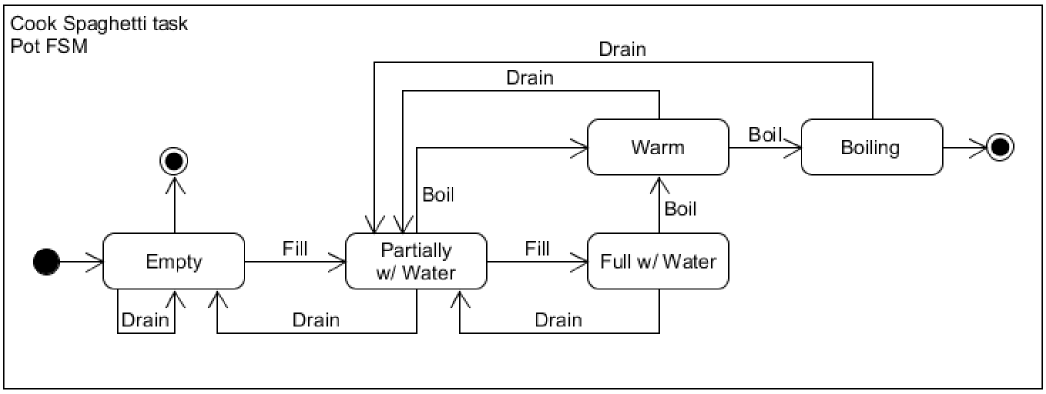


Figure 4.79: Worked example using Figure 4.72 and simple Activities of Daily Living task - Cooking. Finite State Machine of observable entity as part of the task.

Figure 4.79, is a worked example using the framework and the scenario of Cooking. This figure is the presentation of the Observable state of the Pot object under the Cook Spaghetti task, as a Finite State Machine. The benefit is the communication to stakeholders and variable states of observation required for data collection and future analysis and evaluation

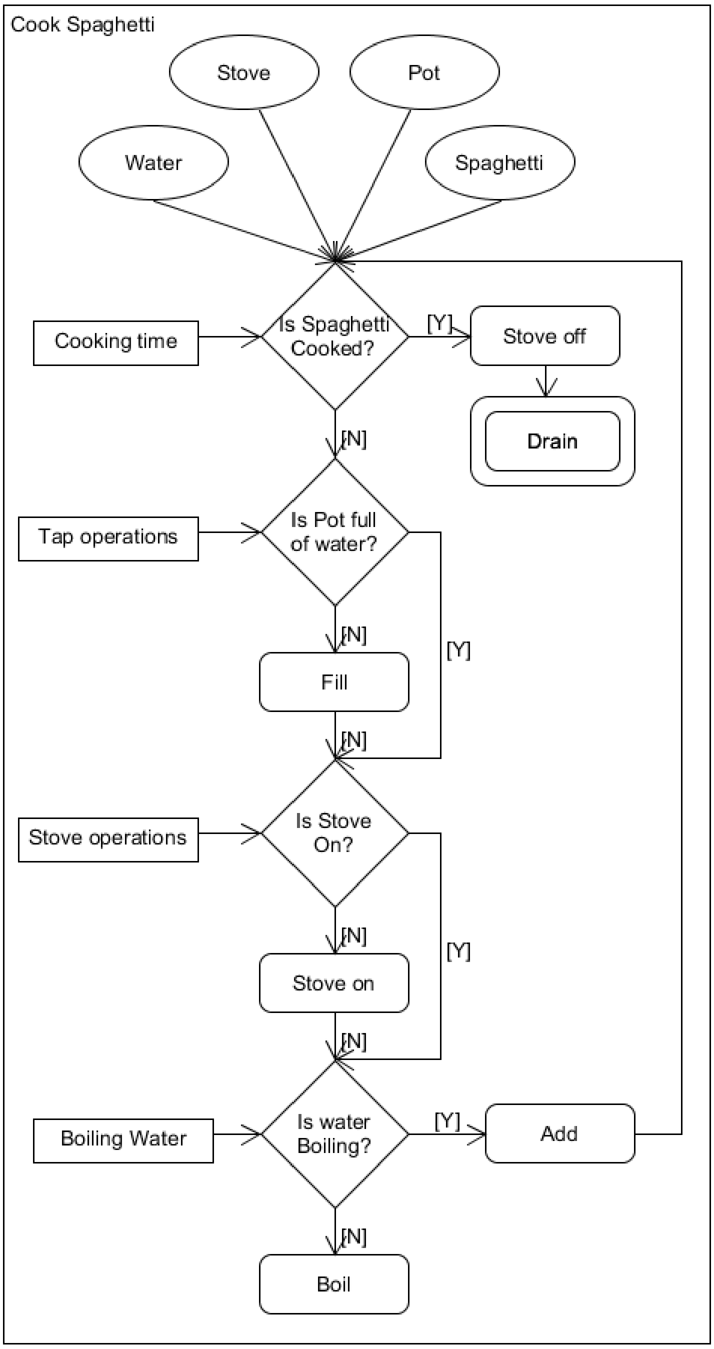


Figure 4.80: Worked example using Figure 4.72 and simple Activities of Daily Living task - Cooking. Described task model based on description and entity extraction.

Figure 4.80, is a worked example using the framework and the scenario of Cooking. This figure is the estimated task model based on the task description for Cook Spaghetti. Multiple states are observed as well as multiple decisions made.

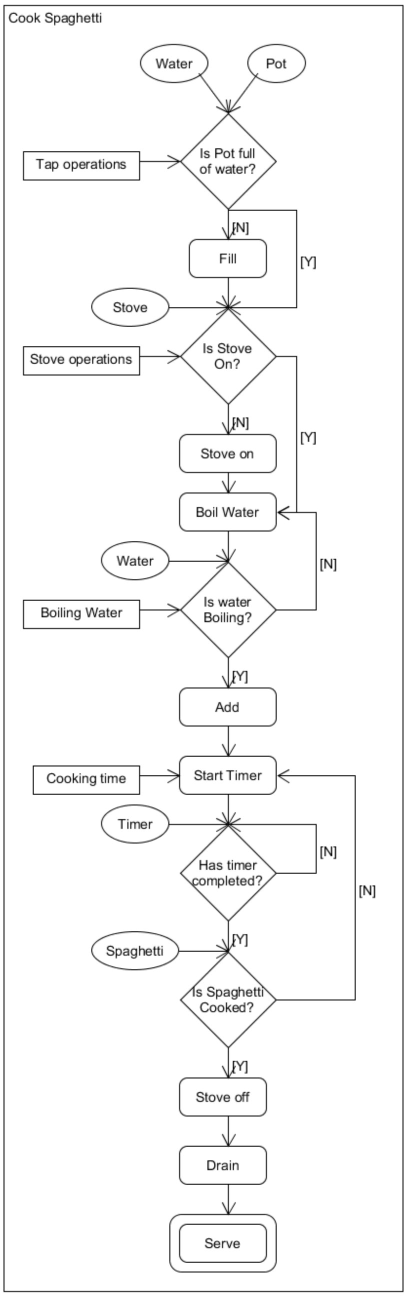


Figure 4.81: Worked example using Figure 4.72 and simple Activities of Daily Living task - Cooking. Reviewed and expanded task model based on collected data.

Figure 4.81, is a worked example using the framework and the scenario of Cooking, and is a continuation Figure 4.80. This figure is the amended "true" task model based on data collected and real world observations.

Criteria / Figure	Breadth	Depth	Alignment	Representation	Reproducibility	Dynamicism	Existing Theory	Effort
Figure 4.82	Low	Low	True	Low	True	Low	True	Low

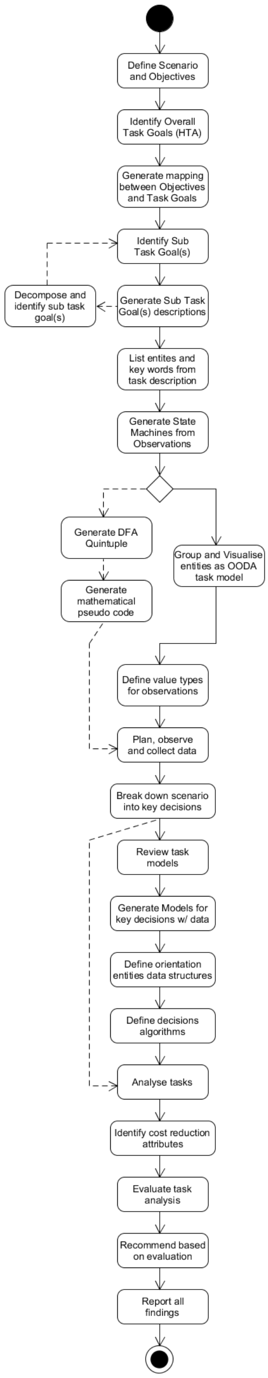


Figure 4.82: Inclusion of Identification of Cost Reduction Attributes step.

Figure 4.82 is a minor amendment due to the inclusion of Identification of Cost Reduction Attributes task. This version is the final DCARR version presented in Section 4.2 and to be applied in Chapter 5. Figure 4.83 builds upon the previous iteration with the single inclusion of an Identify Cost Reduction Attribute task. Figure 4.84 is the phase breakdown of Figure 4.82.

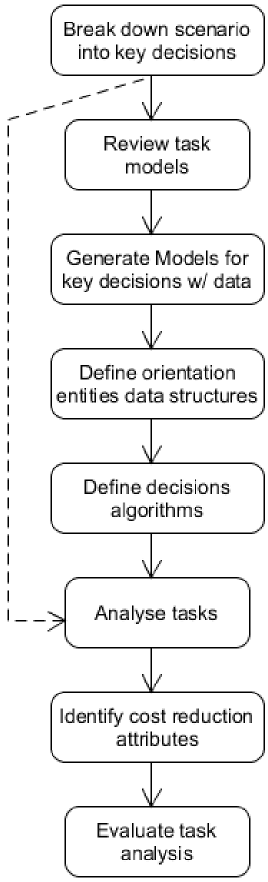


Figure 4.83: Addition of Identification of cost reduction attributes after Analyse Tasks and before Evaluate task analysis.

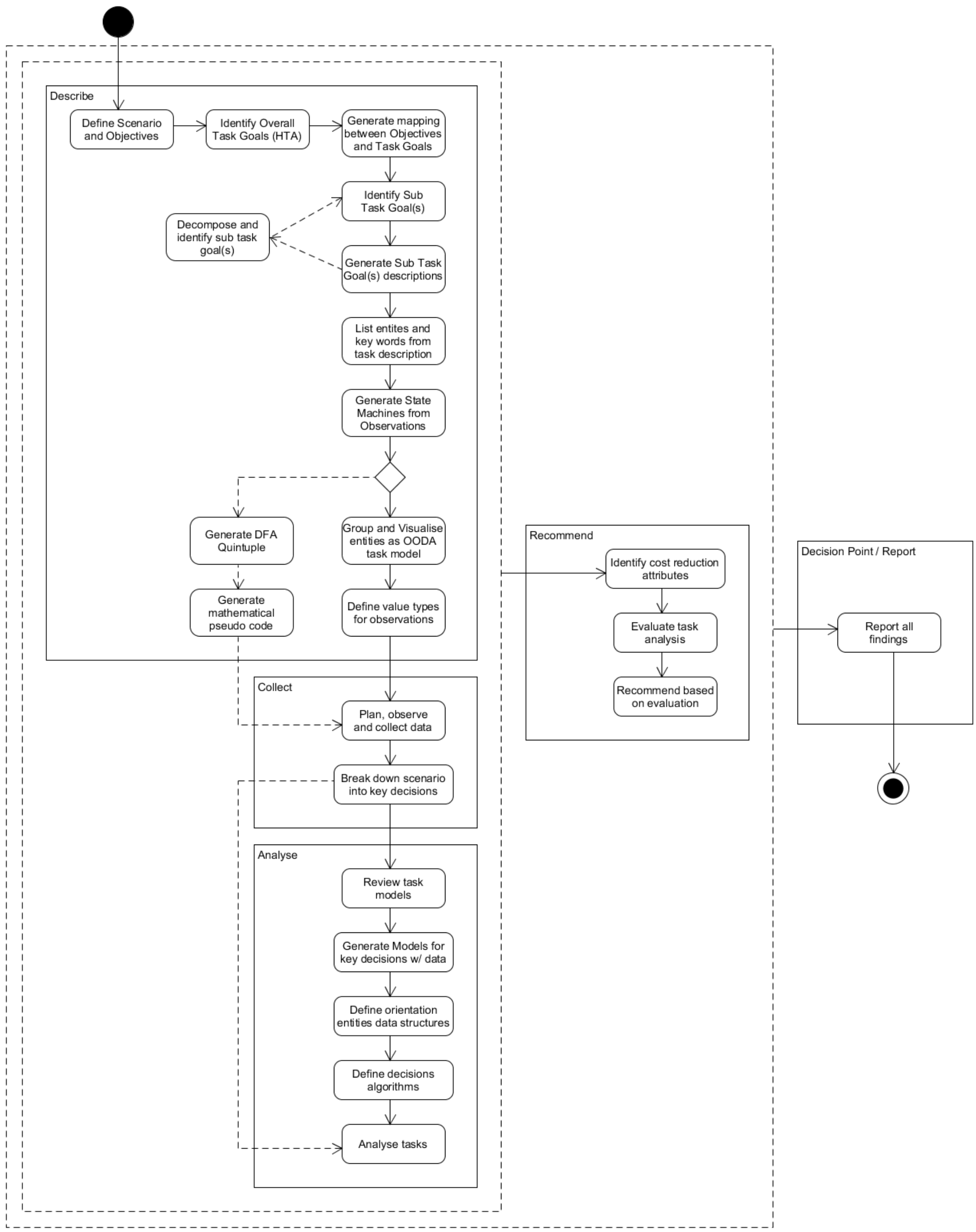


Figure 4.84: Based on Figure 4.82. DCARR Framework separated into Phase Diagrams to highlight the individual steps and processes within each phase.

4.4 Chapter Summary

This chapter presented the Describe, Collect, Analyse, Recommend, and Report (DCARR) framework, developed for the identification, communication, representation, classification and automation ranking of candidate tasks and processes/routines within the domain of White Force facilitated distributed synthetic training, to address the proposed hypothesis of *"How can we multiply White Force capabilities with automation?"*

This chapter presented a high level overview of DCARR, the relationship between the phases of DCARR, and the relationship of DCARR to existing theories and proposed research questions. This chapter also highlighted and presented the major contribution of this research; the complete step-by-step breakdown of each task of DCARR, including worked examples and demonstrations.

Finally, this chapter ended with the explored evolution and development of DCARR, from initial metamodel to final framework. This development, evolution and testing, used the proposed contributions of the Method for Development and Evolution, and Method for Testing and Validation. As part of the development was the application of the proposed contributions of the Criteria and Metrics used for the Method for Development and Evolution and Method for Testing and Validation.

In the following chapter, we aim to present the Method for Validation of DCARR, with members of DST Group's AOSC. This validation will take the form of the summary outcomes from the presentation and communication of DCARR, the motivation behind development, the research questions that have driven the underlying work, the existing theories DCARR is founded upon, and worked examples and scenarios to highlight the benefit and application of DCARR to the domain of White Forces facilitation.

Chapter 5

Framework Validation

This chapter describes the validation of the Describe, Collect, Analyse, Recommend, and Report (DCARR) framework to members of Air Operations Simulation Centre (AOSC) and Human Factors Research at Defence Science and Technology Group, Fisherman's Bend. Included in this chapter is the validation presentation, demonstration material and work shopped example scenarios, as well as the communicated contributions, recommendations and limitations of DCARR, as perceived and communicated by participants, based on their observation of the presentational material. Finally, the chapter finishes with an amendment to the presentation of DCARR to address one of the proposed recommendations made by participants during validation.

5.1 Validation to AOSC

As stated in the Research Methodology (Section 3.3) once the framework had met the measures and criteria (Section 3.3.4 and Section 3.3.6), the final step in the method was to validate the framework with members of AOSC. This final step was an opportunity to determine whether the framework can be communicated adequately to subject-matter experts and members of White Forces, who are the initial drivers and motivation behind the development and requirement of a model driven framework to identify White Force multiplication opportunities.

Due to circumstances beyond the researchers control (COVID-19) and work from home orders, the validation took place as an online interactive presentation/workshop, held over Microsoft Teams on November 26th 2020. The event took place over 3 hours and required that the framework be described, with a simple example to highlight key concepts and ideas. Once an initial understanding was achieved, there was an opportunity for a question and answer session, before moving onto a second more complicated scenario that had been provided by one of the attendees during the early stages of the framework development. Finally, the event finished with a round table discussion and workshop where participants were asked to provide insights and knowledge into the development of further examples to test the applicability of the framework to domain specific activities, of the type that White Force would perform.

Participants of the presentation included the researcher (Simon Vajda), as well as the researchers supervisory team and members of AOSC.

- Mr Graeme Simpkin, Head Air Operations Simulation Group, Air Division, Defence Science and Technology Group
- Dr Christopher Best, Head Human Factors Group, Air Division, Defence Science and Technology Group
- Mr Peter Ross, Research Scientist, Air Division, Defence Science and Technology Group
- Prof Kon Mouzakis, Alfred Deakin Professor and Co-Director, A2I2, Deakin University
- Dr Simon Parker, Head of Operations, A2I2, Deakin University
- Dr Antonio Giardina, Deputy Head, Translational Research and Commercialisation, A2I2, Deakin University
- Dr Leonard Hoon, Deputy Head, Translational Research and Commercialisation, A2I2, Deakin University

5.2 Presentation

The aim of the presentation was to provide the motivation and research questions / hypothesis that drove the development, the related work / background literature and an overview of the DCARR framework. This was followed with two demonstrated examples; a simple example of cooking dinner, to provide a basic example for familiarisation and then a more complicated example, a refuelling scenario, similar to the types of tasks that would appear as part of White Force activities. Finally, the event was concluded with a workshop where participants were asked to contribute knowledge and expertise during further development of a worked example to provide an opportunity to experience and test the framework. This allowed for further discussion and insights from participants.

For the complete presentation, demo material and work shopped scenario, please see appendices listed below. Otherwise, a short summary and overview of the material follows.

- Full Presentation material — Appendix D
- Demo Example 1: Cooking — Appendix E
- Demo Example 2 / Work shopped Scenario: Refuelling — Appendix F

Figure 5.1: Presentation title slide

5.2.1 Motivation, Background and Highlights

- **Presentation purpose**: The purpose of the presentation was to present the developed DCARR framework to members of AOSC and Subject-Matter Experts familiar with White Force tasks and process. This was done to evaluate and validate that the developed framework was usable. This was done through the use of a presentation and interactive workshop session. At any point questions were able to be asked.

- **Framework Goal, Outcome and Process**: DCARR is a framework to identify opportunities (in the form of candidate tasks and processes/routines ranked for automation) in large complex workflows, such as distributed synthetic Live, Virtual and Constructive (LVC) training. These automation opportunities are presented as a report containing ranked and sorted lists of tasks that are identified as suitable for automation based on the identification, communication, representation, classification and automation ranking undertaken from the steps of applying DCARR.

- **Motivation**: Based on the requirements and observations that White Force is required for successful training outcomes, and that it is not always possible to maintain a large force due to expense (cost, training, time, etc), therefore a need is required for technology to enhance or multiply capability. This becomes our hypothesis, on *"How can we multiply White Force capabilities with automation?"*. White Force exists in an inherently complex and chaotic environment of distributed synthetic training (i.e. the domain). This is due to the skills, knowledge and expertise required; the often inconsistent and fault intolerant execution; and the need to inject complex/chaotic elements to create unknown and unexpected scenarios that the training audience need to adept and overcome. To effectively apply automation, White Force tasks and processes/routines require a level of stability. To apply this level of stability, it is necessary to identify, communicate, represent, classify and rank for automation, candidate tasks and processes/routines.

- **Research Questions**: Based on this motivation, the following research questions were addressed:

  1. **How can we systematically identify White Force tasks and processes/routines, including those that are hidden or ad-hoc, for further communication, representation, classification and investigation?**

  2. **What approach can be used to adequately communicate White Force tasks and processes/routines to a wide and varied audience of automation implementors, to facilitate discussion and review for further task representation?**

  3. **What method can be used to suitably capture and represent, in a structured and standardised way, the variables (i.e. information and decision-making) of White Force tasks and processes/routines, to assist with classification of tasks into suitable categories of automation?**

  4. **How can White Force tasks and processes/routines be acceptably classified into cognitively demanding tasks / tasks that resist automation due to instability and are chaotic/complex in nature, versus those that are repetitive, structured and rule based?**

  5. **What selection criteria (e.g. repetitive, rule based, structured, consistent, and does not require a decision, interpretation or assessment of results) can be used to fairly rank and order candidates tasks for automation application?**

  These research questions form the basis for the development of the DCARR framework.

- **Areas of interest and related work**: The proposed framework is based around a number of areas of related work, based on similar domains, applications and concepts (Figure 5.4). Human Factors research provides methods for data collection/task identification and similar domain applications (Event Analysis of Systemic Teamwork (EAST), research work conducted by Human Factors Integration Defence Technology Centre (HFI DTC) that focuses on the analysis of collaborative performance). Cynefin, provides complexity definitions and methods for task classification, examples and concepts of deterministic / nondeterministic tasking. Information Fusion for task representation and breakdown of simple information processing and use of Observe-Orient-Decide-Act loops. UML for tooling and approaches used for task communication and handling abstractions and visualisations. Human Automation Interaction research, for selection criteria for candidate task and process/routine automation, with the focus on human centric automation consideration, treating the human operator as decision maker, and augmenting human decision-making to assist with emergent and complex scenarios and processes.

- **Benefits and Limitations**
  Benefits of DCARR include:

  - Knowledge capture and discussions with stakeholders and experts
  - Driven by the overarching goal of identifying automation opportunities/ranked and ordered candidate tasks and processes/routines for automation
  - Allows for partial application. (E.g. Describe phase can be conducted independent of Collect or Analyse phase to produce beneficial discussion amongst stakeholders. Similarly, Collect phase can be done independent of Describe phase, if an exercise or data collection opportunity presents itself. Also allows for retrospective application.)
  - Allows for visual representations to communicate with a wide range of audiences and is simple enough to be readily usable
  - Forms the foundations for future work, either iterative applications, understanding and better communicating ad hoc processes or for designing automation capabilities to address White Force shortages
  - Reproducible following a simple methodology that's backed up from relevant literature and areas of interest

  Limitations of DCARR include:

  - Expertise and subject intuition - limited by the availability, access and communication with SME's for task definitions/operations and understanding. (Need to understand what is inside peoples heads)
  - Access to participants, experts, White Forces, observations and data collection

- Time intensive during initial application.
- Can also be exhaustive depending on how far/how much detail is required
- Needs data collection application for quantitative measures, analysis and evaluation
- Identifies areas of complexity, nondeterminism and emergent behaviour. Initially proposed as out of scope for investigation and analysis.

5.2.2 Framework Overview and Breakdown

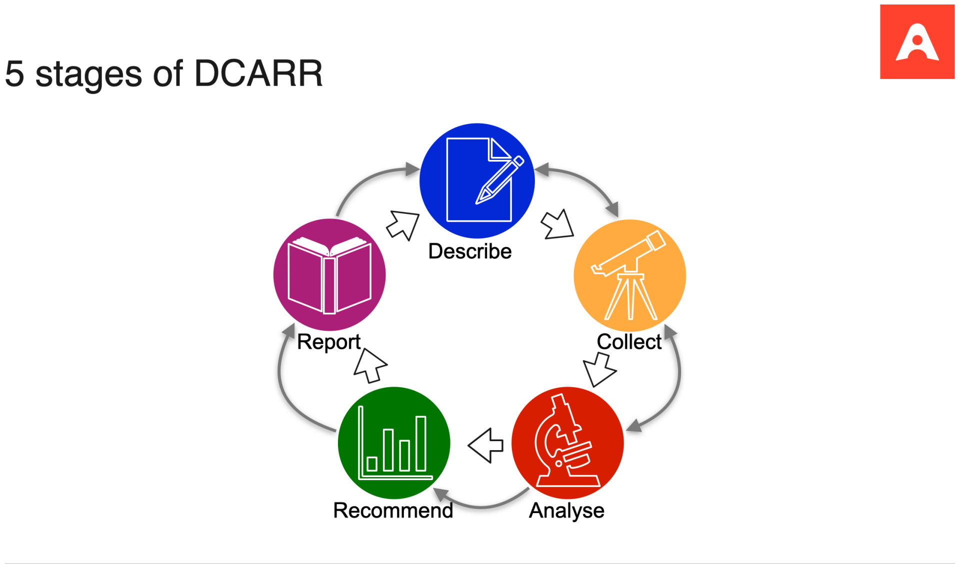


Figure 5.2: Graphical representation of DCARR phases

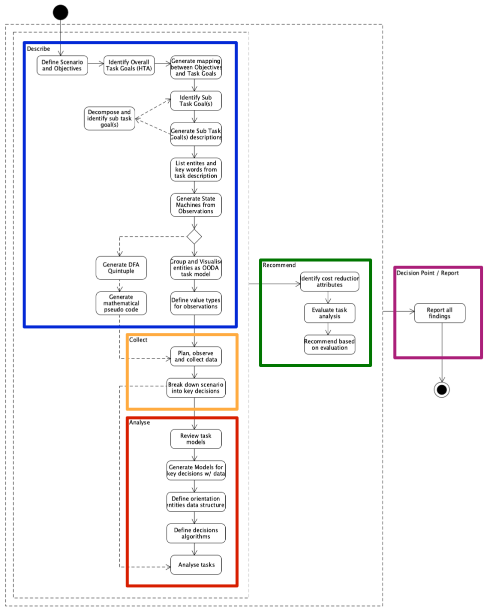


Figure 5.3: DCARR framework and phase highlight

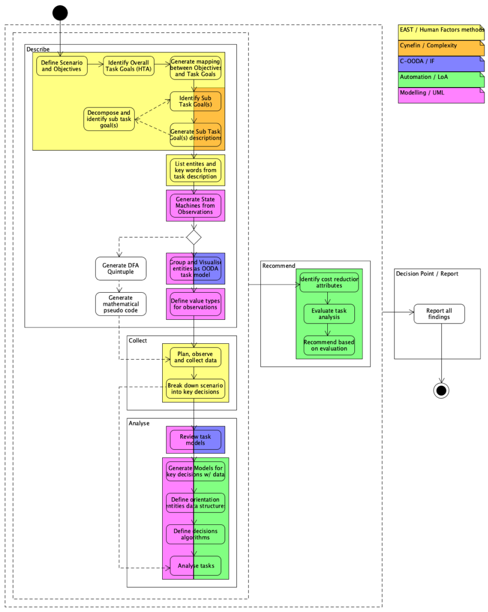


Figure 5.4: DCARR framework and related work highlight

- **DCARR**: DCARR — Describe, Collect, Analyse, Recommend and Report. Split into 5 phases (Figure 5.2 and Figure 5.3) that work together to identify and describe tasks, collect data on identified tasks and areas of interest, analyse data to generate meaningful insights, evaluate the analysis against desired outcomes and goals of the application to create a list of ranked recommendations of multiplication opportunities, and finally compile and create an actionable report or decision point highlighting findings. DCA or Describe, Collect and Analyse, leverages existing knowledge about how to systematically identify, document, and treat mature task candidates, while RR or Recommend and Report takes the identified candidates, to evaluate and prioritise where we could multiply/automate, as well as how we measure if we are succeeding at multiplying organisational capability.

- **Describe in detail**: The *Describe* phase is focused around defining the scenario / exercise under review / study. We use methods from human factors research, such as Hierarchical Task Analysis (HTA) to identify goals and where necessary decompose tasks into self-contained deterministic tasks (using definitions and concepts from Cynefin). Our terminating decision point or deliminator for task decomposition, follows a simple method of task description to identify actions, objects, subjects, tooling and processes of a task. If a task is unable to be described, then decomposition is used to segment and remove non-deterministic / emergent behaviour. From these high level task descriptions it is possible to abstract and visually communicate tasks using simple information processing models such as OODA and UML visualisations in the form of Task Models.

- **Collect in detail**: Once tasks are identified, described and modelled, for further analysis it is necessary to verify models and assumptions are correct through the use of traditional data collection methodologies such as interviews, speak aloud, observations and critical decision method. This phase allows for the planning of events and scenarios, the collection of information from identified data sources (from *Describe* outcomes) and the generation of a timeline to add a temporal element for further analysis.

- **Analyse in detail**: The *Analyse* phase, focuses on analysing, reviewing and verifying of outcomes from *Describe* phase against data collected from *Collect* phase. This verification allows for accurate representations of deterministic tasks as well as an opportunity to identify extra tasks that are required but were missed during Describe. Using the timeline of events and collected data, task model instances generated and story boarded using data to accurate depict the flow and transformation of information and data throughout the exercise or scenario under review. Once an accurate representation of data flow and transformation as well as the scope of deterministic tasks required are represented, analysis can occur where tasks are quantitatively and qualitatively reviewed against 4 different categories - resource usage, frequency, impact and complexity

- **Recommend in detail**: From *Analyse*, we can evaluate the outcomes to generate recommendations against the four areas (resource, frequency, impact and complexity) and we also evaluate against our overall goal the frameworks' application. This evaluation of the analysis against the goal of the frameworks' application will provide an opportunity to generate a ranked list of tasks ranging from most to least suitable for multiplication/automation.

- **Report in detail**: *Report* phase deals with the presentation of recommendations based on outcomes from the previous phases. We take the evaluations from *Recommend*, which are based on the reviews and measures of *Analyse*, which are based on data collections from *Collect*, which are based on task identification from *Describe*, to finally present actionable recommendations.

5.2.3 Demo 1

As mentioned in Section 5.1, the presentation included the use of two scenarios; a simple example to highlight key concepts and ideas, and a second more complicated scenario that had been provided by one of the attendees during the early stages of the framework development. These scenarios provided the basis for the demonstration of the key concepts and ideas of DCARR. Demo 1 is Scenario 4 - Simple scenario 2 - Cooking, outlined in Section 3.3.7.

For the purposes of the demonstration, it was necessary to highlight and showcase examples of the steps of DCARR in relation to the given scenario. Figure 5.5 is a demonstration of steps within the Describe phase and the associated outcomes to the scenario, in particular the steps of scenario mapping and task goal identification. Figure 5.6 is also an example from the Describe phase, this time showing an example state machine built from an associated described observation. Figure 5.7 is an example of the outcomes from the Collect phase, with time elapsed key decisions. Figure 5.8 is an example of Analyse phase, where the collected data is presented as a story board of key decisions represented as task models. Finally, Figure 5.9 is an example of the quantitative measure that occurs during the evaluation step of the Recommend phase.

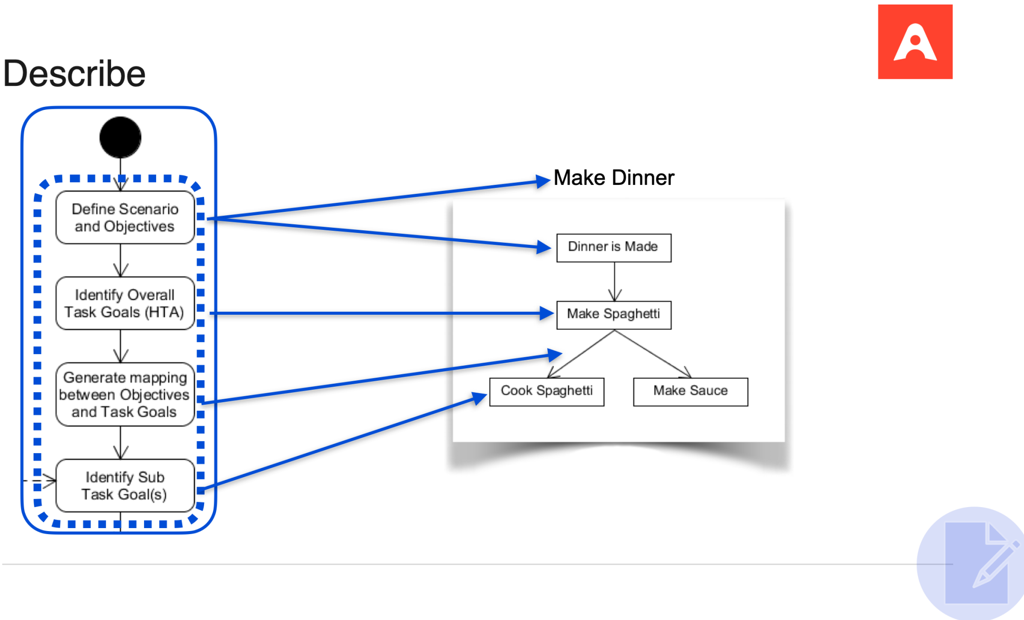


Figure 5.5: Demo 1: Scenario mapping and task goal identification

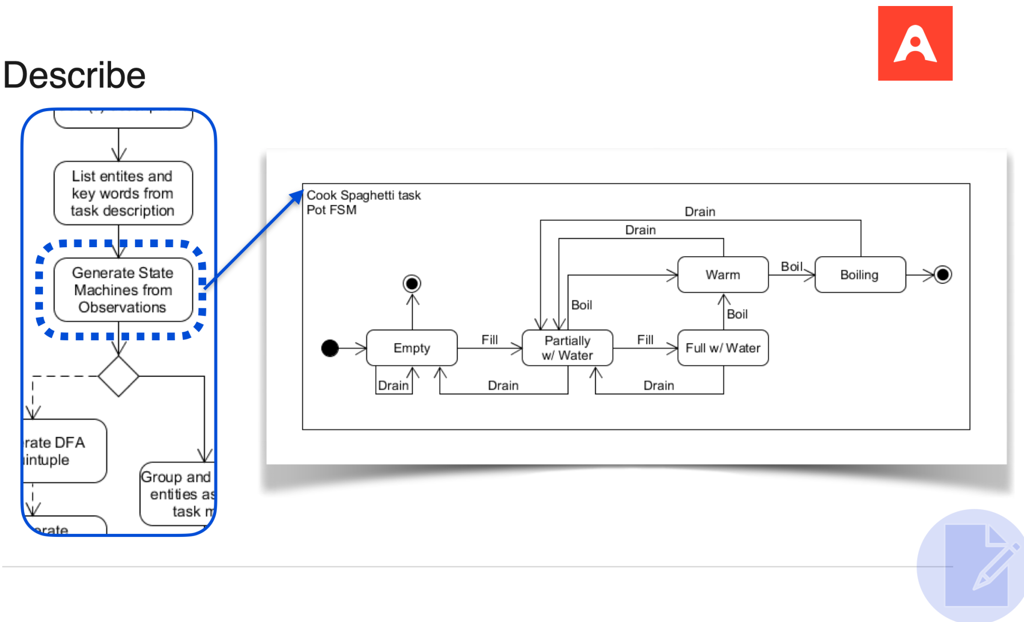


Figure 5.6: Demo 1: State Machines from observation

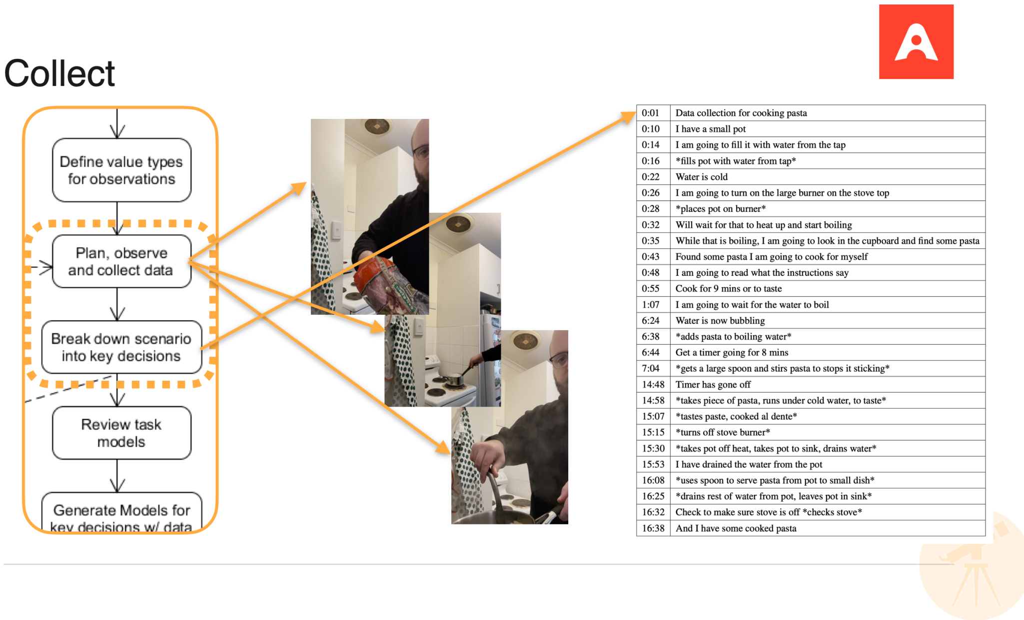


Figure 5.7: Demo 1: Data collection and timeline of critical decisions

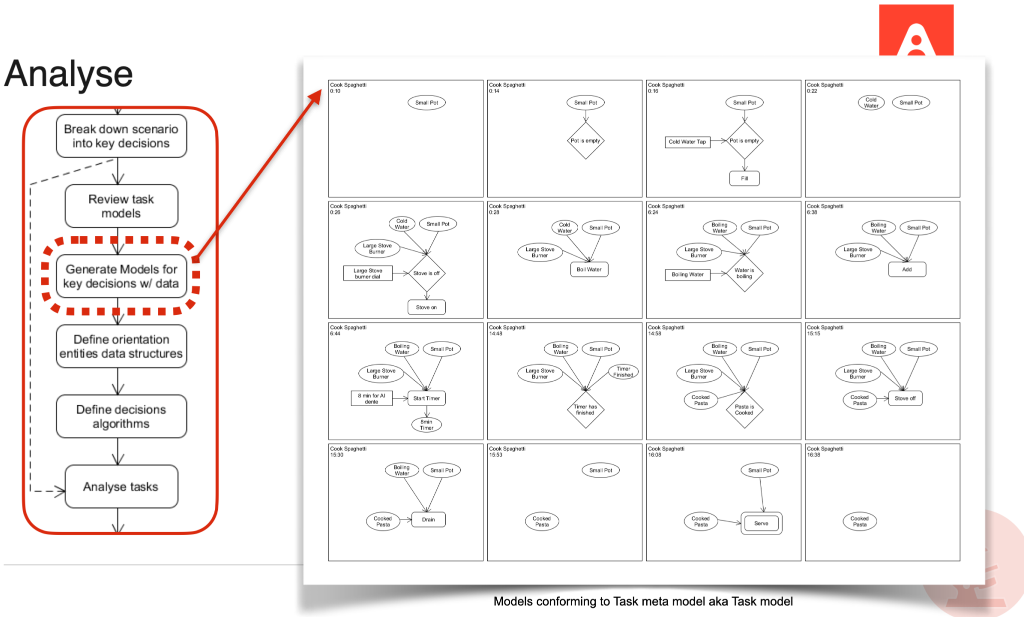
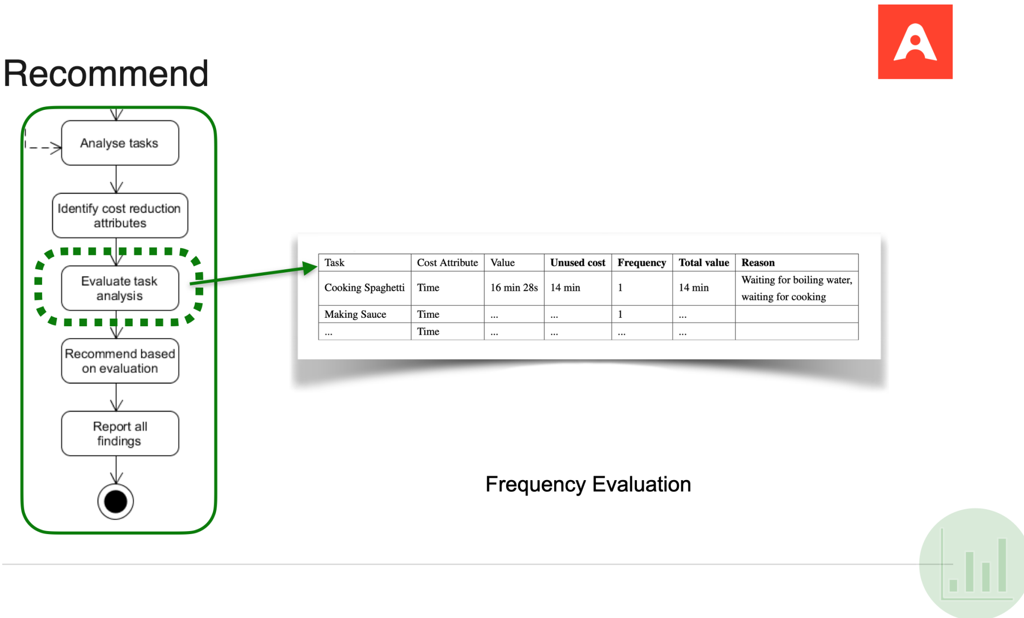


Figure 5.8: Demo 1: Story board of task models with data



Task	Cost Attribute	Value	**Unused cost**	**Frequency**	**Total value**	**Reason**
Cooking Spaghetti	Time	16 min 28s	14 min	1	14 min	Waiting for boiling water, waiting for cooking
Making Sauce	Time	...	...	1	...	
...	Time	...	...	...	...	

Figure 5.9: Demo 1: Sample quantitative/qualitative analysis

5.2.4 Demo 2

The second demonstration uses a complicated scenario provided by one of the attendees during the early stages of the framework development. Demo 2 is based upon Scenario 5 - Simpilot controlling two aircraft, outlined in Section 3.3.7. This scenario is a direct example of the tasks and processes conducted by White Forces (in particular Blue Roleplayers) during LVC distributed training exercises. The benefit of this scenario is that it can draw upon the experience and intuition of participants in collaboration with the researcher, to demonstrate the versatility and features of DCARR.

As with the previous demo, the purpose is to highlight the outcomes and benefits of DCARR. Figure 5.10 is a rehash of the scenario as initial presented and defined in Section 3.3.7. Figure 5.11 is the outcomes of the scenario objective and task goal identification of the Describe phase. Figure 5.12, is the breakdown of the steps of the task, as presented in Section 3.3.7. Figure 5.13 is a naive attempt at generating a task description and presentation of the task description method. From the task description, Figure 5.14 is the generation of a state machine for the described observation. Finally, Figure 5.15 is a naive attempt at representing the task as a task model. Due to the limited resources available to the researcher, the presentation material for this demo was limited to outcomes within the Describe phase.

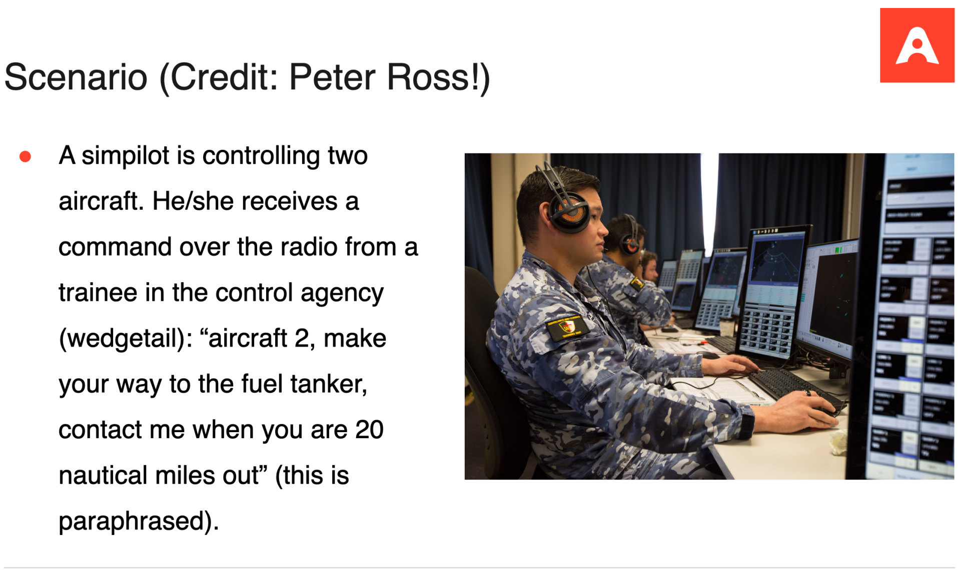


Figure 5.10: Demo 2: Scenario description and information

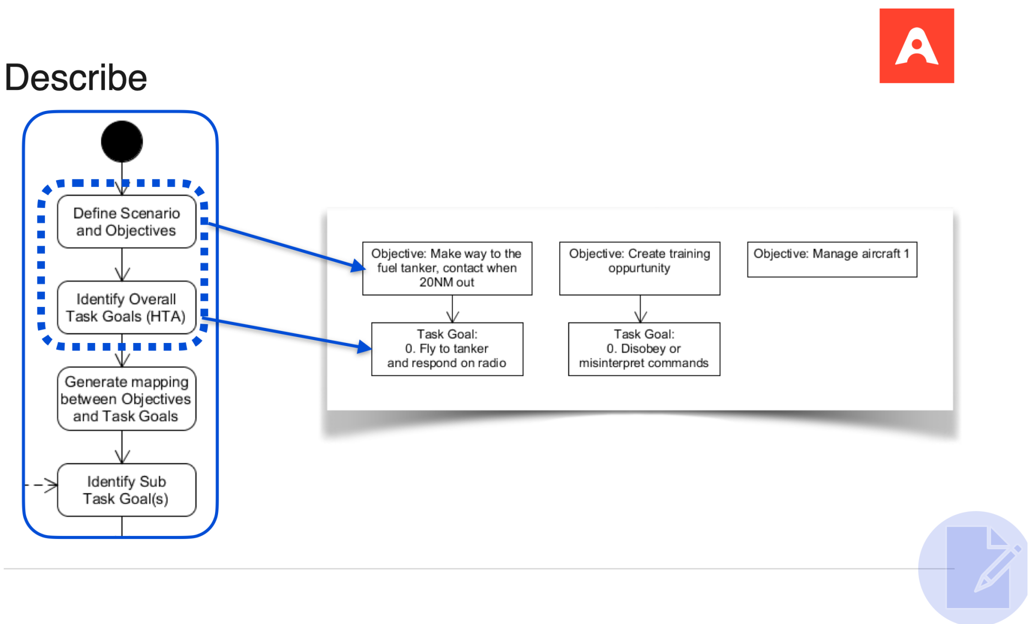


Figure 5.11: Demo 2: Scenario objective and task goal identification

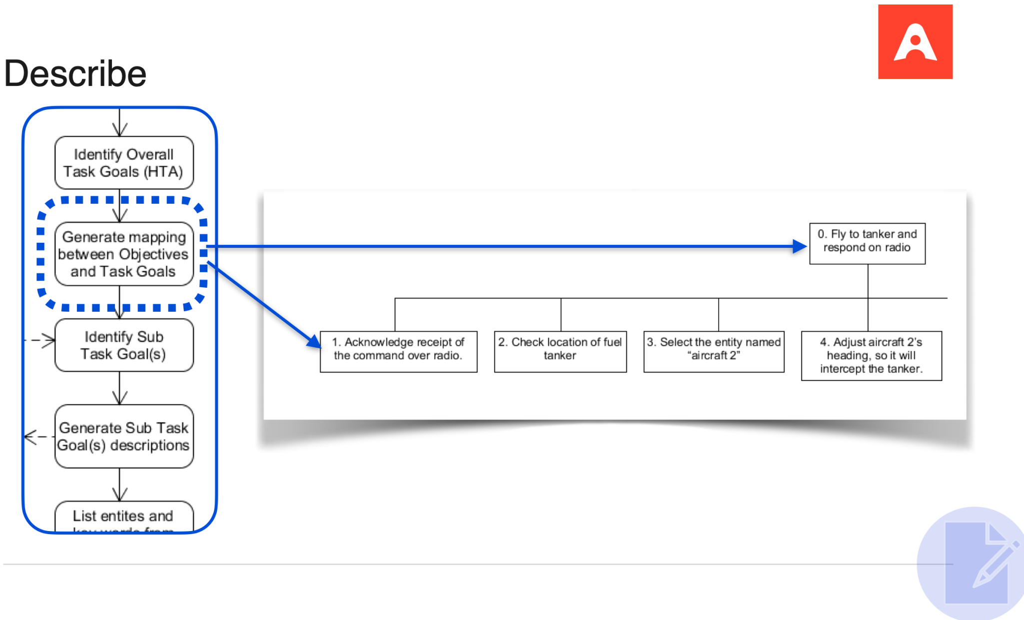


Figure 5.12: Demo 2: Task goal and subtask goal mapping

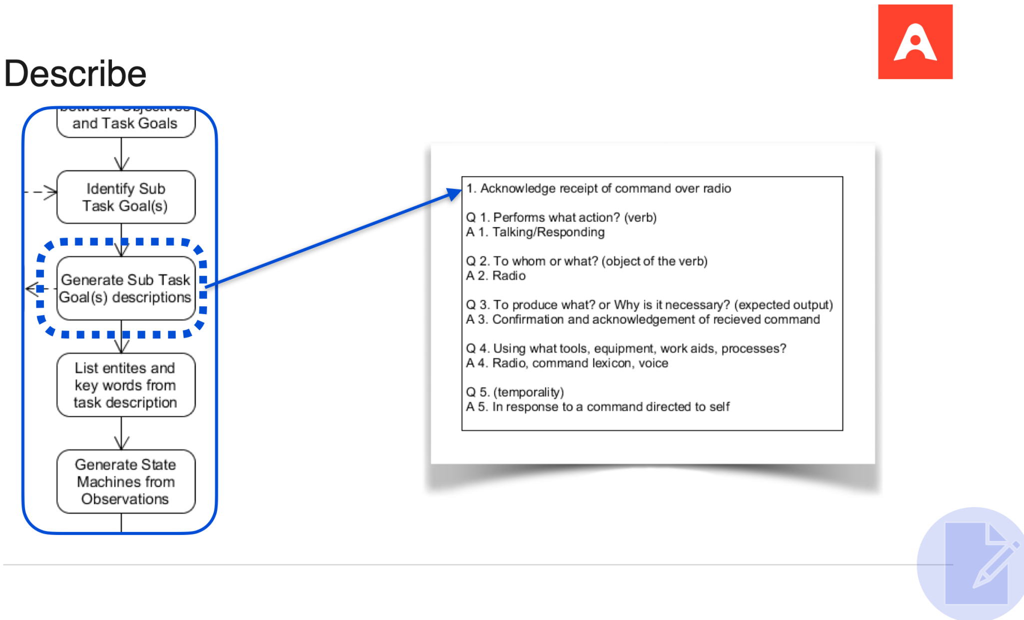


Figure 5.13: Demo 2: Task description method and output

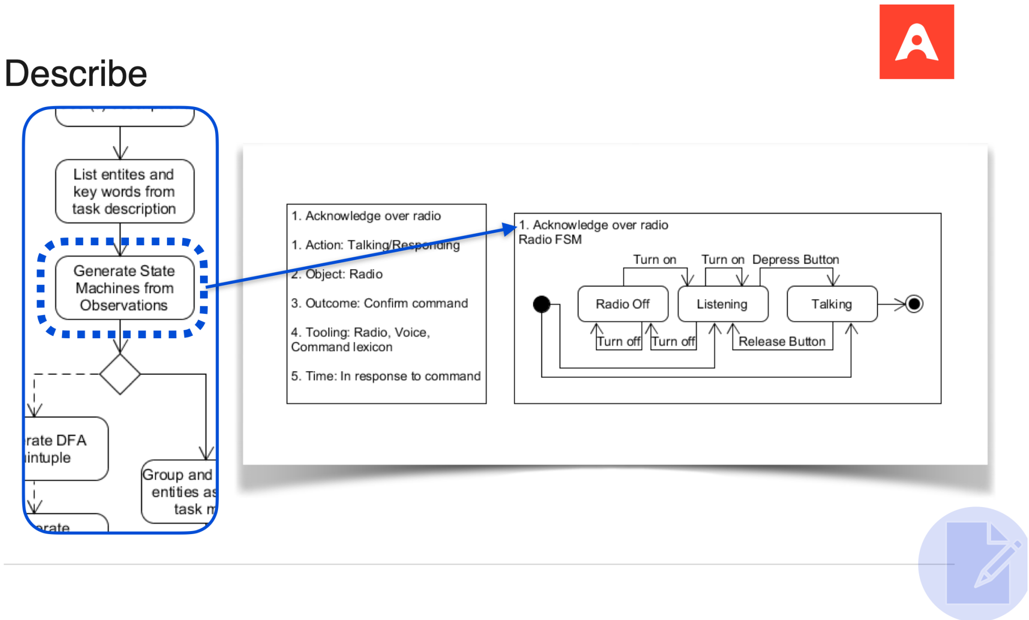


Figure 5.14: Demo 2: State machine generation from observations

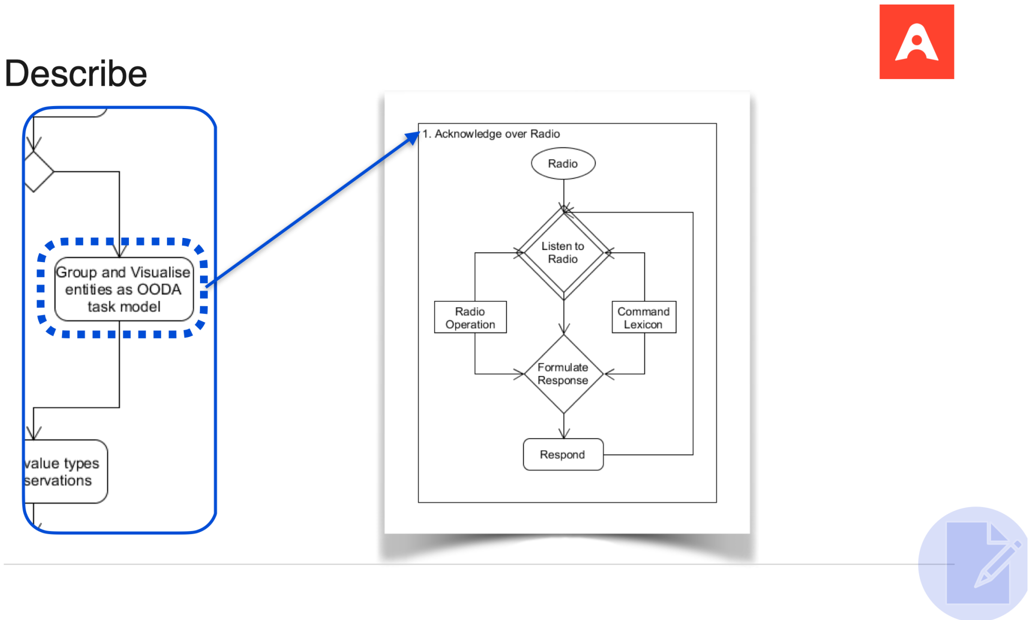


Figure 5.15: Demo 2: Task model based on OODA representation

5.2.5 Workshop

For the workshop component of the presentation, participants decided to continue with Demo 2, and continue generating outcomes for other identified tasks in the familiar scenario of a simpilot controlling two aircraft. The purpose of the workshop was to elicit feedback from participants, regarding usage and comprehension of DCARR. It was also a chance for further questions and demonstration of the time and ease of completion of outcomes for the different phases of DCARR.

In the original scenario only some tasks had been described as examples. In discussion with participants, some additional tasks were selected for description. Figure 5.16 and Figure 5.17 were the steps taken in the generation of a finite state machine for the observable states in the “Config Tool” task. Figure 5.18 is an overview of the set of finite state machines for the observable components of the described tasks.

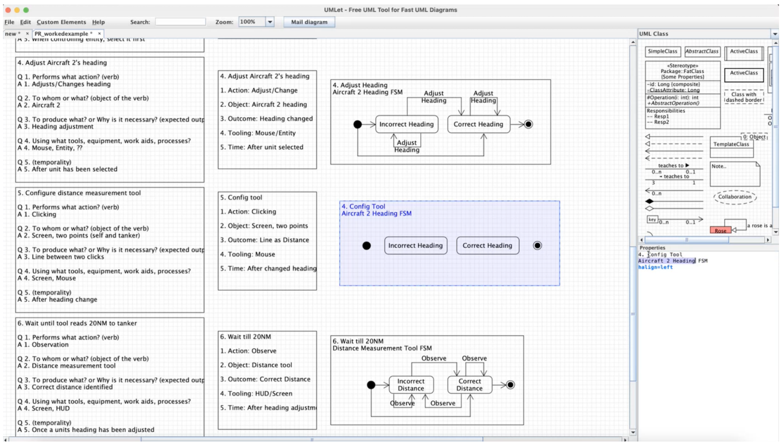


Figure 5.16: Workshop: State machine generation for task "Config tool"

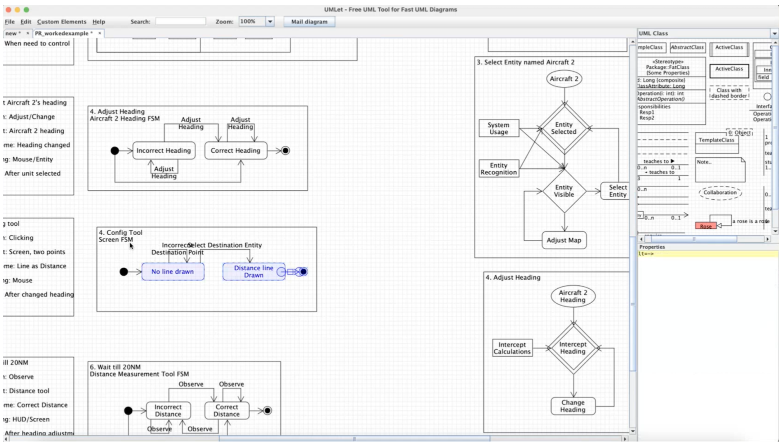


Figure 5.17: Workshop: State machine generation for task "Config tool" part 2

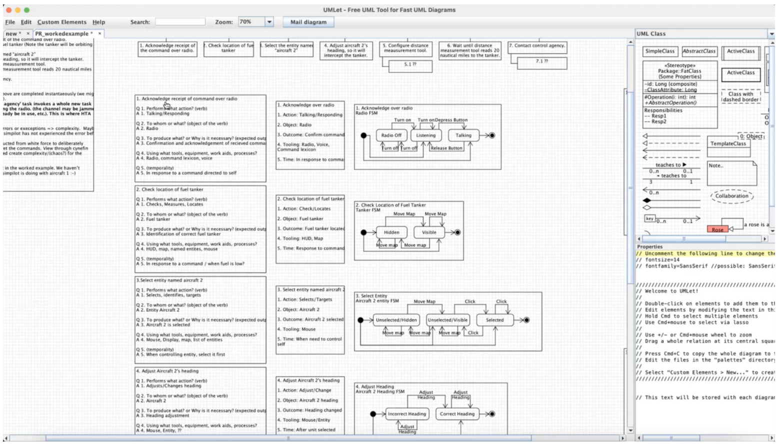


Figure 5.18: Workshop: Overview of completed state machines based on observations

After filling in gaps from Describe outcomes to a satisfactory level, the decision was made to continue with the demo due to time factors. Unable to collect quantitative data, due to the time constraints of the workshop, participants used experience and knowledge to bypass the Collect phase. This saw the workshop move to the Analyse phase, and directly into story boarding of task models (under the assumption that all outcomes of Describe were validated against quantitative data and observations). This allowed for participants to further experience outcome generation of DCARR.

The remainder of the workshop was focused around the discussion of task model story boards, given an arbitrary temporal point (point 1, 2, 3, etc.). Figure 5.19 and Figure 5.20 were the attempts made at generating task models at a temporal point. This discussion highlighted the importance of data collection in the generation of quantitative outcomes of the Analyse phase.

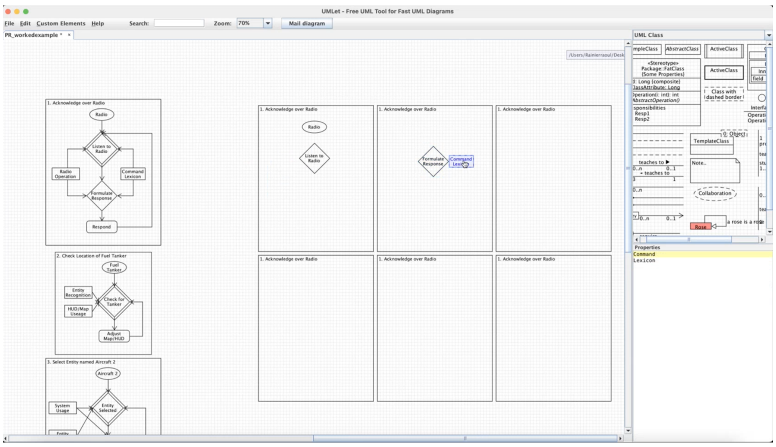


Figure 5.19: Workshop: Discussion of likely data flow and timing

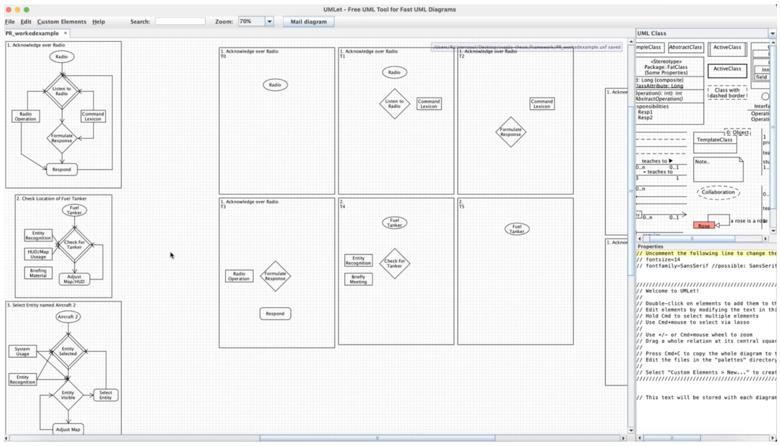


Figure 5.20: Workshop: Story boarding of task with temporal component

5.2.6 Final Points

After a discussion regarding feedback and the response to DCARR (Section 5.3), the presentation and workshop ended with a highlight and overview of the purpose of DCARR and a short summary of what DCARR is/can do. This is reproduced below. After which, the participants and supervisory team were all thanked for their time and patience. This concluded the presentation and validation of the main outcomes of this body of research.

- **Purpose**: Identify automation opportunities
- **Summary**: We describe and decompose tasks until we have a broad selection of deterministic tasks, we collect data to verify and validate our assumptions, we analyse those tasks against impact, frequency, cost and difficulty, we evaluate the analysis against the goal of the framework application and produce a ranked list of tasks from highest to lowest that are appropriate and suitable for automation.
- **Benefits**: Communication amongst stakeholders
- **Limitations**: Initially time intensive
- **Key points**:

  – Provides foundations for automation / capabilities enhancers
  – 5 stages
  – Informed by relevant areas of research and existing theory
  – Reproducible method
  – First pass / attempt at addressing a need for understanding where to multiply

5.3 Feedback and responses

During the course of the presentation and workshop, feedback and responses were provided to the researcher and supervisory team. These were often related to aspects of development and application of DCARR (i.e. design choice, edge case, example task, etc.). The following are the pertinent outcomes unrelated to development and evolution and are the main lessons learned in the Validation of DCARR and represent the final step of the proposed research method (Section 3.3).

5.3.1 Contribution

One of the major contributions that arose from this research, as was noted by one of the attendees, is the novel combination of Human Factors and Computer Science. The observation made, was that the combination of multidisciplinary teams from Human Factors and Computer Science means that there is the potential for a lack of shared understanding as a result of different discipline perspectives. The novelty of this research is the translational assistance between Human Factors and Computer Science with an improved shared understanding between multidisciplinary teams as a result of using language and terminology from both areas (including different views and representations of task and components).

5.3.2 Recommendation

Based on the presentation and discussion, two recommendations were given to the researcher. These constructive observations were based on the initial understanding of DCARR and the perceived limitations of the information as it was presented.

The first recommendation relates to the lack of articulated exit/termination points of DCARR. The inclusion and articulation of clearly defined termination and exit points based on limitations and considerations (lack of collectable data opportunities, lack of clear task decomposition, areas of complexity and nondeterminism, etc.) were included throughout the early stages of development and the evolution of DCARR (Section 4.3.1). With the later stages of DCARR, the researcher removed these clearly defined and articulated termination and exit points, therefore during the validation exercise, the branching aspects of DCARR were not clearly described.

This led onto the second constructive observation, which was the lack of clearly articulated and communicated partial application. Due to the lack of articulated exit and termination points, attendees saw DCARR as a continuous, exhaustive and extensive method that required each and every phase to be completed to achieve any benefit. In reality, the benefit of DCARR is the application of phases given any particular circumstance. For example, during the *Describe* phase, there is substantial benefit in the discussion of stakeholders and participants in generating initial mappings and understandings of tasks. The structured approach for capturing and representing tasks and components amongst different stakeholders (using intuition and determinism to classify as at a high level, candidates suitable for multiplication, and providing methods for drilling down into tasks to a suitable level of representation and communication), is of benefit in informing later stages of application.

5.3.3 Limitations

Because of the lack of communicated exit/termination points and the lack of a description of the potential for partial application, attendees noted, that while appearing to be beneficial, DCARR suffered from an important limitation. It was noted, that ironically, while the tasks and processes that White Force perform in their duties as distributed simulation LVC training facilitators is intensive and exhaustive, what was being presented in the form of DCARR, appeared to further complicate an already difficult process - while claiming to reduce already limited resources by aiming to multiply capabilities. Participants enquired about the estimation for time and cost of application of DCARR, but due to the initial approach and attempt at process multiplication, DCARR is still in its infancy and these estimations need to occur as part of future work and experience with applying DCARR.

5.3.4 Closing comments

At the end of the session, participants were for asked final closing comments and observations. As this research was aimed to address a specific problem (of multiplication of White Force capabilities, through the use of model driven approaches), the final result (DCARR) ended up being a framework for the definition and resolution of White Force capabilities into a ranked list of priority targets for automation application.

At the commencement of this research, given the problem and proposed solution direction, SME's of AOSC were initially expecting a model. Due to the overwhelming size and complexity, including multiple external and internal factors (MilSkil as White Force, RAAF as stakeholder and primary funding, AOSC as researchers and observation in support of LVC training, biannual LVC training events Virtual Black Skies 2018/2020 and opportunities for data collection, research funding and external contractor engagement, and COVID-19, etc.), the ability for a model to be presented, that was adequate and beneficial, was not possible. Instead, the researcher created a framework to be used to generate the model. It was noted, that initially it was not understood how big a job this would be, a model was thought to exist by the end of it, instead a framework exists on how to produce the model.

DCARR, as a *"Structured and repeatable approach to collecting and integrating subjective and objective evidence to deliver a prioritised list of recommendations that target automation"*, is a useful outcome and is the first step in achieving the initial desire of having a model. For future work, it is necessary to discuss with RAAF the need for support to produce the model and some limitations need to be addressed before application. Before presenting to RAAF for resource allocation, time (how long it will take) and cost (how big a task) estimations for application need to exist. These estimations can then be used to determine the cost benefit ratio that will inform future application/future work.

5.4 Updated Framework based on feedback

Section 5.3, outlined a series of contributions, limitations, and recommendations of this research. Of the limitations, it was noted that there was a lack of an initial estimation of time to apply and action DCARR. Of the recommendations, it was noted that there needs to be an adjustment to the presentation and communication of DCARR. This presentation and communication adjustment was to highlight and articulate the exit/termination points of DCARR, as well as the benefit of a partial application approach of how phases work together. Therefore, the following is an adjustment to the presentation of DCARR in an attempt to better articulate to a fresh audience what DCARR is, how it works, and the relationship between phases.

Figure 5.21 is the DCARR model as presented in this chapter. It follows UML Flow chart notation, and tasks and components are directed (i.e. arrows indicated the flow of information and processing). It contains no indication of phases or termination points, other than the initial start and finish points. Figure 5.22, builds upon Figure 5.21, with the demarcation of colour coded phases. Figure 5.23, builds upon Figure 5.22, with the highlighting of key selected tasks and components of DCARR that highlight and are the core of the outcomes of each phase. Figure 5.24, builds upon Figure 5.23, with the removal of non-highlighted tasks and components. Figure 5.25, builds upon Figure 5.24, with the condensing of key tasks and components into a "lite" version of DCARR. Figure 5.26, builds upon Figure 5.25, with the simplification and renaming of tasks and components. Figure 5.27, builds upon Figure 5.26, with the separation of phases into mini phase flow charts (each phase is now represented as a flow chart). Figure 5.28, builds upon Figure 5.27, with the removal of start and finish points, so that each phase is a series of directed tasks and components.

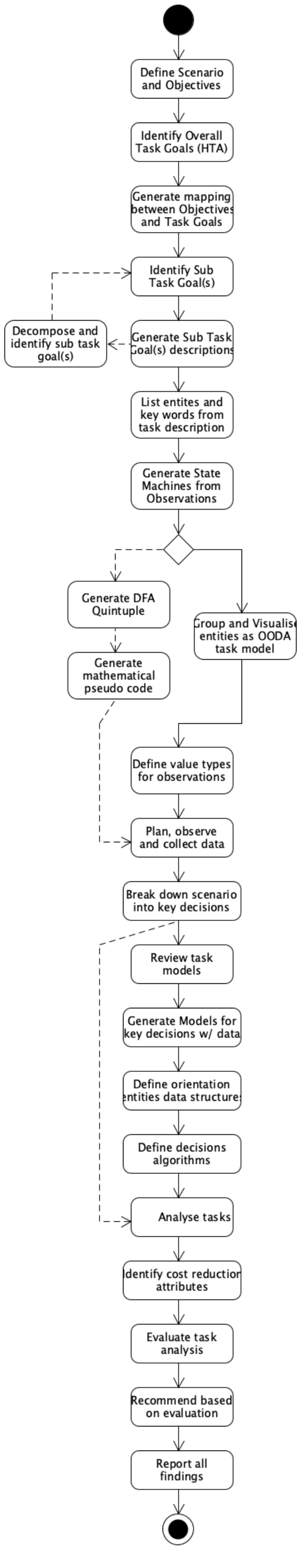


Figure 5.21: DCARR presented as a flow chart UML. Contains minimal branching, phase identification or articulated termination points other than after report generation.

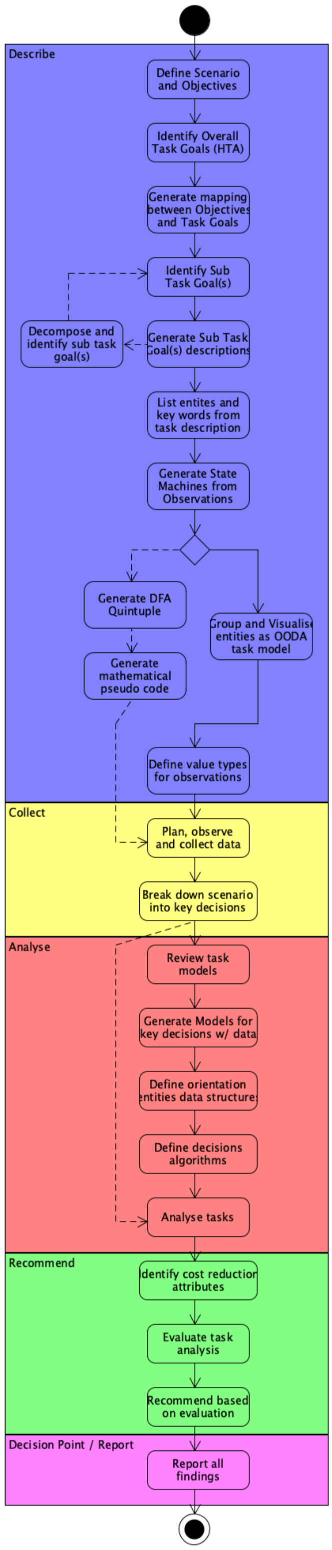


Figure 5.22: DCARR presented as a flow chart UML, with phases highlighted in different colours. Blue for Describe, Yellow for Collect, Red of Analyse, Green for Recommend, and Fuchsia as Report

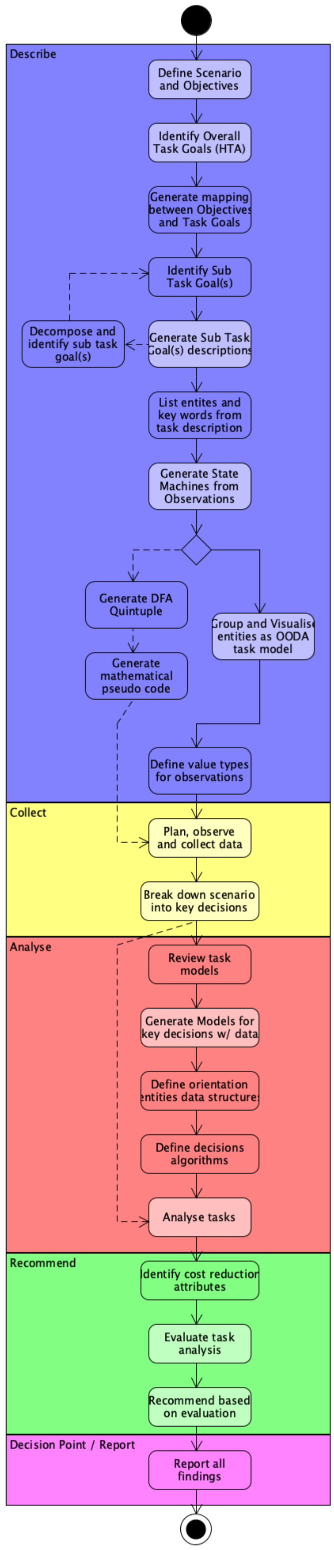


Figure 5.23: DCARR presented as a flow chart UML and coloured phases. Key components and tasks are highlighted.

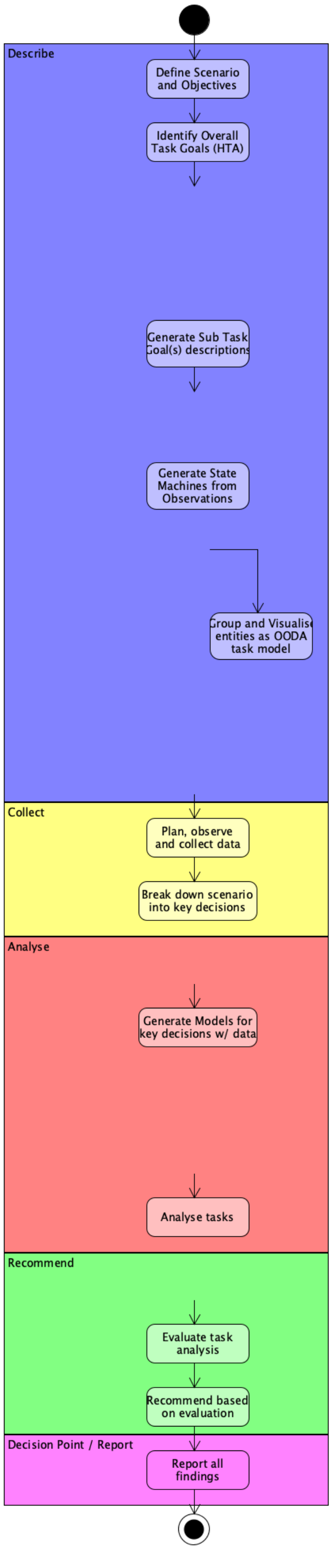


Figure 5.24: DCARR presented as a flow chart UML, with coloured phases and only key tasks and components.

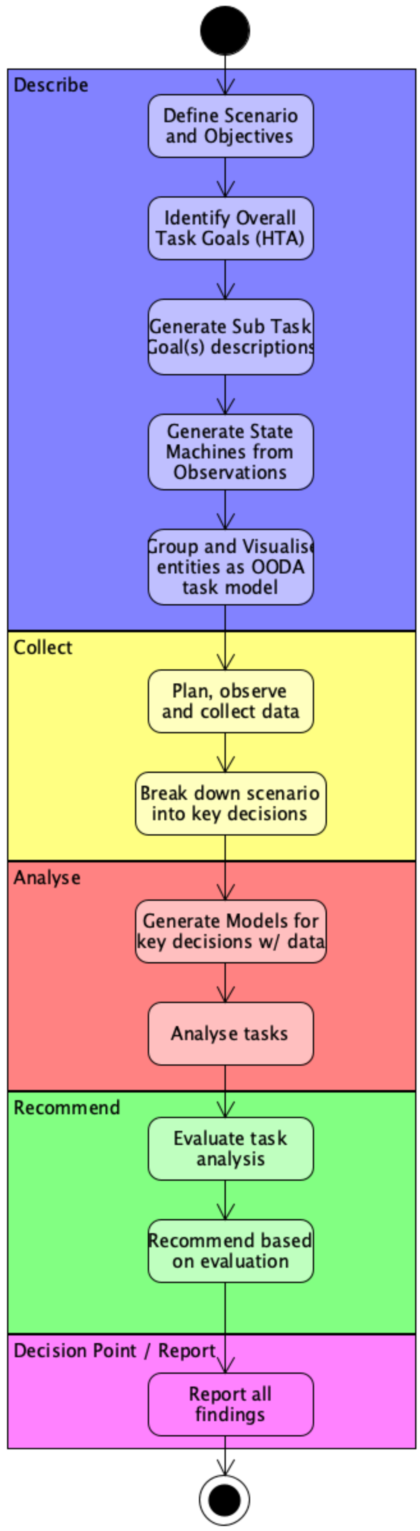


Figure 5.25: Reduction of DCARR into a condensed flow chart + phase breakdown, with only key tasks and components

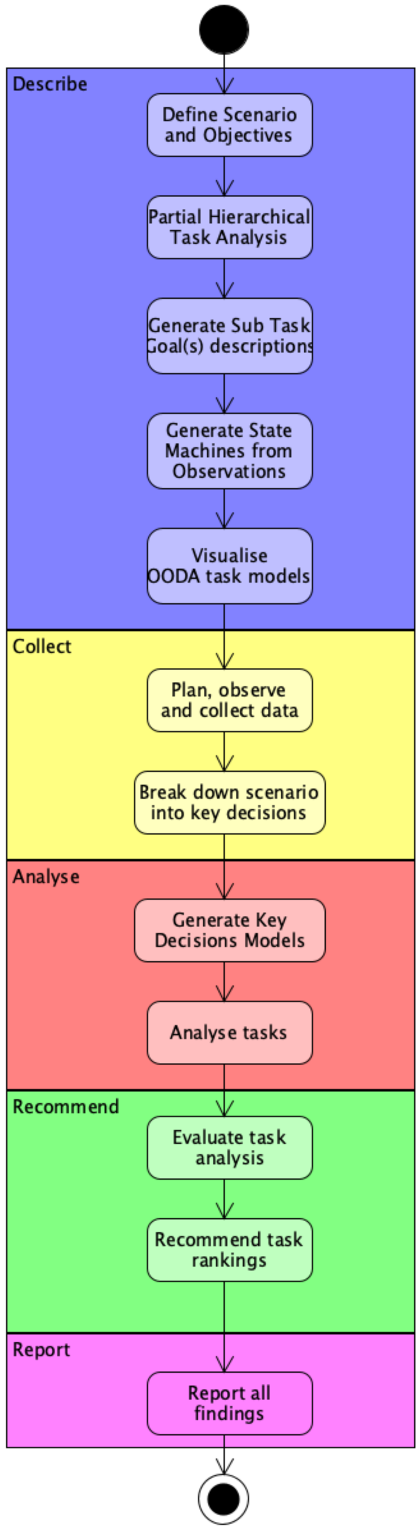


Figure 5.26: Condensed DCARR with simplification of task and components names and identifiers.

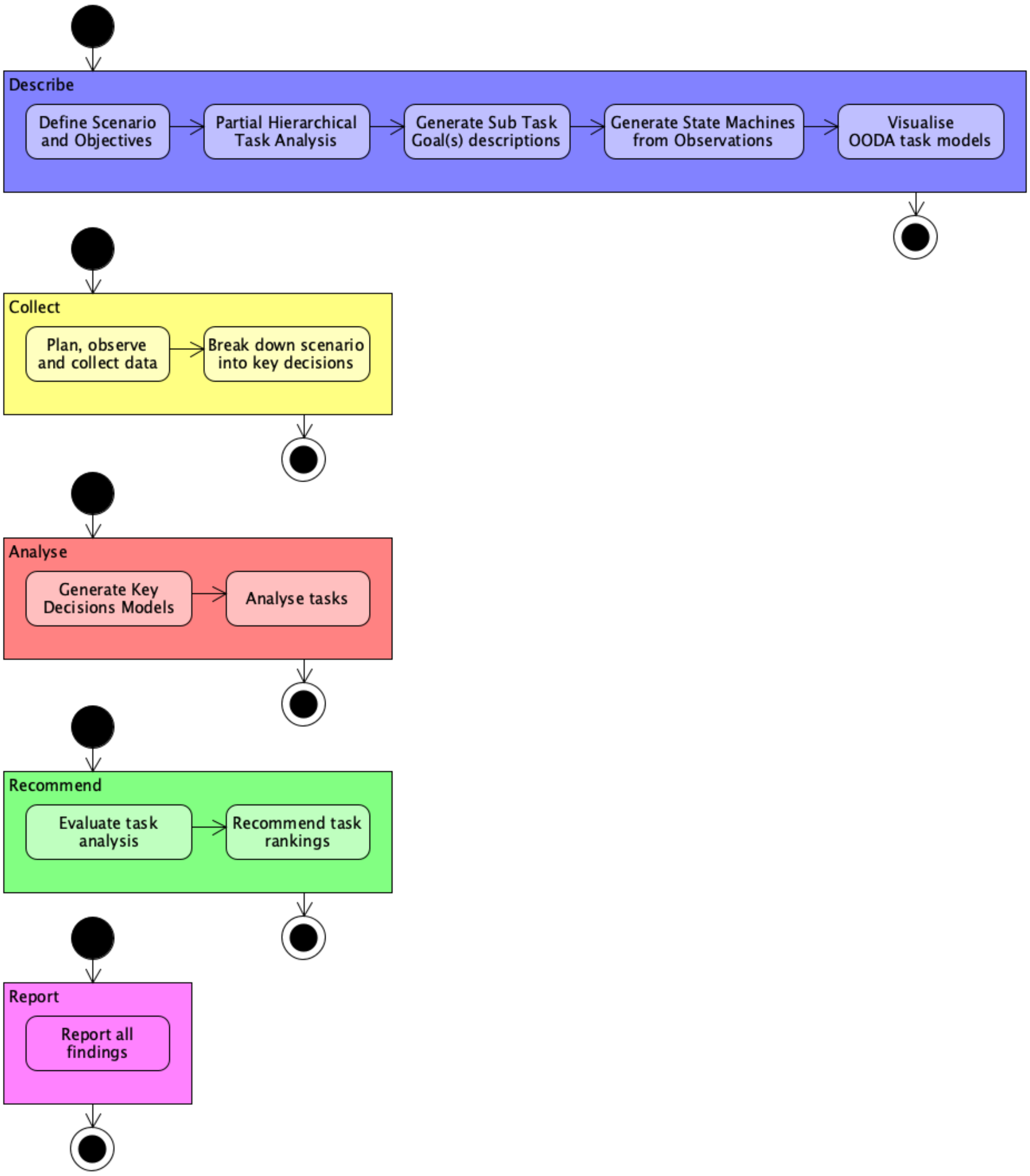


Figure 5.27: Separation of condensed and simplified DCAAR into flow chart phase groupings

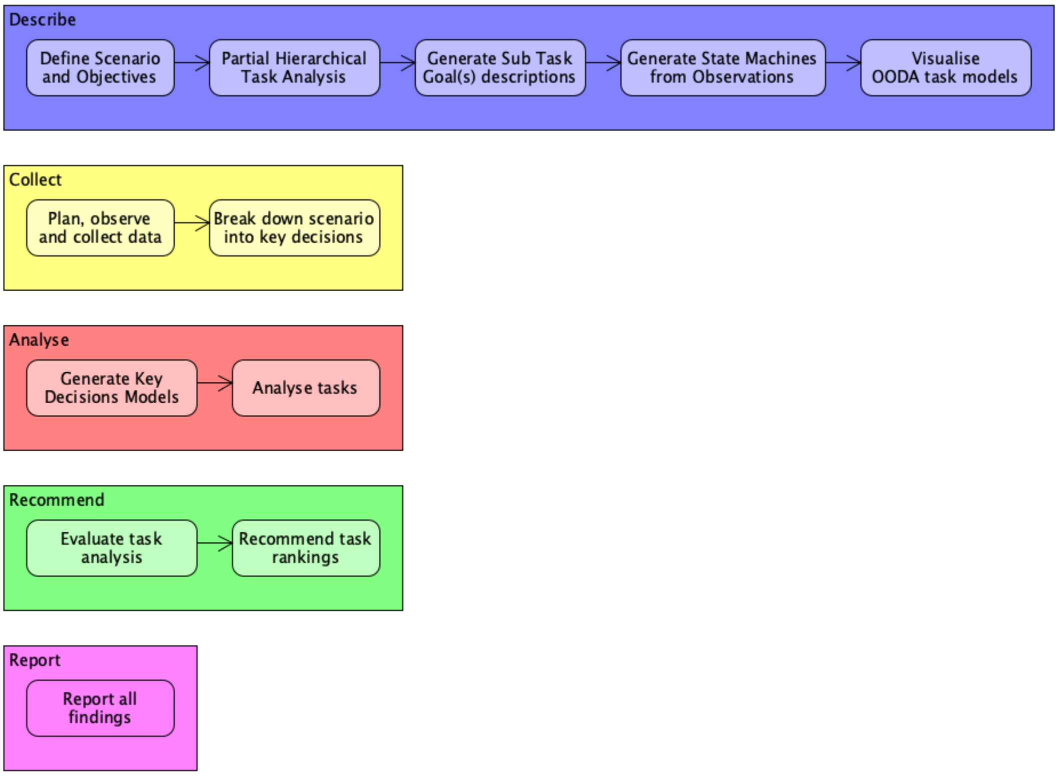


Figure 5.28: Removal of start and finish points from separate phase groupings of condensed and simplified DCARR

Figure 5.29, builds upon Figure 5.28, but is also a new form representation. The medium used to visual DCARR phases has changed from a subset of UML flow chart notation, to "flash cards" represented with a colour coded bounding box and symbol used to denote each phase. Describe is blue with a pen and paper (to symbolise note-taking), Collect is yellow with a telescope (to symbolise observation), Analyse is red with a microscope (to symbolise detailed and minute analysis), Recommend is green with a spreadsheet/bar chart graphic (to symbolise quantitative data representation), and Report is purple with a book or report (to symbolise information collation and documentation). Figure 5.30, is the new and updated phase diagram used in the presentation material for validation to AOSC. It follows the colour and symbology of Figure 5.29. Figure 5.31, builds upon Figure 5.29, with the reinstatement and inclusion of all tasks and components of DCARR that were removed in Figure 5.24. Figure 5.32 is a combination of Figure 5.29 and Figure 5.30. It is the representation of "flash card" phase lists in a circular format similar to the updated DCARR phase diagram.

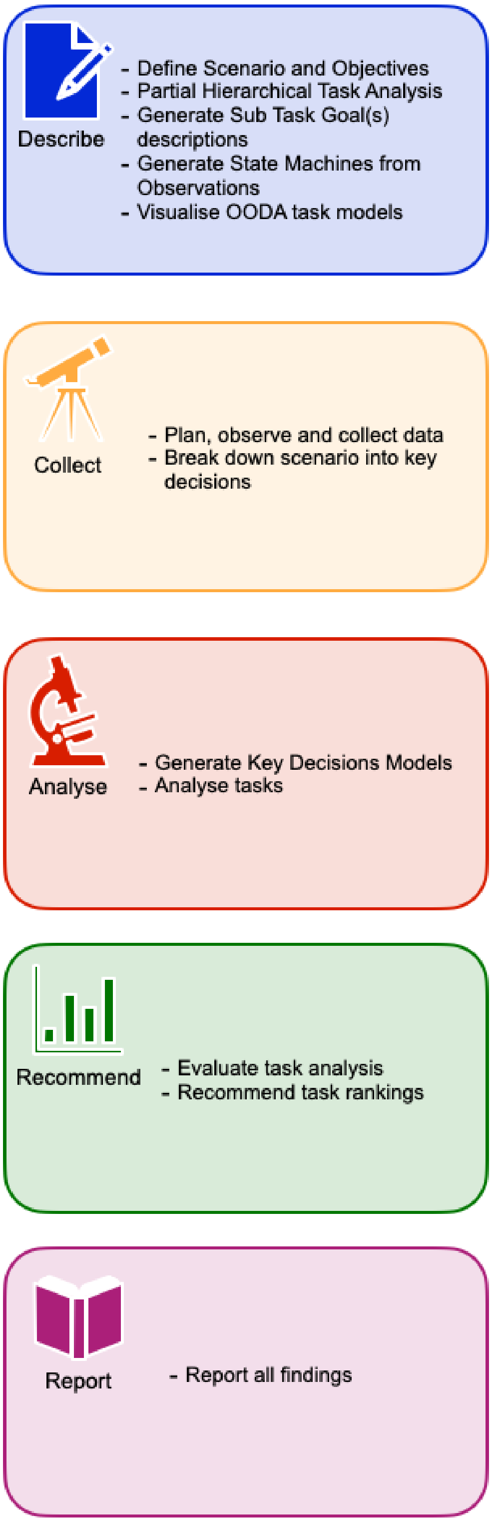


Figure 5.29: Transformation of condensed and simplified DCARR UML flow charts into "flash card" lists. Inclusion of key tasks and components. Makes use of colour and symbols to denote phase

Figure 5.30: Simplification of DCARR Phase diagram into a unique and styled visual graphic. Uses colour and symbols to denote phases.

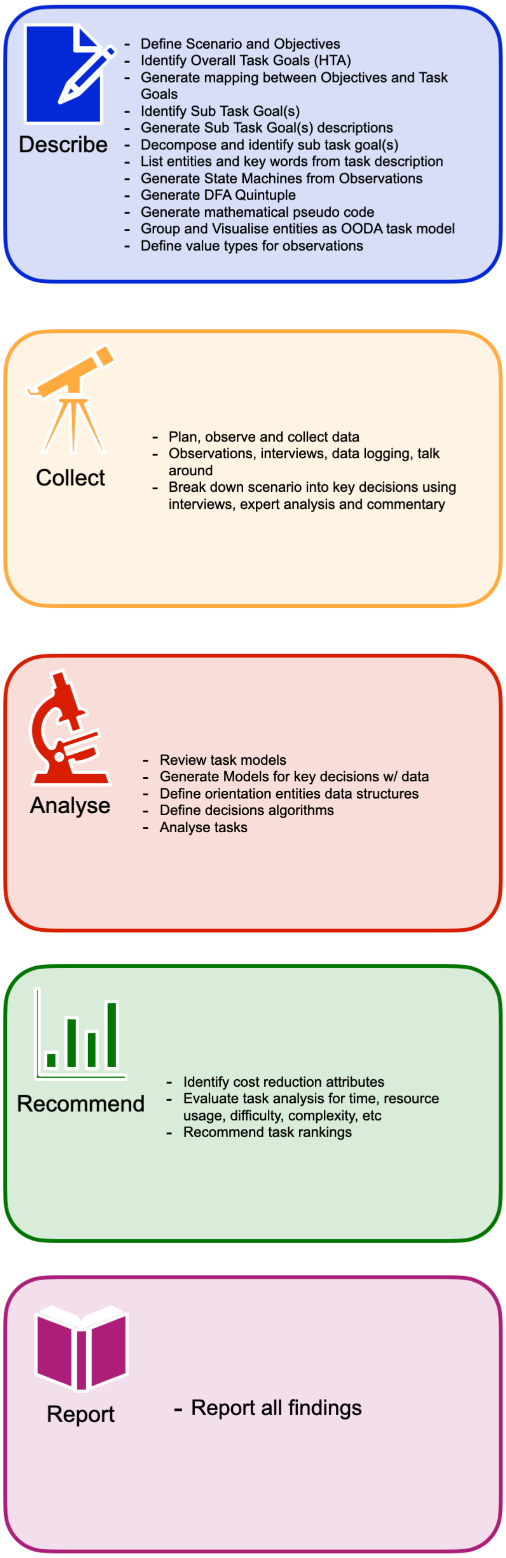


Figure 5.31: Reintroduction of all tasks and components of DCARR into "flash card" list. Maintains colour and symbol to denote phase.

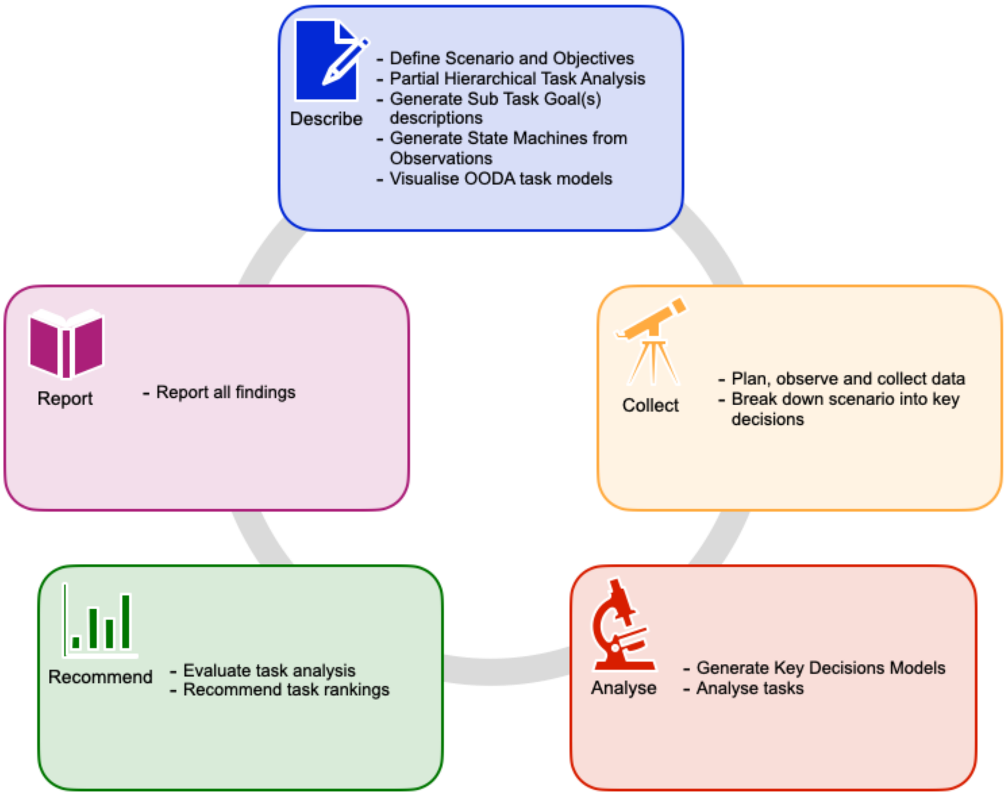


Figure 5.32: Organisation and presentation of key tasks and components "flash card" lists (see Figure 5.29) to mirror updated phase diagram (see Figure 5.30)

Figure 5.33 is the culmination of the amendments to the visual representation of DCARR. It is the combination of the new and updated phase diagram (Figure 5.30) with each task and component of DCARR. The central 5 phases are identified, using colour and symbols, and connected to each root phase node is the branching tasks and components that comprise each phase. Each task and component is linked to highlight and demonstrate the relationship between tasks and components, but the implicit meaning is that each phase is self-contained and has a set of tasks. The linking is non-directional to imply a partial application strategy but are chained to demonstrate order. This figure is an improvement of the initial flow chart, but is to be used as a supplementary visual aid in conjunction with the flow chart representation (Figure 5.21) and step-by-step guide (Section 4.2). It is still limited by the lack of articulation of termination and exit points.

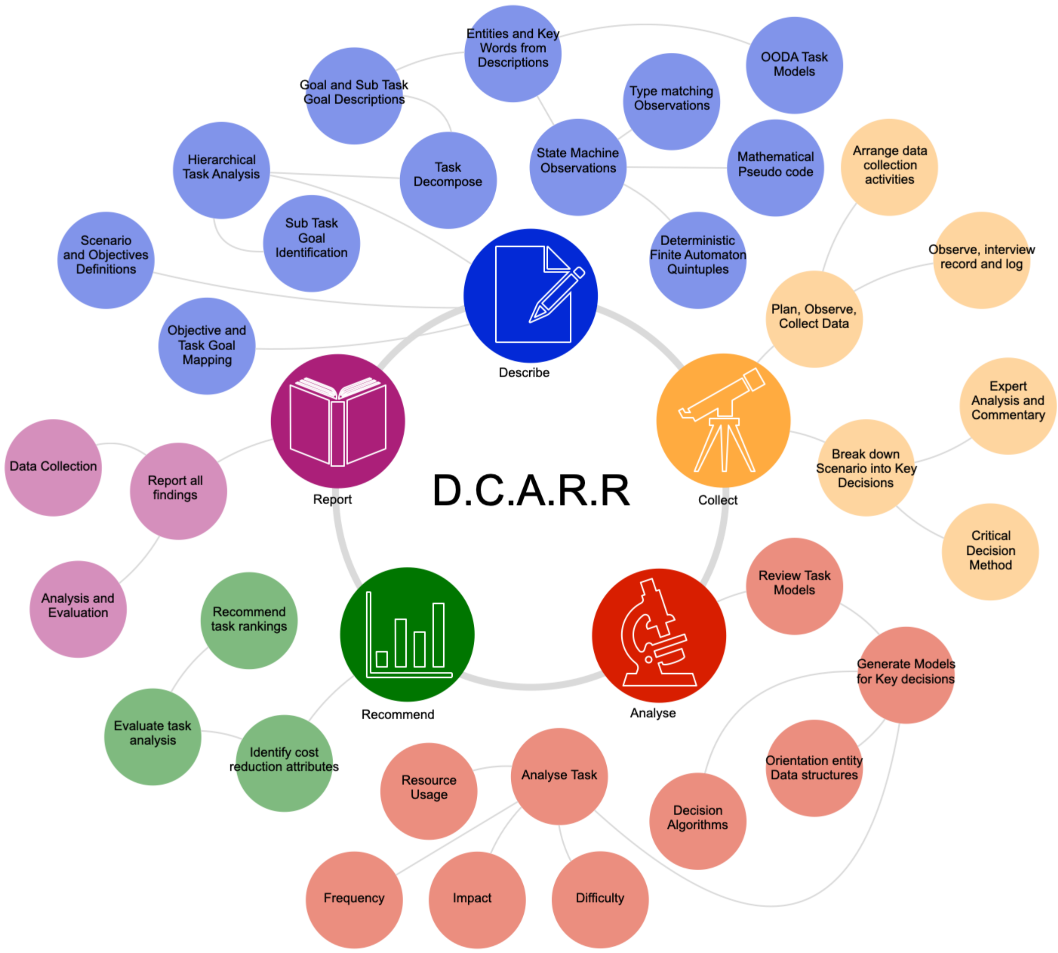


Figure 5.33: Expansion of phase diagram (see 5.30) to include all tasks and components of DCARR. Tasks are linked to denote relationship, and flow from phase root node outwards. Implicit communication of phase relationship and task independence aiming to communicate partial application

5.5 Chapter Summary

This chapter presented the validation of DCARR to subject-matter-experts, members of White Force and researchers from AOSC and Human Factors Research at DST Group. DCARR was developed to address the needs of White Force capability multiplication, which was reflected in the comments and feedback from this validation. DCARR, as a *"Structured and repeatable approach to collecting and integrating subjective and objective evidence to deliver a prioritised list of recommendations that target automation"*, is a useful outcome and is the first step in achieving the initial desire of having a model.

The other major contribution of DCARR, is as a method for multidisciplinary teams to improve shared understanding due to language and terminology used. DCARR, as presented, is lacking in articulated exit/termination points which may confused new users, as well as lacking in communicated partial applicability. Finally, the major limitation of DCARR, as it stands, is the lack of estimation for time and cost for application. In order to gain future support for model production, estimations need to be generated.

This chapter also aimed to present an update to the presentation of DCARR in the form of an updated visual graphic that aims to communicate the distinction between components and phases of DCARR, as well as highlighting the relationship between tasks.

This validation outcome leads into the following chapter, that aims to present a discussion around DCARR, including the original need that drove this research, the major contributions, how the research questions and hypothesis are answered, as well as the presentation of an example application of DCARR, the implications of DCARR and proposed areas of future work.

Chapter 6

Discussion and Future Work

This chapter presents a discussion of the work contained within this thesis. It presents a recap of the initial need that drove the aims of this work, a general discussion and overview of the chapters contained within this thesis, as well as a list of major contributions that have come from this work. A high level overview of DCARR is presented as it relates to the underlying pillars that it is built upon of identification, communication, representation, classification and automation ranking of candidate tasks and processes/routines. We present answers to the proposed research questions and hypothesis that drove this work, and present an example application of DCARR using a simple scenario involving our DCARR practitioner Bob, and small business owner Fred. Finally, we present the implications of DCARR and what it means as a generalised approach to structuring decision-making and, close this chapter with a presentation of areas of Future Work.

6.1 Need for White Force Multipliers

There is a need to multiply the capabilities of White Force. A "White Force Multiplier" refers to *"any technology or technique that leverages a given investment in White Force resources to provide improved White Force capability"* [5]. As simulation training becomes increasingly used, the requirement for diverse and disparate training objectives increases. This can manifest itself as an increase in the requirements for improved quality/fidelity and increased quantity/-variety of training elements.

White Force facilitated simulation training encompasses chaos and complexity. Complexity manifests as the training follows a mix of ad hoc processes and procedures, which are facilitated by a diverse cohort of individuals [5], to converge and deliver consistent outcomes [12, 22, 23, 135]. This is rendered more chaotic when things do not go to plan and White Forces need to adjust for immediate unexpected occurrences and unplanned deviance. This dynamicism in the ecosystem resists the application of automation workflows due to its instability [73, 76, 79, 90, 93, 96, 98, 136, 137].

An overarching systematic methodology or framework does not exist to identify candidate tasks and processes/routines for automation application. Therefore, a systematic methodology or framework is required to developed to identify, communicate, represent, classify and automate the White Force facilitated simulation training domain. By introducing a systematic methodology or framework, to do this, it will bring consistency, rigour, stability, structure, clarity and uniformity to an otherwise complex, chaotic and unstable environment.

6.2 Discussion on Model Driven Framework

Describe, Collect, Analyse, Recommend and Report (DCARR) — a Model Driven Framework — was built to fulfil a need. The need was to address the problem of lack of White Force resourcing (Chapter 1). To address this problem, a review of relevant and related literature and work was undertaken to examine existing methods and approaches for the multiplication of limited resources using automation, and to gain a better understanding of the types of tasks and processes White Force perform. Gaps were identified in the form of methods, techniques, approaches and selection criteria for the identification, communication, representation, classification and automation ranking for White Force tasks and processes/routines to be able to apply stability to the inherently complex and unstable domain of White Forces (Chapter 2).

After a review of related work was completed and gaps identified, a method was designed for the development and testing of a framework to address the five research questions based on the gaps of identification, communication, representation, classification and automation ranking (Chapter 3). Once a framework, that addressed the research questions and proposed criteria, was developed (Chapter 4), the framework was presented and validated with external subject-matter experts (SMEs) and members of Air Operations Simulation Centre at Defence Science and Technology Group, Fisherman's Bend (Chapter 5). The outcomes of this validation, in part, comprise the future work of aiming to address the needs of White Force automation.

The major contributions of this work are as follows:

- The DCARR model driven framework, which aims to identify, communicate, represent, classify and rank for automation candidate tasks and processes/routines found within large scale exercises/events/scenarios/etc.

- The Method for the Development and Evolution of the DCARR Framework

- The Method for the Testing and Validation of the DCARR Framework
- The proposed criteria + metrics used for the Development and Evolution / Testing and Validation of DCARR (Coverage, Alignment, Representation, Reproducibility, Dynamicism, Existing Theory and Effort).

Future work, as presented in this chapter, aims to address identified recommendations and limitations from the validation, as well as areas of application, evaluation, improvement/refinement, extensibility, scalability and generalisation.

6.3 DCARR Overview

DCARR — Describe, Collect, Analyse, Recommend and Report — is a model-driven framework (Figures 6.1) for the identification, communication, representation, classification and automation ranking of White Force tasks and processes/routines. It comprises 5 phases, where each phase has a different outcome. It is designed so that each phase feeds into the following phase, but allows for phase partial application given certain circumstances.

It is a novel approach to candidate identification for automation multiplication, and makes use of definitions and terminology to classify areas of complexity and non-determinism. It makes use of models and modelling (meta task representation) to assist with the visual communication of knowledge flow and task execution.

It follows a structured approach (separated into phases), to allow for reproducibility and is generalised enough for multi domain application. As it is the first developed iteration, it contains the basis for automation recommendations, with the hope that future work can extend automation solution recommendations based on Human Automation Interaction best practices.

Figure 6.1: DCARR framework phase diagram

The **Describe** phase, based on the need for identification, communication and representation, aims to identify and generate the following:

- *Scope of activities under review (Scenario and Objectives)*
- *Decomposition of tasks (Enough to be describable)*
- *Quick identification and selection of deterministic/appropriate tasks (and identification of nondeterministic/out of scope tasks)*
- *Plain language descriptions (to identify object, action, output and context of tasks)*
- *Formatting of tasks for analysis and automation evaluation*
- *Different representations of tasks for insight and review (state machines, DFA Quintuple and pseudocode, UML task models)*

The **Collect** phase, based on the need for identification, communication and representation, aims to:

- *Observe tasks within the scenario under review*

- *Collect data to confirm outputs and identify gaps from* **Describe** *phase*
- *Collect real world data to test and validate future analysis and evaluation.*

The **Analyse** phase, based on the need for identification, communication, representation and classification, aims to:

- *Take task descriptions/representations from the* **Describe** *phase and analyses them*
- *Uses outputs from* **Collect** *phase to confirm descriptions are accurate based on real world data*
- *Analysis looks at* ***resource usage, frequency, difficulty and impact***
- *Generates representations of necessary Orientations and Decisions based on OODA principles for relevant tasks that require further analysis than DFA, State Machines and Pseudocode*
- *Generates a quantitative and qualitative analysis of tasks to be used for evaluation in the* **Recommend** *phase.*

The **Recommend** phase, based on the need for selection of tasks for automation ranking, aims to:

- *Evaluate outputs from* **Analyse** *phase against attributes that can identify cost*
- *Provide a list of suggestions for automation opportunities based on desired outcome from framework application.*

The **Report** phase, based on the need for identification, communication, representation, classification and automation ranking, aims to:

- *Prepare outputs from* ***Describe, Collect, Analyse and Recommend*** *phases and generate a report*
- *Focus of the report should be on the:*
  - *Purpose of application of Framework*
  - *Scenario under review*
  - *Desired outcomes*
  - *List of outputs from* ***Recommend based on desired outcomes and evaluation rankings***

- *Included evidence to support recommendations.*

6.4 Answering Research Questions and Research Hypothesis

To answer our hypothesis of **How can we multiply White Force capabilities with automation?**, we need to address the following research questions:

1. **How can we systematically identify White Force tasks and processes/routines, including those that are hidden or ad-hoc, for further communication, representation, classification and investigation?** In the Describe phase, Hierarchical Task Analysis, in conjunction with data collection techniques from Human Factors research, and discussions amongst shareholders is used to generate a model that identifies the tasks and processes/routines. This identification is backed up, with data collection, including observations and timeline generations from Collect and Analyse phase to corroborate and validate assumptions made to identify hidden and ad hoc tasks.

2. **What approach can be used to adequately communicate White Force tasks and processes/routines to a wide and varied audience of automation implementors, to facilitate discussion and review for further task representation?** DCARR makes use of UML as a modelling technique to communicate not only the steps of DCARR, but the outcomes of tasks identification (HTA), task alignment (objective goal mapping) and task modelling (task representation).

3. **What method can be used to suitably capture and represent, in a structured and standardised way, the variables (i.e. information and decision-making) of White Force tasks and processes/routines, to assist with classification of tasks into suitable categories of automation?** As part of discussion and communication for generation of Describe outcomes, task models are built that aim to capture and represent the observations-orientations-decisions-actions (OODA) of tasks. These models are built using information collected from simple task questionnaires that aim to identify object, subject, tooling, process, outcomes of tasks identified from HTA and represented in UML.

4. **How can White Force tasks and processes/routines be acceptably classified into cognitively demanding tasks / tasks that resist automation due to instability and are chaotic/complex in nature, versus those that are repetitive, structured and rule based?** Cyenfin is a framework for the representation of domains of tasks that follow a structured/ordered vs unstructured/unordered nature and are classified into clear, complicated, complex and chaotic domains. By definition, being able to identify, communicate and represent a task, classifies it into an ordered domain and into a format that is suitable

for representation as a candidate task for automation ranking

5. **What selection criteria (e.g. repetitive, rule based, structured, consistent, and does not require a decision, interpretation or assessment of results) can be used to fairly rank and order candidates tasks for automation application?** Using Human Automation Interaction principles and guidelines, as well as Cyenfin theory and methods for navigating domains, and Levels of Automation including adaptive and adjustable automation approaches, into a set of selection criteria that focuses on the human operator as decision maker and accounts for situational awareness into planning and task execution.

Therefore, to answer **How can we multiply White Force capabilities with automation?** We can do so by applying structure to an otherwise chaotic and complex environment of White Force facilitated simulation training, through the identification, communication, representation, classification and ranking of candidate tasks and processes/routines for automation. To do this we can use HTA and EAST for identification, UML for communication, OODA and IF for representation, Cynefin for classification and LoA and HAI for automation ranking.

6.5 Example Application of DCARR

To put into perspective the steps/phases of DCARR, and how they could relate to a scenario (outside the scenarios used for development, evolution, evaluation and validation), the following is a short example application of DCARR, in relation to requirements and activities in a real world scenario.

6.5.1 Constraints and setup

For this example, Bob, our DCARR practitioner, has been invited by Fred, on behalf of his company "Fred + Co. Inc.", to provide insight into how Fred, can improve productivity at his company. "Fred + Co. Inc." is a small business that specialises in selling homemade soap. Fred has an online website where customers can place orders. Fred is notified by email that someone has placed an order, he then emails the customer and confirms the order (including billing address, contact information and payment details). Once he receives the order confirmation he will go to his bank and process payment. Once the payment has been confirmed, he will make the soap at his home, go to the post office and pay for packaging material, and ship it. The process from order being placed, to order being shipped, usually takes Fred 5 days.

6.5.2 Describe

From the constraints and setup, Bob has a basic understanding of how the business operates. Bob will meet with Fred, and capture the constraints and setup information as the **Scenario and Objectives** (i.e. Send soap to customer), identify the **overall task goal** to achieve the objective (i.e. Soap is sent to customer), and begin to generate a **Mapping between Objective and task goal**. As part of this initial identification, it is required that Fred be involved in the process as he is the stakeholder that contains the information required at this phase.

Bob, with the help of Fred as stakeholder and subject-matter expert, begins the next steps of task decomposition by identifying and confirming the **Sub task goals** (i.e. Order received notification, Order confirmation email, Order confirmed notification, Bank payment processing, Make soap, Post Office shipment). Each of these task goals get turned into **Task Descriptions**, based on the task description method that aims to decompose each task into observations, orientations, decisions and actions. Bob drives the discussion and information extraction based on the methods and steps of DCARR and by being limited to the information Fred can provide.

Once at this stage, Bob will leave Fred, and go **extract keywords and entities**, to **generate state machine observations** and begin the process of **generating OODA task models**. Bob, with task models and state machines, will identify proposed **value types for observations** and will contact Fred to **plan to observe and collect data**.

6.5.3 Collect

Bob with an understanding of the business process, will set up to **observe and collect data**. This will involve data collection methods and techniques, such as interviews, observations, video recordings, screen recordings, transcripts, documentation and records. Fred will go about his process and allow Bob to collect information for further evaluation. Once the process is complete, Bob will generate a **timeline of events and key decisions** and attempt to identify and place task occurrences. Bob will interview and clarify with Fred, where required, to make sure that each event and key decision is captured and assigned a task.

6.5.4 Analyse

With the data collected and broken down into events and key decisions, Bob will **review task models** from Describe phase and update them as required, based on the data collected. Then, he will **generate models for key decisions using the collected data**. These model instances will follow the timelines of events and key decisions from the Collect phase. Using the data from Collect, Bob will define the orientation and decision entities, and where necessary interview and question Fred to the structure/process/technique/approach used, where unknown.

Bob will confirm and validate with Fred that the models (alignment and task decomposition model, individual meta task models and timeline instanced task models) are correct. Once in agreement, Bob will continue the Analyse phase, and analyse tasks to look at and identify **resource usage, frequency, difficulty/complexity and impact**.

6.5.5 Recommend

Bob will contact Fred to ascertain how he hopes to "improve productivity" at his company. Bob will suggest that one approach would be to reduce the time taken to ship his homemade soaps, while another approach could be to increase the number of soaps shipped per 5 day block. This discussion will result in the **identification of the cost reduction attribute**. Once identified, after discussion with Fred, Bob will independently begin to **evaluate task analysis** by evaluating **cost, frequency, difficulty and impact** against the cost reduction attribute. Once each task has been analysed and evaluated, a ranked and ordered list of **recommendations based on evaluation** against desirable attributes can be generated.

6.5.6 Report

Once the recommendations have been generated, Bob will **report all findings**, in particular, to highlight these recommendations based on the evaluations and analysis conducted against the data collected. This report will contain the alignment and task decomposition model, individual meta task models, timeline instanced task models, and breakdown of analysis, evaluation and recommendations based on the desired cost reduction attribute.

6.6 Implications of DCARR

DCARR was developed as a framework to address the need for White Force multipliers, in the form of the selection of appropriate candidates for automation. DCARR aims to provide stability to an otherwise unstable environment. What has emerged from this research and work, is a systematic decision-making framework, based on 5 phases of Describe, Collect, Analyse, Recommend and Report

The underlying pillars that DCARR is built upon and aims to address, are identification, communication, representation, classification and automation ranking. Broadly speaking, these pillars can be applied and can be generalised to a wide variety of different areas, domains, scenarios, exercises, processes, procedures, etc. As an example, identification, communication, representation, classification and automation ranking, can be applied to process optimisation, organisational change management, procurement processes, and systematic decision-making in general.

This implication, that DCARR is widely applicable and generalised, means that other domains can be considered, such as self-driving cars, OSEMN[1] pipelines in data analytics, medical diagnosis, manufacturing, programming, project management, data processing pipelines, project management feasibility studies, machine learning training pipelines, processes and procedures in an applied research institute, and other domains of complexity requiring subject expertise.

[1] https://towardsdatascience.com/a-beginners-guide-to-the-data-science-pipeline-a4904b2d8ad3

Or equally so, domains and processes as simple as those found within an office, a shop, a supermarket or the home.

In its current incarnation and iteration, DCARR is a generalised tool to assist with structuring semi structured or ad hoc tasks and processes/routines of decision-making. With the included methods for the Development and Evolution and the Testing and Validation, and the proposed criteria + metrics, DCARR can be considered merely a tool, (such as a ratcheting screwdriver), that aims to assist people to achieve an automation ranking outcome, (ratcheting screwdriver to screw things in).

DCARR and the contributions of this work (Method for Development and Evolution, and Testing and Validation of the DCARR Framework, and the proposed criteria + metrics of Coverage, Alignment, Representation, Reproducibility, Dynamicism, Existing Theory and Effort), form the basis for a modular pipeline to do something. In this instance, it was used for the identification, communication, representation, classification and automation ranking of candidate tasks and processes/routines perform by White Force facilitated distributed synthetic training.

As demonstrated, DCARR is generalised and applicable to a wide domain and areas of future interest. Each component in DCARR can be swapped out to achieve a different goal, or the process be repeated (using different tooling, scenarios, domains, techniques, methods, selection criteria, related works) to develop something different that aims to achieve the same goal.

As presented, DCARR aimed to address the need for White Force multipliers, by identifying candidates for automation, to improve White Force capabilities. DCARR was developed, using the proposed methods, proposed criteria, proposed metrics, proposed scenarios and proposed areas of related work to achieve this aim. This work was validated with subject-matter experts, and was found to be of interest for further application, once cost and time estimations exist.

6.7 Future Work

Building upon the discussion of the implications of this work, a series of future work areas have been identified that aim to apply, evaluate, improve, refine, extend and scale the initial outcomes and contributions.

Application

DCARR, as presented, is a theoretical framework that contains limitations in the form of application estimations (time and cost). To address this limitation, it is necessary that DCARR be applied and tested, to develop estimations for application cost and time. To do this, DCARR needs to be applied to a series of different problems, that feature different levels of complexity/complicatedness.

This can range from a subset of tasks, found within the domain of White Force facilitated synthetic distributed Live, Virtual and Constructive training exercises. This application, would allow DCARR to be tested and evaluated, to allow for "errors" or "bugs" to be found and fixed, as well as generating data to be used for the generation of cost and time estimations.

This mixed variety of applicable domains for use with DCARR can also be used to ascertain the ratio of the volume/complexity of elements to the time and cost required (i.e. exponential growth of cost/time vs linear growth of cost/time based on scenario/exercise size and complexity). This could allow for discovery of appropriately sized exercises/scenarios/processes/routines for application of DCARR that make the cost and time investment beneficial given the proposed outcomes of ranked and ordered candidate tasks for automation.

Another area of application for DCARR, would be the identification of areas that add value. As part of applying DCARR, a measurement of the cost required and benefit added, through the application of each step of DCARR can be taken (e.g. as part of task identification and classification through task description, substantial value could be added by the identification areas of stability and determinism and by the identification of areas of instability and nondeterminism that require further human decision-making and resource allocation). The benefit of measuring and identifying areas of DCARR that add the most value is the ability to apply a subset of DCARR, given limited time/resource/cost allocation, that can still achieve and generate desirable outcomes. This will also allow for a quick and cost-effective application of DCARR. Further application, can be done to refine and reduce the described steps taken of DCARR, through the similar cost / benefit measurement and analysis, but instead targeting detail and information elicitation, to refine the methods and steps required, to achieve the highest cost / benefit ratio.

Evaluation

Once a set of estimations exist, that address the identified limitations, it is necessary to evaluate DCARR in the originally identified domain of White Forces facilitated synthetic distributed Live, Virtual and Constructive training, to evaluate the effectiveness and application of DCARR in addressing the desired aims and outcomes that DCARR has been created for.

Improvement/Refinement

DCARR is a first pass at addressing the identification, classification, representation, classification and automation ranking of candidate tasks and processes/routines. Improvements and refinements to DCARR, can take the form of changes made to the methods and techniques that DCARR relies upon. Currently, the outcomes of the Describe phase are focused around task decomposition, task representation and take modelling.

Current task identification methods are focused entirely around Hierarchical Task Analysis from Human Factors research. Future work could be done to address the inherent limitations and subjectivity of this method. Automated and semi-automated task identification approaches would greatly enhance the ease of use, and cost and time estimations for application.

One approach to semi-automated task identification would be through the use of model transformations and model transformations languages from Model Driven Engineering (MDE). Tasks and process could be generated using an abstracted modelling language, such as Business Process Modelling Notation (BPMN), and then transformed into low level task hierarchies.

Another area of improvement could be the inclusion of different methods for task representation. Currently, a simplistic form of human information processing, in the form of Observe-Orient-Decide-Act (OODA) loops are used as a meta representation of tasks. As identified as part of Information Fusion research, this model has been extended as a Cognitive model (C-OODA) and Modular model (M-OODA). Further work needs to be done to assess the implication and impact a refined task model would have on DCARR.

Outside of improvements and refinements to the methods and techniques that DCARR relies upon, is an update to the proposed Method for Development and Evolution of DCARR, Method for Testing and Validation of DCARR and proposed criteria and metrics used for the Development and Evolution / Testing and Validation. These criteria are Coverage, Alignment, Representation, Reproducibility, Dynamicism, Existing Theory, and Effort, with DCARR being built and developed to address these concerns. Future work, could look at the appropriateness of these measures, the inclusion of other measures, and the replacement of current measures.

Finally, as with software, evolution and development rarely cease. Therefore, another area of improvement, would be further updates and refinements to DCARR based on feedback, application and evaluation. These could take the form of applicability, usefulness, presentation, communication and overall simplification.

Extensibility

DCARR includes considerations for automation design, taken from Human Automation Interaction. These considerations are used as an initial step for ranking suitable tasks and processes for multiplication. Future work needs to be done to incorporate these considerations into a set of guidelines and rules to be used for automation driven design to address automation. These automation driven design solutions, could take the form of automation implementation guidelines and best practices. This could be extended further with the development of automation guidelines and best practices into patterns, designs and common reusable methods / frameworks / libraries / API's — similar to design patterns within software engineering (e.g. could a "good practice" solution be used independent of the domain or field?).

Scalability
During validation, it was noted of DCARR, that ironically while the tasks and processes that White Force perform in their duties as distributed simulation LVC training facilitators is intensive and exhaustive, what was being presented, appeared to further complicate and exhaust an already exhausted process — while claiming to reduce already limited resources by aiming to multiply capabilities.

Therefore, as part of application and evaluation, testing and validation needs to be conducted on the appropriateness and scalability of DCARR. This scalability relates to the cost / time / benefit ratio for a given a set of tasks and processes/routines. At what point is DCARR able to provide benefit versus the time and cost undertaken and invested (also, at what point is the limit of the input set DCARR can handle?).

Generalisation
DCARR was created and developed at an attempt to solve a resource allocation problem — limited White Force availability. The methods and components of DCARR have the potential to be suitable for a range of different domains but require further application, evaluation and improvements/refinements. These domains have been presented in the Implications of DCARR (Section 6.6).

6.8 Chapter Summary

This chapter provided an overview of the need for White Force capability multipliers to improve the limited resource allocation of White Force staffing. This was followed by the overview of each chapter within this work, and list of major contributions. DCARR was presented at a high level, and included an indication of the outcomes and tasks performed under each phase.

The research questions and research hypothesis that drove this research were addressed, with identified techniques/methods/approaches/frameworks/selection criteria from areas of related work that drove the identification (HTA and EAST), communication (UML), representation (IF and OODA), classification (Cynefin) and automation ranking (HAI and LOA) of candidate tasks and processes/routines.

An example scenario and descriptive approach of Bob, our DCARR guru, applying DCARR to small business owner Fred's shop "Fred + Co. Inc." to assist him with improving productivity for the ordering and shipping of his homemade soaps, was provided as an example application to demonstrate, in practice, the required actor interactions between stakeholders/SMEs and practitioner.

The implications of DCARR, as a generalised decision-making framework/tool, as well as example lateral applications of DCARR for a wide variety of domains of future interest was presented and discussed. As an initial tool to solve White Force capability multiplication through the use of identification of candidate tasks and processes/routines for automation, it can be argued that DCARR, built upon the pillars of identification, communication, representation, classification and automation ranking, can have its components swapped and tooling adjusted to fit nearly any conceived domain of decision-making. As a first step, it can achieve the originally designed goal, once estimations for time and cost are identified.

Finally, this chapter closed with the proposed areas of future work based on the contributions and outcomes from the validation of DCARR. The areas of future work revolve around the central consideration for the application of DCARR for the generation of time and cost estimations. This is the major focus of any future work related to DCARR. Other identified areas of future work relate to the evaluation, improvement and refinement of methods/techniques/components of DCARR, as well as testing the scalability of DCARR related to the dimensionality and size of exercises/scenarios/processes/routines that can be processed and the cost/time/benefit ratios of ideal domains for investigation.

Chapter 7

Conclusion

In this thesis, we have presented White Force as a critical and limited resource that needs capability multiplication (Chapter 1). Automation is purported to offer the benefit of consistency, alleviation of mundane activities and cognitive load on staff, and accelerate throughput. Automation is a common approach to scale up and facilitate capability multiplication. But it is not, a "one size fits all" solution or magic bullet. Automation requires stability (rule-based nature), rigidity (structured data), consistency (routine tasks) and frequency (repetitiveness), amongst other attributes. Therefore, to understand how we can multiply White Force capabilities with automation, it was necessary to understand how to identify, communicate, represent, classify and rank for automation the tasks and processes/routines that exist in the White Force synthetic training domain. This is because, to effectively apply automation, White Force tasks and processes/routines require a level of stability (Chapter 2).

To achieve this it was necessary to answer a set of Research Questions (Chapter 3).

1. **How can we systematically identify White Force tasks and processes/routines, including those that are hidden or ad-hoc, for further communication, representation, classification and investigation?** By systematically identifying and ensuring coverage of tasks (including ad-hoc and hidden), we aim to reduce and minimise miscommunication and/or misrepresentation.

2. **What approach can be used to adequately communicate White Force tasks and processes/routines to a wide and varied audience of automation implementors, to facilitate discussion and review for further task representation?** Communication of tasks is important for the need to share and communicate knowledge in a digestible format by the varied audience (including shareholders, stakeholders, and capability mul-

tipliers). Information needs to be understood by an intelligent but potentially non-expert / technical audience.

3. **What method can be used to suitably capture and represent, in a structured and standardised way, the variables (i.e. information and decision-making) of White Force tasks and processes/routines, to assist with classification of tasks into suitable categories of automation?** The need for representation, is a precursor to function / task automation design and implementation, with identified observations, orientations, decisions and actions, acting as input/output parameters and variables, algorithms, procedures and functions, and abstracted knowledge.

4. **How can White Force tasks and processes/routines be acceptably classified into cognitively demanding tasks / tasks that resist automation due to instability and are chaotic/complex in nature, versus those that are repetitive, structured and rule based?** The onus of task classification is due to the context of how automation will be implemented. Differing organisations may have different resourcing and requirements available for automation multiplication, and may be subject to external or internal influences regarding the desired classification application

5. **What selection criteria (e.g. repetitive, rule based, structured, consistent, and does not require a decision, interpretation or assessment of results) can be used to fairly rank and order candidates tasks for automation application?** The selection criteria can be considered arbitrary. This is based on the needs and requirements of who is assessing it, the desired outcomes of application (reduction of resource usage or multiplication of capability, etc.), and internal and external factors (resource allocation, cost, time, shareholder buy in, etc.). This is because there is no "one size fits all" solution or magic bullet for automation.

To address the above questions into a consistent and repeatable framework, it was necessary to use an exploratory sequential mixed methods approach, that encompassed two methods (i) Methodology for Framework Construction, and (ii) Methodology for Testing and Evaluation (Section 3.3). As part of these methods, a set of criteria (Section 3.3.2) were developed in consultation with subject-matter-experts at Defence Science and Technology (DST) Group, to be used to shape and form the construction of the proposed framework. From these criteria, a set of metrics and scenarios were developed for testing and evaluation of the proposed framework (Section 3.3.6).

The outcome of this methodology was the creation of the Describe, Collect, Analyse, Recommend, and Report (DCARR) framework (Chapter 4). DCARR (Section 4.2) aims to apply stability to White Force tasks and processes/routines, through the identification of current White Force activities (including the hidden and ad hoc tasks and processes/routines). From this identification, DCARR can be used to model and represent (to communicate) the decision-making and required variables (observations, action outcomes, prior knowledge, expertise, experience, etc.). From this model of task functionality, DCARR can be used to classify tasks into nominal groups of difficulty or complexity, and separate tasks into those that are suitable for automation vs those that required human decision-making and judgement. Once classified, DCARR can be used to assess and apply automation best practice/principles (human automation interaction guidelines, levels of automation, robotic process automation, etc.) to the selected tasks to generate a ranked and ordered list of candidates of automation for capability multiplication.

Once developed, DCARR was validated through the presentation to members of White Force and researchers from Air Operations Simulation Centre (AOSC) and Human Factors Research at DST Group (Chapter 5). DCARR, is described as a *"Structured and repeatable approach to collecting and integrating subjective and objective evidence to deliver a prioritised list of recommendations that target automation"*, and was regarded by participants, as a useful outcome and as the first step in achieving the initial desire of having a model of White Force tasks and processes/routines.

Therefore, to answer to our Research Questions (Chapter 6), DCARR can be used in the following way:

1. **How can we systematically identify White Force tasks and processes/routines, including those that are hidden or ad-hoc, for further communication, representation, classification and investigation?** In the Describe phase, Hierarchical Task Analysis, in conjunction with data collection techniques from Human Factors research, and discussions amongst shareholders is used to generate a model that identifies the tasks and processes/routines. This identification is backed up, with data collection, including observations and timeline generations from Collect and Analyse phase to corroborate and validate assumptions made to identify hidden and ad hoc tasks.

2. **What approach can be used to adequately communicate White Force tasks and processes/routines to a wide and varied audience of automation implementors, to facilitate discussion and review for further task representation?** DCARR makes use of UML as a modelling technique to communicate not only the steps of DCARR, but the outcomes of tasks identification (HTA), task alignment (objective goal mapping) and task modelling (task representation).

3. **What method can be used to suitably capture and represent, in a structured and**

standardised way, the variables (i.e. information and decision-making) of White Force tasks and processes/routines, to assist with classification of tasks into suitable categories of automation? As part of discussion and communication for generation of Describe outcomes, task models are built that aim to capture and represent the observations-orientations-decisions-actions (OODA) of tasks. These models are built using information collected from simple task questionnaires that aim to identify object, subject, tooling, process, outcomes of tasks identified from HTA and represented in UML.

4. **How can White Force tasks and processes/routines be acceptably classified into cognitively demanding tasks / tasks that resist automation due to instability and are chaotic/complex in nature, versus those that are repetitive, structured and rule based?** Cyenfin is a framework for the representation of domains of tasks that follow a structured/ordered vs unstructured/unordered nature and are classified into clear, complicated, complex and chaotic domains. By definition, being able to identify, communicate and represent a task, classifies it into an ordered domain and into a format that is suitable for representation as a candidate task for automation ranking

5. **What selection criteria (e.g. repetitive, rule based, structured, consistent, and does not require a decision, interpretation or assessment of results) can be used to fairly rank and order candidates tasks for automation application?** Using Human Automation Interaction principles and guidelines, as well as Cyenfin theory and methods for navigating domains, and Levels of Automation including adaptive and adjustable automation approaches, into a set of selection criteria that focuses on the human operator as decision maker and accounts for situational awareness into planning and task execution.

The implications of DCARR, as a generalised decision-making framework/tool for a wide variety of domains of future interest was presented and discussed (Chapter 6). As an initial tool to solve White Force capability multiplication, it is beneficial, but it can be argued that DCARR can have its components swapped and tooling adjusted to fit nearly any conceived domain of decision-making. As a first step, it can achieve the originally designed goal, once proposed areas of future work, based on the central consideration for the application of DCARR, for the generation of time and cost estimations are addressed.

In summary, this work aimed to address the overarching need for White Force multipliers. To do this, gaps were identified in literature to address the identification, communication, representation, classification and automation ranking of White Force tasks and processes/routines. This is due to the inherent instability found within the domain. Therefore, a set of research questions were proposed and an exploratory sequential mixed methods approach used to develop the major contributions of this work:

- The DCARR model driven framework, which aims to identify, communicate, represent, classify and rank for automation candidate tasks and processes/routines found within large scale exercises/events/scenarios/etc.

- The Method for the Development and Evolution of the DCARR Framework

- The Method for the Testing and Validation of the DCARR Framework

- The proposed criteria + metrics used for the Development and Evolution / Testing and Validation of DCARR (Coverage, Alignment, Representation, Reproducibility, Dynamicism, Existing Theory and Effort).

These outcomes were validated with SME's at AOSC, DST Group, as an initial attempt to address White Force capability multiplication, through the identification, communication, representation, classification and automation ranking of White Force tasks and processes/routines. Due to the approach used to generate these contributions, DCARR can be considered as a generalised decision-making framework, applicable to a wide variety of domains. Future work needs to address application of DCARR, for the generation of time and cost estimations.

Thank you

S.Vajda

References

[1] MilSkil, "Mass Air Brief Virtual Pitch Black 2018," Air Warfare Centre, Tech. Rep., 2018.

[2] R. Hubal and T. Parsons, "Synthetic environments for skills training and practice," *The Sciences of Learning and Instructional Design: Constructive Articulation Between Communities*, no. July, pp. 152–185, 2017.

[3] USD(AT&L), "DoD Modeling and Simulation (M&S) Glossary," *DTIC Document*, 1998. [Online]. Available: http://acqnotes.com/attachments/dod5000.59-mdodmodelingandsimulationglossary.pdf

[4] US DoD M&S Coordination Office, "US DoD M&S Glossary," Modeling and Simulation Coordination Office, Tech. Rep., 2011.

[5] G. Simpkin, P. Ross, and B. Macpherson, "On the Need for White Force Multipliers," *NATO Modelling & Simulation Group (NMSG) Multi-Workshop*, no. Figure 1, pp. 1–15, 2013. [Online]. Available: http://ftp.rta.nato.int/public/PubFullText/RTO/MP/STO-MP-MSG-111/MP-MSG-111-20.pdf

[6] Royal Australian Air Force, "Plan Jericho - Program of Work," Royal Australian Air Force, Tech. Rep., 2015. [Online]. Available: http://airpower.airforce.gov.au/Publications/Plan-Jericho-Program-of-Workhttp://airpower.airforce.gov.au/APDC/media/PDF-Files/AirForcePublications/AF18-Plan-Jericho-Program-of-Work.pdf

[7] DST Group, "Shaping Defence Science and Technology in the Aerospace Domain: 2017 - 2027," Royal Australian Air Force, Tech. Rep., 2017. [Online]. Available: https://www.dst.defence.gov.au/publication/shaping-defence-science-and-technology-aerospace-domain-2017-2027https://www.dst.defence.gov.au/sites/default/files/publications/documents/DSC{_}1697-Aerospace{_}Domain{_}S{%}26T{_}Strategy{_}PRO2{_}LR.pdf

[8] H. Dudfield and J. Kearse, "Managing distributed training events–lessons learned from the mission training through distributed simulation capability concept demonstrator," *Proceedings of IITSEC*, 2009.

[9] S. E. Gehr, M. Schurig, L. Jacobs, J. van der Pal, W. Bennett Jr, and B. Schreiber,

Assessing the training potential of MTDS in Exercise First WAVE. Neuilly-sur-Seine: NATO RTO, 2005.

[10] P. Crane, W. Bennett Jr, M. France, E. Smith, R. Anderson, J. Borgvall, M. Castor, N. Lagerback, P. L. Robbie, C. Best *et al.*, “Preparing for large-force exercises with distributed simulation: A panel presentation,” AIR FORCE RESEARCH LAB MESA AZ 711 HUMAN PERFORMANCE WING, Tech. Rep., 2010.

[11] R. L. Rusing, “Prepare the fighter force-red flag/composite force,” ARMY COMMAND AND GENERAL STAFF COLL FORT LEAVENWORTH KS, Tech. Rep., 1980.

[12] G. Simpkin, B. Macpherson, P. Ross, M. Vukovic, and J. Parker, “Preliminary White Force Task Analysis Extract from Exercise Black Skies 2016 White Force Observations.” [Unpublished Presentation], 2016.

[13] P. Ryan, P. Ross, W. Oliver, and L. Zalcman, “STATUS OF ADVANCED DISTRIBUTED SIMULATION ARCHITECTURES,” Air Operations Division, Defence Science & Technology Organisation (DSTO), Tech. Rep., 2015.

[14] S. Abrams, “TENA As An Enabler For Joint Combined Training,” *SimTecT 2008*, vol. 07, 2008.

[15] P. Ryan and L. Zalcman, “The DIS vs HLA Debate : What’s in it for Australia? What is Distributed Simulation?” *Proceedings of SimTect 2003*, 2003.

[16] O. Topçu and H. Oğuztüzün, *Guide to Distributed Simulation with HLA*, L. G. Birta, Ed. Springer, 2017. [Online]. Available: http://link.springer.com/10.1007/978-3-319-61267-6

[17] O. Topçu, U. Durak, H. Oguztüzün, and L. Yilzman, *Distributed simulation - A model driven engineering approach*, L. G. Birta, Ed. Springer, 2016.

[18] I. Petz, “Communication protocols in distributed simulation.” *SCYR 2010 - 10th Scientific Conference of Young Researchers*, 2010.

[19] P. Ryan, P. Ross, and W. Oliver, “High fidelity simulation using the updated distributed interactive simulation standard,” *International Journal of Intelligent Defence Support Systems*, vol. 5, no. 2, pp. 109–126, 2014.

[20] M. Mccall and B. Murray, “IEEE 1278 Distributed Interactive Simulation (DIS),” SISO, Tech. Rep. May, 2010.

[21] L. Zalcman, P. Ross, and J. Blacklock, “Link-16 interoperability for air combat officer training at raaf williamtown and raaf sale using the adgesim and awsim trainers,” in *SimTecT2010 Conference, Brisbane, Australia*, 2010.

[22] C. Shanahan, C. Best, M. Finch, A. Stott, E. Tracey, and S. Hasenbosch, “Exercise black skies 2008 : Enhancing live training through virtual preparation Part one : An evaluation

of training effectiveness," DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION VICTORIA (AUSTRALIA), Tech. Rep., 2008.

[23] E. Tracey, S. Hasenbosch, J. Vince, D. Pope, A. Stott, C. Best, C. Shanahan, and M. Finch, "Exercise Black Skies 2008 : Enhancing live training through virtual preparation Part Two : An evaluation of Tools and Techniques," DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION VICTORIA (AUSTRALIA), Tech. Rep., 2008.

[24] C. Best, C. Eidman, P. Crane, C. Kam, M. Skinner, S. Hasenbosch, E. Burchat, M. Finch, C. Shanahan, and M. Zamba, "Exercise Pacific Link 2 : Distributed Training for Air Battle Managers," *Twelfth Australian Aeronautical Conference*, no. 7104, 2007.

[25] C. Best, S. Hasenbosch, M. Skinner, P. Crane, E. Burchat, S. E. Gehr, C. Kam, C. Shanahan, and M. Zamba, "Coalition Distributed Mission Training : Exercise Pacific Link 2," *AIAC12 – Twelfth Australian International Aerospace Congress*, no. March, pp. 1–17, 2007.

[26] J. Blacklock and L. Zalcman, "The virtual air environment, interim training capability," *DSTO Client Report*, no. DSTO-CR-0279, 2003.

[27] J. Sorroche, "Tactical digital information link-technical advice and lexicon for enabling simulation (tadil-tales)," in *10th International Command and Control Research and Technology Symposium The Future of C2*, 2005.

[28] M. K. And and J. Mackett, "Moving Link 16 to the Tactical Edge," *Anzidlsoc2016*, no. November, 2016. [Online]. Available: http://www.anzidlsoc.com.au/11.Link16atTacticalEdgeNov72016Viasat.pdf

[29] T. Kühne, "Matters of (meta-) modeling," *Software and Systems Modeling*, vol. 5, no. 4, pp. 369–385, 2006.

[30] C. Atkinson and T. Kühne, "Model-driven development: A metamodeling foundation," *IEEE Software*, vol. 20, no. 5, pp. 36–41, 2003.

[31] B. Selic, "What will it take? A view on adoption of model-based methods in practice," *Software and Systems Modeling*, vol. 11, no. 4, pp. 513–526, 2012.

[32] D. Moody, "The physics of notations: Toward a scientific basis for constructing visual notations in software engineering," *IEEE Transactions on Software Engineering*, vol. 35, no. 6, pp. 756–779, 2009.

[33] O. M. Group, "Uml 2.4.1 superstructure specification," Object Management Group, Tech. Rep., 2011.

[34] IEEE Computer Society, *IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process Multi-Architecture Overlay (DMAO)*. IEEE, 2013, no. January. [Online]. Available: http://ieeexplore.ieee.org/servlet/opac?punumber=6654217

[35] M. Belaunde, C. Casanave, D. DSouza, K. Duddy, W. El Kaim, A. Kennedy, W. Frank, D. Frankel, R. Hauch, and S. Hendryx, "MDA Guide Version 1.0," *Object Management Group*, vol. 234, no. May, p. 51, 2003.

[36] O. M. Group, "Meta Object Facility (MOF) Specification v 2.4.1," Object Management Group, Tech. Rep., 2011.

[37] ——, "Meta-modeling and the omg meta object facility (mof)," Object Management Group, Tech. Rep., 2017.

[38] J. Z. Pan, S. Staab, U. Aßmann, J. Ebert, and Y. Zhao, "Ontology-driven software development," *Ontology-Driven Software Development*, pp. 1–337, 2013.

[39] J. Sprinkle, B. Rumpe, H. Vangheluwe, and G. Karsai, "Metamodelling: state of the art and research challenges," *Model-Based Engineering of Embedded Real-Time Systems, Lecture Notes in Computer Science, 6100*, vol. 6100, pp. 57–76, 2010. [Online]. Available: http://www.springerlink.com/content/e16233n5mvq50r1v/

[40] J. W. Ko, K. Y. Chung, and J. S. Han, "Model transformation verification using similarity and graph comparison algorithm," *Multimedia Tools and Applications*, vol. 74, no. 20, pp. 8907–8920, 2015.

[41] Y. Z. Y. Zhao, Y. F. Y. Fan, X. B. X. Bai, Y. W. Y. Wang, H. C. H. Cai, and W. D. W. Ding, "Towards formal verification of UML diagrams based on graph transformation," *IEEE International Conference on E-Commerce Technology for Dynamic E-Business*, 2004.

[42] V. Zue and J. Glass, "Conversational interfaces: advances and challenges," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1166–1180, 2000. [Online]. Available: http://ieeexplore.ieee.org/document/880078/

[43] O. M. Group, "Meta object facility (mof) 2.0 query/view/transformation specification," Object Management Group, Tech. Rep., 2008.

[44] P. Luyckx, P. Hellinckx, and H. Vangheluwe, "Model Driven Engineering for Building Automation, a Research Agenda," *2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pp. 485–491, 2013. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6681277

[45] T. Vepsäläinen and S. Kuikka, "Model-Driven Development of Automation and Control Applications: Modeling and Simulation of Control Sequences," *Advances in Software Engineering*, vol. 2014, pp. 1–14, 2014. [Online]. Available: https://www.hindawi.com/archive/2014/470201/

[46] S. Holzmuller-Laue, B. Gode, and K. Thurow, "Model-driven complex workflow automation for laboratories," *IEEE International Conference on Automation Science and Engineering*, pp. 758–763, 2013.

[47] C. Buckl, D. Sojer, and A. Knoll, “FTOS: Model-driven development of fault-tolerant automation systems,” *Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2010*, 2010.

[48] M. Maurmaier, “Leveraging model-driven development for automation systems development,” *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 733–737, 2008. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4638481

[49] X. Qafmolla, “Automation of Web Services Development Using Model Driven Techniques,” *Architecture*, vol. 3, pp. 190–194, 2010. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=05452059

[50] M. L. Alvarez, I. Sarachaga, A. Burgos, E. Estevez, and M. Marcos, “A Methodological Approach to Model-Driven Design and Development of Automation Systems,” *IEEE Transactions on Automation Science and Engineering*, pp. 1–13, 2016.

[51] J. Diaz, J. Perez, C. Fernandez-Sanchez, and J. Garbajosa, “Model-to-code transformation from product-line architecture models to AspectJ,” *Proceedings - 39th Euromicro Conference Series on Software Engineering and Advanced Applications, SEAA 2013*, pp. 98–105, 2013.

[52] S. De Cesare, D. Juric, and M. Lycett, “Toward the automation of business process ontology generation,” *Proceedings - 16th IEEE Conference on Business Informatics, CBI 2014*, vol. 1, pp. 70–77, 2014.

[53] A. S. de Oliveira, A. Charfi, B. Schmeling, and G. Serme, “A model-driven approach for accountability in business processes,” *Lecture Notes in Business Information Processing*, vol. 175 LNBIP, no. January 2009, pp. 184–199, 2014.

[54] A. Fouad, K. Phalp, J. M. Kanyaru, and S. Jeary, “Embedding requirements within Model-Driven Architecture,” *Software Quality Journal*, vol. 19, no. 2, pp. 411–430, 2011.

[55] A. Rodrigues Da Silva, “Model-driven engineering: A survey supported by the unified conceptual model,” *Computer Languages, Systems and Structures*, vol. 43, pp. 139–155, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.cl.2015.06.001

[56] O. M. Group, “Business process definition metamodel volume ii: Process definitions,” Object Management Group, Tech. Rep., 2008.

[57] J. Freund, D. Gagné, F. Menge, M. Kloppmann, S. Nijssen, G. Navarro-Suarez, I. Trickovic, and S. A. White, “BPMN 2.0 by Example,” Object Management Group, Tech. Rep. June, 2010. [Online]. Available: http://www.omg.org/spec/BPMN/2.0/examples/PDF/

[58] M. Owen and J. Raj, “Bpmn and business process management, introduction to the new business process modeling standard,” Popkin Software, Tech. Rep., 2003.

[59] M. Chinosi and A. Trombetta, "BPMN: An introduction to the standard," *Computer Standards and Interfaces*, vol. 34, no. 1, pp. 124–134, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.csi.2011.06.002

[60] M. Kunze and M. Weske, *Behavioural models: From modelling finite automata to analysing business processes*. Springer, 2016.

[61] J. Cardoso, "Business Process Control-Flow Complexity," *International Journal of Web Services Research*, vol. 5, no. June, pp. 49–76, 2008.

[62] R. Martinho, R. Rijo, and A. Nunes, "Complexity Analysis of a Business Process Automation: Case Study on a Healthcare Organization," *Procedia Computer Science*, vol. 64, pp. 1226–1231, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2015.08.510

[63] E. Rolón, J. Cardoso, F. García, F. Ruiz, and M. Piattini, "Analysis and validation of control-flow complexity measures with BPMN process models," *Lecture Notes in Business Information Processing*, vol. 29 LNBIP, no. 4, pp. 58–70, 2009.

[64] M. Vagia, A. A. Transeth, and S. A. Fjerdingen, "A literature review on the levels of automation during the years. what are the different taxonomies that have been proposed?" *Applied ergonomics*, vol. 53, pp. 190–202, 2016.

[65] A. Krijnen, *The Toyota way: 14 management principles from the world's greatest manufacturer*. Taylor & Francis, 2007.

[66] R. Shah and P. T. Ward, "Lean manufacturing: Context, practice bundles, and performance," *Journal of Operations Management*, vol. 21, no. 2, pp. 129–149, 2003.

[67] J. F. Krafcik, "Triumph of the lean production system," *Sloan management review*, vol. 30, no. 1, pp. 41–52, 1988.

[68] P. S. Pande, R. P. Neuman, and R. R. Cavanagh, *The Six Sigma Way: How GE, Motorola, and Other Top Companies are Honing Their Performance*. McGraw-Hill, 2000, vol. 34.

[69] E. Lander and J. K. Liker, "The toyota production system and art: making highly customized and creative products the toyota way," *International journal of production research*, vol. 45, no. 16, pp. 3681–3698, 2007.

[70] J. Antony, R. Snee, and R. Hoerl, "Lean Six Sigma: yesterday, today and tomorrow," *International Journal of Quality & Reliability Management*, vol. 34, no. 7, pp. 1073–1093, 2017. [Online]. Available: http://www.emeraldinsight.com/doi/10.1108/IJQRM-03-2016-0035

[71] A. Laureani and J. Antony, "Leadership characteristics for Lean Six Sigma," *Total Quality Management and Business Excellence*, vol. 28, no. 3-4, pp. 405–426, 2017.

[72] N. Moray, T. Inagaki, and M. Itoh, "Adaptive automation, trust, and self-confidence in

fault management of time-critical tasks." *Journal of experimental psychology: Applied*, vol. 6, no. 1, p. 44, 2000.

[73] R. Parasuraman, R. Molloy, and I. L. Singh, "Performance consequences of automation-induced 'complacency'," *The International Journal of Aviation Psychology*, vol. 3, no. 1, pp. 1–23, 1993.

[74] M. W. Scerbo, "Theoretical perspectives on adaptive automation," in *Automation and Human Performance: Theory and Applications*. CRC Press, 1996.

[75] R. Parasuraman, S. Galster, P. Squire, H. Furukawa, and C. Miller, "A flexible delegation-type interface enhances system performance in human supervision of multiple robots: Empirical studies with roboflag," *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.*, vol. 35, no. 4, pp. 481–493, 2005.

[76] C. M. Burns, "Automation and the Human Factors Race to Catch Up," *Journal of Cognitive Engineering and Decision Making*, vol. 12, no. 1, pp. 83–85, 2018.

[77] T. B. Sheridan and R. Parasuraman, "Human-Automation Interaction," *Reviews of Human Factors and Ergonomics*, vol. 1, no. 1, pp. 89–129, 2005. [Online]. Available: http://journals.sagepub.com/doi/10.1518/155723405783703082

[78] P. J. Smith, "Conceptual Frameworks to Guide Design," *Journal of Cognitive Engineering and Decision Making*, vol. 12, no. 1, pp. 50–52, 2018.

[79] G. A. Jamieson and G. Skraaning, "Levels of Automation in Human Factors Models for Automation Design: Why We Might Consider Throwing the Baby Out With the Bathwater," *Journal of Cognitive Engineering and Decision Making*, vol. 12, no. 1, pp. 42–49, 2018.

[80] D. B. Kaber and M. R. Endsley, "The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task," *Theoretical issues in ergonomics science*, vol. 5, no. 2, pp. 113–153, 2004.

[81] M. Johnson, J. M. Bradshaw, and P. J. Feltovich, "Tomorrow's Human–Machine Design Tools: From Levels of Automation to Interdependencies," *Journal of Cognitive Engineering and Decision Making*, vol. 12, no. 1, pp. 77–82, 2018.

[82] T. Inagaki and T. B. Sheridan, "A critique of the SAE conditional driving automation definition, and analyses of options for improvement," *Cognition, Technology and Work*, no. Sae 2016, pp. 1–10, 2018. [Online]. Available: https://doi.org/10.1007/s10111-018-0471-5

[83] S. International, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," SAE International Warrendale, PA, Tech. Rep., 2016.

[84] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "A Model for Types and Levels of Human Interaction," *IEEE Transactions on Systems Man and Cybernetics Part A*

Systems and Humans, vol. 30, no. 3, pp. 286–297, 2000.

[85] D. B. Kaber, "Issues in Human–Automation Interaction Modeling: Presumptive Aspects of Frameworks of Types and Levels of Automation," *Journal of Cognitive Engineering and Decision Making*, vol. 12, no. 1, pp. 7–24, 2018.

[86] M. R. Endsley, "From Here to Autonomy: Lessons Learned from Human-Automation Research," *Human Factors*, vol. 59, no. 1, pp. 5–27, 2017.

[87] J. D. Lee and K. A. See, "Trust in Automation: Designing for Appropriate Reliance," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 46, no. 1, pp. 50–80, 2004. [Online]. Available: http://hfs.sagepub.com/cgi/doi/10.1518/hfes.46.1.50{_}30392

[88] R. Parasuraman and D. H. Manzey, "Complacency and bias in human use of automation: An attentional integration," *Human Factors*, vol. 52, no. 3, pp. 381–410, 2010.

[89] R. E. Patterson, "Intuitive Cognition and Models of Human-Automation Interaction," *Human Factors*, vol. 59, no. 1, pp. 101–115, 2017.

[90] R. Parasuraman, "Designing automation for human use: Empirical studies and quantitative models," *Ergonomics*, vol. 43, no. 7, pp. 931–951, 2000.

[91] D. Manzey, J. Reichenbach, and L. Onnasch, "Human Performance Consequences of Automated Decision Aids: The Impact of Degree of Automation and System Experience," *Journal of Cognitive Engineering and Decision Making*, vol. 6, no. 1, pp. 57–87, 2012.

[92] L. Onnasch, C. D. Wickens, H. Li, and D. Manzey, "Human performance consequences of stages and levels of automation: An integrated meta-analysis," *Human Factors*, vol. 56, no. 3, pp. 476–488, 2014.

[93] M. R. Endsley, "Level of Automation Forms a Key Aspect of Autonomy Design," *Journal of Cognitive Engineering and Decision Making*, vol. 12, no. 1, pp. 29–34, 2018.

[94] M. W. Scerbo and M. Mouloua, "Automation technology and human performa. current research and trends," Old Dominion University Research Foundation and Office of Naval Research, Tech. Rep., 1998.

[95] E. J. de Visser, R. Pak, and T. H. Shaw, "From 'automation' to 'autonomy': the importance of trust repair in human–machine interaction," *Ergonomics*, vol. 0139, pp. 1–19, 2018. [Online]. Available: https://doi.org/10.1080/00140139.2018.1457725

[96] C. Wickens, "Automation Stages & Levels, 20 Years After," *Journal of Cognitive Engineering and Decision Making*, vol. 12, no. 1, pp. 35–41, 2018.

[97] M. R. Endsley, "Automation and Situation Awareness," *Automation and Human Performance: Theory and Applications*, pp. 163–181, 1996.

[98] R. Parasuraman and D. H. Manzey, "Complacency and bias in human use of automation: An attentional integration," *Human Factors*, vol. 52, no. 3, pp. 381–410, 2010.

[99] A. Chavaillaz, D. Wastell, and J. Sauer, "System reliability, performance and trust in adaptable automation," *Applied Ergonomics*, vol. 52, pp. 333–342, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.apergo.2015.07.012

[100] J. Sauer, P. Nickel, and D. Wastell, "Designing automation for complex work environments under different levels of stress," *Applied Ergonomics*, vol. 44, no. 1, pp. 119–127, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.apergo.2012.05.008

[101] G. Simpkin, P. Ross, and B. Macpherson, "On the Need for White Force Multipliers," *NATO Modelling & Simulation Group (NMSG) Multi-Workshop*, no. Figure 1, pp. 1–15, 2013. [Online]. Available: http://ftp.rta.nato.int/public/PubFullText/RTO/MP/STO-MP-MSG-111/MP-MSG-111-20.pdf

[102] D. J. D. Snowden and M. M. E. Boone, "A leader's framework for decision making," *Harvard Business Review*, vol. 85, no. 11, p. 68, 2007. [Online]. Available: http://aacu-secure.nisgroup.com/meetings/ild/documents/Symonette.MakeAssessmentWork.ALeadersFramework.pdf{%}5Cnhttp://www.amazon.com/dp/0691084424

[103] M. Couture, "Complexity and Chaos - State-of-the-Art; List of Works, Experts, Organizations, Projects, Journals, Conferences and Tools," *Defence R&D Canada —- Valcartier*, p. 366, 2007.

[104] R. V. O'Connor and M. Lepmets, "Exploring the use of the cynefin framework to inform software development approach decisions," *International Conference on Software and Systems Process, ICSSP 2015*, vol. 24-26-Augu, pp. 97–101, 2015. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84958538253{&}partnerID=40{&}md5=1fb88559409f9b6203cef81aaf07fab7

[105] E. K. Van Beurden, A. M. Kia, A. Zask, U. Dietrich, and L. Rose, "Making sense in a complex landscape: How the cynefin framework from complex adaptive systems theory can inform health promotion practice," *Health Promotion International*, vol. 28, no. 1, pp. 73–83, 2013.

[106] J. Vollmar, M. Gepp, H. Palm, and A. Cala, "Engineering framework for the future: Cynefin for engineers," *2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings*, 2017.

[107] N. Stanton, P. M. Salmon, and L. A. Rafferty, *Human factors methods: a practical guide for engineering and design*. Ashgate Publishing, Ltd., 2013.

[108] N. A. Stanton, A. Hedge, K. Brookhuis, E. Salas, and H. W. Hendrick, *Handbook of human factors and ergonomics methods*. CRC press, 2004.

[109] N. Stanton, C. Baber, and D. Harris, *Modelling command and control: Event analysis of*

systemic teamwork. Ashgate Publishing, Ltd., 2008.

[110] N. A. D. Stanton, P. D. Salmon, and G. H. D. Walker, *Systems thinking in practice: applications of the event analysis of systemic teamwork method*. CRC Press, 2018.

[111] N. A. Stanton, G. H. Walker, D. Jenkins, M. S. Young, R. Stewart, and L. Wells, "A human factors approach to analysing military command and control," in *11th ICCRTS (International Command and Control Research and Technology Symposium)*, 2010.

[112] E. Blasch and D. A. Lambert, *High-level information fusion management and systems design*. Artech House, 2012.

[113] ——, "High-level information fusion management and systems design," *IEEE Aerospace and Electronic Systems Magazine*, vol. 27, no. 9, pp. 4–20, 2012.

[114] D. Lambert and C. Nowak, "The mephisto conceptual framework," DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION EDINBURGH (AUSTRALIA) COMMAND ..., Tech. Rep., 2008.

[115] R. Breton and R. Rousseau, "The c-ooda: A cognitive version of the ooda loop to represent c2 activities," in *Proceedings of the 10th International Command and Control Research Technology Symposium*, 2005.

[116] ——, "The m-ooda: Modelling of the ooda loop as a modular functional system," *Def. Res. and Dev. CA-Valcartier, DRDC TM*, vol. 130, 2008.

[117] S. E. McIntosh, "The wingman-philosopher of mig alley: John boyd and the ooda loop," *Air Power History*, vol. 58, no. 4, pp. 24–33, 2011.

[118] B. Brehmer, "The dynamic ooda loop: Amalgamating boyd's ooda loop and the cybernetic approach to command and control," in *Proceedings of the 10th international command and control research technology symposium*. Citeseer, 2005, pp. 365–368.

[119] E. P. Blasch, R. Breton, P. Valin, and E. Bosse, "User information fusion decision making analysis with the C-OODA model," *Fusion 2011 - 14th International Conference on Information Fusion*, 2011.

[120] E. P. Blasch, R. Breton, and P. Valin, "Using the C-OODA model for CIMIC analysis," *National Aerospace and Electronics Conference, Proceedings of the IEEE*, pp. 130–138, 2011.

[121] R. Breton and R. Rousseau, "The C-OODA: a Cognitive Version of the Ooda Loop To Represent C 2 Activities [ppt presentation]," *Proceedings of the 10th International Command and Control Research Technology Symposium*, p. 25 pp., 2005.

[122] T. B. Sheridan and R. Parasuraman, "Human-automation interaction," *Reviews of Human Factors and Ergonomics*, vol. 1, no. 1, pp. 89–129, 2005. [Online]. Available: https://doi.org/10.1518/155723405783703082

[123] R. Breton and É. Bossé, "The cognitive costs and benefits of automation," DEFENCE RESEARCH AND DEVELOPMENT CANADAVALCARTIER (QUEBEC), Tech. Rep., 2003.

[124] E. P. Blasch, R. Breton, and P. Valin, "Using the c-ooda model for cimic analysis," in *Proceedings of the 2011 IEEE National Aerospace and Electronics Conference (NAECON)*. IEEE, 2011, pp. 130–138.

[125] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "A model for types and levels of human interaction with automation," *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans*, vol. 30, no. 3, pp. 286–297, 2000.

[126] M. R. Endsley, B. Bolté, and D. G. Jones, *Designing for situation awareness: An approach to user-centered design*. CRC press, 2003.

[127] T. Kühne, "Matters of (meta-) modeling," *Software & Systems Modeling*, vol. 5, no. 4, pp. 369–385, 2006.

[128] C. Atkinson and T. Kuhne, "Model-driven development: a metamodeling foundation," *IEEE software*, vol. 20, no. 5, pp. 36–41, 2003.

[129] D. J. Snowden and M. E. Boone, "A leader's framework for decision making," *Harvard business review*, vol. 85, no. 11, p. 68, 2007.

[130] E. P. Blasch, R. Breton, P. Valin, and E. Bosse, "User information fusion decision making analysis with the c-ooda model," in *14th International Conference on Information Fusion*. IEEE, 2011, pp. 1–8.

[131] J. Roy and S. Wark, *Concepts, models, and tools for information fusion*. Artech House, 2007.

[132] R. Breton, "The modelling of three levels of cognitive controls with the cognitive-ooda loop framework," *Def. Res. & Dev. CA-Valcartier, DRDC TR*, vol. 111, 2008.

[133] C. Baber, D. Harris, and N. A. Stanton, *Modelling command and control: Event analysis of systemic teamwork*. Ashgate Publishing, Ltd., 2012.

[134] M. R. Endsley, "Situation awareness misconceptions and misunderstandings," *Journal of Cognitive Engineering and Decision Making*, vol. 9, no. 1, pp. 4–32, 2015. [Online]. Available: https://doi.org/10.1177/1555343415572631

[135] J. Vince and J. Parker, "Usability Evaluation of Air Warfare Assessment & Review Toolset in Exercise Black Skies 2012," 2012.

[136] R. Parasuraman, M. Mouloua, and B. Hilburn, *Adaptive aiding and adaptive task allocation enhance human-machine interaction*, 1999.

[137] M. M. Cummings, "Informing Autonomous System Design Through the Lens of Skill-,

Rule-, and Knowledge-Based Behaviors," *Journal of Cognitive Engineering and Decision Making*, vol. 12, no. 1, pp. 58–61, 2018.

[138] J. Bruno, S. Johnson, and J. Hesley, "Robotic disruption and the new revenue cycle: robotic process automation represents an immediate new opportunity for healthcare organizations to perform repetitive, ongoing revenue cycle processes more efficiently and accurately," *Healthcare Financial Management*, vol. 71, no. 9, pp. 54–62, 2017.

[139] J. Wewerka and M. Reichert, "Robotic process automation–a systematic literature review and assessment framework," *arXiv preprint arXiv:2012.11951*, 2020.

[140] A. Jimenez-Ramirez, H. A. Reijers, I. Barba, and C. D. Valle, "A method to improve the early stages of the robotic process automation lifecycle," in *International Conference on Advanced Information Systems Engineering*. Springer, 2019, pp. 446–461.

[141] A. Leshob, A. Bourgouin, and L. Renard, "Towards a process analysis approach to adopt robotic process automation," in *2018 IEEE 15th international conference on e-business engineering (ICEBE)*. IEEE, 2018, pp. 46–53.

[142] J. Annett, "Hierarchical Task Analysis," pp. 17–35, 2003.

[143] N. a. Stanton, ""Best Known Task Analysis Technique"," *Applied Ergonomics*, vol. 1, no. 3, pp. 1 – 56, 2006.

[144] V. A. Banks and N. A. Stanton, "Analysis of driver roles: modelling the changing role of the driver in automated driving systems using EAST," *Theoretical Issues in Ergonomics Science*, vol. 20, no. 3, pp. 284–300, 2019. [Online]. Available: https://doi.org/10.1080/1463922X.2017.1305465

[145] W. D. Gray and S. S. Kirschenbaum, "Analyzing a Novel Expertise: An Unmarked Road," *Cognitive task analysis*, no. January 2000, pp. pp. 275—-290, 2000.

[146] G. H. Walker, H. Gibson, N. A. Stanton, C. Baber, P. Salmon, and D. Green, "Event analysis of systemic teamwork (EAST): A novel integration of ergonomics methods to analyse C4i activity," *Ergonomics*, vol. 49, no. 12-13, pp. 1345–1369, 2006.

[147] M. Schmitz, C. Stummer, and M. Gerke, "Smart automation as enabler of digitalization? a review of rpa/ai potential and barriers to its realization," *Future Telco*, pp. 349–358, 2019.

[148] M. Lacity, L. P. Willcocks, and A. Craig, "Robotic process automation at telefonica o2," 2015.

[149] M. Cohen, A. Rozario *et al.*, "Exploring the use of robotic process automation (rpa) in substantive audit procedures," *The CPA Journal*, vol. 89, no. 7, pp. 49–53, 2019.

[150] M. Patel, A. Shukla, R. Porwal, and R. Kotecha, "Customized automated email response bot using machine learning and robotic process automation," in *2nd International Con-*

ference on Advances in Science & Technology (ICAST), 2019.

[151] M. C. Lacity and L. P. Willcocks, "A new approach to automating services," *MIT Sloan Management Review*, vol. 58, no. 1, pp. 41–49, 2016.

[152] R. Breton, "The modeling of three levels of cognitive controls with the Cognitive-OODA loop framework," no. September, p. 72 pp., 2008. [Online]. Available: http://pubs-www.drenet.dnd.ca/BASIS/pcandid/www/engpub/DDW?W=AUTHOR+=+'Breton,+R.'{&}M=12{&}R=Y{&}U=1

[153] E. Blasch, J. Llinas, D. Lambert, P. Valin, S. Das, C. Chong, M. Kokar, and E. Shahbazian, "High Level Information Fusion developments, issues, and grand challenges," *13th Conference on Information Fusion, Fusion 2010*, 2010.

[154] P. Hallikainen, R. Bekkhus, and S. L. Pan, "How opuscapita used internal rpa capabilities to offer services to clients." *MIS Quarterly Executive*, vol. 17, no. 1, 2018.

[155] E. Byrne, "Knowing Behavior Helps Insure Models Against Breakage: A Commentary on Kaber's "Issues in Human–Automation Interaction Modeling"," *Journal of Cognitive Engineering and Decision Making*, vol. 12, no. 1, pp. 67–69, 2018.

[156] D. B. Kaber, "Reflections on Commentaries on "Issues in Human–Automation Interaction Modeling"," *Journal of Cognitive Engineering and Decision Making*, vol. 12, no. 1, pp. 86–93, 2018.

[157] A. Sebok and C. D. Wickens, "Implementing Lumberjacks and Black Swans into Model-Based Tools to Support Human-Automation Interaction," *Human Factors*, vol. 59, no. 2, pp. 189–203, 2017.

[158] B. T. B. Sheridan and R. Parasuraman, "Human-Automation Interaction," pp. 89–129, 2002.

[159] T. B. Sheridan, "Comments on " Issues in Human – Automation Interaction Modeling : Presumptive Aspects of Frameworks of Types and Levels of Automation " by David B. Kaber," *Jourmal of Cognitive Engineering and Decision Making*, vol. 12, no. 1, pp. 25–28, 2018.

[160] F. Dehais, M. Causse, F. Vachon, and S. Tremblay, "Cognitive conflict in human-automation interactions: A psychophysiological study," *Applied Ergonomics*, vol. 43, no. 3, pp. 588–595, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.apergo.2011.09.004

[161] G. A. Jamieson and G. Skraaning, "Levels of Automation in Human Factors Models for Automation Design: Why We Might Consider Throwing the Baby Out With the Bathwater," *Journal of Cognitive Engineering and Decision Making*, vol. 12, no. 1, pp. 42–49, 2018.

[162] L. P. Willcocks, M. Lacity, and A. Craig, "The it function and robotic process automa-

tion," 2015.

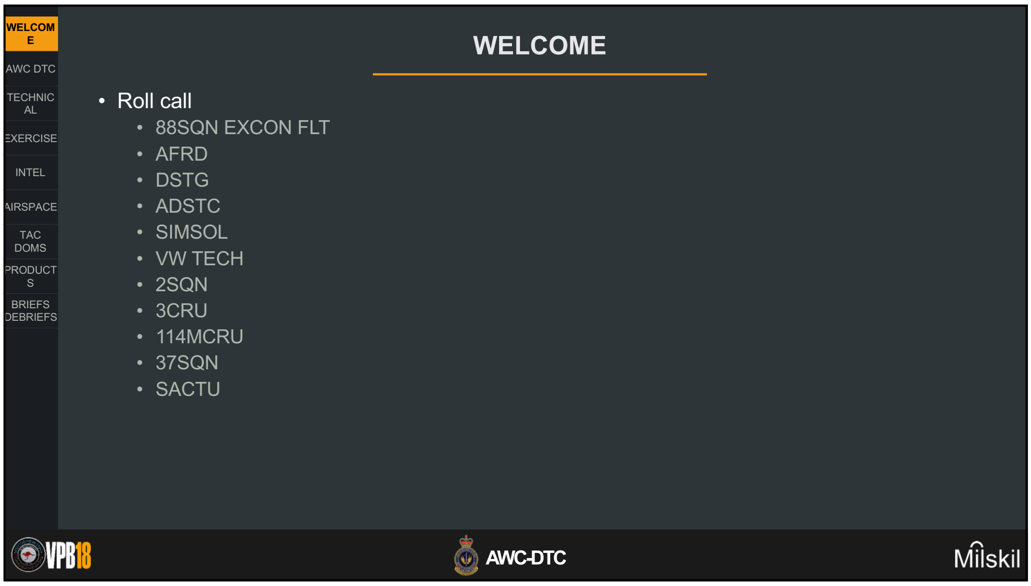
[163] U. N. A. T. Command, *CNATRA P-826 Flight Training Instruction, Basic Fighter Manoeuvring*.

[164] R. L. Shaw, *Fighter Combat Tactics and Manoeuvring*. Naval Institute Press, Annapolis, MD, 1985.

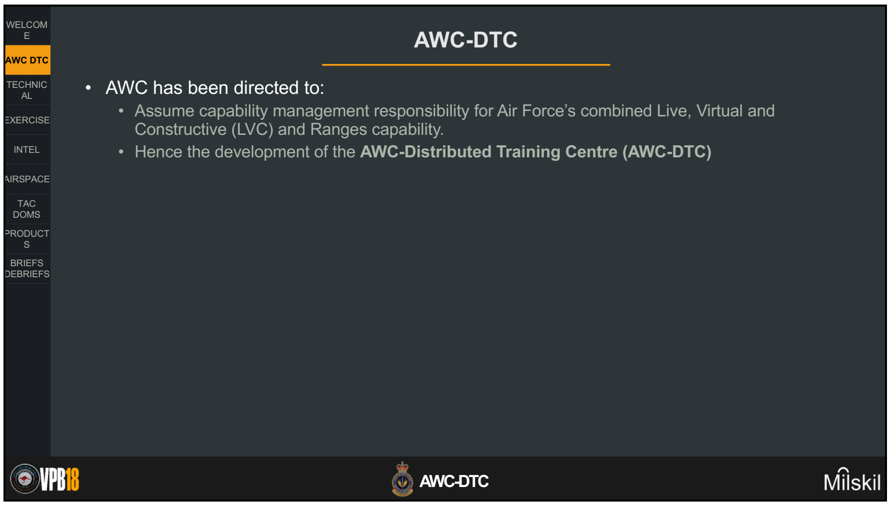
Appendix A

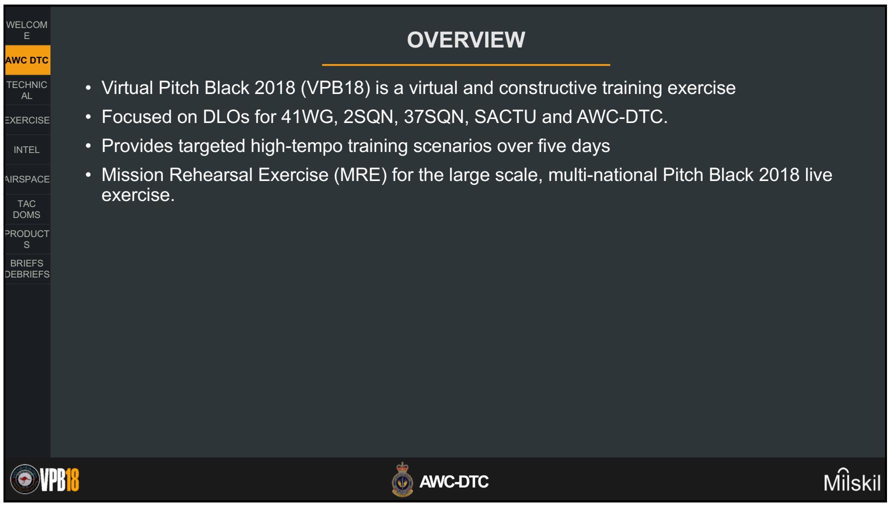
Briefing Material: Exercise Virtual Pitch Black [1]

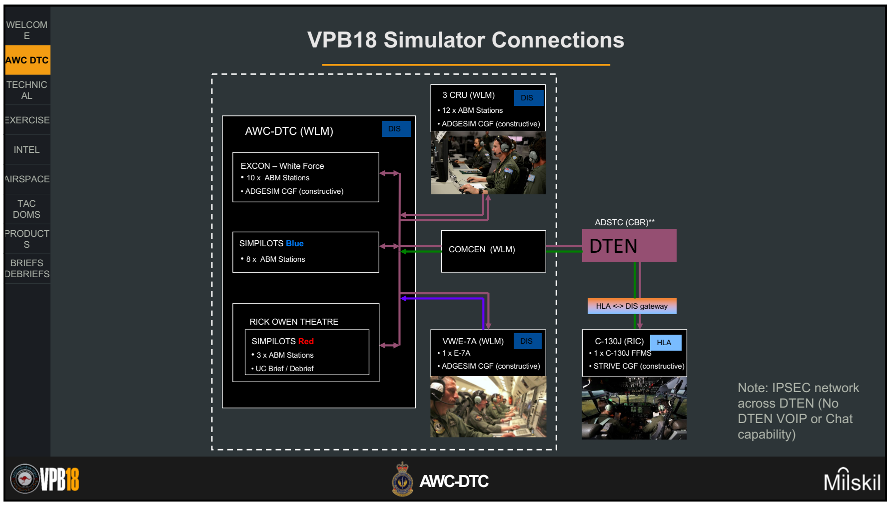

VIRTUAL PITCH BLACK 2018
AWC - DTC
MASS AIR BRIEF
29 JUN 2018
AIR WARFARE CENTRE
ROYAL AUSTRALIAN AIR FORCE
READY THE WARFIGHTER
VPB18
AIR WARFARE CENTRE - DISTRIBUTED TRAINING CENTRE

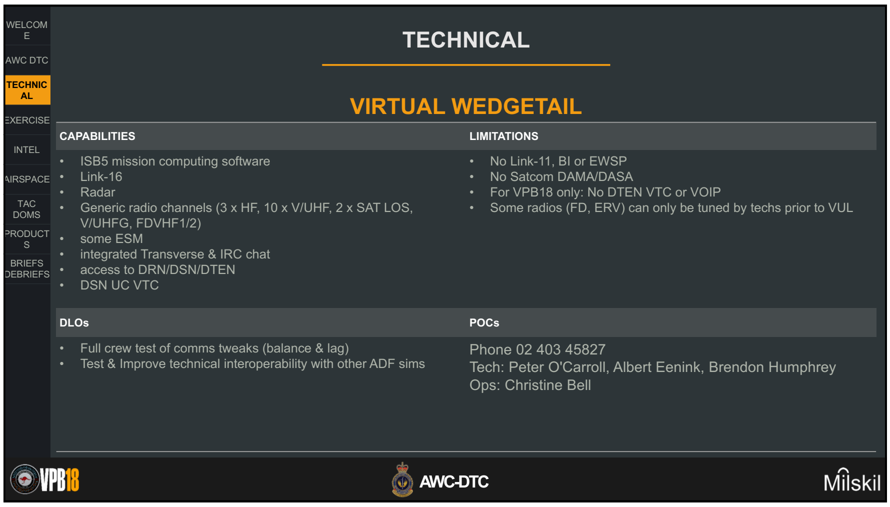

WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
WELCOME
Roll call
88SQN EXCON FLT
AFRD
DSTG
ADSTC
SIMSOL
VW TECH
2SQN
3CRU
114MCRU
37SQN
SACTU
VPB18
AWC-DTC
Milskil


WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
WHY LVC
• Increasingly 4th / 5th Gen platforms cannot be adequately exercised in Live environment due to inability to generate complexity, density, scale, fidelity and increasing security constraints.
VPB18
AWC-DTC
Milskil


WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
AWC-DTC
• AWC has been directed to:
• Assume capability management responsibility for Air Force's combined Live, Virtual and Constructive (LVC) and Ranges capability.
• Hence the development of the AWC-Distributed Training Centre (AWC-DTC)
VPB18
AWC-DTC
Milskil

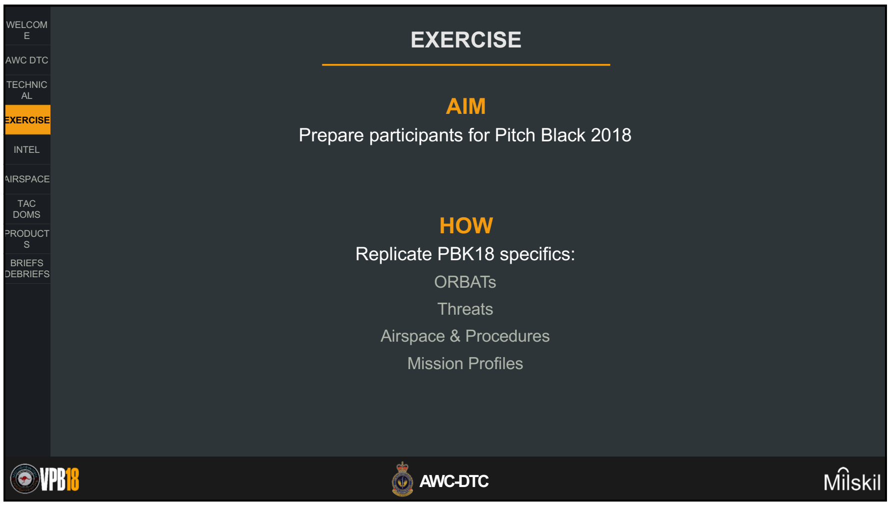

WELCOM
E
AWC DTC
TECHNIC
AL
EXERCISE
INTEL
AIRSPACE
TAC
DOMS
PRODUCT
S
BRIEFS
DEBRIEFS
OVERVIEW
• Virtual Pitch Black 2018 (VPB18) is a virtual and constructive training exercise
• Focused on DLOs for 41WG, 2SQN, 37SQN, SACTU and AWC-DTC.
• Provides targeted high-tempo training scenarios over five days
• Mission Rehearsal Exercise (MRE) for the large scale, multi-national Pitch Black 2018 live exercise.
VPB18
AWC-DTC
Milskil

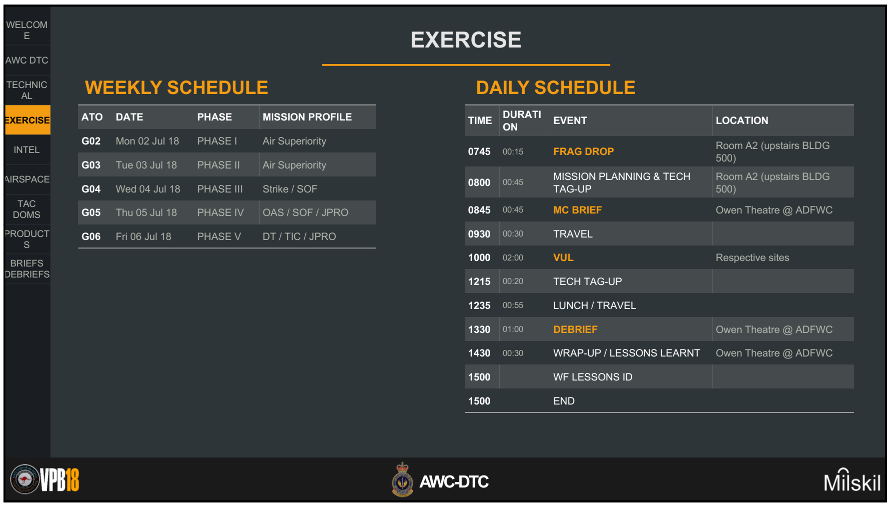

WELCOM
E
AWC DTC
TECHNIC
AL
EXERCISE
INTEL
AIRSPACE
TAC
DOMS
PRODUCT
S
BRIEFS
DEBRIEFS
VPB18 Simulator Connections
AWC-DTC (WLM)
DIS
EXCON – White Force
• 10 x ABM Stations
• ADGESIM CGF (constructive)
SIMPILOTS Blue
• 8 x ABM Stations
RICK OWEN THEATRE
SIMPILOTS Red
• 3 x ABM Stations
• UC Brief / Debrief
3 CRU (WLM)
DIS
• 12 x ABM Stations
• ADGESIM CGF (constructive)
COMCEN (WLM)
VW/E-7A (WLM)
DIS
• 1 x E-7A
• ADGESIM CGF (constructive)
ADSTC (CBR)**
DTEN
HLA <-> DIS gateway
C-130J (RIC)
HLA
• 1 x C-130J FFMS
• STRIVE CGF (constructive)
Note: IPSEC network across DTEN (No DTEN VOIP or Chat capability)
VPB18
AWC-DTC
Milskil

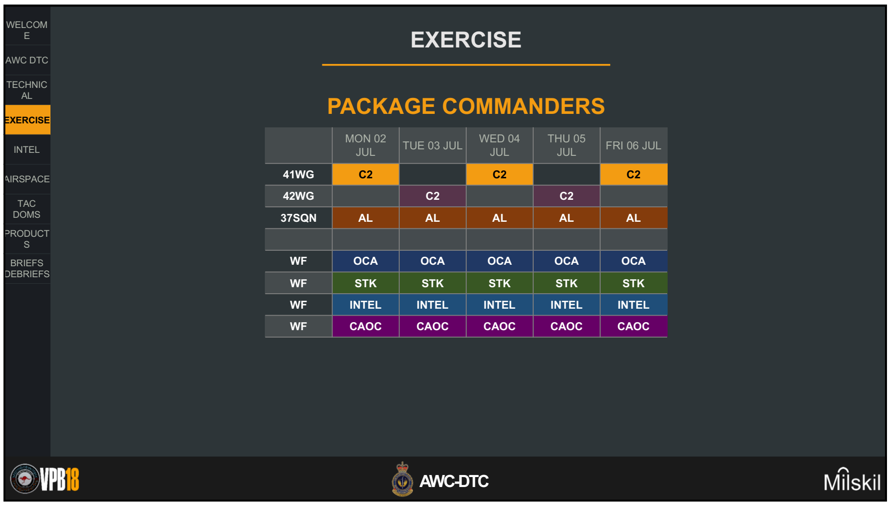

WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
TECHNICAL
VIRTUAL WEDGETAIL
CAPABILITIES
• ISB5 mission computing software
• Link-16
• Radar
• Generic radio channels (3 x HF, 10 x V/UHF, 2 x SAT LOS, V/UHFG, FDVHF1/2)
• some ESM
• integrated Transverse & IRC chat
• access to DRN/DSN/DTEN
• DSN UC VTC
LIMITATIONS
• No Link-11, BI or EWSP
• No Satcom DAMA/DASA
• For VPB18 only: No DTEN VTC or VOIP
• Some radios (FD, ERV) can only be tuned by techs prior to VUL
DLOs
• Full crew test of comms tweaks (balance & lag)
• Test & Improve technical interoperability with other ADF sims
POCs
Phone 02 403 45827
Tech: Peter O'Carroll, Albert Eenink, Brendon Humphrey
Ops: Christine Bell
VPB18
AWC-DTC
Milskil


WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
TECHNICAL
DISTRIBUTED TRAINING CENTRE & EASTROC
CAPABILITIES
• CGF – Blue and Red OK
• TDS - Good
• Good comms with VW and C130
• Link 16 Good
LIMITATIONS
• No 'Secure' (crypto) on simulated radios
DLOs
POCs
White Force room: 02 403 46059
Syndicate Room 2: 02 403 46279
Owen Lecture theater: 02 403 46005
Coord on Beige 2
ENG stations on Intercom for assistance
VPB18
AWC-DTC
Milskil

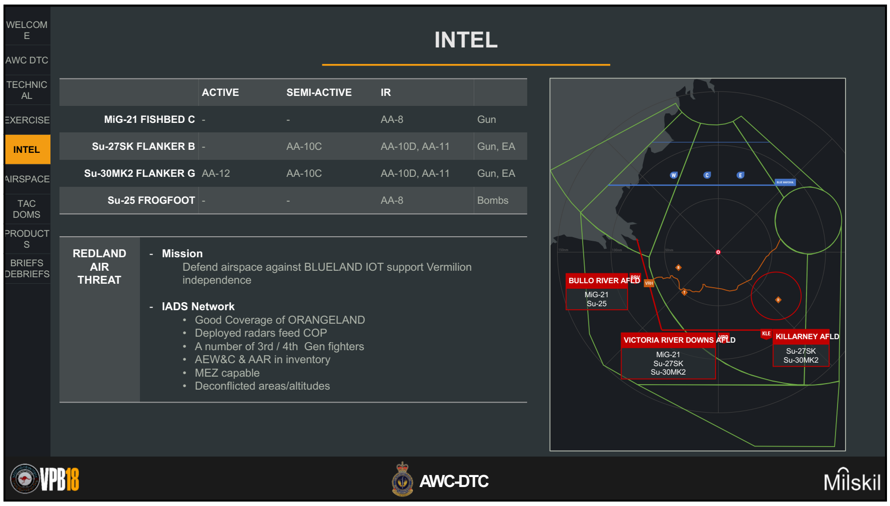

WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
TECHNICAL
C130
CAPABILITIES
Radios:
- 2 x VHF (+ secure)
- 2 x UHF (+ secure)
- 2 x HF
LAIRCM
LIMITATIONS
Due to crew training limitations, there will be no:
- RWR response to RADAR emitters
- Link 16 capability
Due to visual database limitations, only the following airfields will be used:
- LZ 1: YBRD Bradshaw (high fidelity)
- LZ 2: YTBR Timber Creek (low fidelity)
- DZ 1: YDMR Delamere Station; airdrop only, no TO or LDG.
DLOs
• Provide planning and execution training opportunities to junior crews in an LFE environment
• Provide junior crews experience in developing and adhering to mission contracts in an LFE environment
• Provide opportunity to employ ABNOPS skillsets in LF mission scenarios
POCs
Primary: Mervin Sayseng (CAE) 02 4587 1647
Secondary: CPL James Wood (RAAF CISCON), 02 4587 1454
VPB18
AWC-DTC
Milskil

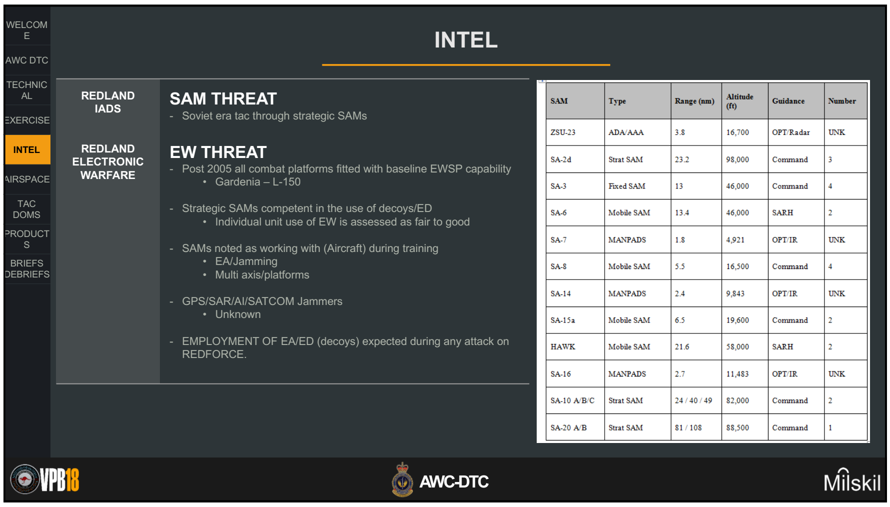

WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
EXERCISE
AIM
Prepare participants for Pitch Black 2018
HOW
Replicate PBK18 specifics:
ORBATs
Threats
Airspace & Procedures
Mission Profiles
VPB18
AWC-DTC
Milskil


WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
EXERCISE
WEEKLY SCHEDULE
ATO | DATE | PHASE | MISSION PROFILE
G02 | Mon 02 Jul 18 | PHASE I | Air Superiority
G03 | Tue 03 Jul 18 | PHASE II | Air Superiority
G04 | Wed 04 Jul 18 | PHASE III | Strike / SOF
G05 | Thu 05 Jul 18 | PHASE IV | OAS / SOF / JPRO
G06 | Fri 06 Jul 18 | PHASE V | DT / TIC / JPRO
DAILY SCHEDULE
TIME | DURATION | EVENT | LOCATION
0745 | 00:15 | FRAG DROP | Room A2 (upstairs BLDG 500)
0800 | 00:45 | MISSION PLANNING & TECH TAG-UP | Room A2 (upstairs BLDG 500)
0845 | 00:45 | MC BRIEF | Owen Theatre @ ADFWC
0930 | 00:30 | TRAVEL |
1000 | 02:00 | VUL | Respective sites
1215 | 00:20 | TECH TAG-UP |
1235 | 00:55 | LUNCH / TRAVEL |
1330 | 01:00 | DEBRIEF | Owen Theatre @ ADFWC
1430 | 00:30 | WRAP-UP / LESSONS LEARNT | Owen Theatre @ ADFWC
1500 | | WF LESSONS ID |
1500 | | END |
VPB18
AWC-DTC
Milskil

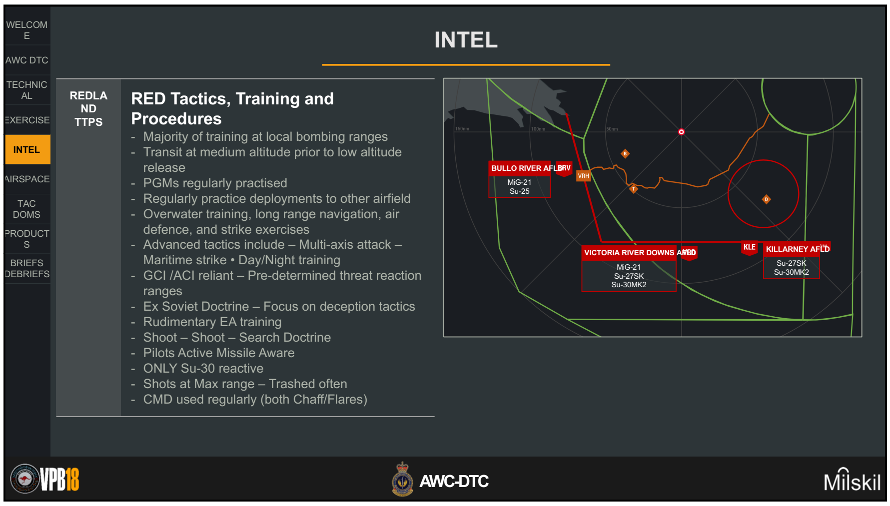

WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
EXERCISE
PACKAGE COMMANDERS
| MON 02 JUL | TUE 03 JUL | WED 04 JUL | THU 05 JUL | FRI 06 JUL
41WG | C2 | | C2 | | C2
42WG | | C2 | | C2 |
37SQN | AL | AL | AL | AL | AL
WF | OCA | OCA | OCA | OCA | OCA
WF | STK | STK | STK | STK | STK
WF | INTEL | INTEL | INTEL | INTEL | INTEL
WF | CAOC | CAOC | CAOC | CAOC | CAOC
VPB18
AWC-DTC
Milskil

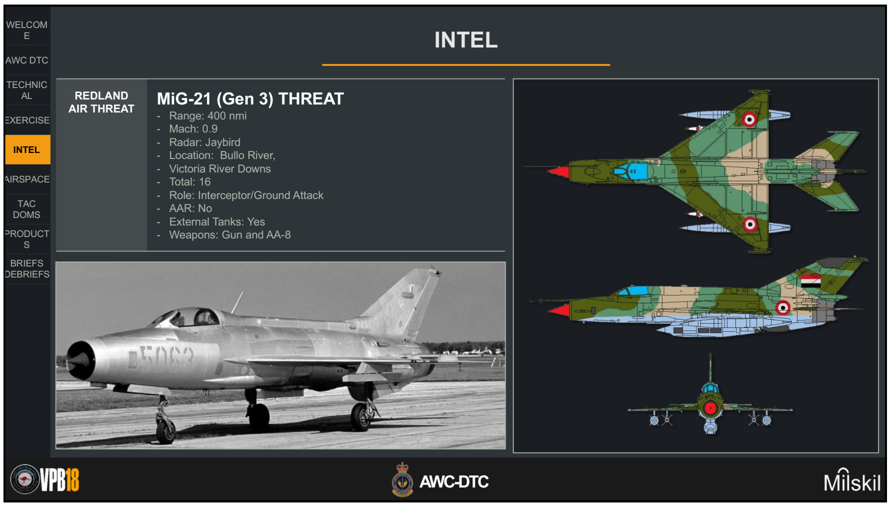

WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
INTEL
ROAD TO WAR
STRATEGIC SITUATION
VERMILION (VLA) has declared independence from ORANGELAND
VERMILION consists of a large percentage of REDLAND ethnic personnel (93%).
REDLAND is supporting the independence push both overtly and covertly.
REDLAND has stated that the protection of REDLANDIANS (in ORANGELAND) is
REDLAND assets are active in VERMILION and surrounds.
ON THE GROUND
VLA is fighting ORANGELAND troops on two fronts - Koolendong Port – Delamere
ORANGELAND has requested air support from BLUELAND ISO these fronts
ORANGELAND has requested BLUELAND assistance against REDLAND involver
No intent to escalate conflict.
No targeting inside REDLAND borders
UN is the lead agency for IDP/Refugee efforts
OPERATIONAL ASSUMPTIONS
ORANGELAND is unable to support BLUELAND operations in its current state
BLUELAND is able to utilise APODs/SPODs in ORANGELAND as required
Operations inside REDLAND will not be authorised
Some civilian transport will continue to operate in a non-permissive environment
United Nations operations will occur inside ORANGELAND without BLUELAND involvement
VPB18
AWC-DTC
Milskil


WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
INTEL
ACTIVE | SEMI-ACTIVE | IR
MiG-21 FISHBED C | - | - | AA-8 | Gun
Su-27SK FLANKER B | - | AA-10C | AA-10D, AA-11 | Gun, EA
Su-30MK2 FLANKER G | AA-12 | AA-10C | AA-10D, AA-11 | Gun, EA
Su-25 FROGFOOT | - | - | AA-8 | Bombs
REDLAND AIR THREAT
- Mission
Defend airspace against BLUELAND IOT support Vermilion independence
- IADS Network
• Good Coverage of ORANGELAND
• Deployed radars feed COP
• A number of 3rd / 4th Gen fighters
• AEW&C & AAR in inventory
• MEZ capable
• Deconflicted areas/altitudes
BULLO RIVER AFLD
MiG-21
Su-25
VICTORIA RIVER DOWNS AFLD
MiG-21
Su-27SK
Su-30MK2
KILLARNEY AFLD
Su-27SK
Su-30MK2
VPB18
AWC-DTC
Milskil

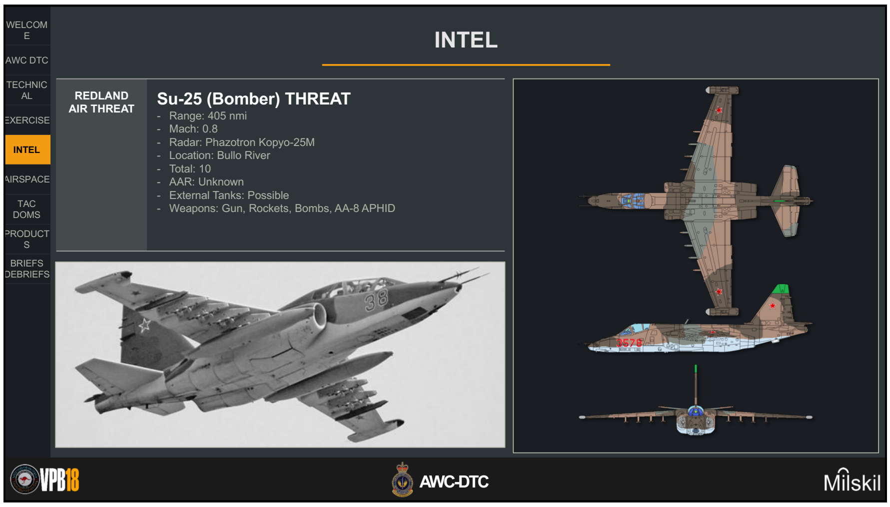


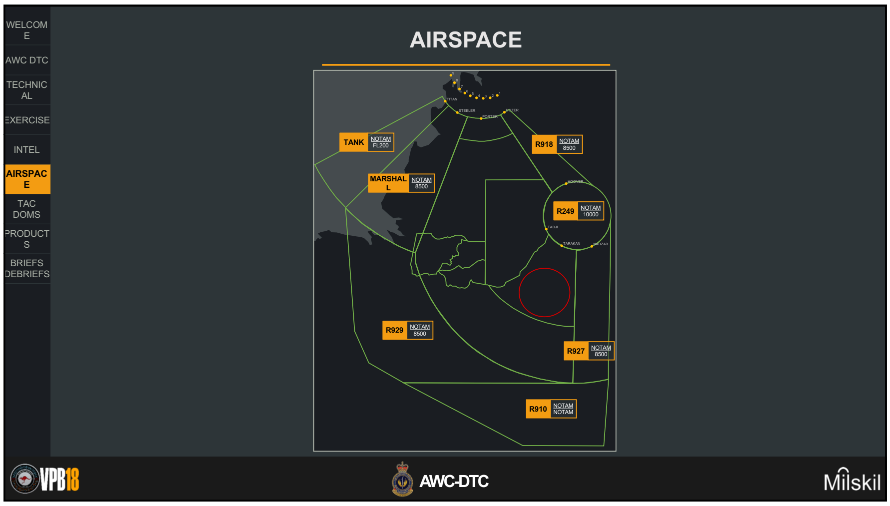
SAM	Type	Range (nm)	Altitude (ft)	Guidance	Number
ZSU-23	ADA/AAA	3.8	16,700	OPT/Radar	UNK
SA-2d	Strat SAM	23.2	98,000	Command	3
SA-3	Fixed SAM	13	46,000	Command	4
SA-6	Mobile SAM	13.4	46,000	SARH	2
SA-7	MANPADS	1.8	4,921	OPT/IR	UNK
SA-8	Mobile SAM	5.5	16,500	Command	4
SA-14	MANPADS	2.4	9,843	OPT/IR	UNK
SA-15a	Mobile SAM	6.5	19,600	Command	2
HAWK	Mobile SAM	21.6	58,000	SARH	2
SA-16	MANPADS	2.7	11,483	OPT/IR	UNK
SA-10 A/B/C	Strat SAM	24 / 40 / 49	82,000	Command	2
SA-20 A/B	Strat SAM	81 / 108	88,500	Command	1



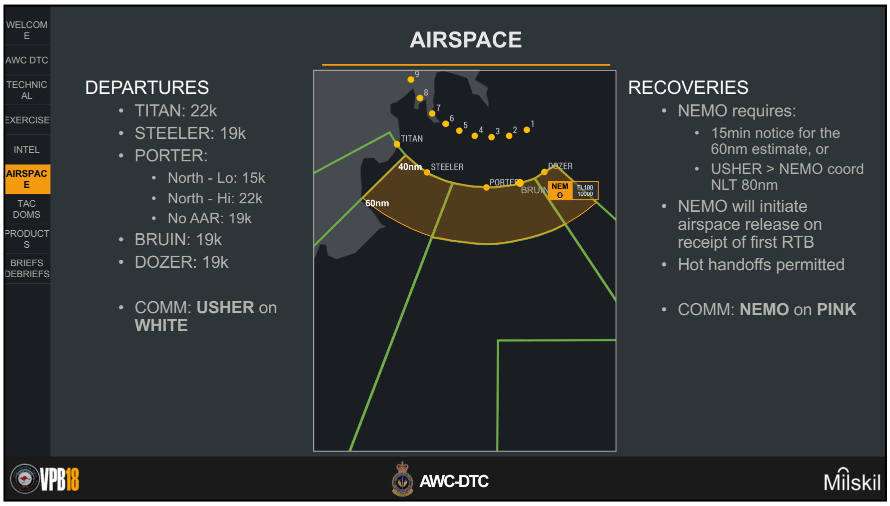

WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
INTEL
REDLAND AIR THREAT
MiG-21 (Gen 3) THREAT
- Range: 400 nmi
- Mach: 0.9
- Radar: Jaybird
- Location: Bullo River,
- Victoria River Downs
- Total: 16
- Role: Interceptor/Ground Attack
- AAR: No
- External Tanks: Yes
- Weapons: Gun and AA-8
VPB18
AWC-DTC
Milskil

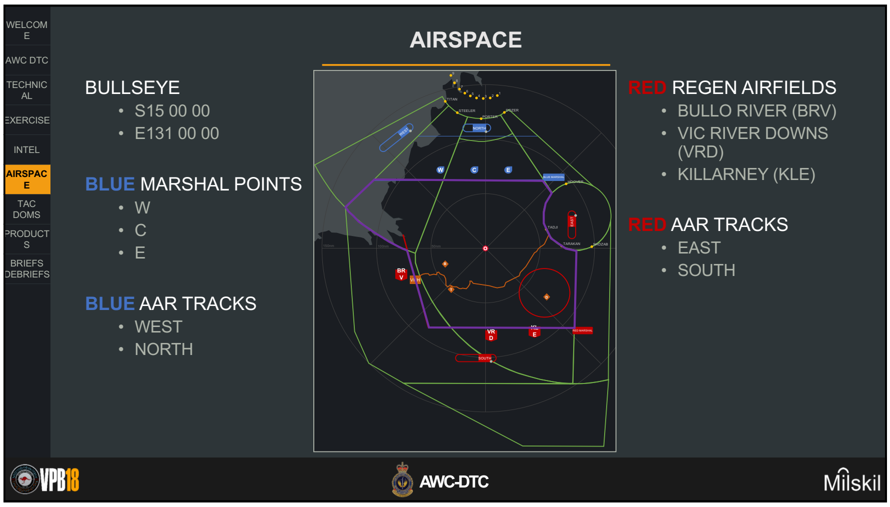

WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
INTEL
REDLAND AIR THREAT
Su-27/30 (Gen 4) THREAT
- Range: 460 nmi – Mach: 2.3
- Radar: Slotback II
- Location: Victoria River Downs, Killarney AFLD
- Total: 48
- Role: Air Superiority
- AAR: No
- External Tanks: No
- Weapons: Gun, AA-10 C/D, AA-11, AA-12A/B
VPB18
AWC-DTC
Milskil


WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
INTEL
REDLAND AIR THREAT
Su-25 (Bomber) THREAT
- Range: 405 nmi
- Mach: 0.8
- Radar: Phazotron Kopyo-25M
- Location: Bullo River
- Total: 10
- AAR: Unknown
- External Tanks: Possible
- Weapons: Gun, Rockets, Bombs, AA-8 APHID
VPB18
AWC-DTC
Milskil

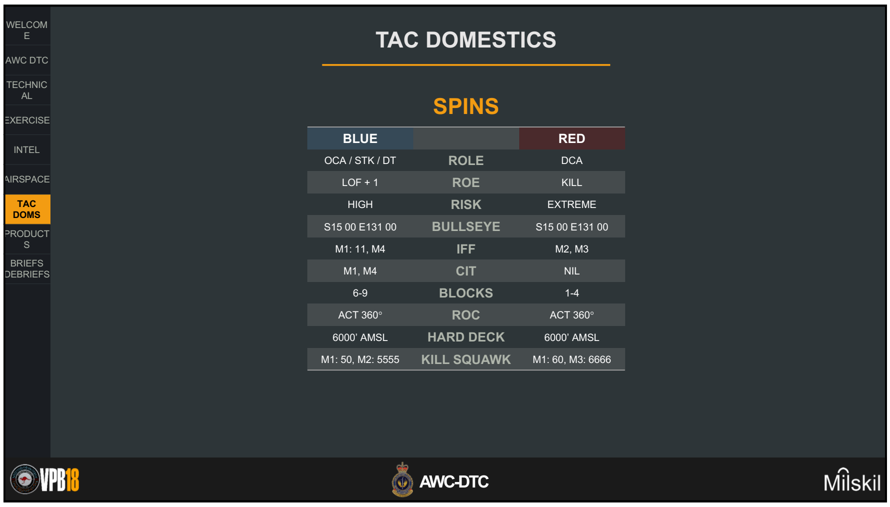

WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
AIRSPACE
TITAN
STEELER
PORTER
DOZER
TANK NOTAM FL200
MARSHALL NOTAM 8500
R918 NOTAM 8500
HOOVER
R249 NOTAM 10000
TADJI
TARAKAN
NADZAB
R929 NOTAM 8500
R927 NOTAM 8500
R910 NOTAM NOTAM
VPB18
AWC-DTC
Milskil


WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
AIRSPACE
TITAN
STEELER
PORTER
DOZER
R915
R914
R225
R916
D210
R247
R268
R269
R270
R917
R248 B
R248 A
R232
TADJI
TARAKAN
VPB18
AWC-DTC
Milskil

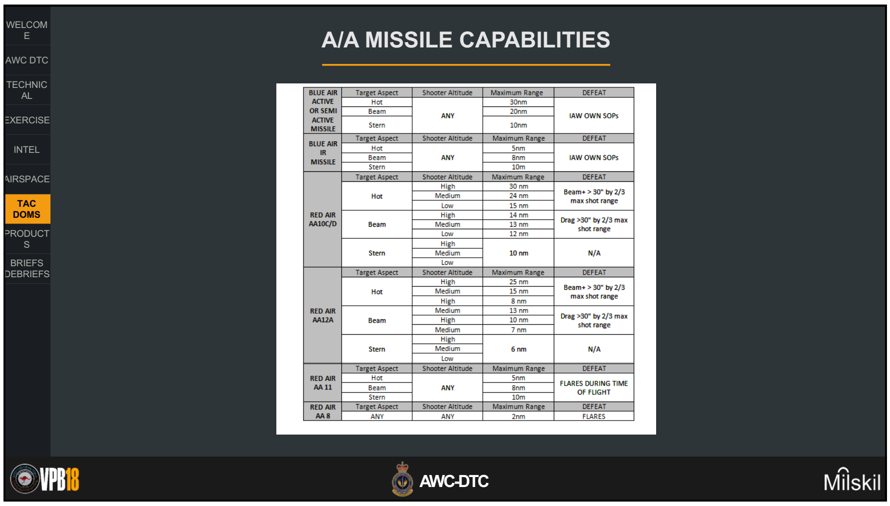

WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
AIRSPACE
DEPARTURES
TITAN: 22k
STEELER: 19k
PORTER:
North - Lo: 15k
North - Hi: 22k
No AAR: 19k
BRUIN: 19k
DOZER: 19k
COMM: USHER on WHITE
TITAN
STEELER
PORTER
DOZER
40nm
60nm
NEMO
RECOVERIES
NEMO requires:
15min notice for the 60nm estimate, or
USHER > NEMO coord NLT 80nm
NEMO will initiate airspace release on receipt of first RTB
Hot handoffs permitted
COMM: NEMO on PINK
VPB18
AWC-DTC
Milskil


WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
AIRSPACE
BULLSEYE
• S15 00 00
• E131 00 00
BLUE MARSHAL POINTS
• W
• C
• E
BLUE AAR TRACKS
• WEST
• NORTH
RED REGEN AIRFIELDS
• BULLO RIVER (BRV)
• VIC RIVER DOWNS (VRD)
• KILLARNEY (KLE)
RED AAR TRACKS
• EAST
• SOUTH
VPB18
AWC-DTC
Milskil

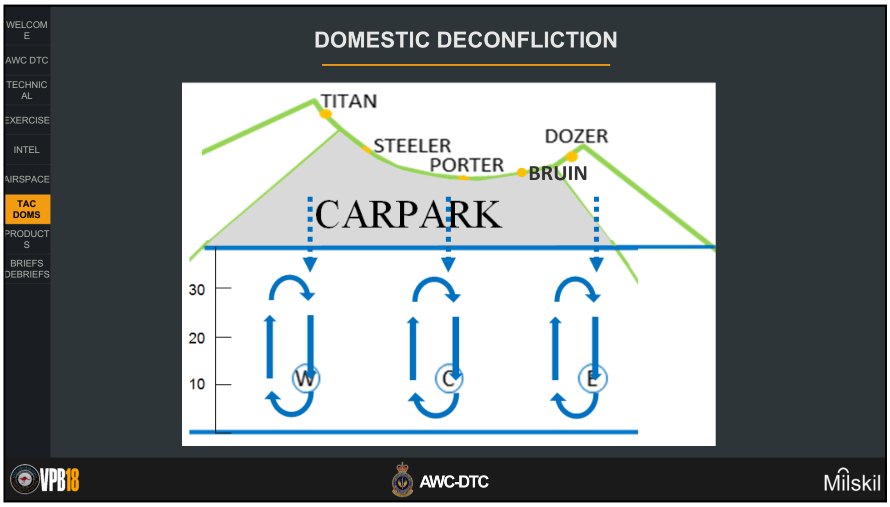

WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
TAC DOMESTICS
BLUE ORBAT
SHOGUN RAAF F/A-18A
WARLOCK RAAF F/A-18A
SHIKRA RSAF F-15SG
TRISONIC IAF Su-30 Non-link
TAZER FAF Rafale
GHOSTRIDER RMAF F-18D Non-link
BRUTAL RAAF E/A-18G
CANNON RAAF F/A-18F
VIKING RAAF F/A-18A
PHOENIX RSAF F-16 Non-link
SHARK RTAF JAS-39 Non-link
WOODY USAF F-16G
RYDER TNI-AU F-16 Non-link
TROJAN RAAF C-130J
BENZENE RMAF A400M Non-link
WINDSOR RAAF KC-30A
TWISTER RSAF KC-135 Non-link
OUTBACK RAAF E-7A
TAIPAN RAAF MCRC
SLIM PACAF B-52
VPB18
AWC-DTC
Milskil

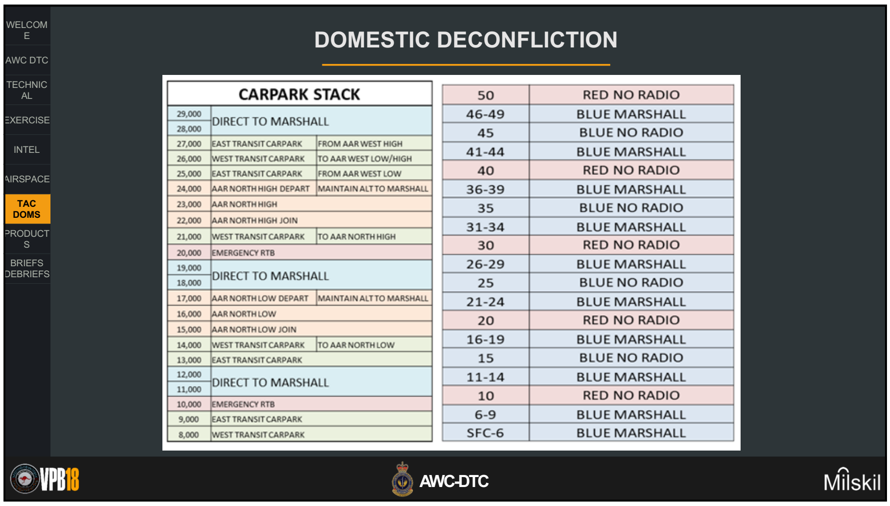


BLUE		RED
OCA / STK / DT	ROLE	DCA
LOF + 1	ROE	KILL
HIGH	RISK	EXTREME
S15 00 E131 00	BULLSEYE	S15 00 E131 00
M1: 11, M4	IFF	M2, M3
M1, M4	CIT	NIL
6-9	BLOCKS	1-4
ACT 360°	ROC	ACT 360°
6000' AMSL	HARD DECK	6000' AMSL
M1: 50, M2: 5555	KILL SQUAWK	M1: 60, M3: 6666



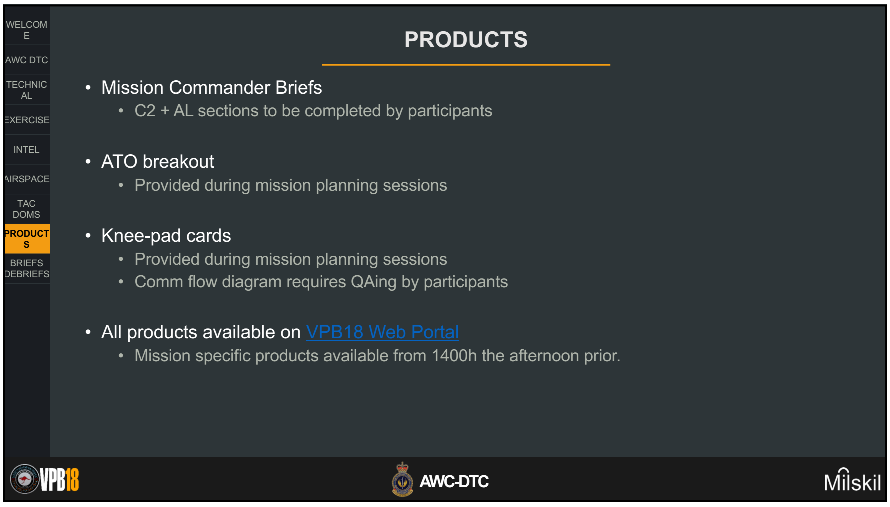
BLUE AIR ACTIVE OR SEMI ACTIVE MISSILE	Target Aspect	Shooter Altitude	Maximum Range	DEFEAT
	Hot	ANY	30nm	IAW OWN SOPs
	Beam		20nm	
	Stern		10nm	
BLUE AIR IR MISSILE	Target Aspect	Shooter Altitude	Maximum Range	DEFEAT
	Hot	ANY	5nm	IAW OWN SOPs
	Beam		8nm	
	Stern		10m	
RED AIR AA10C/D	Target Aspect	Shooter Altitude	Maximum Range	DEFEAT
	Hot	High	30 nm	Beam+ > 30" by 2/3 max shot range
		Medium	24 nm	
		Low	15 nm	
	Beam	High	14 nm	Drag >30" by 2/3 max shot range
		Medium	13 nm	
		Low	12 nm	
	Stern	High	10 nm	N/A
		Medium		
		Low		
RED AIR AA12A	Target Aspect	Shooter Altitude	Maximum Range	DEFEAT
	Hot	High	25 nm	Beam+ > 30" by 2/3 max shot range
		Medium	15 nm	
		High	8 nm	
	Beam	Medium	13 nm	Drag >30" by 2/3 max shot range
		High	10 nm	
		Medium	7 nm	
	Stern	High	6 nm	N/A
		Medium		
		Low		
RED AIR AA 11	Target Aspect	Shooter Altitude	Maximum Range	DEFEAT
	Hot	ANY	5nm	FLARES DURING TIME OF FLIGHT
	Beam		8nm	
	Stern		10m	
RED AIR AA 8	Target Aspect	Shooter Altitude	Maximum Range	DEFEAT
	ANY	ANY	2nm	FLARES

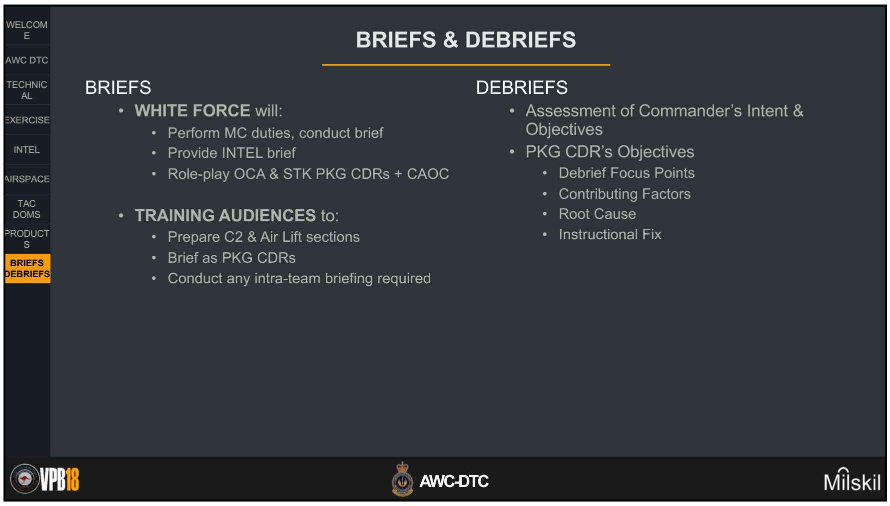


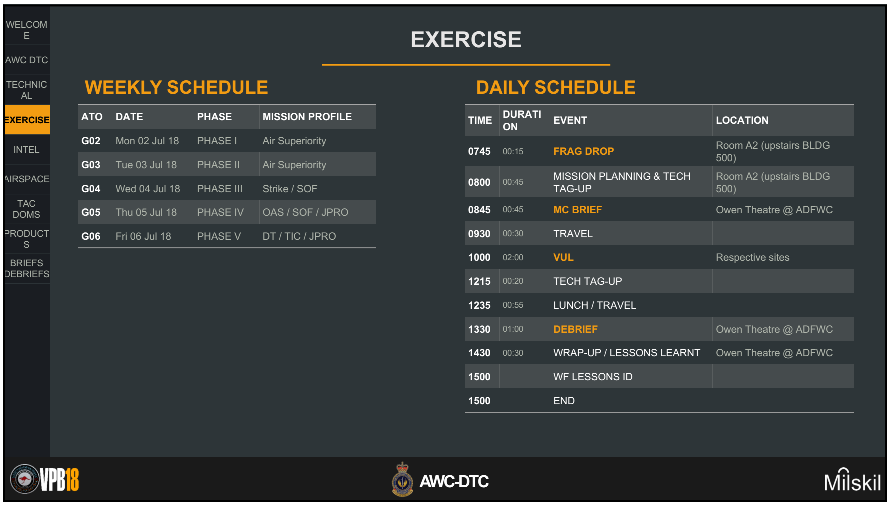


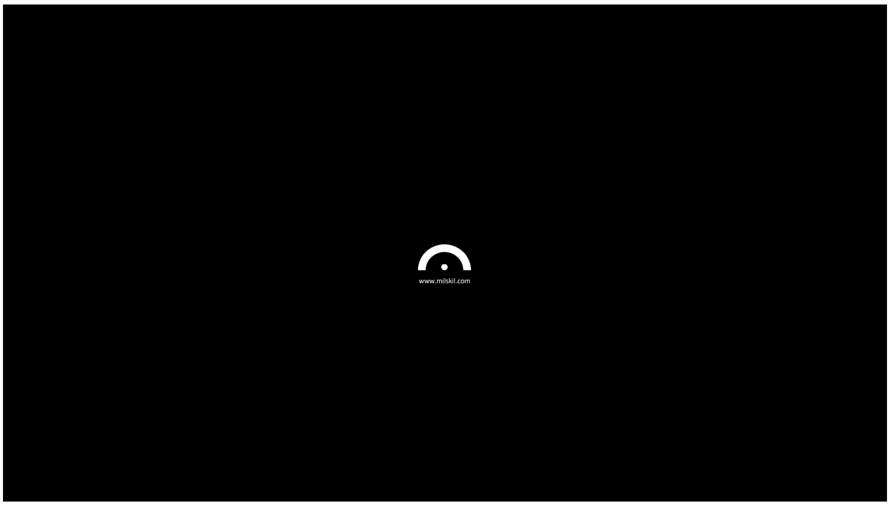
CARPARK STACK		
29,000	DIRECT TO MARSHALL	
28,000		
27,000	EAST TRANSIT CARPARK	FROM AAR WEST HIGH
26,000	WEST TRANSIT CARPARK	TO AAR WEST LOW/HIGH
25,000	EAST TRANSIT CARPARK	FROM AAR WEST LOW
24,000	AAR NORTH HIGH DEPART	MAINTAIN ALT TO MARSHALL
23,000	AAR NORTH HIGH	
22,000	AAR NORTH HIGH JOIN	
21,000	WEST TRANSIT CARPARK	TO AAR NORTH HIGH
20,000	EMERGENCY RTB	
19,000	DIRECT TO MARSHALL	
18,000		
17,000	AAR NORTH LOW DEPART	MAINTAIN ALT TO MARSHALL
16,000	AAR NORTH LOW	
15,000	AAR NORTH LOW JOIN	
14,000	WEST TRANSIT CARPARK	TO AAR NORTH LOW
13,000	EAST TRANSIT CARPARK	
12,000	DIRECT TO MARSHALL	
11,000		
10,000	EMERGENCY RTB	
9,000	EAST TRANSIT CARPARK	
8,000	WEST TRANSIT CARPARK	

50	RED NO RADIO
46-49	BLUE MARSHALL
45	BLUE NO RADIO
41-44	BLUE MARSHALL
40	RED NO RADIO
36-39	BLUE MARSHALL
35	BLUE NO RADIO
31-34	BLUE MARSHALL
30	RED NO RADIO
26-29	BLUE MARSHALL
25	BLUE NO RADIO
21-24	BLUE MARSHALL
20	RED NO RADIO
16-19	BLUE MARSHALL
15	BLUE NO RADIO
11-14	BLUE MARSHALL
10	RED NO RADIO
6-9	BLUE MARSHALL
SFC-6	BLUE MARSHALL


WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
PRODUCTS
• Mission Commander Briefs
• C2 + AL sections to be completed by participants
• ATO breakout
• Provided during mission planning sessions
• Knee-pad cards
• Provided during mission planning sessions
• Comm flow diagram requires QAing by participants
• All products available on VPB18 Web Portal
• Mission specific products available from 1400h the afternoon prior.
VPB18
AWC-DTC
Milskil


WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
BRIEFS & DEBRIEFS
BRIEFS
• WHITE FORCE will:
• Perform MC duties, conduct brief
• Provide INTEL brief
• Role-play OCA & STK PKG CDRs + CAOC
• TRAINING AUDIENCES to:
• Prepare C2 & Air Lift sections
• Brief as PKG CDRs
• Conduct any intra-team briefing required
DEBRIEFS
• Assessment of Commander's Intent & Objectives
• PKG CDR's Objectives
• Debrief Focus Points
• Contributing Factors
• Root Cause
• Instructional Fix
VPB18
AWC-DTC
Milskil


WELCOME
AWC DTC
TECHNICAL
EXERCISE
INTEL
AIRSPACE
TAC DOMS
PRODUCTS
BRIEFS DEBRIEFS
EXERCISE
WEEKLY SCHEDULE
ATO DATE PHASE MISSION PROFILE
G02 Mon 02 Jul 18 PHASE I Air Superiority
G03 Tue 03 Jul 18 PHASE II Air Superiority
G04 Wed 04 Jul 18 PHASE III Strike / SOF
G05 Thu 05 Jul 18 PHASE IV OAS / SOF / JPRO
G06 Fri 06 Jul 18 PHASE V DT / TIC / JPRO
DAILY SCHEDULE
TIME DURATION EVENT LOCATION
0745 00:15 FRAG DROP Room A2 (upstairs BLDG 500)
0800 00:45 MISSION PLANNING & TECH TAG-UP Room A2 (upstairs BLDG 500)
0845 00:45 MC BRIEF Owen Theatre @ ADFWC
0930 00:30 TRAVEL
1000 02:00 VUL Respective sites
1215 00:20 TECH TAG-UP
1235 00:55 LUNCH / TRAVEL
1330 01:00 DEBRIEF Owen Theatre @ ADFWC
1430 00:30 WRAP-UP / LESSONS LEARNT Owen Theatre @ ADFWC
1500 WF LESSONS ID
1500 END
VPB18
AWC-DTC
Milskil


www.milskil.com

Appendix B

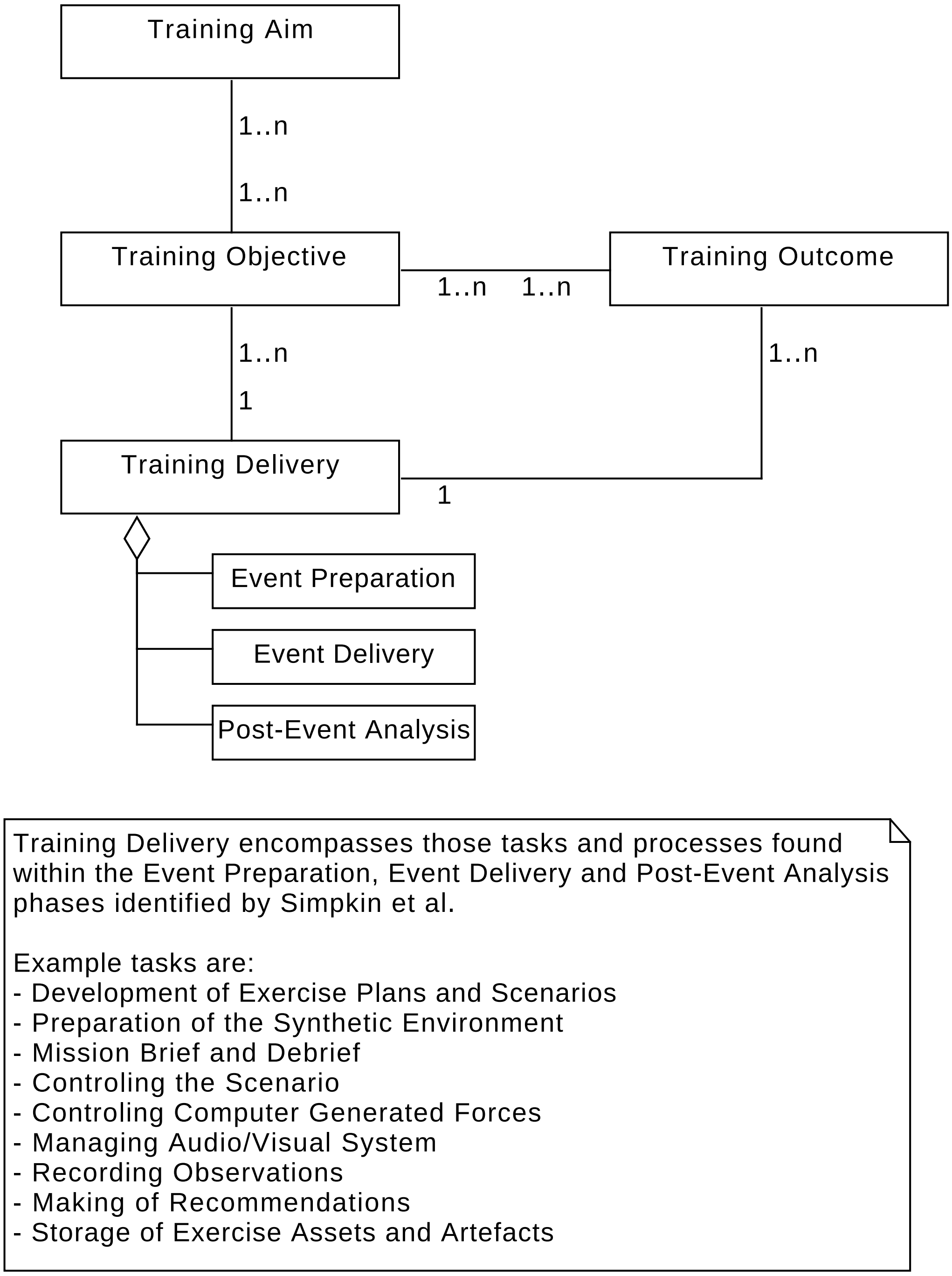
Framework Development Table

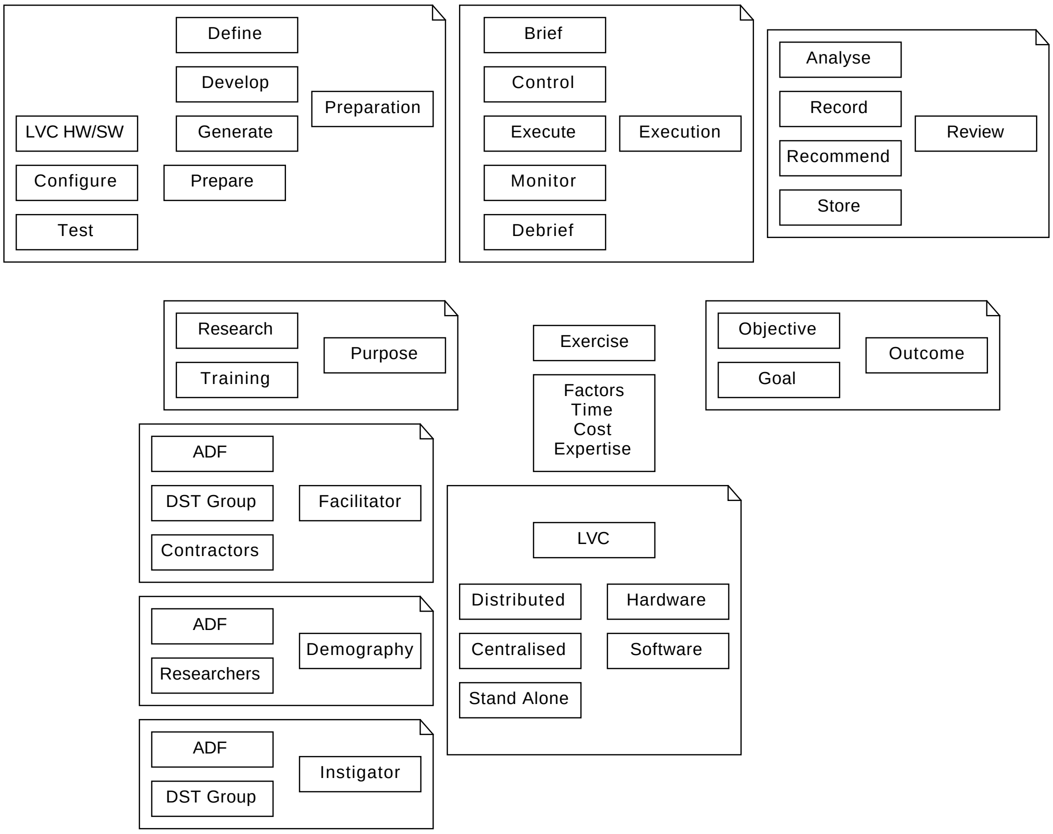
Framework Development and Evolution

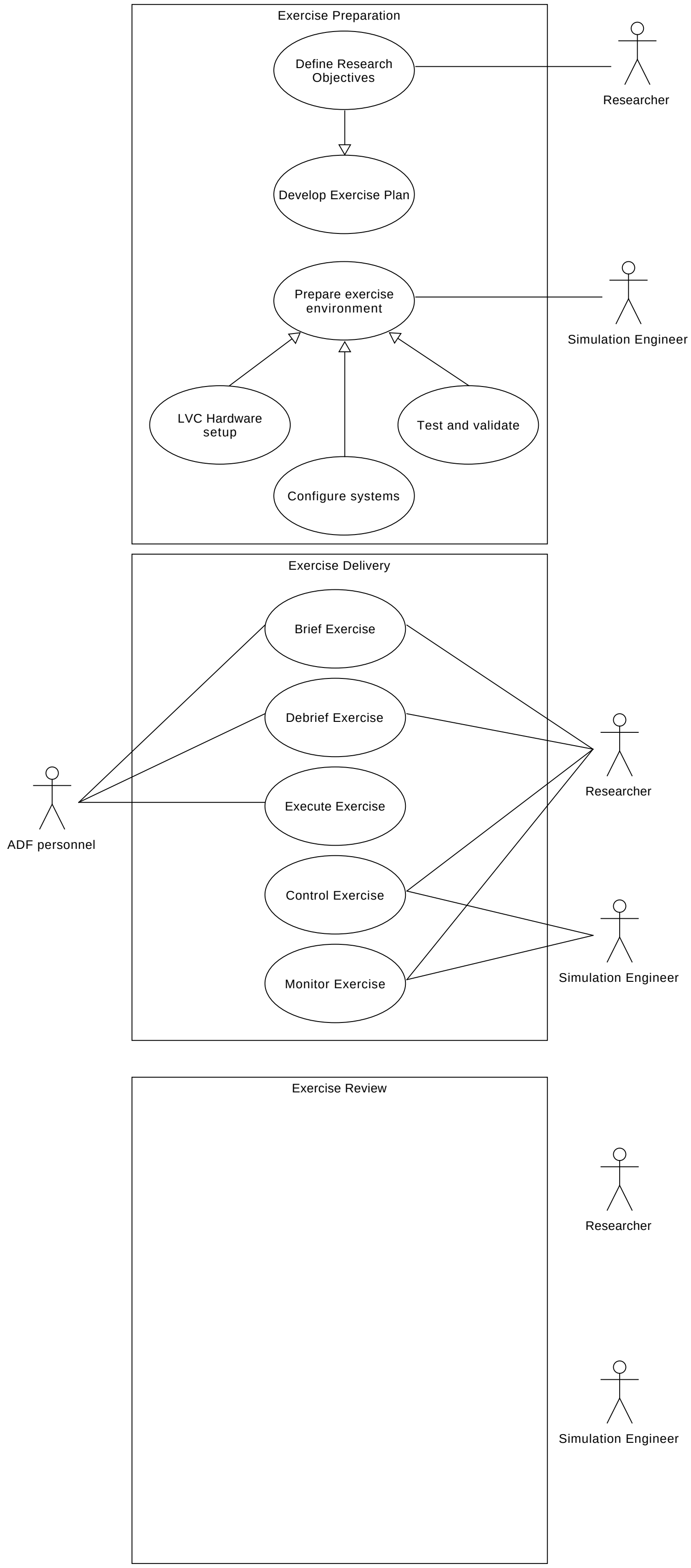
Figure	Version	Scenario used for development and testing	Coverage - Breadth	Coverage - Depth	Alignment	Representation	Effort	Reproducability	Existing theory	Dynamicism	Caption/Comment	Criteria overview	Reference
			Effort LMH, Deterministic T/F	Effort LMH, tasks exist T/F	Exists T/F	Effort LMH, Cover+Align T/F	Type LMH	Structured T/F	Theory T/F	Structure LMH			
4.4	Meta Model V0	Scenario 1: LVC Synthetic Training	FALSE	FALSE	FALSE	FALSE	HIGH	FALSE	FALSE	LOW	Initial basic phase diagram of event delivery. Extracted from "*On the Need for White Force Multipliers*" paper by Simpkin et. al.		\ref{4}
4.5	Meta Model V1	Scenario 1: LVC Synthetic Training	FALSE	FALSE	TRUE	FALSE	HIGH	FALSE	FALSE	LOW	First attempt to Meta Modeling training delivery. Based on Meta modeling concepts and theories from UML methodology. Inclusion of concepts of task, objectives, participants and outcomes.		\ref{5}
4.6	Meta Model V2	Scenario 1: LVC Synthetic Training	FALSE	HIGH, TRUE	TRUE	FALSE	HIGH	FALSE	FALSE	LOW	Refined attempt of meta modeling LVC synthetic training delivery using modeling techniques. Simplified to focus on Alignment between Exercise, Objective and Task. Initial task decomposition into a simple model of information processing		\ref{6}
4.7	Meta Model V2 - Example 1	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	FALSE	LOW	First practical application of initial meta model, using Scenario 2: High Yoyo as a subset task of the types of tasks typically found within LVC training. As we can see, Breadth and Depth are not limited, and too many tasks at inappropriate coverage are defined. Effort required is high	(criteria: coverage, breadth and depth)	\ref{7}
4.8	Meta Model V2 - Example 2	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	FALSE	LOW	Example using one task. Depth is verbose and unnecessary. Attempting to apply to all breadth tasks, too much effort.	(criteria: coverage, breadth and depth)	\ref{8}
4.9	Meta Model V2 - Example 3	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	FALSE	LOW	Decomposition of a single task identified through depth coverage. Can refer to this as a micro task. Too much unnecessary detail and effort required	(criteria: depth)	\ref{9}
4.10	Meta Model V2 - Example 4	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	FALSE	LOW	Similar micro task decomposition. Grouped elements together. Too much unnecessary detail and effort required	(criteria: depth)	\ref{10}
4.11	Meta Model V3	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	FALSE	LOW	Attempt to define and segment tasks into Macro, Meso and Micro. Attempt to address appropriate depth coverage.	(criteria: coverage, depth)	\ref{11}
4.12	Meta Model V3 - Example 1	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	FALSE	LOW	Using Macro, Meso and Micro task separation, example using High Yoyo to address depth. High level summary	(criteria: coverage, depth)	\ref{12}
4.13	Meta Model V3 - Example 2	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	TRUE	LOW	Example with the Introduction of OODA as replacement for simple model for information processing	(criteria: coverage, depth)	\ref{13}
4.14	Meta Model V3 - Example 3	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	TRUE	LOW	Introduction of OODA into meta model of Task	(criteria: coverage, depth, existing theory)	\ref{14}
4.15	Meta Model V3 - Example 4	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	TRUE	LOW	Introduction of OODA into meta model of Macro, Meso and Micro tasks.	(criteria: coverage, depth, existing theory)	\ref{15}
4.16	Meta Model V4 - Example 1	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	TRUE	MEDIUM	Overall exercise meta model begins to demonstrate dynamicism. Inclusion of OODA as a element of Task. Different approaches to visually communicate OODA	(criteria: representation)	\ref{16}
4.17	Meta Model V4 - Example 2	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	TRUE	MEDIUM	Inclusion of OODA as a element of Macro, Meso and Micro tasks. Different approaches to visually communicate OODA	(criteria: representation)	\ref{17}
4.18	Meta Model V4 - Example 3	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	TRUE	MEDIUM	Classification of Macro tasks as Objective tasks, Meso tasks as Process tasks and Micro tasks as Action tasks. Different approaches to visually communicate OODA	(criteria: representation)	\ref{18}
4.19	Meta Model V4 - Example 4	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	TRUE	MEDIUM	Classification of Objective tasks as general task, Process tasks as Process and Action tasks as Action. Different approaches to visually communicate OODA	(criteria: representation)	\ref{19}
4.20	Meta Model V4 - Example 5	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	TRUE	MEDIUM	Action loop, as part of action - formally micro task. Approach to communicate task decomposition as OODA	(criteria: representation)	\ref{20}
4.21	Meta Model V4 - Example 6	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	TRUE	MEDIUM	Example using Scenario: High Yoyo. Representation is "better". Still too much effort. No coverage selection	(criteria: representation)	\ref{21}
4.22	Meta Model V5	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	TRUE	MEDIUM	Breakdown of Action loop into individual O-O-D-A elements at meta model stage.	(criteria: representation)	\ref{22}
4.23	Meta Model V5 - Example 1	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	TRUE	MEDIUM	Example using decomposed Action loop as individual O-O-D-A elements	(criteria: representation)	\ref{23}
4.24	Meta Model V5 - Example 2	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	TRUE	MEDIUM	Example using Action loop	(criteria: representation)	\ref{24}
4.25	Meta Model V5 - Example 3	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	TRUE	MEDIUM	Representation of OODA and Action Loop as Flow Chart. Using UML methods.	(criteria: representation)	\ref{25}
4.26	Meta Model V6	Scenario 2: High Yoyo	HIGH, FALSE	HIGH, TRUE	TRUE	HIGH, TRUE	HIGH	FALSE	TRUE	MEDIUM	Introduction of Hierarchical Task Analysis as Human Factors method for appropriate task coverage depth and breadth	(criteria: existing theory, coverage, depth, overview, representation)	\ref{26}
4.27	Framework V1.0	Scenario 2: High Yoyo	HIGH, FALSE	MEDIUM, TRUE	TRUE	HIGH, TRUE	HIGH	TRUE	TRUE	MEDIUM	Introduction of Event Analysis of Systemic Teamwork structured framework as a similar domain application and structured approach. Transform meta model into a reproducible Framework.	(criteria: reproducibility) (criteria: domain agnostic)	\ref{27}
4.28	Framework V1.0	Scenario 2: High Yoyo	HIGH, FALSE	MEDIUM, TRUE	TRUE	HIGH, TRUE	HIGH	TRUE	TRUE	MEDIUM	Structured step by step directions to follow. Clearly defines and forces alignment (Objective overview)	(criteria: explicit overview requirement, reproducibility)	\ref{28}
4.29	Framework V1.1	Scenario 1: LVC Synthetic Training	MEDIUM, TRUE	MEDIUM, TRUE	TRUE	HIGH, TRUE	HIGH	TRUE	TRUE	MEDIUM	Introduction of Cynefin theory of emergent behavior and complexity. Aims to address coverage breadth and depth.	(criteria: breadth and depth, existing theory)	\ref{29}
4.30	Framework V1.1 - Example 1	Scenario 1: LVC Synthetic Training	MEDIUM, TRUE	MEDIUM, TRUE	TRUE	HIGH, TRUE	HIGH	TRUE	TRUE	MEDIUM	Task decomposition to subtask goals until able to extract OODA elements, keywords and entities. Attempt at representation.	(criteria: coverage, reproducibility, existing theory)	\ref{30}
4.31	Framework V1.2	Scenario 1: LVC Synthetic Training	MEDIUM, TRUE	MEDIUM, TRUE	TRUE	MEDIUM, TRUE	MEDIUM	TRUE	TRUE	MEDIUM	Inclusion of Information Fusion knowledge representation as OODA UML Task models based on M-OODA.	(criteria: coverage, representation, reproducibility, existing theory)	\ref{31}
4.32	Framework V1.2	Scenario 1: LVC Synthetic Training	MEDIUM, TRUE	MEDIUM, TRUE	TRUE	MEDIUM, TRUE	MEDIUM	TRUE	TRUE	HIGH	First major iteration of framework. Still missing structured outcomes and actionable items.		\ref{32}
4.33	Framework V1.2 - Example 1	Testing Scenario 1: Cleaning/Washing	MEDIUM, TRUE	MEDIUM, TRUE	TRUE	MEDIUM, TRUE	MEDIUM	TRUE	TRUE	HIGH	First use case outside of high yoyo example. Application to test and validate coverage, depth, breadth, alignment, representation, dynamicism, existing theory, reproducibility and effort.		\ref{33}
4.34	Framework V1.2 - Example 2	Testing Scenario 1: Cleaning/Washing	MEDIUM, TRUE	MEDIUM, TRUE	TRUE	MEDIUM, TRUE	MEDIUM	TRUE	TRUE	HIGH	Example of Alignment and Coverage - Breadth and Depth.		\ref{34}
4.35	Framework V1.2 - Example 3	Testing Scenario 1: Cleaning/Washing	MEDIUM, TRUE	MEDIUM, TRUE	TRUE	MEDIUM, TRUE	MEDIUM	TRUE	TRUE	HIGH	Example using method for Coverage - Breadth -> Task descriptions. If definable therefore deterministic.		\ref{35}
4.36	Framework V1.2 - Example 4	Testing Scenario 1: Cleaning/Washing	MEDIUM, TRUE	MEDIUM, TRUE	TRUE	MEDIUM, TRUE	MEDIUM	TRUE	TRUE	HIGH	Example single OODA task model based on key concepts from task description.		\ref{36}
4.37	Framework V1.2 - Example 5	Testing Scenario 1: Cleaning/Washing	MEDIUM, TRUE	MEDIUM, TRUE	TRUE	MEDIUM, TRUE	MEDIUM	TRUE	TRUE	HIGH	Collection of example OODA task models based on key concepts from task descriptions.		\ref{37}
4.38	Framework V2	Testing Scenario 1: Cleaning/Washing	MEDIUM, TRUE	MEDIUM, TRUE	TRUE	MEDIUM, TRUE	MEDIUM	TRUE	TRUE	HIGH	Iteration to address analysis and evaluation of task models from Coverage, Alignment and Representation.		\ref{38}
4.39	Framework V3	Testing Scenario 1: Cleaning/Washing	LOW, TRUE	LOW, TRUE	TRUE	LOW, TRUE	MEDIUM	TRUE	TRUE	HIGH	Further iteration to clearly articulate the flow of information through the steps of the framework. Refinement and improvement.		\ref{39}
4.40	Framework V3 - Example 1	Testing Scenario 1: Cleaning/Washing	LOW, TRUE	LOW, TRUE	TRUE	LOW, TRUE	MEDIUM	TRUE	TRUE	HIGH	Example application of refined framework version to already examined simple test scenario.		\ref{40}
4.41	Framework V3 - Example 2	Testing Scenario 1: Cleaning/Washing	LOW, TRUE	LOW, TRUE	TRUE	LOW, TRUE	MEDIUM	TRUE	TRUE	HIGH	Example of simple method and approach for the deterministic evaluation of Coverage - Breadth. Task description analysis to generate OODA elements.		\ref{41}
4.42	Framework V3 - Example 3	Testing Scenario 1: Cleaning/Washing	LOW, TRUE	LOW, TRUE	TRUE	LOW, TRUE	MEDIUM	TRUE	TRUE	HIGH	Example generation of Finite State Machines to define fully deterministic observable states. Forces Coverage - Breadth		\ref{42}
4.43	Framework V3 - Example 4	Testing Scenario 1: Cleaning/Washing	LOW, TRUE	LOW, TRUE	TRUE	LOW, TRUE	MEDIUM	TRUE	TRUE	HIGH	Example simple data collection transcription. Time stamped to identify key ideas and decisions.		\ref{43}
4.44	Framework V3 - Example 5	Testing Scenario 1: Cleaning/Washing	LOW, TRUE	LOW, TRUE	TRUE	LOW, TRUE	MEDIUM	TRUE	TRUE	HIGH	Example mapping of task models using collected data and time stamps over a period of time. Visual representation of OODA in action.		\ref{44}
4.45	Framework V3 - Example 6	Testing Scenario 1: Cleaning/Washing	LOW, TRUE	LOW, TRUE	TRUE	LOW, TRUE	MEDIUM	TRUE	TRUE	HIGH	Example of updated task model based on data collected.		\ref{45}
4.46	Framework V3 - Example 7	Testing Scenario 1: Cleaning/Washing	LOW, TRUE	LOW, TRUE	TRUE	LOW, TRUE	MEDIUM	TRUE	TRUE	HIGH	Example of expanded timeline using relationship arrows to properly communicate observable states, orientation entities, decisions and actions in "real time".		\ref{46}
4.47	Framework V3	Testing Scenario 1: Cleaning/Washing	LOW, TRUE	LOW, TRUE	TRUE	LOW, TRUE	MEDIUM	TRUE	TRUE	HIGH	High level breakdown and representation of current iteration of framework as a minimal phase diagram, color coded to align with research areas. Phase diagram indicates direction of information flow and ability to move backwards and forwards during the Describe, Collect and Analyse phases.		\ref{47}
4.48	Framework V4	Testing Scenario 2: Cooking	LOW, TRUE	LOW, TRUE	TRUE	LOW, TRUE	LOW	TRUE	TRUE	HIGH	Iterated Framework flowchart to simplify and clearly articulate the overall architecture and structure of DCARR framework for easy consumption. Iteration used as basis for full step by step write up of DCARR framework.		\ref{48}
4.49	Framework V4 - Example 1	Testing Scenario 2: Cooking	LOW, TRUE	LOW, TRUE	TRUE	LOW, TRUE	LOW	TRUE	TRUE	HIGH	Example using a different Testing Scenario: Cooking. Demonstrates Alignment and Coverage.		\ref{49}
4.50	Framework V4 - Example 2	Testing Scenario 2: Cooking	LOW, TRUE	LOW, TRUE	TRUE	LOW, TRUE	LOW	TRUE	TRUE	HIGH	Example of observable state as Finite State Machine.		\ref{50}
4.51	Framework V4 - Example 3	Testing Scenario 2: Cooking	LOW, TRUE	LOW, TRUE	TRUE	LOW, TRUE	LOW	TRUE	TRUE	HIGH	Example of described tasks as OODA Task models		\ref{51}
4.52	Framework V4 - Example 4	Testing Scenario 2: Cooking	LOW, TRUE	LOW, TRUE	TRUE	LOW, TRUE	LOW	TRUE	TRUE	HIGH	Plotting of Task models against timeline to identify key decisions.		\ref{52}

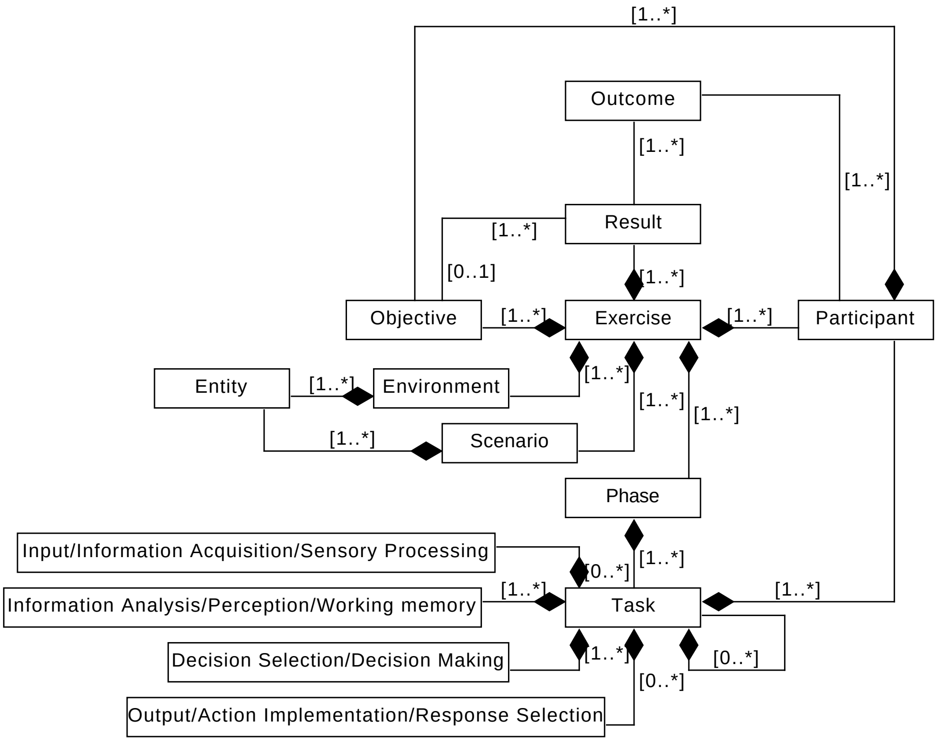
Appendix C

Framework Development and Evolution Supplementary Figures

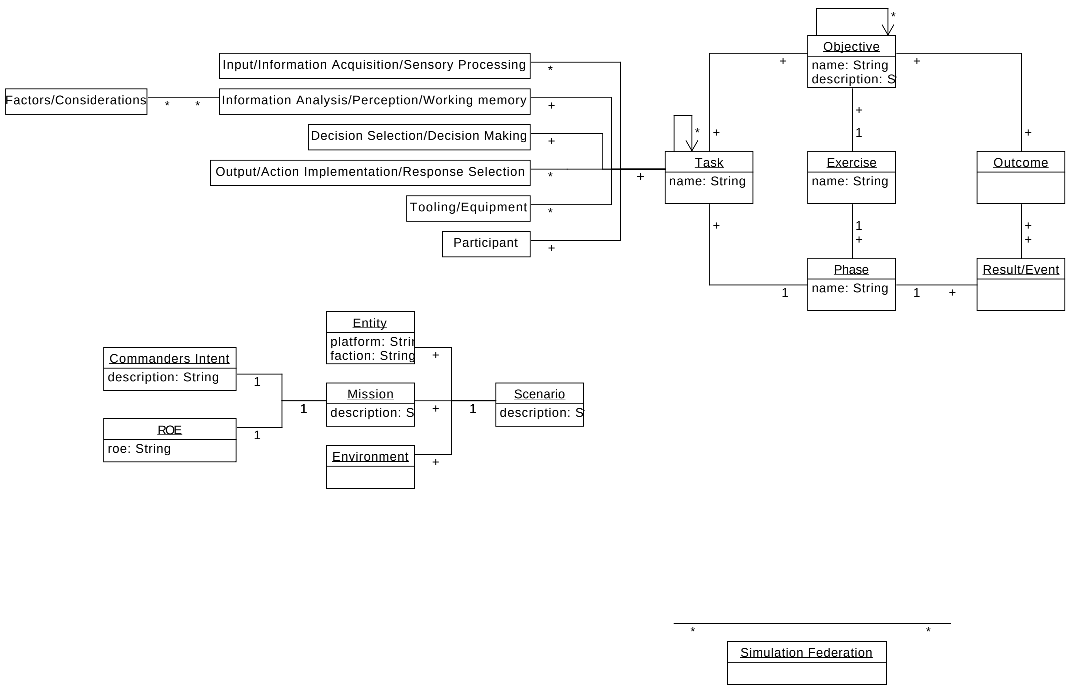

Training Aim
1..n
1..n
Training Objective
Training Outcome
1..n
1..n
1..n
1
1..n
Training Delivery
1
Event Preparation
Event Delivery
Post-Event Analysis
Training Delivery encompasses those tasks and processes found within the Event Preparation, Event Delivery and Post-Event Analysis phases identified by Simpkin et al.
Example tasks are:
- Development of Exercise Plans and Scenarios
- Preparation of the Synthetic Environment
- Mission Brief and Debrief
- Controling the Scenario
- Controling Computer Generated Forces
- Managing Audio/Visual System
- Recording Observations
- Making of Recommendations
- Storage of Exercise Assets and Artefacts


Define
Develop
Generate
Preparation
LVC HW/SW
Configure
Prepare
Test
Brief
Control
Execute
Execution
Monitor
Debrief
Analyse
Record
Recommend
Store
Review
Research
Training
Purpose
Exercise
Factors
Time
Cost
Expertise
Objective
Goal
Outcome
ADF
DST Group
Contractors
Facilitator
LVC
Distributed
Centralised
Stand Alone
Hardware
Software
ADF
Researchers
Demography
ADF
DST Group
Instigator

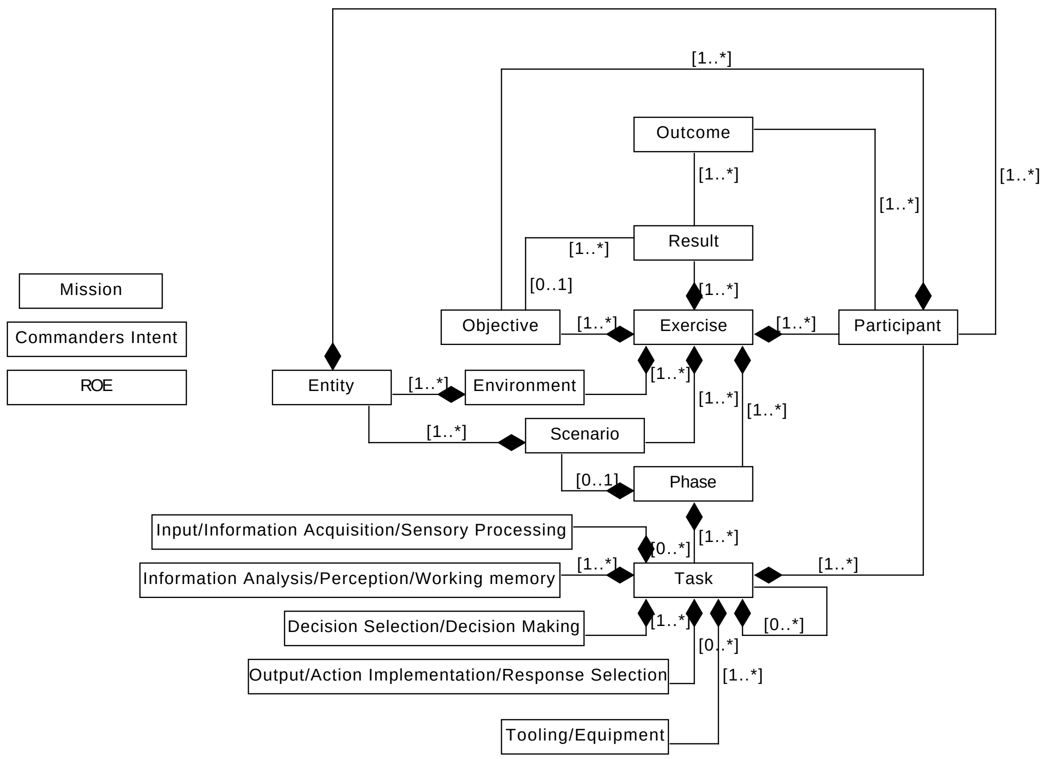

Exercise Preparation
Define Research Objectives
Researcher
Develop Exercise Plan
Prepare exercise environment
Simulation Engineer
LVC Hardware setup
Test and validate
Configure systems
Exercise Delivery
Brief Exercise
Debrief Exercise
Researcher
ADF personnel
Execute Exercise
Control Exercise
Simulation Engineer
Monitor Exercise
Exercise Review
Researcher
Simulation Engineer



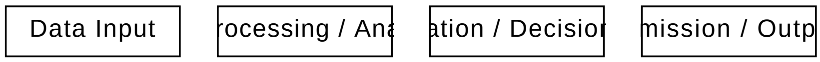
At the heart is the exercise
An exercise can have more than one result
It can also have more than one objective
The result and oobjective are not a 1=1 ratio
The objective of an exercise may not be achieved - that is still a result
An result of the excerise may be unexpected
Regardless of the result, each participant will achieve an outcome.
Each exercise is made up of participants (trainees and whiteforce)
An exercise is conducted with a scenario, in an envirnonment, containing entities
There are multiple phases that occur during an exercise (preparation, delivery and post event review)
Each phase contains a series of tasks, where tasks are completed by particpants

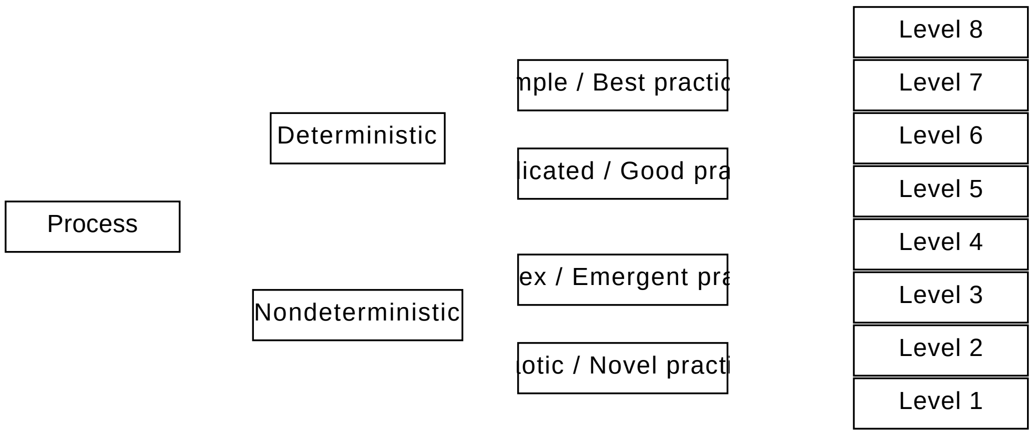


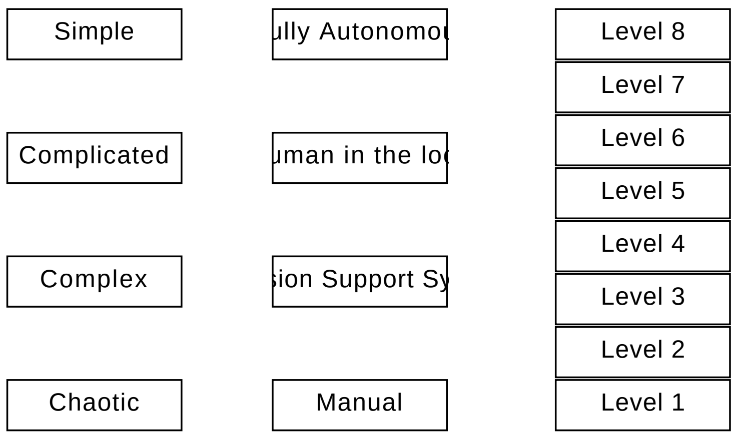
Input/Information Acquisition/Sensory Processing
Factors/Considerations
Information Analysis/Perception/Working memory
Decision Selection/Decision Making
Output/Action Implementation/Response Selection
Tooling/Equipment
Participant
Objective
name: String
description: S
Task
name: String
Exercise
name: String
Outcome
Phase
name: String
Result/Event
Entity
platform: Strir
faction: String
Commanders Intent
description: String
Mission
description: S
Scenario
description: S
ROE
roe: String
Environment
Simulation Federation

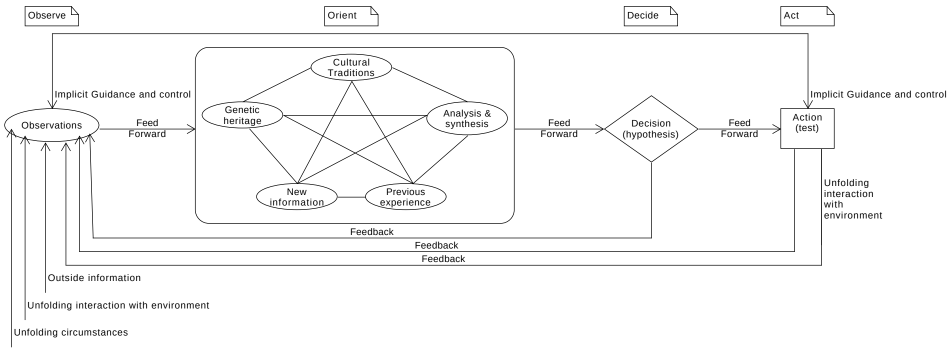


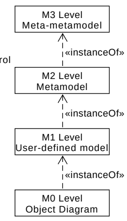
Mission
Commanders Intent
ROE
Outcome
Result
Objective
Exercise
Participant
Entity
Environment
Scenario
Phase
Task
Input/Information Acquisition/Sensory Processing
Information Analysis/Perception/Working memory
Decision Selection/Decision Making
Output/Action Implementation/Response Selection
Tooling/Equipment
[1..*]
[0..1]
[0..*]

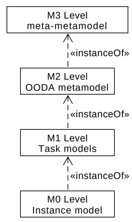

Data Input
rocessing / Ana
ation / Decisio
mission / Outp

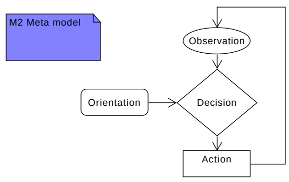

Process
Deterministic
Nondeterministic
nple / Best practic
icated / Good pra
ex / Emergent pra
otic / Novel pract
Level 8
Level 7
Level 6
Level 5
Level 4
Level 3
Level 2
Level 1

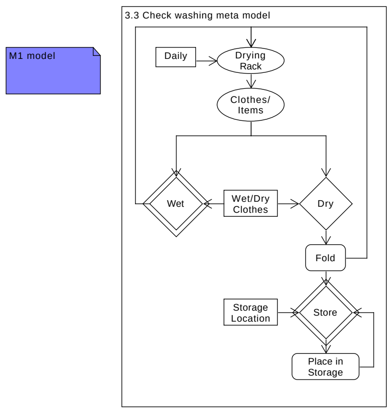

Simple
Complicated
Complex
Chaotic
ully Autonomo
uman in the lo
ion Support Sy
Manual
Level 8
Level 7
Level 6
Level 5
Level 4
Level 3
Level 2
Level 1

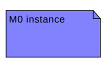

Observe
Orient
Decide
Act
Implicit Guidance and control
Observations
Feed Forward
Cultural Traditions
Genetic heritage
Analysis & synthesis
New information
Previous experience
Feed Forward
Decision (hypothesis)
Feed Forward
Action (test)
Implicit Guidance and control
Unfolding interaction with environment
Feedback
Feedback
Feedback
Outside information
Unfolding interaction with environment
Unfolding circumstances

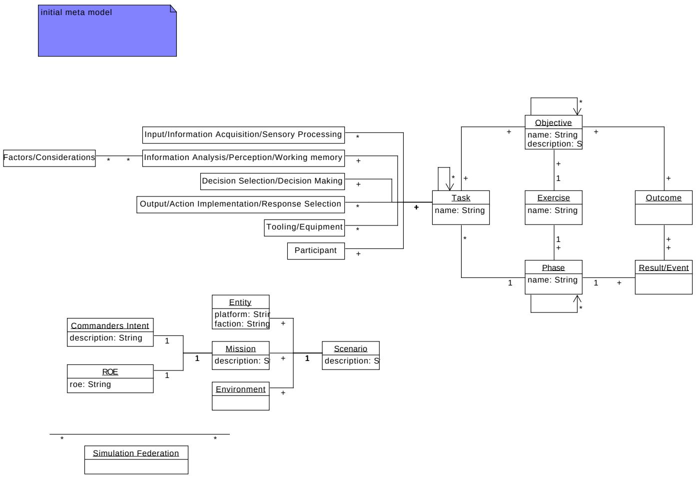

M3 Level
Meta-metamodel
«instanceOf»
M2 Level
Metamodel
«instanceOf»
M1 Level
User-defined model
«instanceOf»
M0 Level
Object Diagram


M3 Level
meta-metamodel
«instanceOf»
M2 Level
OODA metamodel
«instanceOf»
M1 Level
Task models
«instanceOf»
M0 Level
Instance model

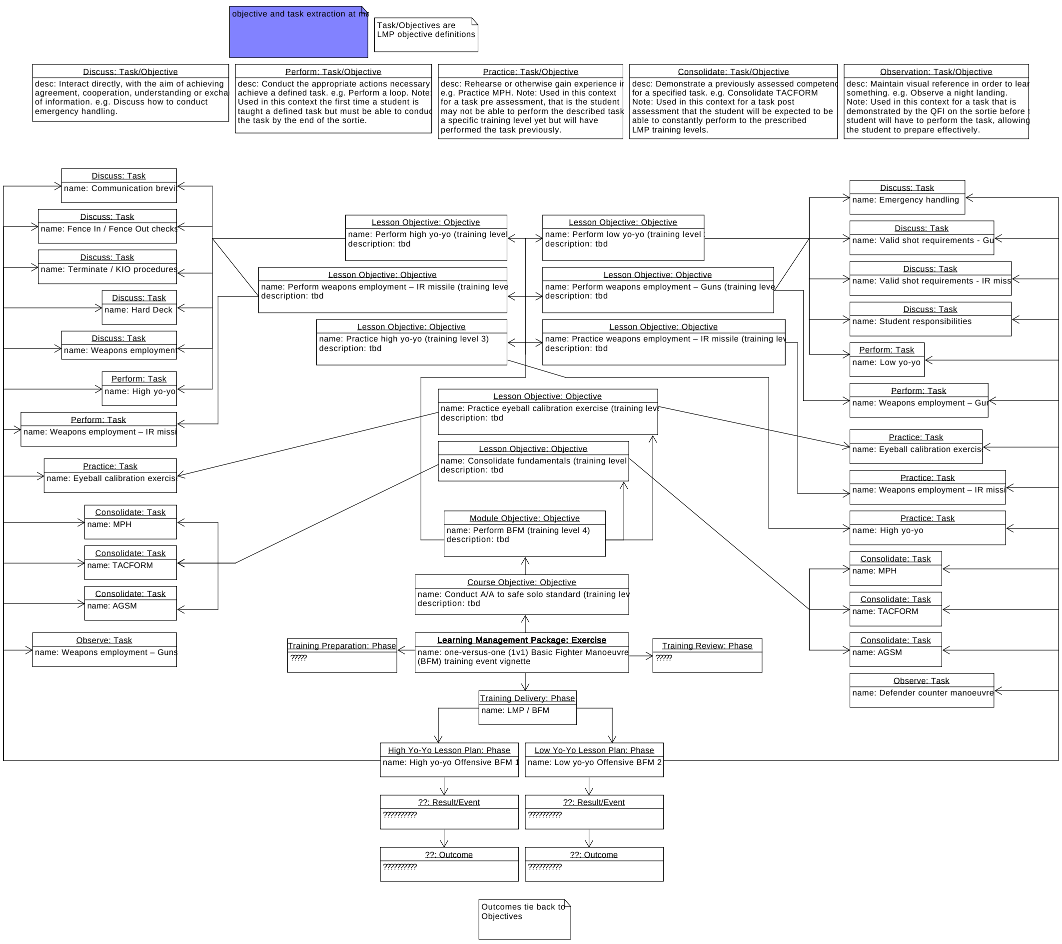

M2 Meta model
Observation
Orientation
Decision
Action


M1 model
3.3 Check washing meta model
Daily
Drying Rack
Clothes/ Items
Wet
Wet/Dry Clothes
Dry
Fold
Storage Location
Store
Place in Storage

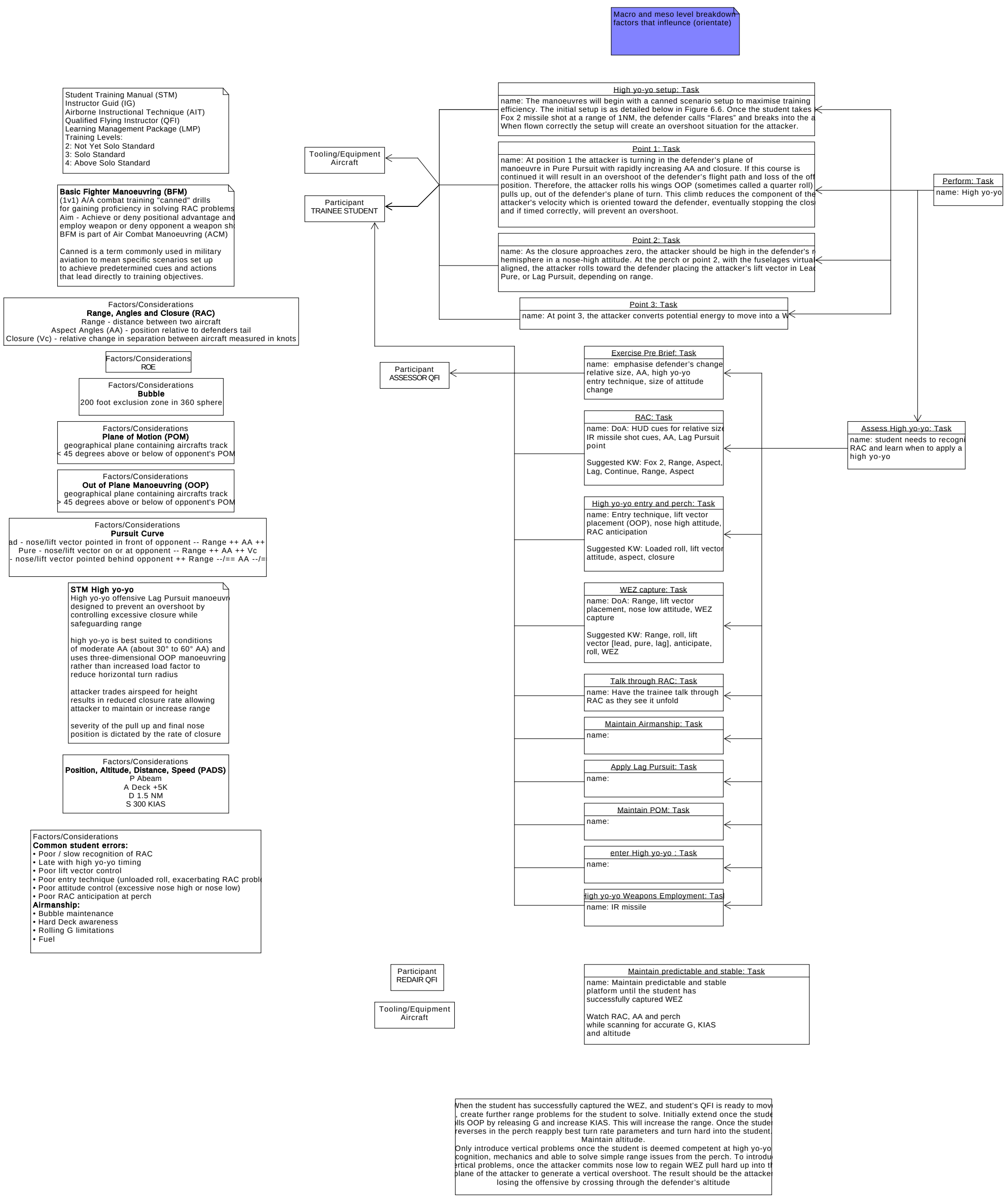

M0 instance

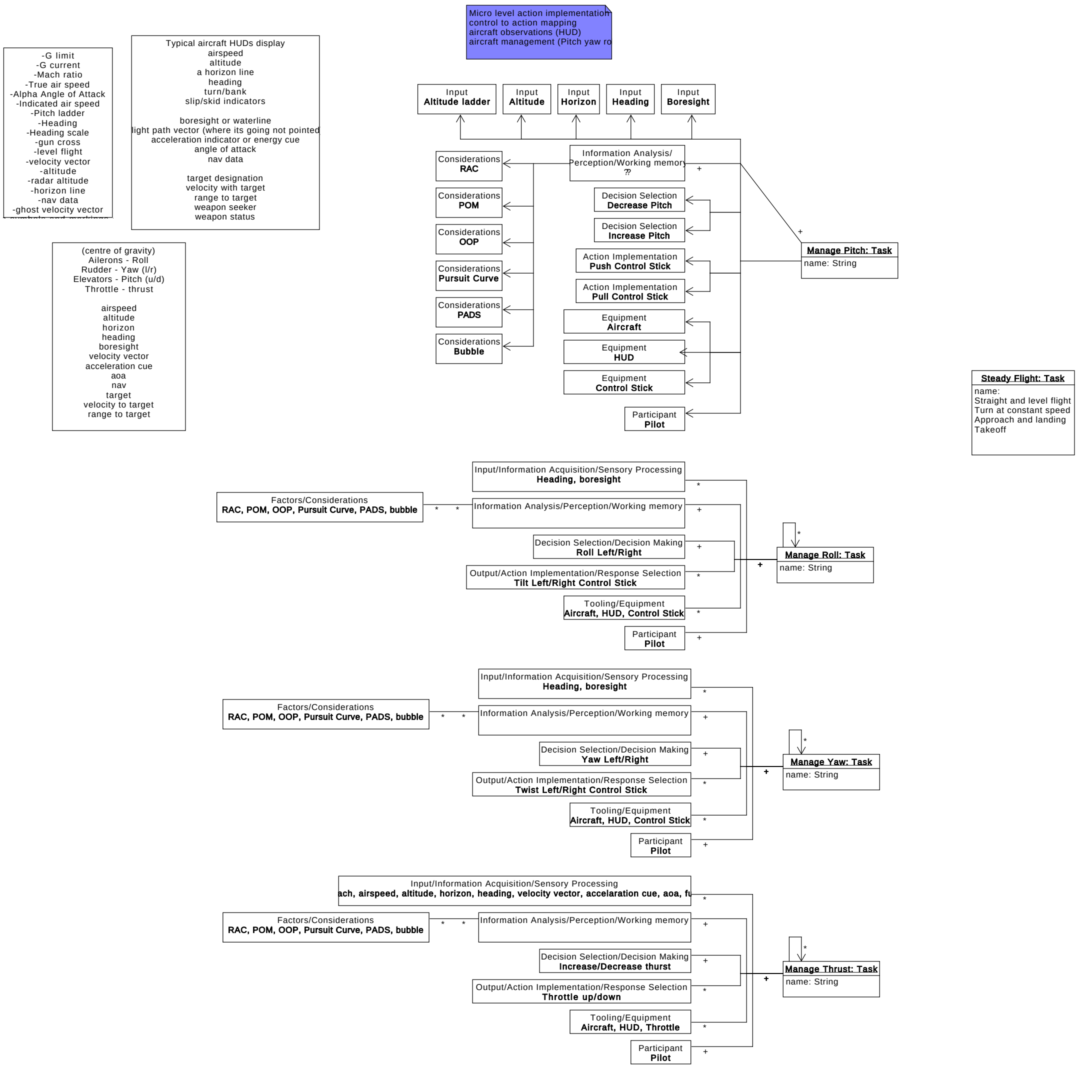


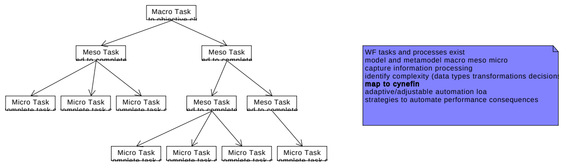
initial meta model
Objective
name: String
Task
name: String
Exercise
name: String
Outcome
Phase
name: String
Result/Event
Input/Information Acquisition/Sensory Processing
Information Analysis/Perception/Working memory
Decision Selection/Decision Making
Output/Action Implementation/Response Selection
Tooling/Equipment
Participant
Factors/Considerations
Entity
Commanders Intent
description: String
Mission
Scenario
ROE
roe: String
Environment
Simulation Federation



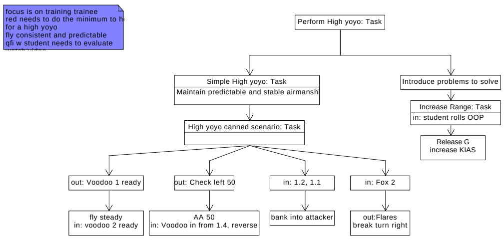
Task/Objectives are LMP objective definitions
Discuss: Task/Objective
desc: Interact directly, with the aim of achieving agreement, cooperation, understanding or excha of information. e.g. Discuss how to conduct emergency handling.
Perform: Task/Objective
desc: Conduct the appropriate actions necessary achieve a defined task. e.g. Perform a loop. Note: Used in this context the first time a student is taught a defined task but must be able to conduc the task by the end of the sortie.
Practice: Task/Objective
desc: Rehearse or otherwise gain experience i e.g. Practice MPH. Note: Used in this context for a task pre assessment, that is the student may not be able to perform the described task a specific training level yet but will have performed the task previously.
Consolidate: Task/Objective
desc: Demonstrate a previously assessed competenc for a specified task. e.g. Consolidate TACFORM Note: Used in this context for a task post assessment that the student will be expected to be able to constantly perform to the prescribed LMP training levels.
Observation: Task/Objective
desc: Maintain visual reference in order to lear something. e.g. Observe a night landing. Note: Used in this context for a task that is demonstrated by the QFI on the sortie before t student will have to perform the task, allowing the student to prepare effectively.
Discuss: Task
name: Communication brevi
name: Fence In / Fence Out checks
name: Terminate / KIO procedures
name: Hard Deck
name: Weapons employmen
Perform: Task
name: High yo-yo
name: Weapons employment – IR missi
Practice: Task
name: Eyeball calibration exercis
Consolidate: Task
name: MPH
name: TACFORM
name: AGSM
Observe: Task
name: Weapons employment – Guns
Lesson Objective: Objective
name: Perform high yo-yo (training level
description: tbd
name: Perform low yo-yo (training level
name: Perform weapons employment – IR missile (training leve
name: Perform weapons employment – Guns (training leve
name: Practice high yo-yo (training level 3)
name: Practice weapons employment – IR missile (training lev
name: Practice eyeball calibration exercise (training lev
name: Consolidate fundamentals (training level
Module Objective: Objective
name: Perform BFM (training level 4)
Course Objective: Objective
name: Conduct A/A to safe solo standard (training lev
Learning Management Package: Exercise
name: one-versus-one (1v1) Basic Fighter Manoeuvre (BFM) training event vignette
Training Preparation: Phase
?????
Training Review: Phase
?????
Training Delivery: Phase
name: LMP / BFM
High Yo-Yo Lesson Plan: Phase
name: High yo-yo Offensive BFM 1
Low Yo-Yo Lesson Plan: Phase
name: Low yo-yo Offensive BFM 2
??: Result/Event
??????????
??: Outcome
??????????
Outcomes tie back to Objectives
name: Emergency handling
name: Valid shot requirements - Gu
name: Valid shot requirements - IR miss
name: Student responsibilities
name: Low yo-yo
name: Weapons employment – Gu
name: Eyeball calibration exercis
name: Weapons employment – IR missi
name: High yo-yo
name: Defender counter manoeuvre


Macro and meso level breakdown
factors that infleunce (orientate)
Student Training Manual (STM)
Instructor Guid (IG)
Airborne Instructional Technique (AIT)
Qualified Flying Instructor (QFI)
Learning Management Package (LMP)
Training Levels:
2: Not Yet Solo Standard
3: Solo Standard
4: Above Solo Standard
Basic Fighter Manoeuvring (BFM)
(1v1) A/A combat training "canned" drills
for gaining proficiency in solving RAC problems
Aim - Achieve or deny positional advantage and
employ weapon or deny opponent a weapon sh
BFM is part of Air Combat Manoeuvring (ACM)
Canned is a term commonly used in military
aviation to mean specific scenarios set up
to achieve predetermined cues and actions
that lead directly to training objectives.
Factors/Considerations
Range, Angles and Closure (RAC)
Range - distance between two aircraft
Aspect Angles (AA) - position relative to defenders tail
Closure (Vc) - relative change in separation between aircraft measured in knots
Factors/Considerations
ROE
Factors/Considerations
Bubble
200 foot exclusion zone in 360 sphere
Factors/Considerations
Plane of Motion (POM)
geographical plane containing aircrafts track
< 45 degrees above or below of opponent's POM
Factors/Considerations
Out of Plane Manoeuvring (OOP)
geographical plane containing aircrafts track
> 45 degrees above or below of opponent's POM
Factors/Considerations
Pursuit Curve
ad - nose/lift vector pointed in front of opponent -- Range ++ AA ++
Pure - nose/lift vector on or at opponent -- Range ++ AA ++ Vc
- nose/lift vector pointed behind opponent ++ Range --/== AA --/=
STM High yo-yo
High yo-yo offensive Lag Pursuit manoeuv
designed to prevent an overshoot by
controlling excessive closure while
safeguarding range
high yo-yo is best suited to conditions
of moderate AA (about 30° to 60° AA) and
uses three-dimensional OOP manoeuvring
rather than increased load factor to
reduce horizontal turn radius
attacker trades airspeed for height
results in reduced closure rate allowing
attacker to maintain or increase range
severity of the pull up and final nose
position is dictated by the rate of closure
Factors/Considerations
Position, Altitude, Distance, Speed (PADS)
P Abeam
A Deck +5K
D 1.5 NM
S 300 KIAS
Factors/Considerations
Common student errors:
• Poor / slow recognition of RAC
• Late with high yo-yo timing
• Poor lift vector control
• Poor entry technique (unloaded roll, exacerbating RAC probl
• Poor attitude control (excessive nose high or nose low)
• Poor RAC anticipation at perch
Airmanship:
• Bubble maintenance
• Hard Deck awareness
• Rolling G limitations
• Fuel
Tooling/Equipment
Aircraft
Participant
TRAINEE STUDENT
High yo-yo setup: Task
name: The manoeuvres will begin with a canned scenario setup to maximise training
efficiency. The initial setup is as detailed below in Figure 6.6. Once the student takes
Fox 2 missile shot at a range of 1NM, the defender calls “Flares” and breaks into the a
When flown correctly the setup will create an overshoot situation for the attacker.
Point 1: Task
name: At position 1 the attacker is turning in the defender’s plane of
manoeuvre in Pure Pursuit with rapidly increasing AA and closure. If this course is
continued it will result in an overshoot of the defender’s flight path and loss of the off
position. Therefore, the attacker rolls his wings OOP (sometimes called a quarter roll)
pulls up, out of the defender's plane of turn. This climb reduces the component of the
attacker's velocity which is oriented toward the defender, eventually stopping the clos
and if timed correctly, will prevent an overshoot.
Point 2: Task
name: As the closure approaches zero, the attacker should be high in the defender's r
hemisphere in a nose-high attitude. At the perch or point 2, with the fuselages virtual
aligned, the attacker rolls toward the defender placing the attacker’s lift vector in Lead
Pure, or Lag Pursuit, depending on range.
Point 3: Task
name: At point 3, the attacker converts potential energy to move into a W
Perform: Task
name: High yo-yo
Participant
ASSESSOR QFI
Exercise Pre Brief: Task
name: emphasise defender’s change
relative size, AA, high yo-yo
entry technique, size of attitude
change
RAC: Task
name: DoA: HUD cues for relative size
IR missile shot cues, AA, Lag Pursuit
point
Suggested KW: Fox 2, Range, Aspect,
Lag, Continue, Range, Aspect
Assess High yo-yo: Task
name: student needs to recogni
RAC and learn when to apply a
high yo-yo
High yo-yo entry and perch: Task
name: Entry technique, lift vector
placement (OOP), nose high attitude,
RAC anticipation
Suggested KW: Loaded roll, lift vector
attitude, aspect, closure
WEZ capture: Task
name: DoA: Range, lift vector
placement, nose low attitude, WEZ
capture
Suggested KW: Range, roll, lift
vector [lead, pure, lag], anticipate,
roll, WEZ
Talk through RAC: Task
name: Have the trainee talk through
RAC as they see it unfold
Maintain Airmanship: Task
name:
Apply Lag Pursuit: Task
name:
Maintain POM: Task
name:
enter High yo-yo : Task
name:
High yo-yo Weapons Employment: Tas
name: IR missile
Participant
REDAIR QFI
Tooling/Equipment
Aircraft
Maintain predictable and stable: Task
name: Maintain predictable and stable
platform until the student has
successfully captured WEZ
Watch RAC, AA and perch
while scanning for accurate G, KIAS
and altitude
When the student has successfully captured the WEZ, and student’s QFI is ready to mov
, create further range problems for the student to solve. Initially extend once the stude
lls OOP by releasing G and increase KIAS. This will increase the range. Once the stude
reverses in the perch reapply best turn rate parameters and turn hard into the student.
Maintain altitude.
Only introduce vertical problems once the student is deemed competent at high yo-yo
cognition, mechanics and able to solve simple range issues from the perch. To introdu
rtical problems, once the attacker commits nose low to regain WEZ pull hard up into t
plane of the attacker to generate a vertical overshoot. The result should be the attacke
losing the offensive by crossing through the defender’s altitude

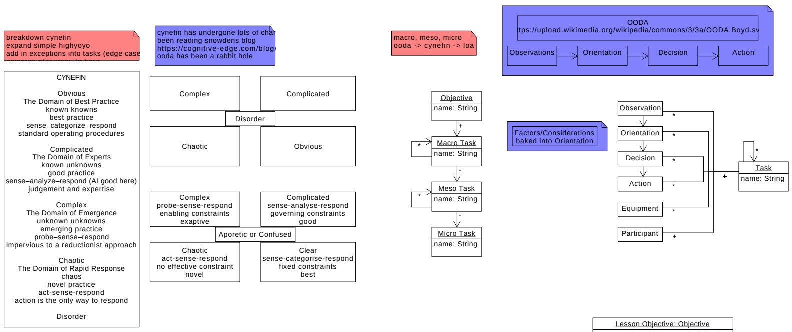

Micro level action implementation
control to action mapping
aircraft observations (HUD)
aircraft management (Pitch yaw ro
-G limit
-G current
-Mach ratio
-True air speed
-Alpha Angle of Attack
-Indicated air speed
-Pitch ladder
-Heading
-Heading scale
-gun cross
-level flight
-velocity vector
-altitude
-radar altitude
-horizon line
-nav data
-ghost velocity vector
Typical aircraft HUDs display
airspeed
altitude
a horizon line
heading
turn/bank
slip/skid indicators
boresight or waterline
light path vector (where its going not pointed
acceleration indicator or energy cue
angle of attack
nav data
target designation
velocity with target
range to target
weapon seeker
weapon status
(centre of gravity)
Ailerons - Roll
Rudder - Yaw (l/r)
Elevators - Pitch (u/d)
Throttle - thrust
airspeed
altitude
horizon
heading
boresight
velocity vector
acceleration cue
aoa
nav
target
velocity to target
range to target
Input Altitude ladder
Input Altitude
Input Horizon
Input Heading
Input Boresight
Considerations RAC
Considerations POM
Considerations OOP
Considerations Pursuit Curve
Considerations PADS
Considerations Bubble
Information Analysis/ Perception/Working memory ??
Decision Selection Decrease Pitch
Decision Selection Increase Pitch
Action Implementation Push Control Stick
Action Implementation Pull Control Stick
Equipment Aircraft
Equipment HUD
Equipment Control Stick
Participant Pilot
Manage Pitch: Task
name: String
Steady Flight: Task
name:
Straight and level flight
Turn at constant speed
Approach and landing
Takeoff
Input/Information Acquisition/Sensory Processing Heading, boresight
Factors/Considerations RAC, POM, OOP, Pursuit Curve, PADS, bubble
Information Analysis/Perception/Working memory
Decision Selection/Decision Making Roll Left/Right
Output/Action Implementation/Response Selection Tilt Left/Right Control Stick
Tooling/Equipment Aircraft, HUD, Control Stick
Participant Pilot
Manage Roll: Task
name: String
Input/Information Acquisition/Sensory Processing Heading, boresight
Factors/Considerations RAC, POM, OOP, Pursuit Curve, PADS, bubble
Information Analysis/Perception/Working memory
Decision Selection/Decision Making Yaw Left/Right
Output/Action Implementation/Response Selection Twist Left/Right Control Stick
Tooling/Equipment Aircraft, HUD, Control Stick
Participant Pilot
Manage Yaw: Task
name: String
Input/Information Acquisition/Sensory Processing ach, airspeed, altitude, horizon, heading, velocity vector, accelaration cue, aoa, fu
Factors/Considerations RAC, POM, OOP, Pursuit Curve, PADS, bubble
Information Analysis/Perception/Working memory
Decision Selection/Decision Making Increase/Decrease thurst
Output/Action Implementation/Response Selection Throttle up/down
Tooling/Equipment Aircraft, HUD, Throttle
Participant Pilot
Manage Thrust: Task
name: String



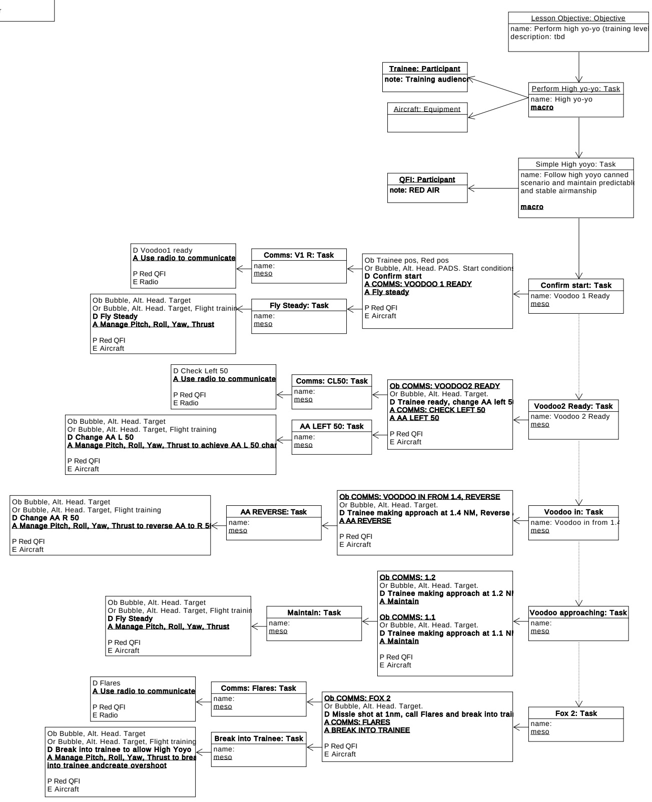
Macro Task
Meso Task
Meso Task
Micro Task
Micro Task
Micro Task
Meso Task
Meso Task
Micro Task
Micro Task
Micro Task
Micro Task
WF tasks and processes exist
model and metamodel macro meso micro
capture information processing
identify complexity (data types transformations decisions
map to cynefin
adaptive/adjustable automation loa
strategies to automate performance consequences



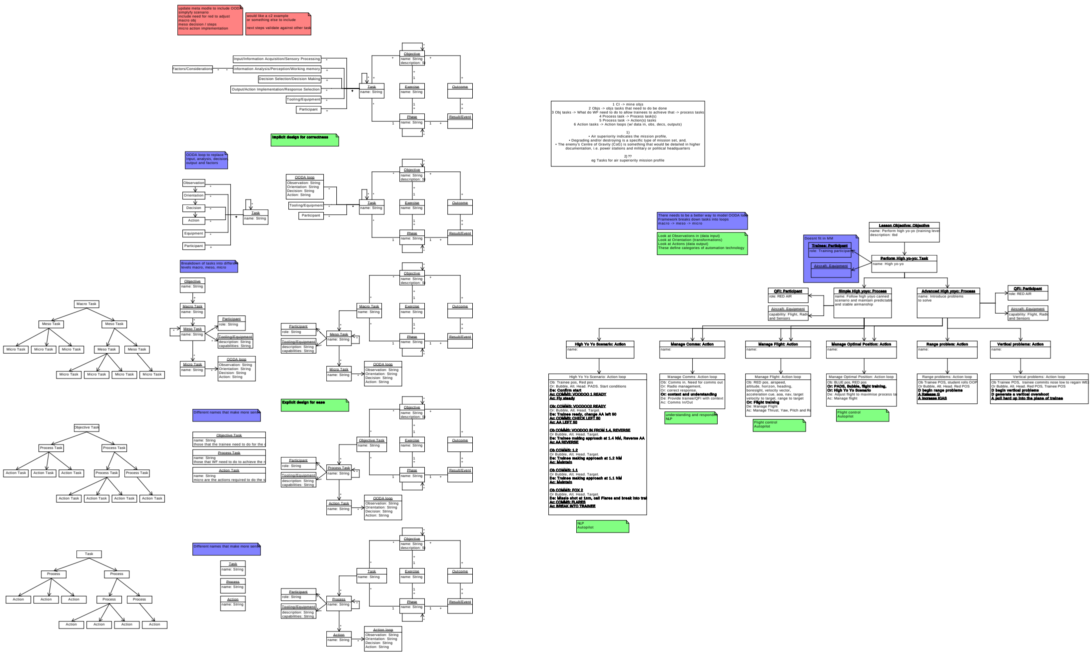
focus is on training trainee
red needs to do the minimum to h
for a high yoyo
fly consistent and predictable
qfi w student needs to evaluate
Perform High yoyo: Task
Simple High yoyo: Task
Maintain predictable and stable airmanshi
Introduce problems to solve
Increase Range: Task
in: student rolls OOP
Release G
increase KIAS
High yoyo canned scenario: Task
out: Voodoo 1 ready
out: Check left 50
in: 1.2, 1.1
in: Fox 2
fly steady
in: voodoo 2 ready
AA 50
in: Voodoo in from 1.4, reverse
bank into attacker
out:Flares
break turn right

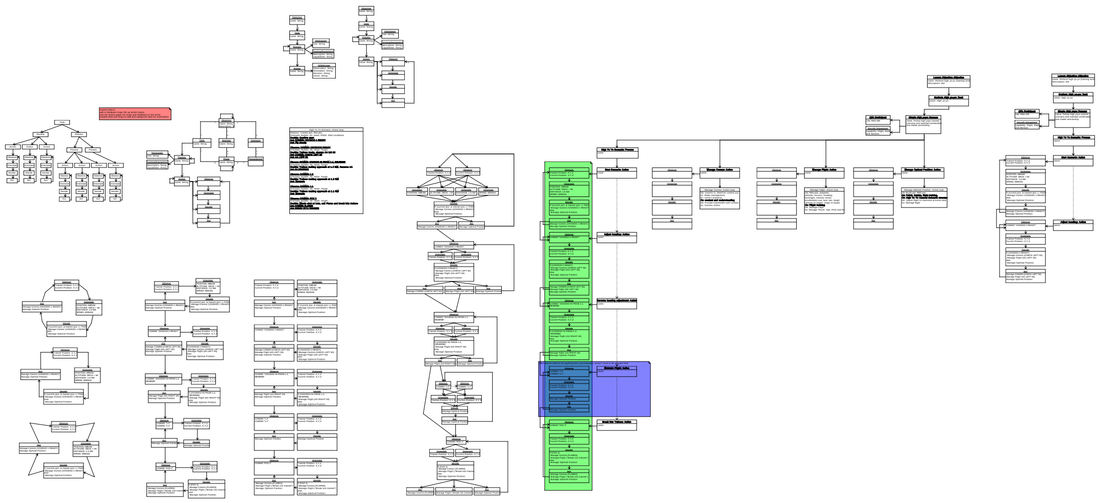


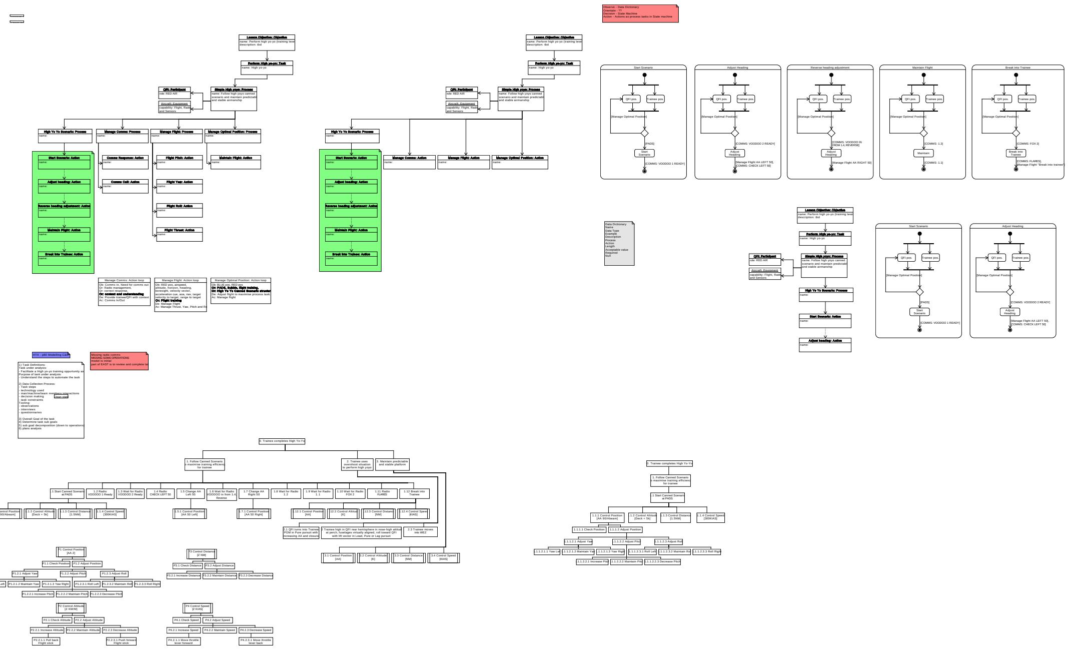
breakdown cynefin
expand simple highyoyo
add in exceptions into tasks (edge case
cynefin has undergone lots of cha
been reading snowdens blog
https://cognitive-edge.com/blog
ooda has been a rabbit hole
macro, meso, micro
ooda -> cynefin -> loa
OODA
ttps://upload.wikimedia.org/wikipedia/commons/3/3a/OODA.Boyd.sv
Observations
Orientation
Decision
Action
CYNEFIN
Obvious
The Domain of Best Practice
known knowns
best practice
sense–categorize–respond
standard operating procedures
Complicated
The Domain of Experts
known unknowns
good practice
sense–analyze–respond (AI good here)
judgement and expertise
Complex
The Domain of Emergence
unknown unknowns
emerging practice
probe–sense–respond
impervious to a reductionist approach
Chaotic
The Domain of Rapid Response
chaos
novel practice
act-sense-respond
action is the only way to respond
Disorder
Complex
Complicated
Disorder
Chaotic
Obvious
Complex
probe-sense-respond
enabling constraints
exaptive
Complicated
sense-analyse-respond
governing constraints
good
Aporetic or Confused
Chaotic
act-sense-respond
no effective constraint
novel
Clear
sense-categorise-respond
fixed constraints
best
Objective
name: String
Macro Task
name: String
Meso Task
name: String
Micro Task
name: String
Factors/Considerations
baked into Orientation
Observation
Orientation
Decision
Action
Equipment
Participant
Task
name: String

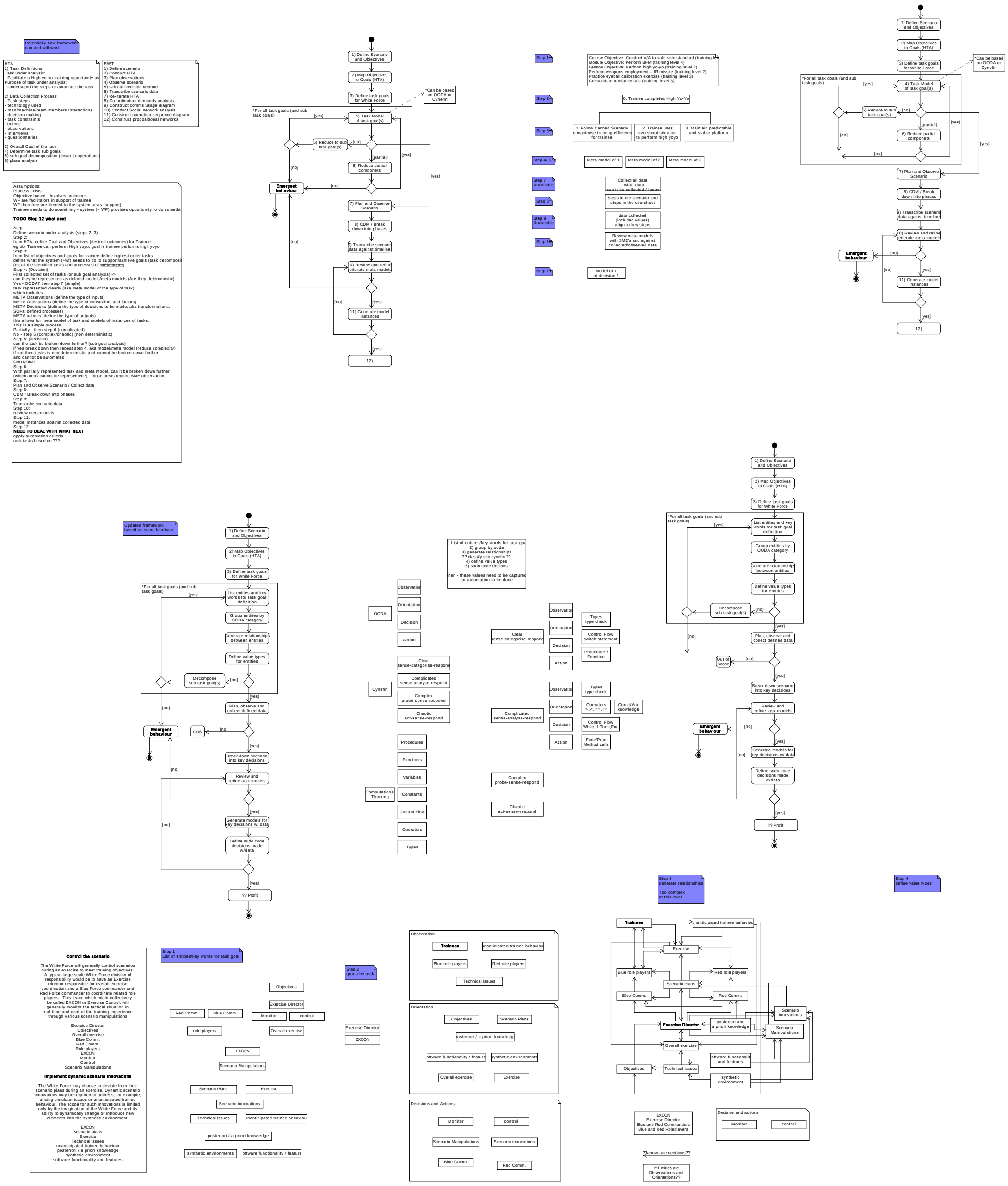


Lesson Objective: Objective
name: Perform high yo-yo (training leve
description: tbd
Trainee: Participant
note: Training audienc
Aircraft: Equipment
Perform High yo-yo: Task
name: High yo-yo
macro
Simple High yoyo: Task
name: Follow high yoyo canned
scenario and maintain predictabl
and stable airmanship
macro
QFI: Participant
note: RED AIR
D Voodoo1 ready
A Use radio to communicate
P Red QFI
E Radio
Comms: V1 R: Task
name:
meso
Ob Trainee pos, Red pos
Or Bubble, Alt. Head. PADS. Start conditions
D Confirm start
A COMMS: VOODOO 1 READY
A Fly steady
P Red QFI
E Aircraft
Confirm start: Task
name: Voodoo 1 Ready
meso
Ob Bubble, Alt. Head. Target
Or Bubble, Alt. Head. Target, Flight trainin
D Fly Steady
A Manage Pitch, Roll, Yaw, Thrust
P Red QFI
E Aircraft
Fly Steady: Task
name:
meso
D Check Left 50
A Use radio to communicate
P Red QFI
E Radio
Comms: CL50: Task
name:
meso
Ob COMMS: VOODOO2 READY
Or Bubble, Alt. Head. Target.
D Trainee ready, change AA left 5
A COMMS: CHECK LEFT 50
A AA LEFT 50
P Red QFI
E Aircraft
Voodoo2 Ready: Task
name: Voodoo 2 Ready
meso
Ob Bubble, Alt. Head. Target
Or Bubble, Alt. Head. Target, Flight training
D Change AA L 50
A Manage Pitch, Roll, Yaw, Thrust to achieve AA L 50 cha
P Red QFI
E Aircraft
AA LEFT 50: Task
name:
meso
Ob Bubble, Alt. Head. Target
Or Bubble, Alt. Head. Target, Flight training
D Change AA R 50
A Manage Pitch, Roll, Yaw, Thrust to reverse AA to R 5
P Red QFI
E Aircraft
AA REVERSE: Task
name:
meso
Ob COMMS: VOODOO IN FROM 1.4, REVERSE
Or Bubble, Alt. Head. Target.
D Trainee making approach at 1.4 NM, Reverse
A AA REVERSE
P Red QFI
E Aircraft
Voodoo in: Task
name: Voodoo in from 1.
meso
Ob COMMS: 1.2
Or Bubble, Alt. Head. Target.
D Trainee making approach at 1.2 N
A Maintain
Ob COMMS: 1.1
Or Bubble, Alt. Head. Target.
D Trainee making approach at 1.1 N
A Maintain
P Red QFI
E Aircraft
Ob Bubble, Alt. Head. Target
Or Bubble, Alt. Head. Target, Flight trainin
D Fly Steady
A Manage Pitch, Roll, Yaw, Thrust
P Red QFI
E Aircraft
Maintain: Task
name:
meso
Voodoo approaching: Task
name:
meso
D Flares
A Use radio to communicate
P Red QFI
E Radio
Comms: Flares: Task
name:
meso
Ob COMMS: FOX 2
Or Bubble, Alt. Head. Target.
D Missle shot at 1nm, call Flares and break into trai
A COMMS: FLARES
A BREAK INTO TRAINEE
P Red QFI
E Aircraft
Fox 2: Task
name:
meso
Ob Bubble, Alt. Head. Target
Or Bubble, Alt. Head. Target, Flight training
D Break into trainee to allow High Yoyo
A Manage Pitch, Roll, Yaw, Thrust to brea
into trainee andcreate overshoot
P Red QFI
E Aircraft
Break into Trainee: Task
name:
meso

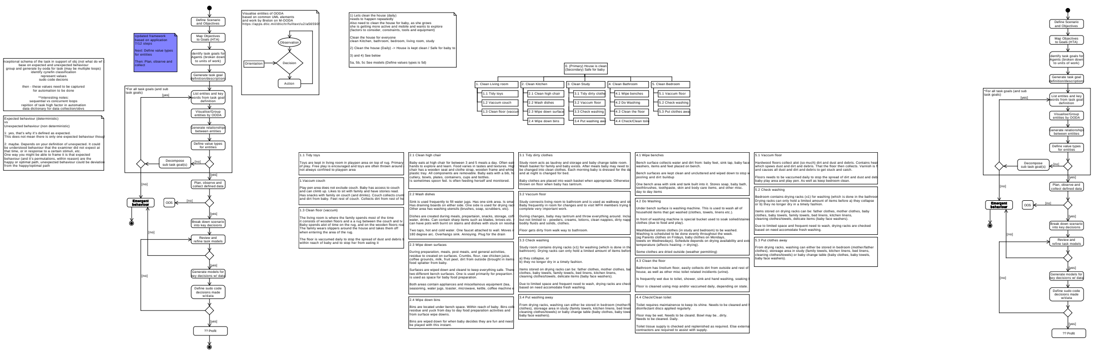

update meta modle to include OODA
simplyfy scenario
include need for red to adjust
macro obj
meso decision / steps
micro action implementation
would like a c2 example
or something else to include
next steps validate against other task
Input/Information Acquisition/Sensory Processing
Factors/Considerations
Information Analysis/Perception/Working memory
Decision Selection/Decision Making
Output/Action Implementation/Response Selection
Tooling/Equipment
Participant
Implicit design for correctness
1 CI -> mine objs
2 Objs -> objs tasks that need to do be done
3 Obj tasks -> What do WF need to do to allow trainees to achieve that -> process tasks
4 Process task -> Process task(s)
5 Process task -> Action(s) tasks
6 Action tasks -> Action loops (w/ data in, obs, decs, outputs)
1)
• Air superiority indicates the mission profile,
• Degrading and/or destroying is a specific type of mission set, and,
• The enemy's Centre of Gravity (CoG) is something that would be detailed in higher documentation, i.e. power stations and military or political headquarters
2) ??
eg Tasks for air superiority mission profile
There needs to be a better way to model OODA
Framework breaks down tasks into loops
macro -> meso -> micro
Look at Observations in (data input)
Look at Orientation (transformations)
Look at Actions (data output)
These define categories of automation technology

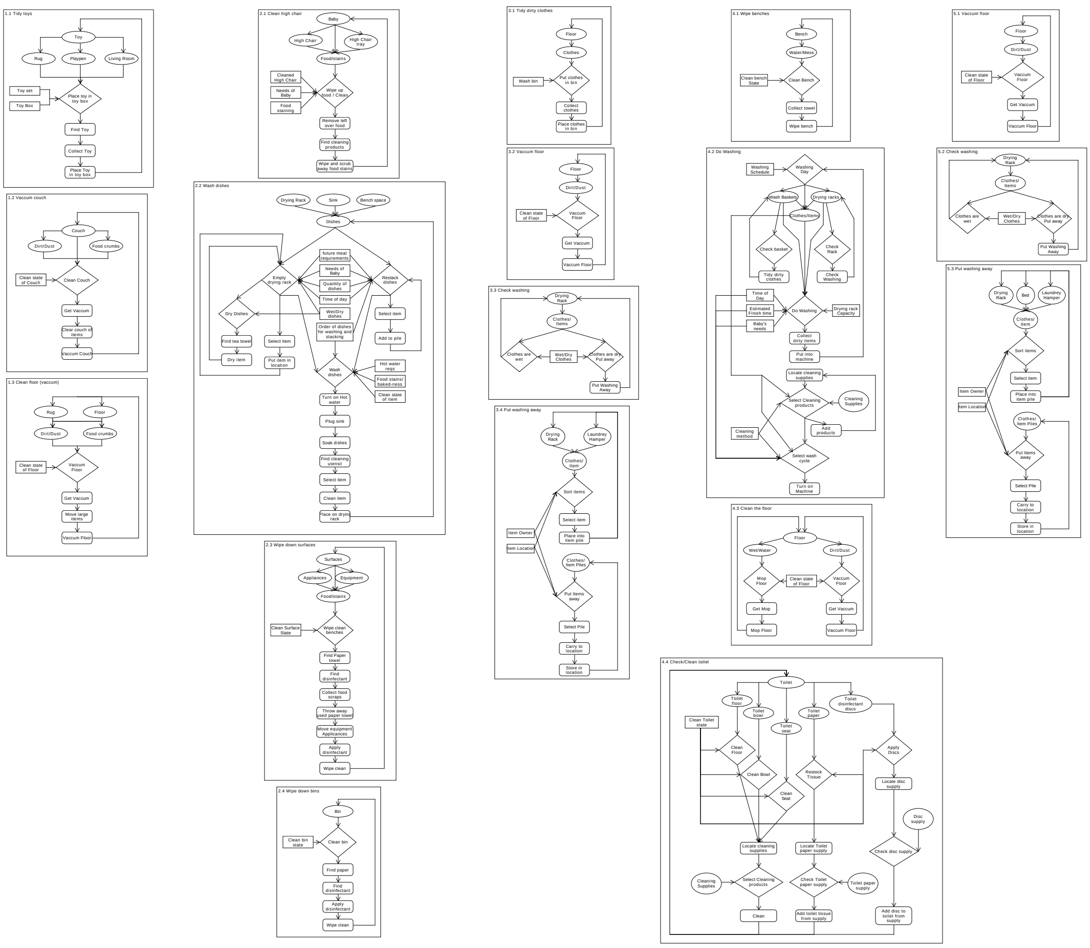


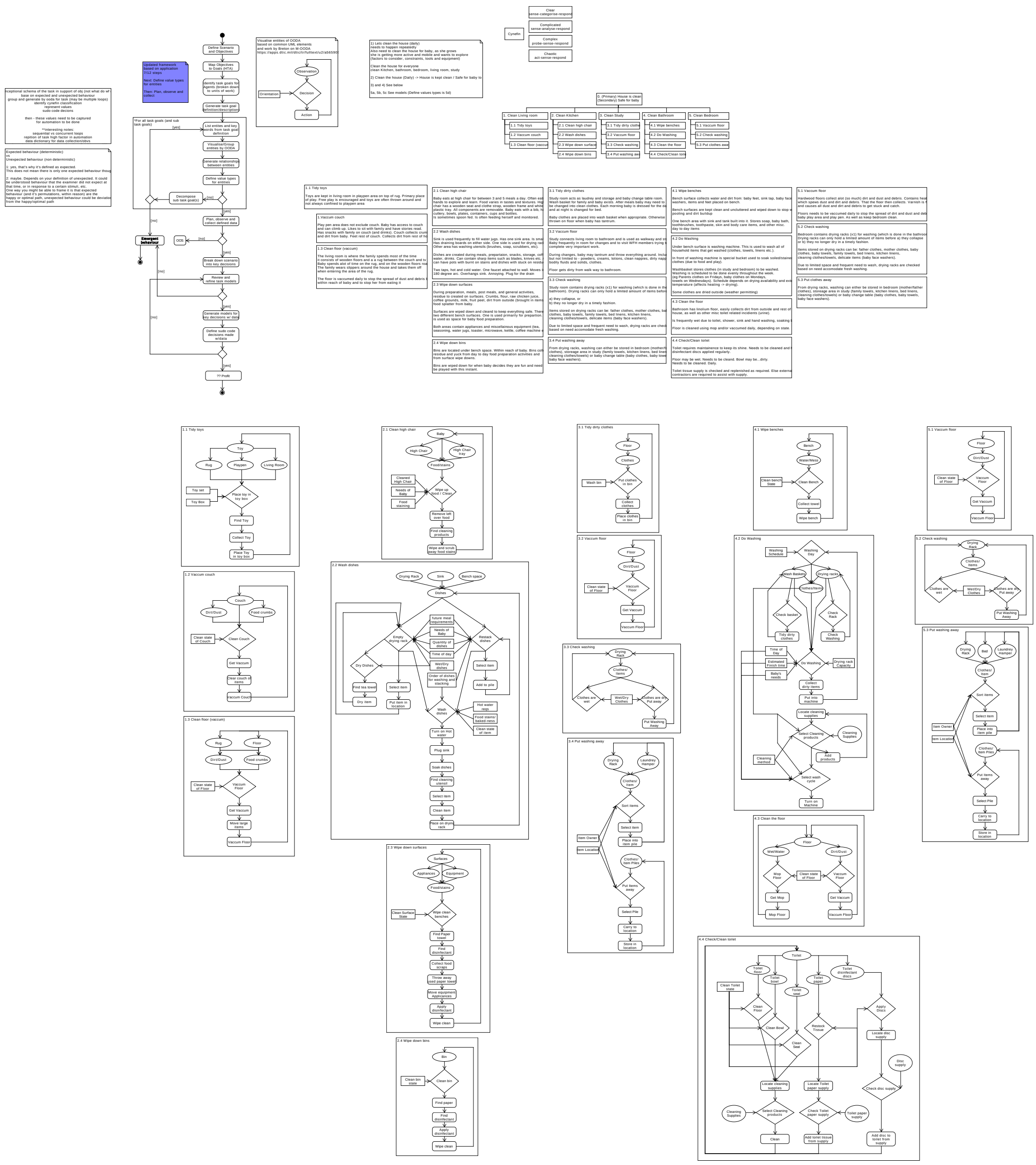


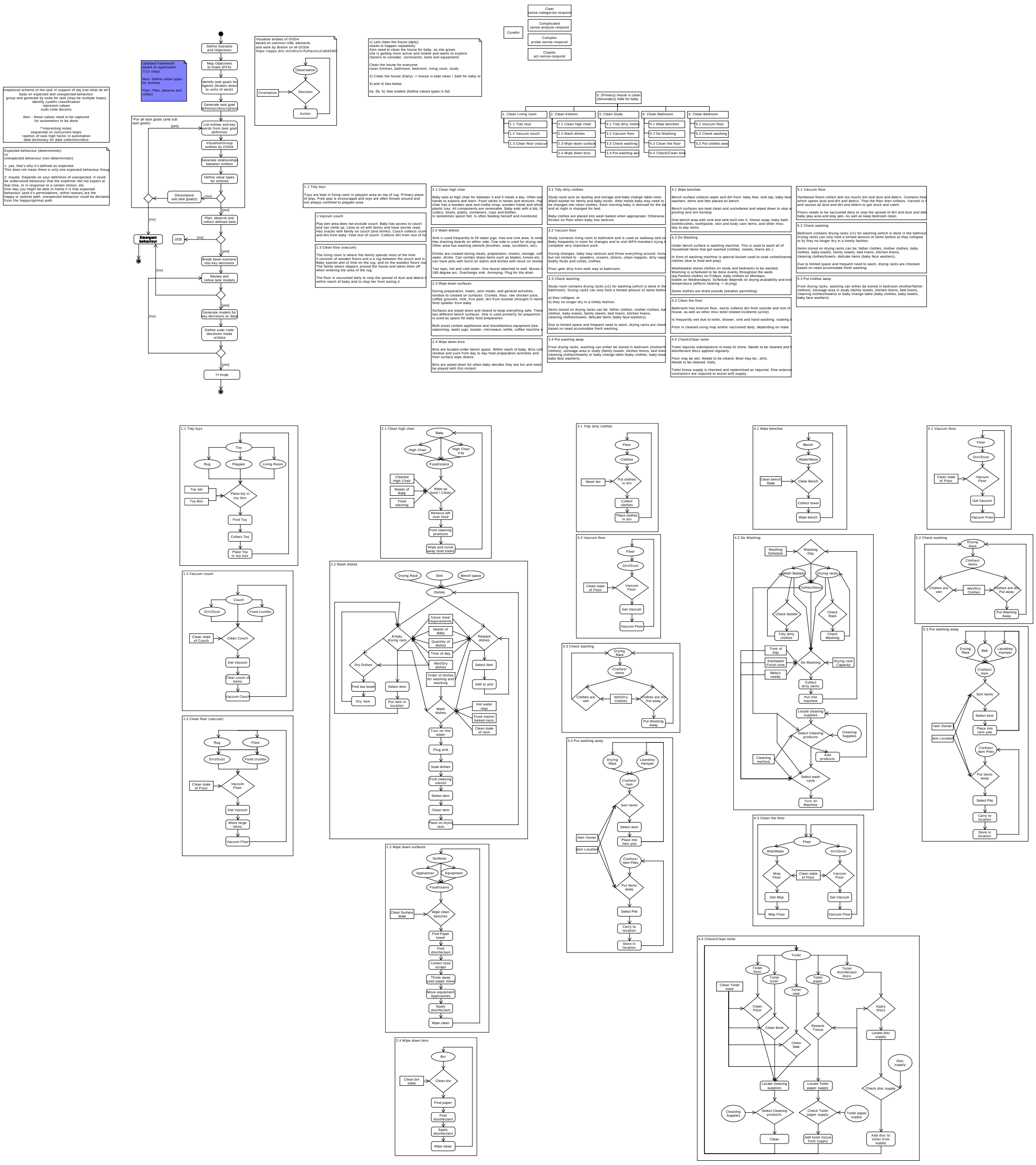


Control the scenario
The White Force will generally control scenarios during an exercise to meet training objectives. A typical large-scale White Force division of responsibility would be to have an Exercise Director responsible for overall exercise coordination and a Blue Force commander and Red Force commander to coordinate related role players. This team, which might collectively be called EXCON or Exercise Control, will generally monitor the tactical situation in real-time and control the training experience through various scenario manipulations
Exercise Director
Objectives
Overall exercise
Blue Comm.
Red Comm.
Role players
EXCON
Monitor
Control
Scenario Manipulations
Implement dynamic scenario innovations
The White Force may choose to deviate from their scenario plans during an exercise. Dynamic scenario innovations may be required to address, for example, arising simulator issues or unanticipated trainee behaviour. The scope for such innovations is limited only by the imagination of the White Force and its ability to dynamically change or introduce new elements into the synthetic environment.
EXCON
Scenario plans
Exercise
Technical issues
unanticipated trainee behaviour
posteriori / a priori knowledge
synthetic environment
software functionality and features
Step 1
Course Objective: Conduct A/A to safe solo standard (training lev
Module Objective: Perform BFM (training level 4)
Lesson Objective: Perform high yo-yo (training level 2)
Perform weapons employment – IR missile (training level 2)
Practice eyeball calibration exercise (training level 3)
Consolidate fundamentals (training level 3)
Step 2
0. Trainee completes High Yo-Yo
Step 3
1. Follow Canned Scenario o maximise training efficienc for trainee
2. Trainee uses overshoot situation to perform high yoyo
3. Maintain predictable and stable platform
Step 4(,5,6)
Meta model of 1
Meta model of 2
Meta model of 3
Step 7 Unavilable
Collect all data - what data can it be collected / logged
Step 8
Steps in the scenario and steps in the overshoot
Step 9 Unavilable
data collected (included values) align to key steps
Step 10
Review meta models with SME's and against collected/observed data
Step 11
Model of 1 at decision 1


Define Scenario and Objectives
Map Objectives to Goals (HTA)
Identify task goals for Agents (broken down to units of work)
Generate task goal definition/descriptions
Updated framework based on application 7/12 steps
Next: Define value types for entities
Then: Plan, observe and collect
Visualise entities of OODA based on common UML elements and work by Breton on M-OODA
Observation
Orientation
Decision
Action
*For all task goals (and sub task goals)
List entites and key words from task goal definition
Visualise/Group entities by OODA
Generate relationships between entities
Define value types for entities
Decompose sub task goal(s)
[yes]
[no]
Plan, observe and collect defined data
Emergent behaviour
OOS
Break down scenario into key decisions
Review and refine task models
1.1 Tidy toys
1.Vaccum couch
1.3 Clean floor (vaccum)

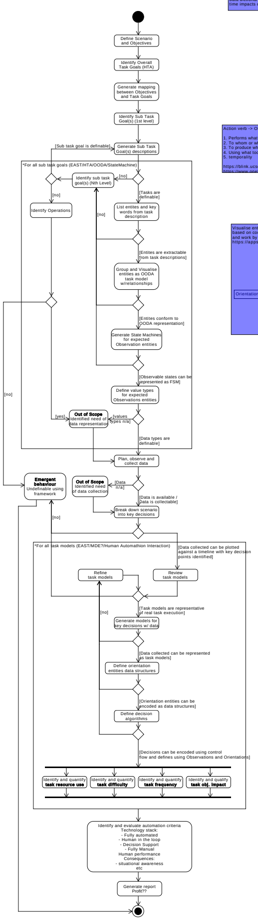

1.1 Tidy toys
1.2 Vaccum couch
1.3 Clean floor (vaccum)
2.1 Clean high chair
2.2 Wash dishes
2.3 Wipe down surfaces
2.4 Wipe down bins
3.1 Tidy dirty clothes
3.2 Vaccum floor
3.3 Check washing
3.4 Put washing away
4.1 Wipe benches
4.2 Do Washing
4.3 Clean the floor
4.4 Check/Clean toilet
5.1 Vaccum floor
5.2 Check washing
5.3 Put washing away





Work done for meeting 19/6/20

Interesting notes:
sequential vs concurrent loops
repition of task high factor in automation
data dictionary for data collection/obvs
time impacts on all events (starting, duration, finishing, repitition)

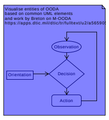


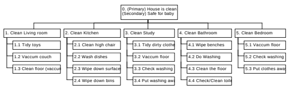
Action verb -> Object of the action -> Purpose or Result of the action

1. Performs what action? (verb)
2. To whom or what? (object of the verb)
3. To produce what? or Why is it necessary? (expected output)
4. Using what tools, equipment, work aids, processes?
5. temporality

https://blink.ucsd.edu/_files/HR-tab/compensation/writingtask.pdf
https://www.onetcenter.org/dl_files/TaskUpdating.pdf

Visualise entities of OODA
based on common UML elements
and work by Breton on M-OODA
https://apps.dtic.mil/dtic/tr/fulltext/u2/a56590





1.1 Tidy toys

Collect toys that are not in the toy box and return them to toy box until the room/floor is clean

1.Vaccum couch

Vaccum the couch using the vaccum cleaner to remove traces of food and misc. dirt until it is clean

1.3 Clean floor (vaccum)

Use the vaccum cleaner to clean the floor on a daily basis to collect food, dirt and other misc material so that it is clean

2.1 Clean high chair

Remove remnents of food/drink from the high chair, using cleanin products, face washers and tissues until the high chair is clean and for baby to use

2.2 Wash dishes

Empty, Soap, Wash and Clean dishes, pots, pans, bottles and other kitchen and household items in the sink, using cleaning products. clean, store items on the drying rack until such time that they can returned and stored to the correct place.

2.3 Wipe down surfaces

Wipe and clean bench tops, surfaces and household items that may dirty, from food scraps. Use cleaning products, face washers and p towels until clean

2.4 Wipe down bins

Wipe down the outside of bins, using cleaning products and paper until clean.

3.1 Tidy dirty clothes

Collect clothes from the floor of the study and place them in the wash bask all clothes have been collected. If clothes need to be soaked, place them in soaking bucket

3.2 Vaccum floor

Use the vaccum cleaner to vaccum the floor, on a daily basis, to stop the bu up of dirt and other misc. material. Vaccum until clean. If required, due to liquid or waste, use a steam mop or mop and bucket to clean, until clean.

3.3 Check washing

Check washing on drying rack to see if dry/dry enough to store away.
If so, either put washing away or fold and store in piles

3.4 Put washing away

Collect washing from drying rack or from collated piles and return to storag until all items are away

4.1 Wipe benches

Wipe bench surfaces using cleaning products and paper towels or towels to remove water, dirt, grime or other misc. until clean

4.2 Do Washing

Collect and place dirty clothing items in the washing machine, use appro cleaning products based on clothing items, as well as wash cycle and wa clothes. Wash clothes based on a schedule so that clothes are done in batches and that the family always has clothes to wear.

4.3 Clean the floor

Use the vaccum cleaner to vaccum the floor on a daily basis until clean.
If required, due to water and misc. material, use steam mop or mop and to clean floor until clean.

4.4 Check/Clean toilet

Check toilet for stains and dirt build up. Check toilet paper supply.
If required, use cleaning products and equipment to clean toilet to remov stains and build up, until clean. If required restock toilet paper from toile paper supply, until full or one extra.

5.1 Vaccum floor

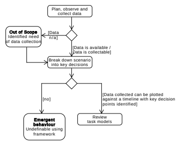
Use the vaccum cleaner to clean the floor on a daily basis collect food, dirt and other misc material so that it is clea

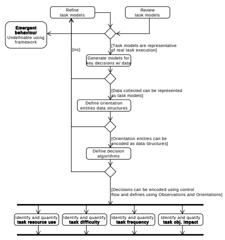
5.2 Check washing

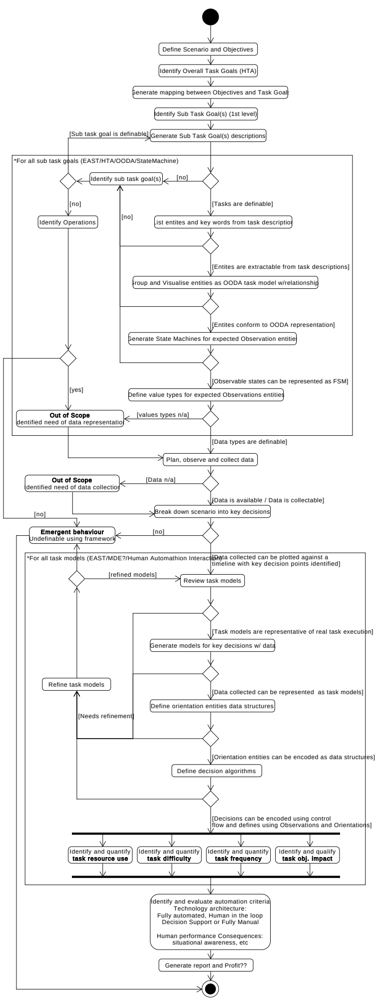
Check washing on drying rack to see if dry/dry enough to away. If so, either put washing away or fold and store in p

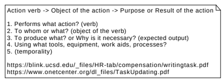
5.3 Put clothes away

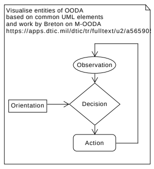
Collect washing from drying rack or from collated piles an return to storage areas until all items are away

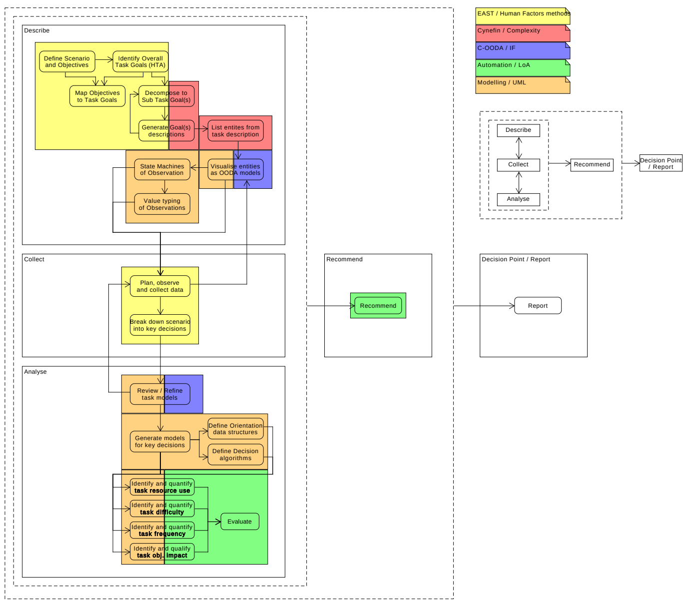


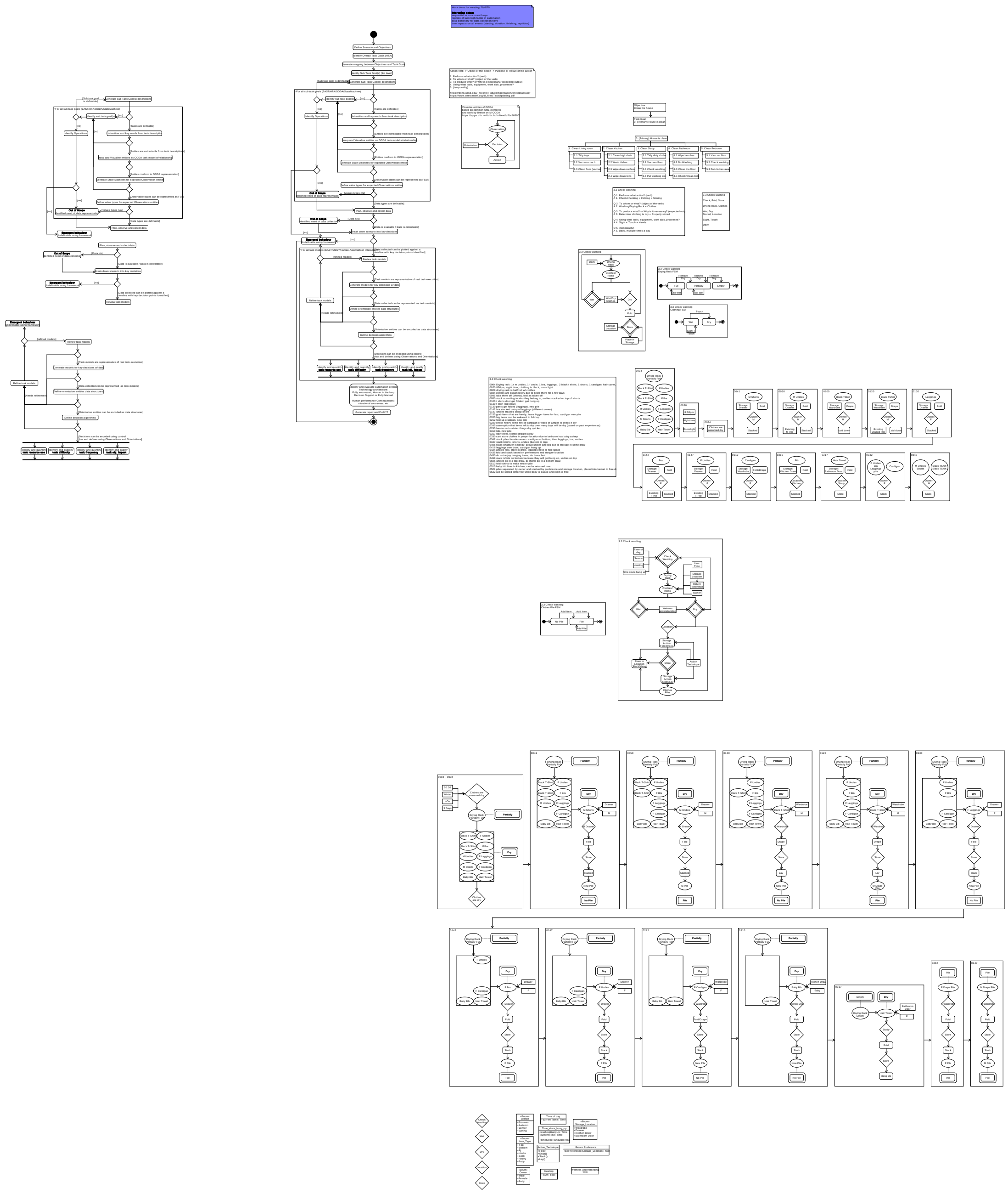


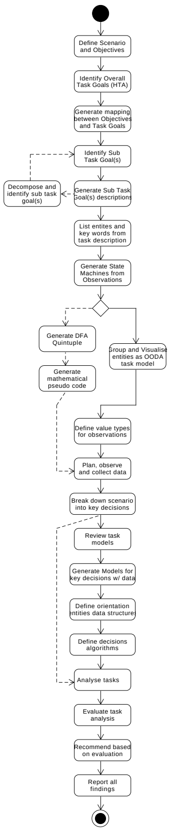

Define Scenario and Objectives
Identify Overall Task Goals (HTA)
Generate mapping between Objectives and Task Goal
Identify Sub Task Goal(s) (1st level)
[Sub task goal is definable]
Generate Sub Task Goal(s) descriptions
*For all sub task goals (EAST/HTA/OODA/StateMachine)
Identify sub task goal(s)
[no]
[no]
Identify Operations
[Tasks are definable]
[no]
List entites and key words from task descriptior
[Entites are extractable from task descriptions]
Group and Visualise entities as OODA task model w/relationship
[Entites conform to OODA representation]
Generate State Machines for expected Observation entitie
[yes]
[Observable states can be represented as FSM]
Define value types for expected Observations entities
Out of Scope
Identified need of data representation
[values types n/a]
[Data types are definable]
Plan, observe and collect data
Out of Scope
Identified need of data collection
[Data n/a]
[Data is available / Data is collectable]
Break down scenario into key decisions
[no]
Emergent behaviour
Undefinable using framework
[no]
*For all task models (EAST/MDE?/Human Automathion Interaction)
[Data collected can be plotted against a timeline with key decision points identified]
[refined models]
Review task models
[Task models are representative of real task execution]
Generate models for key decisions w/ data
Refine task models
[Data collected can be represented as task models]
Define orientation entities data structures
[Needs refinement]
[Orientation entities can be encoded as data structures]
Define decision algorithms
[Decisions can be encoded using control flow and defines using Observations and Orientations]
Identify and quantify task resource use
Identify and quantify task difficulty
Identify and quantify task frequency
Identify and qualify task obj. impact
Identify and evaluate automation criteria
Technology architecture:
Fully automated, Human in the loop
Decision Support or Fully Manual
Human performance Consequences:
situational awareness, etc
Generate report and Profit??

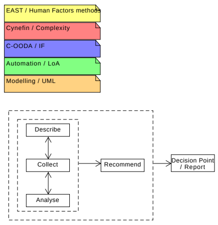

Action verb -> Object of the action -> Purpose or Result of the action
1. Performs what action? (verb)
2. To whom or what? (object of the verb)
3. To produce what? or Why is it necessary? (expected output)
4. Using what tools, equipment, work aids, processes?
5. (temporality)
https://blink.ucsd.edu/_files/HR-tab/compensation/writingtask.pdf
https://www.onetcenter.org/dl_files/TaskUpdating.pdf

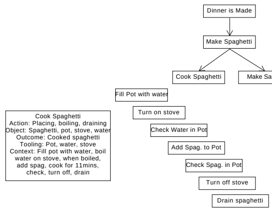

Visualise entities of OODA
based on common UML elements
and work by Breton on M-OODA
https://apps.dtic.mil/dtic/tr/fulltext/u2/a56590
Observation
Orientation
Decision
Action

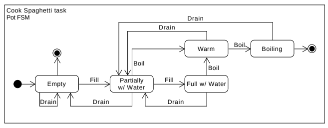

Describe
Define Scenario and Objectives
Identify Overall Task Goals (HTA)
Map Objectives to Task Goals
Decompose to Sub Task Goal(s)
Generate Goal(s) descriptions
List entites from task description
State Machines of Observation
Visualise entities as OODA models
Value typing of Observations
Collect
Plan, observe and collect data
Break down scenario into key decisions
Analyse
Review / Refine task models
Generate models for key decisions
Define Orientation data structures
Define Decision algorithms
Identify and quantify task resource use
Identify and quantify task difficulty
Identify and quantify task frequency
Identify and qualify task obj. impact
Evaluate
Recommend
Recommend
Decision Point / Report
Report
EAST / Human Factors methods
Cynefin / Complexity
C-OODA / IF
Automation / LoA
Modelling / UML
Describe
Collect
Analyse
Recommend
Decision Point / Report

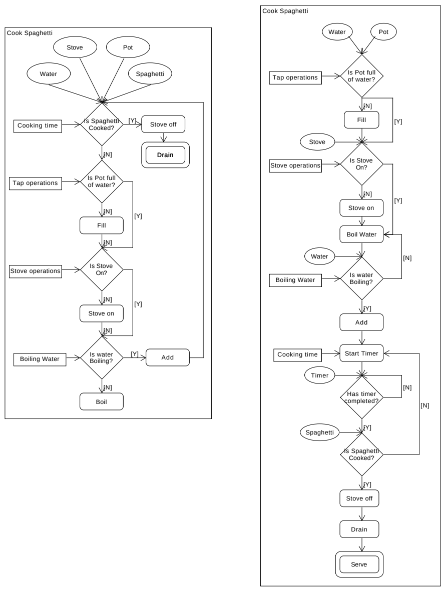


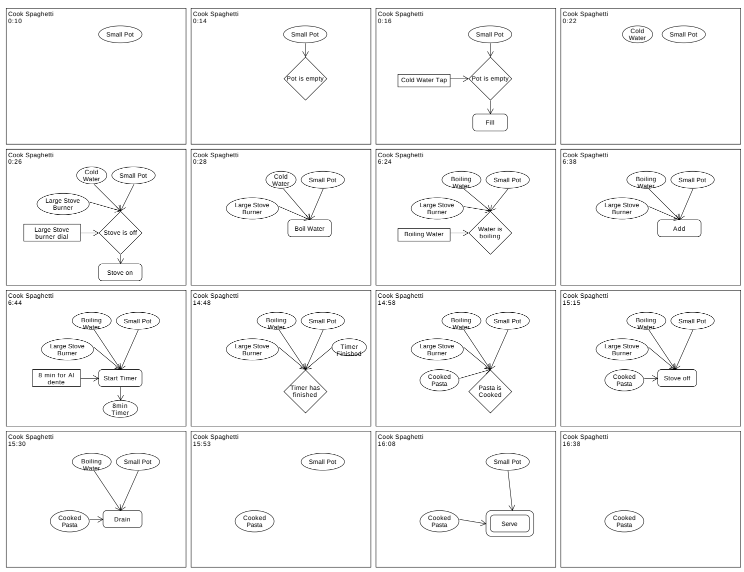

Define Scenario and Objectives
Identify Overall Task Goals (HTA)
Generate mapping between Objectives and Task Goals
Identify Sub Task Goal(s)
Decompose and identify sub task goal(s)
Generate Sub Task Goal(s) description
List entites and key words from task description
Generate State Machines from Observations
Generate DFA Quintuple
Generate mathematical pseudo code
Group and Visualise entities as OODA task model
Define value types for observations
Plan, observe and collect data
Break down scenario into key decisions
Review task models
Generate Models for key decisions w/ data
Define orientation entities data structure
Define decisions algorithms
Analyse tasks
Evaluate task analysis
Recommend based on evaluation
Report all findings

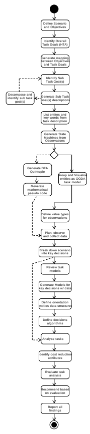

EAST / Human Factors methods
Cynefin / Complexity
C-OODA / IF
Automation / LoA
Modelling / UML
Describe
Collect
Analyse
Recommend
Decision Point / Report

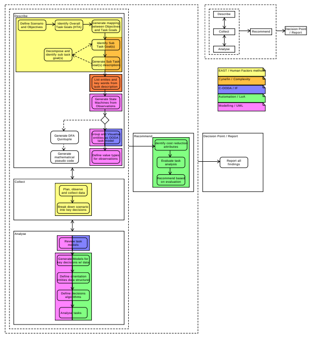

Dinner is Made
Make Spaghetti
Cook Spaghetti
Make Sauce
Fill Pot with water
Turn on stove
Check Water in Pot
Add Spag. to Pot
Check Spag. in Pot
Turn off stove
Drain spaghetti
Cook Spaghetti
Action: Placing, boiling, draining
Object: Spaghetti, pot, stove, water
Outcome: Cooked spaghetti
Tooling: Pot, water, stove
Context: Fill pot with water, boil water on stove, when boiled, add spag, cook for 11mins, check, turn off, drain

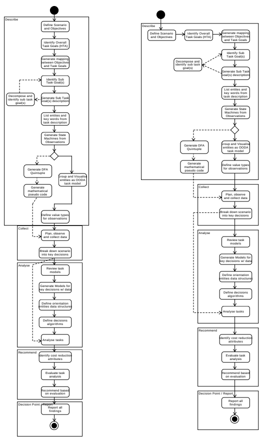

Cook Spaghetti task
Pot:FSM
Empty
Partially w/ Water
Full w/ Water
Warm
Boiling
Fill
Boil
Drain

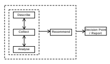

Cook Spaghetti
Stove
Pot
Water
Spaghetti
Cooking time
Is Spaghetti Cooked?
Stove off
Drain
Tap operations
Is Pot full of water?
Fill
Stove operations
Is Stove On?
Stove on
Boiling Water
Is water Boiling?
Add
Boil
Timer
Start Timer
Has timer completed?
Serve
[Y]
[N]

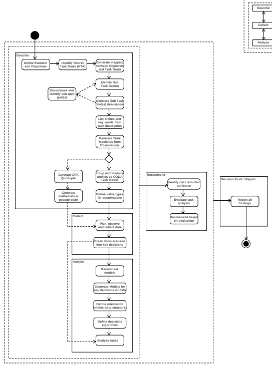

Cook Spaghetti
0:10
0:14
0:16
0:22
0:26
0:28
6:24
6:38
6:44
14:48
14:58
15:15
15:30
15:53
16:08
16:38
Small Pot
Pot is empty
Cold Water Tap
Fill
Cold Water
Large Stove Burner
Large Stove burner dial
Stove is off
Stove on
Boil Water
Boiling Water
Water is boiling
Add
8 min for Al dente
Start Timer
8min Timer
Timer Finished
Timer has finished
Cooked Pasta
Pasta is Cooked
Stove off
Drain
Serve

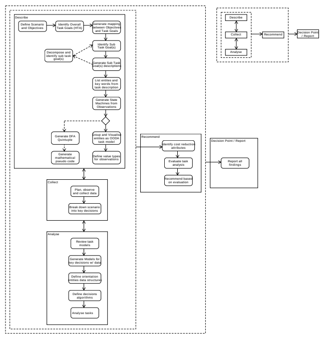


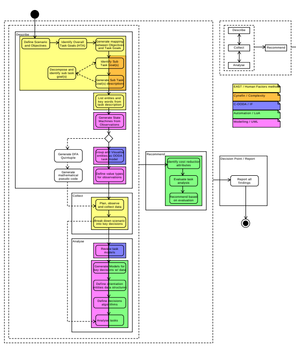


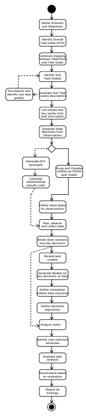


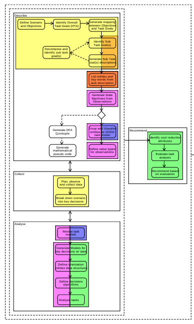


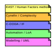


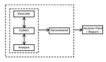


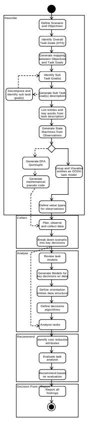


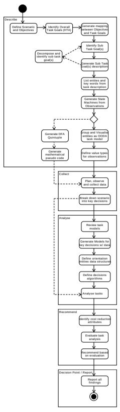


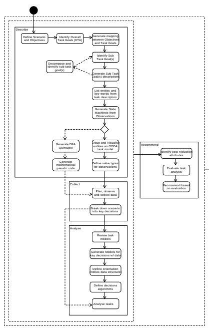


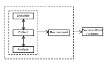


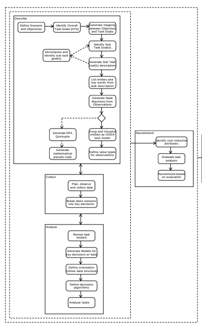


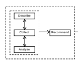


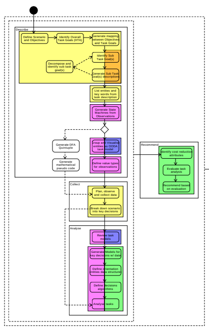


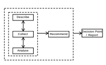












7/8/20

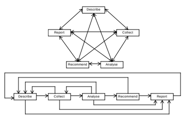
%OODA loop high level concepts
%this diagram is almost OODA loop
%could be circular
%the last step of reporting could feed back into describe
%high level description for decarr
each then has its own model
%less waterfall and more spiral
%data collection will be constant

%high level grouping
%each then has a description
%report can be follow on actions
%high level description
each then has a model

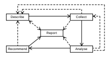
easy entry and exit points
%data collection iterative
%as part of overview, use LH comments

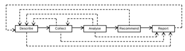
%- for each task and phase

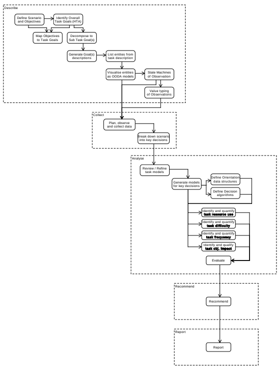
What the task produces
Who needs to be involved, Subject Matter Experts, Domain Experts, Analysts
How to perform the task
When to perform the task (after task a, b, c, etc)
Can this be performed in parallel



TODO:
each then has a model
easy entry and exit points



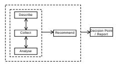




Describe
Phase of identify objectives, tasks and processes.
Tasks are modelled at the highest level to capture OODA representaions to inform Collect
Observations are defined.
What
Task descriptions, models, sources of data of interest
Observations are representated as states and/or data types
Who
Domain experts, Data Analysts
How
Task description method, HTA, OODA task models, State machines, computational thinkin
When
Exploritory task to be conducted first
Concurrent
Outputs can be reported. Multiple instances can occur

Collect
Data collection phase. Based on ares of interest from Describe phase. Uses Task models t
what data is of interest for OODA representations
What
Data points, logs, videos, screen casts, audio logs, time stamped data, observations, etc
Key decisions and important actions identified
Who
Researchers undertaking framework application. Subjects of data collection i.e SMEs,
How
Typical Usability study approaches for data collection. Ethics required.
Cricitical decision method used to identify key decisions
When
Preferable to occur after Describe phase.
Otherwise occurs during events when tasks and processes occur
Concurrent
Outputs reported. Occurs when available. After Describe phase, to limit scope of collectio

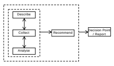
Analyse
Phase to review and compare task models vs real events. Combines task models with data
generate model instances. Identifies and defines decisions and orientations (OODA) with d
Metrics are able to be generated to identify frequency, resources use, impact and difficulty
Evaluation of metrics and task OODAs for rankings and auto. archiecture recommendation
What
Refined task models based on real data
Instance models based on collected data and key decisions
Metrics identifying impact, frequency, difficulty, resource impact
Rankings of tasks
Automation architecture recommendations
Who
Researcher undertaking framework application, Data analysts, Domain Experts, SME
How
data analysis and review, UML Modelling
When
Occurs after data collection. Whenever there is available data
Concurrent
Can have multiple instances. Outputs reported

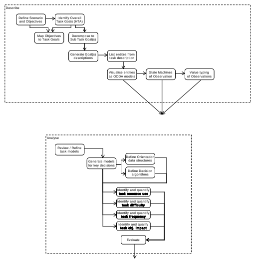


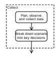
Recommend
Based on output from Analysis phase. Uses rankings of task metrics, task models etc
Influenced by technology availability, implementation capability etc.
Based on needs and use case of business.
Based on automation considerations and understandings, Levels of automation
What
Recommendations for which tasks should be automated or optimised.
Who
Researcher undertaking framework application, Domain expert, SMEs
How
Interpretation of analysis results, understanding of LoA and automation considerations.
Understanding of automation technology architecture benefits and limitations.
When
Occurs based on Analyse phase. Usually last task to be completed at end of frameowkr ap
Focus of Report phase.
Can occur multiple times based on scenario/objectives/processes/task under review.
Concurrent
Can be one or many instances. Outputs reported

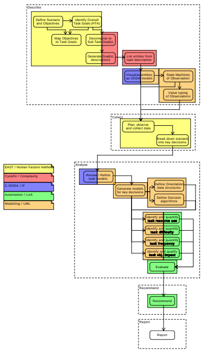
Report
Compilation of outputs from every other phase. Structured and readable
What
Report containing compiled and collated and structured outputs of every phase
Who
Research undertaking framework application
How
LaTeX or Word, writing a document. Reviews all available and used data
When
Occurs throughout framework application (to make final report easier)
Concurrent
Occurs at the outputs of every other phase.

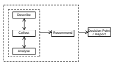
TODO:
each then has a model
easy entry and exit points

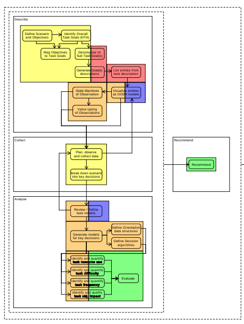


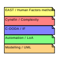


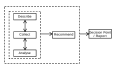


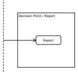


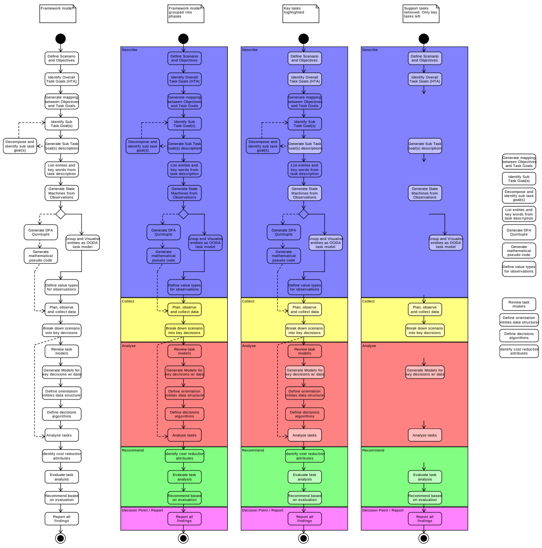


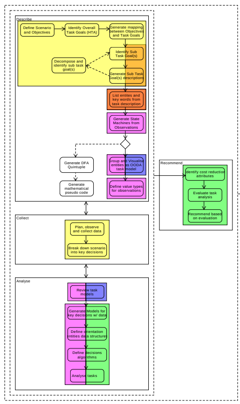


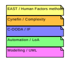


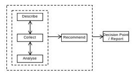


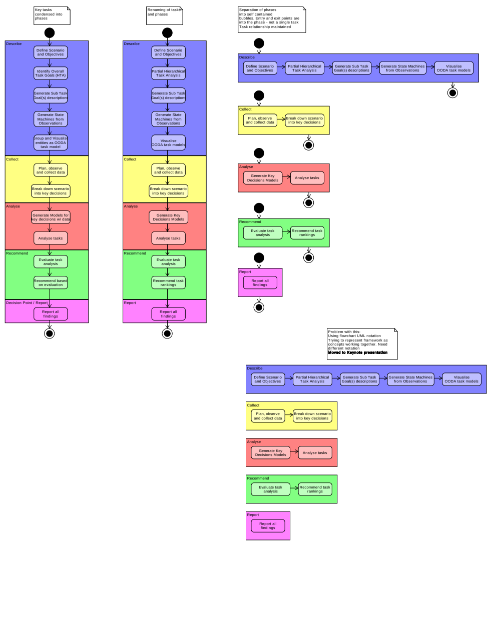






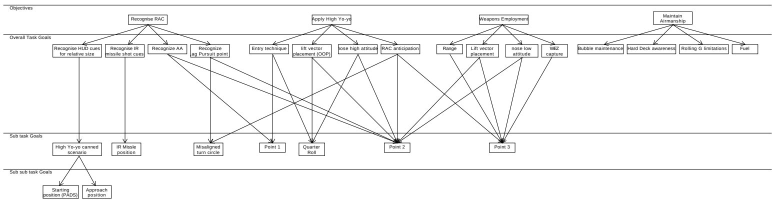






High Yo-yo scenario: (QFI w/ Student perspective)
The main emphasis for the student is to be able recognise RAC and learn when to ap
high yo-yo.
Have the trainee talk through RAC as they see it unfold. Weapons employment
is subsidiary to recognition of RAC, flying skills and techniques.
The student should state when they would apply Lag Pursuit and maintain POM, then
intentionally continue to the point of a high yo-yo for learning purposes.

Objectives and goals:
Recognise RAC
Talk through RAC
- HUD cues for relative size,
- IR missile shot cues
- AA
- Lag Pursuit point
maintian POM
Continue to point of High Yo yo
Apply High Yo-yo
- Entry technique
- lift vector placement (OOP)
- nose high attitude
- RAC anticipation



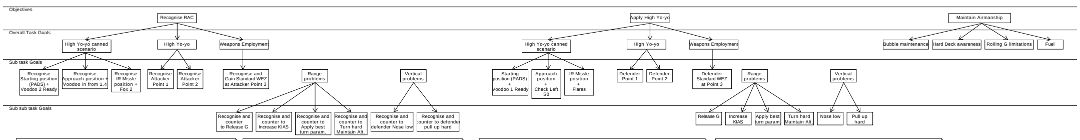
What is the action performed? (verb)
To whom, or what is the action performed on/to? (object of the verb)
What is produced by the action? Or Why is it necessary? (expected output
What are the tools, equipment, work aids, processes etc used?
What is the context of this task in relation to the scenario? (temporality)

High Yo-yo scenario: (QFI w/ Student perspective)
The main emphasis for the student is to be able recognise RAC and learn when to ap
high yo-yo.
Have the trainee talk through RAC as they see it unfold. Weapons employment
is subsidiary to recognition of RAC, flying skills and techniques.
The student should state when they would apply Lag Pursuit and maintain POM, then
intentionally continue to the point of a high yo-yo for learning purposes.

Objectives and goals:
Recognise RAC
Talk through RAC
- HUD cues for relative size,
- IR missile shot cues
- AA
- Lag Pursuit point
maintian POM
Continue to point of High Yo yo
Apply High Yo-yo
- Entry technique
- lift vector placement (OOP)
- nose high attitude
- RAC anticipation

What is the action performed? (verb)
To whom, or what is the action performed on/to? (object of the verb)
What is produced by the action? Or Why is it necessary? (expected out
What are the tools, equipment, work aids, processes etc used?
What is the context of this task in relation to the scenario? (temporalit

Range(NM), Aspect Angle(AA), Closure(KIAS)
yaw/heading/sideslip, pitch/flightpath/AA, roll/bank
weight, thrust, aerodynamic force
Plane of Motion, Out of Plane Manoeuvring (POM, OOP
Lead, Pure, Lag pursuit cuves



Objective: RECOGNISE RAC
Task: Recognise Starting position (PADS) + Voodoo 2 Ready
--*Action*: Observe, hear, look, listen, watch, guide, instruct
--*Object*: HUD, aircraft, instruments, defender, attacker, trainee
--*Output*: Trainee recognises and communicates to instructor QFI current RAC, gets into starting relative to red air, Abeam, Deck + 5k, 1.5NM, 300 KIAS, Voodoo 2 ready
--*Tooling*: Aircraft, headset, HUD, instruments, RAC, current position, red air position
--*Context*: observes trainee and coordinates with red air and trainee to assume starting position high yo yo canned scenario, communicates with trainee to ascertain current RAC to begin scenario

Objective: RECOGNISE RAC
Task: Recognise Approach position + Voodoo in from 1.4
--*Action*: Observe, hear, look, listen, watch, guide, instruct
--*Object*: HUD, aircraft, instruments, defender, attacker, trainee
--*Output*: Trainee recognises and communicates to instructor QFI current RAC, change in defend air heading, changes attack heading and recognises the change in range
--*Tooling*: Aircraft, headset, HUD, instruments, RAC, current position, red air position
--*Context*: Red air begins change in heading to facilitate Highyoyo maneouver. Trainee closes ra to get into IR missile position

Objective: RECOGNISE RAC
Task: Recognise Attacker Point 1
--*Action*: Observe, hear, look, listen, watch, guide, instruct
--*Object*: HUD, aircraft, instruments, defender, attacker, trainee
--*Output*: Trainee recognises and communicates to instructor QFI current RAC, trainee has called trainee recognises increasing AA and closure, trainee rolls OOP, pulls out of red plane of turn. Trai communicates RAC as high in red airs rear hemisphere, nose high attitude
--*Tooling*: Aircraft, headset, HUD, instruments, RAC, current position, red air position
--*Context*: Red air turns hard, attacker turns OOP, and pulls up. High part of High yoyo

Objective: RECOGNISE RAC
Task: Recognise Attacker Point 2
--*Action*: Observe, hear, look, listen, watch, guide, instruct
--*Object*: HUD, aircraft, instruments, defender, attacker, trainee
--*Output*: Trainee recognises and communicates to instructor QFI current RAC, trainee is at perc communicates fuselages virtually aligned, trainee communicates roll toward red air and lift vector i pure or lag pursuit, depending in communicated range
--*Tooling*: Aircraft, headset, HUD, instruments, RAC, current position, red air position

Objective: APPLY HIGH YOYO
Task: Starting position (PADS) + Voodoo 1 Ready
--*Action*: Flight, yaw/pitch/roll, thrust
--*Object*: Controls, flight stick, aircraft,
--*Output*: get into starting position relative to trainee, Abeam, Deck + 5k, 1.5NM, 300 KIAS, Voodoo
--*Tooling*: Aircraft, HUD, RAC, position, trainee position
--*Context*: work with trainee to assume starting position for high yo yo canned scenario

Objective: APPLY HIGH YOYO
Task: Approach position + Check Left 50
--*Action*: Flight, yaw/pitch/roll, thrust
--*Object*: Controls, flight stick, aircraft,
--*Output*: change heading left 50, turns into trainee,confirmation and range closure, maintains stea
--*Tooling*: Aircraft, HUD, RAC, position, trainee position
--*Context*: setup for trainee to close range and get into position for FOX 2 call

Objective: APPLY HIGH YOYO
Task: Defender Point 1
--*Action*: Flight, yaw/pitch/roll, thrust
--*Object*: Controls, flight stick, aircraft,
--*Output*: maintains break into trainee, watches RAC, AA and perch. Maintains accurate G, KIAS and
--*Tooling*: Aircraft, HUD, RAC, position, trainee position
--*Context*: Provides oppurtunity for trainee to reach perch and regain AA. Steady and predictable

Objective: APPLY HIGH YOYO
Task: Defender Point 2
--*Action*: Flight, yaw/pitch/roll, thrust
--*Object*: Controls, flight stick, aircraft,
--*Output*: maintains break, watches RAC, AA and high perch. Maintains accurate G, KIAS and altitud
--*Tooling*: Aircraft, HUD, RAC, position, trainee position
--*Context*: Trainee nose high perch, begins nose low lift vector placement. Steady and predictable

Objective: RECOGNISE RAC
Task: Recognise Starting position (PADS) + Voodoo 2 Ready
Observations: HUD, trainee comms, red air
Orientations: Training material, canned scenario
Decisions: Has trainee recognised and communicated RAC?
Actions: Speak/Control?

Objective: RECOGNISE RAC
Task: Recognise Approach position + Voodoo in from 1.4
Observations: HUD, trainee comms, red air
Orientations: Training material, canned scenario
Decisions: Has trainee recognised and communicated RAC?
Actions: Speak/Control?

Objective: RECOGNISE RAC
Task: Recognise IR Missle position + Fox 2
Observations: HUD, trainee comms, red air
Orientations: Training material, canned scenario
Decisions: Has trainee recognised and communicated RAC?
Actions: Speak/Control?

Objective: RECOGNISE RAC
Task: Recognise Attacker Point 1
Observations: HUD, trainee comms, red air
Orientations: Training material, canned scenario
Decisions: Has trainee recognised and communicated RAC?
Actions: Speak/Control?

Objective: RECOGNISE RAC
Task: Recognise Attacker Point 2
Observations: HUD, trainee comms, red air
Orientations: Training material, canned scenario
Decisions: Has trainee recognised and communicated RAC?
Actions: Speak/Control?

Objective: RECOGNISE RAC
Task: Recognise and Gain Standard WEZ at Attacker Point 3
Observations: HUD, trainee comms, red air
Orientations: Training material, canned scenario
Decisions: Has trainee recognised and communicated RAC?
Actions: Speak/Control?

Objective: APPLY HIGH YOYO
Task: Starting position (PADS) + Voodoo 1 Ready
--*Action*: Flight, yaw/pitch/roll, thrust
--*Object*: Controls, flight stick, aircraft,
--*Output*: get into starting position relative to trainee, Abeam, Deck + 5k, 1.5NM, 300 KIAS, Voodoo
--*Tooling*: Aircraft, HUD, RAC, position, trainee position
--*Context*: work with trainee to assume starting position for high yo yo canned scenario

Objective: APPLY HIGH YOYO
Task: Approach position + Check Left 50
--*Action*: Flight, yaw/pitch/roll, thrust
--*Object*: Controls, flight stick, aircraft,
--*Output*: change heading left 50, turns into trainee,confirmation and range closure, maintains stea
--*Tooling*: Aircraft, HUD, RAC, position, trainee position
--*Context*: setup for trainee to close range and get into position for FOX 2 call

Objective: APPLY HIGH YOYO
Task: Defender Point 1
--*Action*: Flight, yaw/pitch/roll, thrust
--*Object*: Controls, flight stick, aircraft,
--*Output*: maintains break into trainee, watches RAC, AA and perch. Maintains accurate G, KIAS and
--*Tooling*: Aircraft, HUD, RAC, position, trainee position
--*Context*: Provides oppurtunity for trainee to reach perch and regain AA. Steady and predictable

Objective: APPLY HIGH YOYO
Task: Defender Point 2
--*Action*: Flight, yaw/pitch/roll, thrust
--*Object*: Controls, flight stick, aircraft,
--*Output*: maintains break, watches RAC, AA and high perch. Maintains accurate G, KIAS and altitud
--*Tooling*: Aircraft, HUD, RAC, position, trainee position
--*Context*: Trainee nose high perch, begins nose low lift vector placement. Steady and predictable

Scenario

A simpilot is controlling two aircraft. He/she receives a command over the radio from a trainee in the control agency (wedgetail): "aircraft 2, make your way to the fuel tanker, contact me when you are 20 nautical miles out" (this is paraphrased).

From the perspective of the trainee, in a Large Force Engagement (LFE) exercise this is a 'clear' task, as those trainee already have proficiency. However for new students undergoing intial employment training, especially for the first time, this task may be 'complicated'.

From the perspective of the simpilot, he/she now has a goal to "fly to tanker and respond on radio". Again, for a real pilot, or experienced simpilot this is 'clear'. But for anyone else, I feel that it gets 'complicated', fast!

Tasks sequence in order to satisfy the above goal:

- Acknowledge receipt of the command over radio.
- Check location of fuel tanker (Note the tanker will be orbiting in a circle.)
- Select the entity named "aircraft 2"
- Adjust aircraft 2's heading, so it will intercept the tanker.
- Configure distance meausurement tool.
- Wait until distance measurement tool reads 20 nautical miles to the tanker.
- Contact control agency.

Discussion:
Not all of the tasks above are completed instantaeously (we might call that synchronous).
The 'contact control agency' task invokes a whole new task heirachy for operating the radio. (the channel may be jammed, the channel may already be in use, etc.). This is where HTA comes in handy.

Tasks may produce errors or execeptions => complexity. Maybe even chaos when the simpilot has not experienced the error before.

Simpilot may be instructed from white force to deliberately disobey or misinterpret the commands. View through cynefin lens, this is to intended create complexity/(chaos?) for the trainee.

There are two aircraft in the worked example. We haven't considered what the simpilot is doing with aircraft 1 :-)

Objective: Make way to the fuel tanker, contact when 20NM out

Task Goal:
0. Fly to tanker and respond on radio

Objective: Create training oppurtunity

Task Goal:
0. Disobey or misinterpret commands

Objective: Manage aircraft 1

0. Fly to tanker and respond on radio

1. Acknowledge receipt of the command over radio.
2. Check location of fuel tanker
3. Select the entity named "aircraft 2"
4. Adjust aircraft 2's heading, so it will intercept the tanker.
5. Configure distance meausurement tool.
6. Wait until distance measurement tool reads 20 nautical miles to the tanker.
7. Contact control agency.
   - 7.1 ??

1. Acknowledge receipt of command over radio

Q 1. Performs what action? (verb)
A 1. Talking/Responding

Q 2. To whom or what? (object of the verb)
A 2. Radio

Q 3. To produce what? or Why is it necessary? (expected outp
A 3. Confirmation and acknowledgement of recieved comma

Q 4. Using what tools, equipment, work aids, processes?
A 4. Radio, command lexicon, voice

Q 5. (temporality)
A 5. In response to a command directed to self

1. Acknowledge over radio

1. Action: Talking/Responding

2. Object: Radio

3. Outcome: Confirm command

4. Tooling: Radio, Voice, Command lexicon

5. Time: In response to comma

2. Check location of fuel tanker

Q 1. Performs what action? (verb)
A 1. Checks, Measures, Locates

Q 2. To whom or what? (object of the verb)
A 2. Fuel tanker

Q 3. To produce what? or Why is it necessary? (expected outp
A 3. Identification of correct fuel tanker

Q 4. Using what tools, equipment, work aids, processes?
A 4. HUD, map, named entites, mouse

Q 5. (temporality)
A 5. In response to a command / when fuel is low?

2. Check location of fuel tanker

1. Action: Check/Locates

2. Object: Fuel tanker

3. Outcome: Fuel tanker located

4. Tooling: HUD, Map

5. Time: Response to command

3.Select entity named aircraft 2

Q 1. Performs what action? (verb)
A 1. Selects, identifies, targets

Q 2. To whom or what? (object of the verb)
A 2. Entity Aircraft 2

Q 3. To produce what? or Why is it necessary? (expected outp
A 3. Aircraft 2 is selected

Q 4. Using what tools, equipment, work aids, processes?
A 4. Mouse, Display, map, list of entities

Q 5. (temporality)
A 5. When controlling entity, select it first

3. Select entity named aircraft 2

1. Action: Selects/Targets

2. Object: Aircraft 2

3. Outcome: Aircraft 2 selected

4. Tooling: Mouse

5. Time: When need to control self

4. Adjust Aircraft 2's heading

Q 1. Performs what action? (verb)
A 1. Adjusts/Changes heading

Q 2. To whom or what? (object of the verb)
A 2. Aircraft 2

Q 3. To produce what? or Why is it necessary? (expected outp
A 3. Heading adjustment

Q 4. Using what tools, equipment, work aids, processes?
A 4. Mouse, Entity, ??

Q 5. (temporality)
A 5. After unit has been selected

4. Adjust Aircraft 2's heading

1. Action: Adjust/Change

2. Object: Aircraft 2 heading

3. Outcome: Heading changed

4. Tooling: Mouse/Entity

5. Time: After unit selected

5. Configure distance measurement tool

Q 1. Performs what action? (verb)
A 1. Clicking

Q 2. To whom or what? (object of the verb)
A 2. Screen, two points (self and tanker)

Q 3. To produce what? or Why is it necessary? (expected outp
A 3. Line between two clicks

Q 4. Using what tools, equipment, work aids, processes?
A 4. Screen, Mouse

Q 5. (temporality)
A 5. After heading change

5. Config tool

1. Action: Clicking

2. Object: Screen, two points

3. Outcome: Line as Distance

4. Tooling: Mouse

5. Time: After changed heading

6. Wait until tool reads 20NM to tanker

Q 1. Performs what action? (verb)
A 1. Observation

Q 2. To whom or what? (object of the verb)
A 2. Distance measurement tool

Q 3. To produce what? or Why is it necessary? (expected outp
A 3. Correct distance identified

Q 4. Using what tools, equipment, work aids, processes?
A 4. Screen, HUD

Q 5. (temporality)
A 5. Once a units heading has been adjusted

6. Wait till 20NM

1. Action: Observe

2. Object: Distance tool

3. Outcome: Correct Distance

4. Tooling: HUD/Screen

5. Time: After heading adjustm

7. Contact control Agency

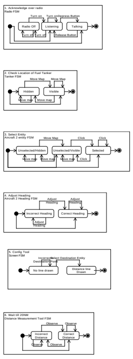
Q 1. Performs what action? (verb)
A 1.

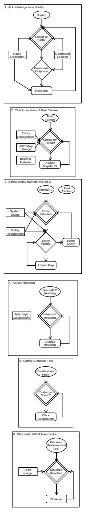
Q 2. To whom or what? (object of the verb)
A 2.

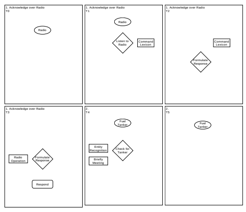
Q 3. To produce what? or Why is it necessary? (expected outp
A 3.

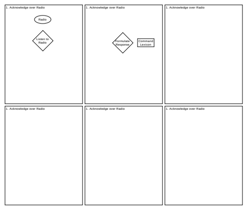
Q 4. Using what tools, equipment, work aids, processes?
A 4.

Q 5. (temporality)
A 5.









Appendix D

DCARR Presentation to AOSC

DCARR - Describe, Collect, Analyse, Recommend and Report

A framework for the identification of automation opportunities in large scale complex processes

Simon Vajda

APPLIED ARTIFICIAL INTELLIGENCE INSTITUTE

DEAKIN UNIVERSITY

1

Purpose

- Interactive
- Test communication
- Identify gaps
- Validation

2

Overview

- **Goal**:
  Identify Automation opportunities
- **Outcome**:
  Decision point based on recommendations
- **Process**:
  5 phased framework - DCARR

3

Motivation

White Forces are expensive and expertise is scarce.

This work aims to multiply existing capabilities through identification of suitable areas of automation.

4

Research Questions

- What tasks exist?
- What is/isn't suitable for automation?
- What is the mapping between tasks and automation suitability?
- What is the framework that ties the above together?

5

Areas of Interest

- Human Factors
- Cynefin
- Information Fusion
- UML / Meta modelling
- Human Automation Interaction

6

Benefits

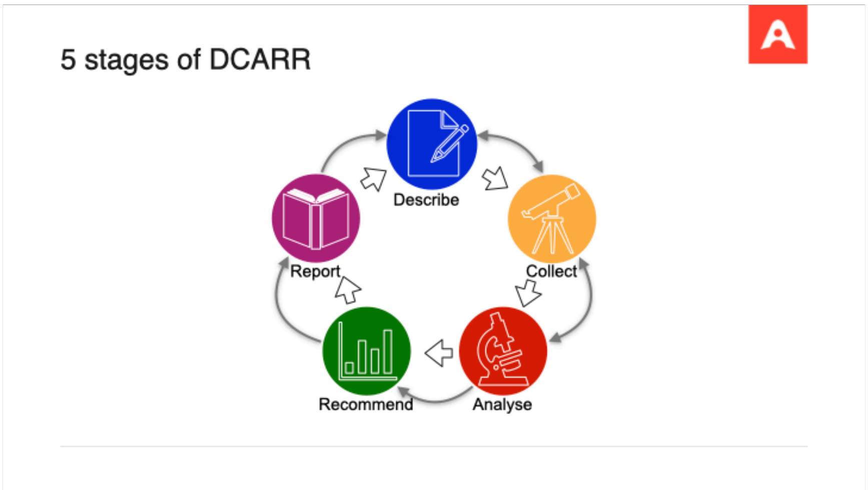
- Knowledge capture
- Discussion
- Goal driven
- Multiple exit points
- Information representation
- Basis for development
- Reproducible method

7

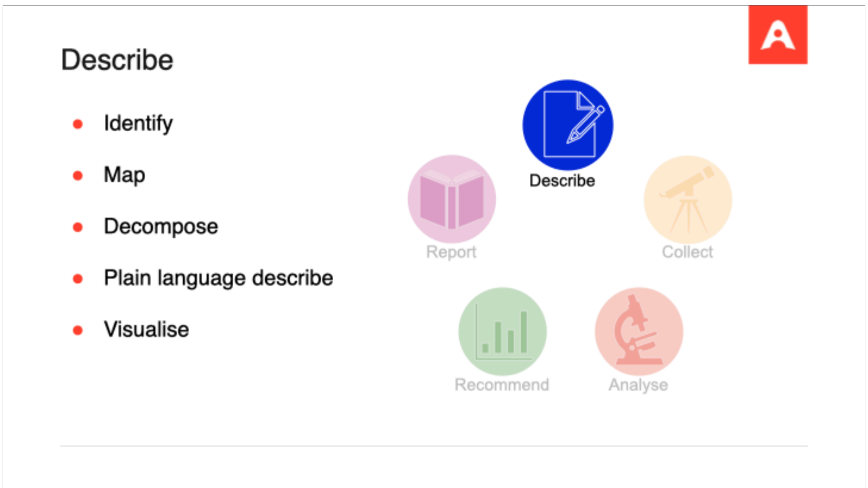
Limitations

- Requires investment - Access, Expertise, Time
- Data driven
- Out of Scope

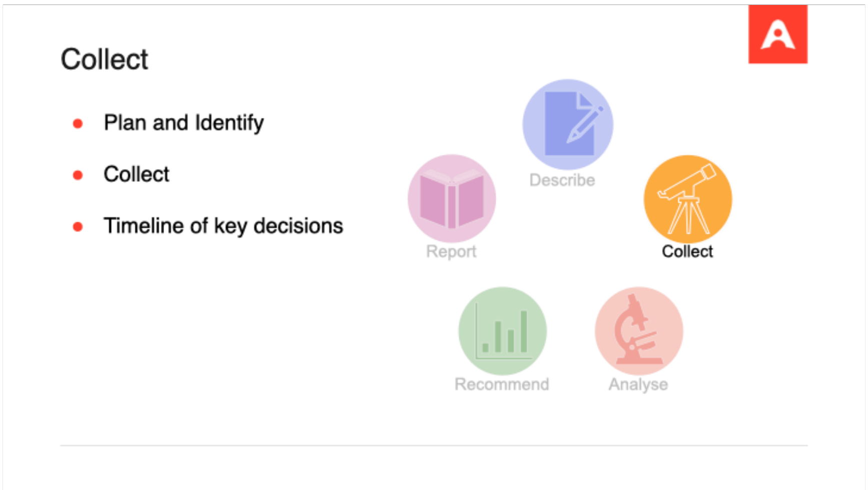
8


5 stages of DCARR
Describe
Collect
Analyse
Recommend
Report

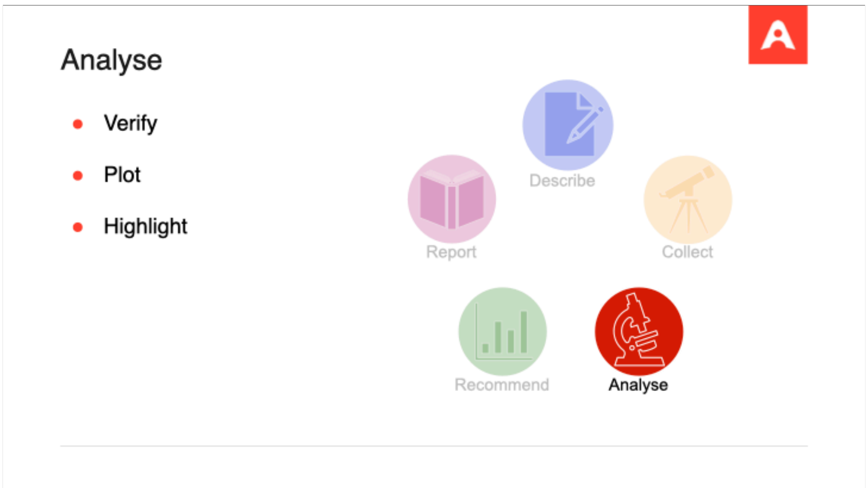
9


Describe
Identify
Map
Decompose
Plain language describe
Visualise
Describe
Report
Collect
Recommend
Analyse

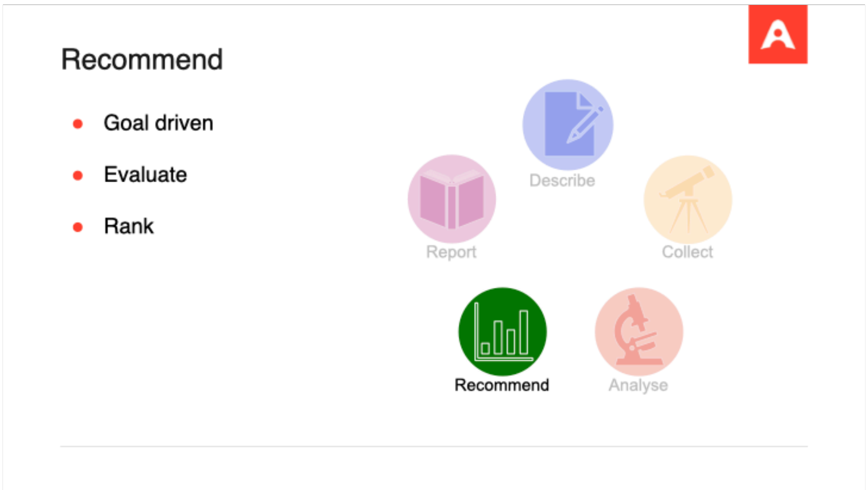
10


Collect
Plan and Identify
Collect
Timeline of key decisions
Describe
Report
Collect
Recommend
Analyse

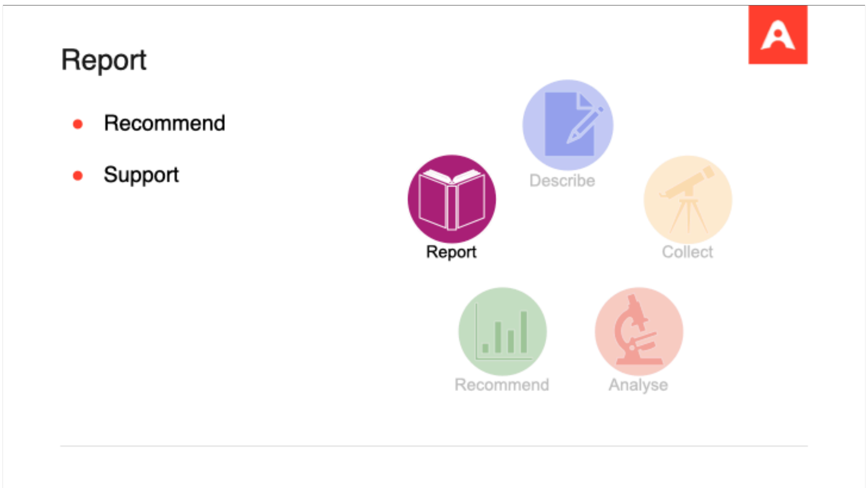
11


Analyse
Verify
Plot
Highlight
Describe
Report
Collect
Recommend
Analyse

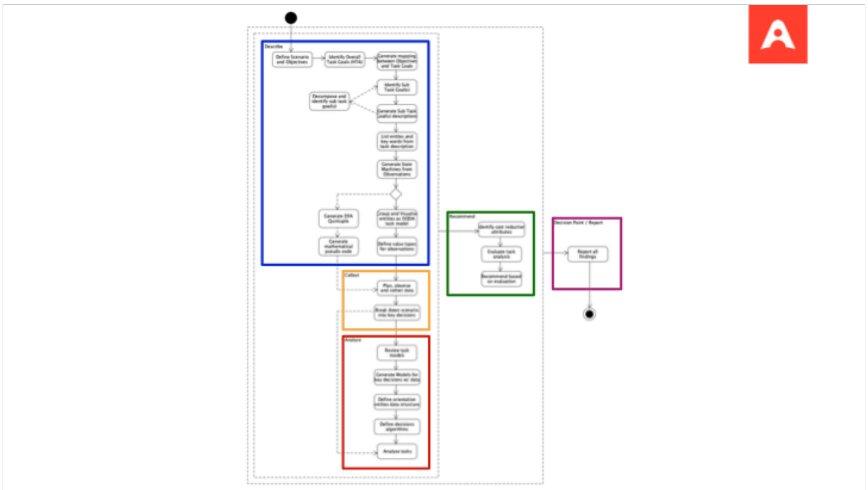
12


Recommend
Goal driven
Evaluate
Rank
Describe
Report
Collect
Recommend
Analyse

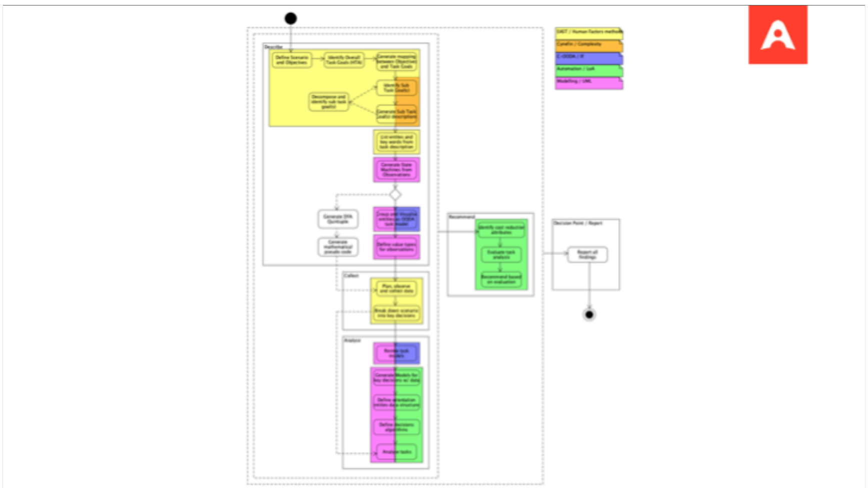
13


Report
Recommend
Support
Describe
Report
Collect
Recommend
Analyse

14



15

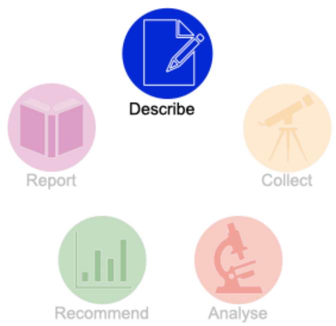


16

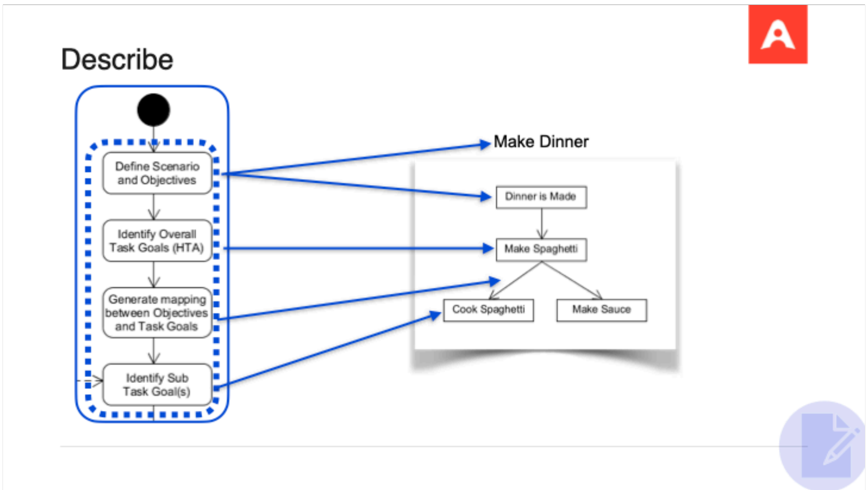
Demo 1

17

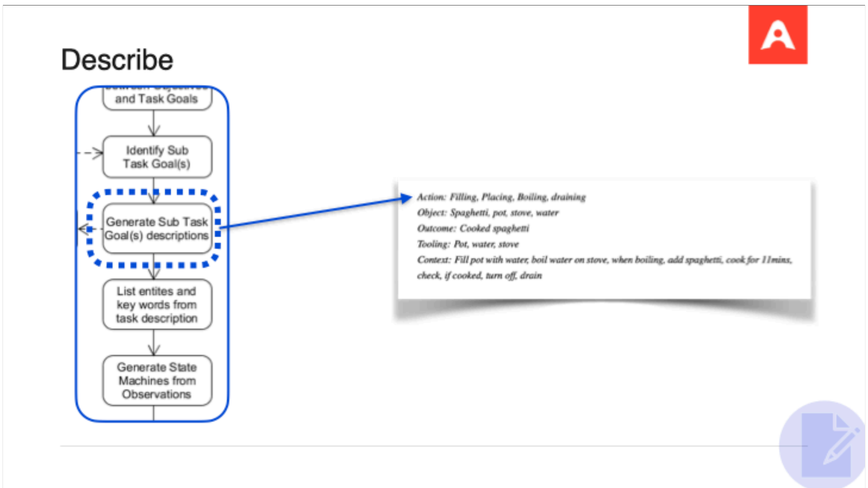
Describe



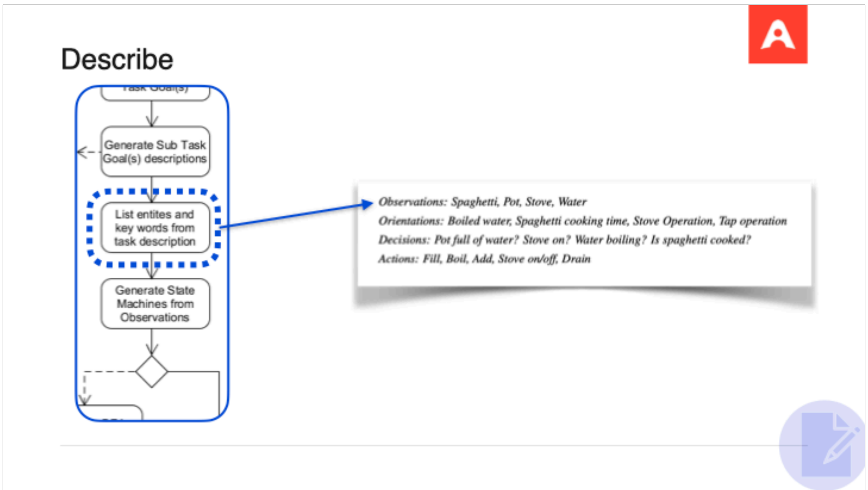
18


Describe
Define Scenario and Objectives
Identify Overall Task Goals (HTA)
Generate mapping between Objectives and Task Goals
Identify Sub Task Goal(s)
Make Dinner
Dinner is Made
Make Spaghetti
Cook Spaghetti
Make Sauce

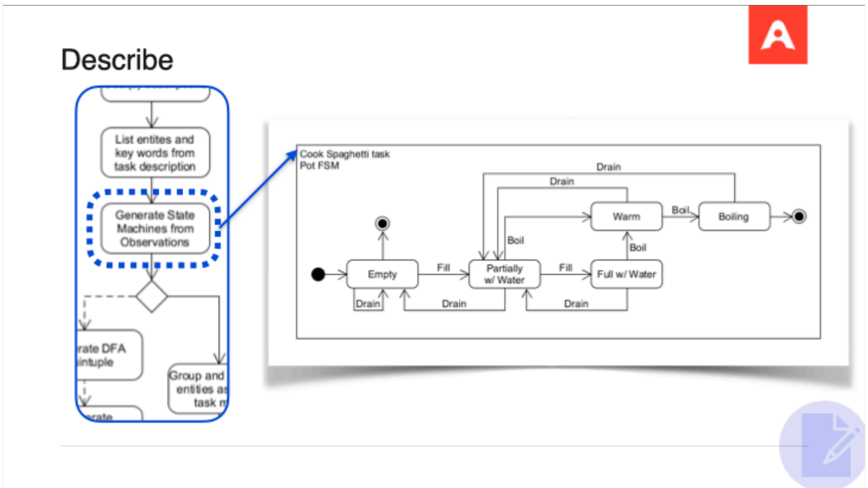
19


Describe
and Task Goals
Identify Sub Task Goal(s)
Generate Sub Task Goal(s) descriptions
List entites and key words from task description
Generate State Machines from Observations
Action: Filling, Placing, Boiling, draining
Object: Spaghetti, pot, stove, water
Outcome: Cooked spaghetti
Tooling: Pot, water, stove
Context: Fill pot with water, boil water on stove, when boiling, add spaghetti, cook for 11mins,
check, if cooked, turn off, drain

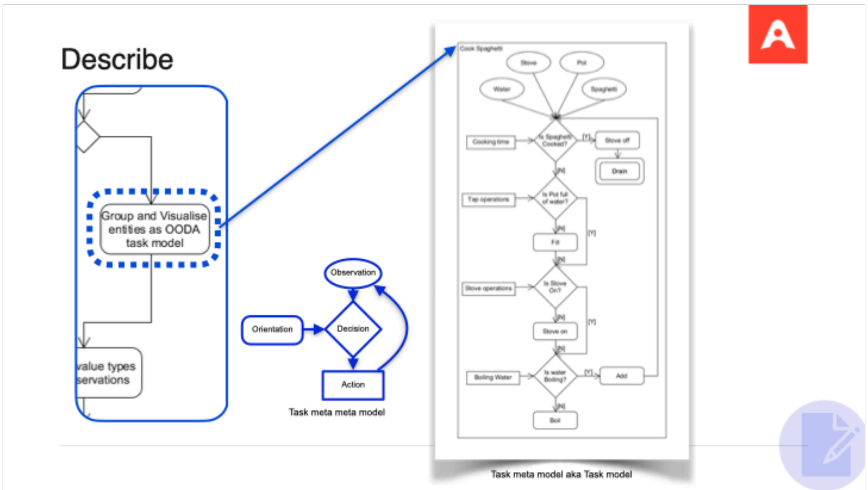
20


Describe
Generate Sub Task Goal(s) descriptions
List entites and key words from task description
Generate State Machines from Observations
Observations: Spaghetti, Pot, Stove, Water
Orientations: Boiled water, Spaghetti cooking time, Stove Operation, Tap operation
Decisions: Pot full of water? Stove on? Water boiling? Is spaghetti cooked?
Actions: Fill, Boil, Add, Stove on/off, Drain

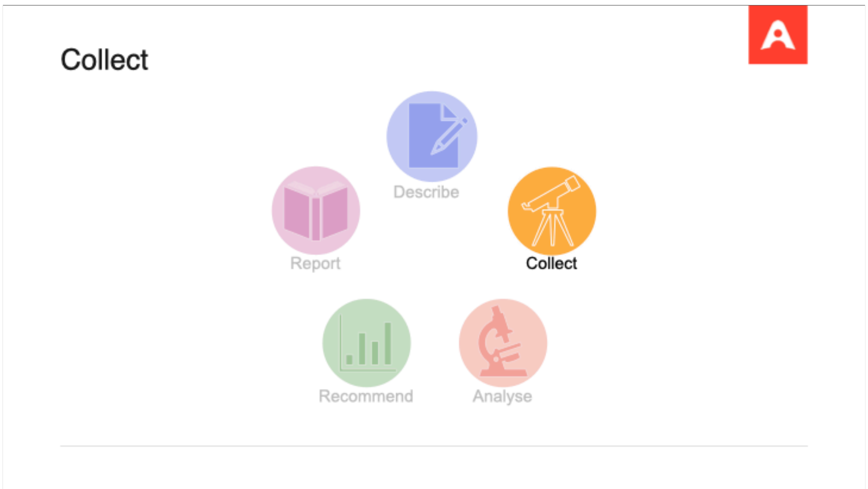
21


Describe
List entites and key words from task description
Generate State Machines from Observations
Cook Spaghetti task
Pot FSM
Empty
Partially w/ Water
Full w/ Water
Warm
Boiling
Fill
Boil
Drain

22


Describe
Group and Visualise entities as OODA task model
value types
servations
Observation
Orientation
Decision
Action
Task meta meta model
Task meta model aka Task model

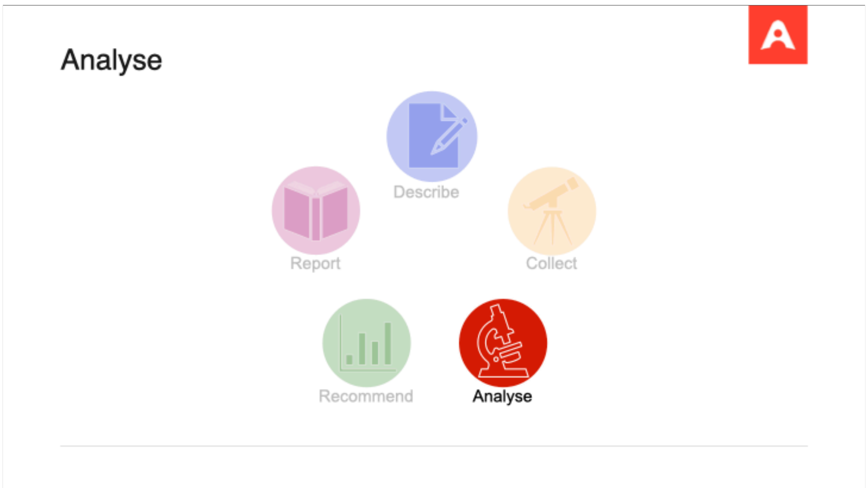
23


Collect
Describe
Report
Collect
Recommend
Analyse

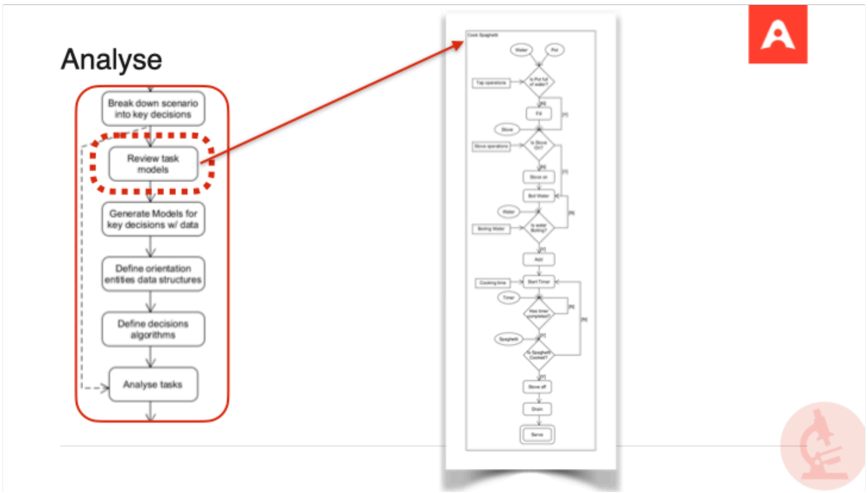
24


Collect
Define value types for observations
Plan, observe and collect data
Break down scenario into key decisions
Review task models
Generate Models for key decisions w/ data
0:01 Data collection for cooking pasta
0:10 I have a small pot
0:14 I am going to fill it with water from the tap
0:16 *fills pot with water from tap*
0:22 Water is cold
0:26 I am going to turn on the large burner on the stove top
0:28 *places pot on burner*
0:32 Will wait for that to heat up and start boiling
0:35 While that is boiling, I am going to look in the cupboard and find some pasta
0:43 Found some pasta I am going to cook for myself
0:48 I am going to read what the instructions say
0:55 Cook for 9 mins or to taste
1:07 I am going to wait for the water to boil
6:24 Water is now bubbling
6:38 *adds pasta to boiling water*
6:44 Get a timer going for 8 mins
7:04 *gets a large spoon and stirs pasta to stops it sticking*
14:48 Timer has gone off
14:58 *takes piece of pasta, runs under cold water, to taste*
15:07 *tastes paste, cooked al dente*
15:15 *turns off stove burner*
15:30 *takes pot off heat, takes pot to sink, drains water*
15:53 I have drained the water from the pot
16:08 *uses spoon to serve pasta from pot to small dish*
16:25 *drains rest of water from pot, leaves pot in sink*
16:32 Check to make sure stove is off *checks stove*
16:38 And I have some cooked pasta

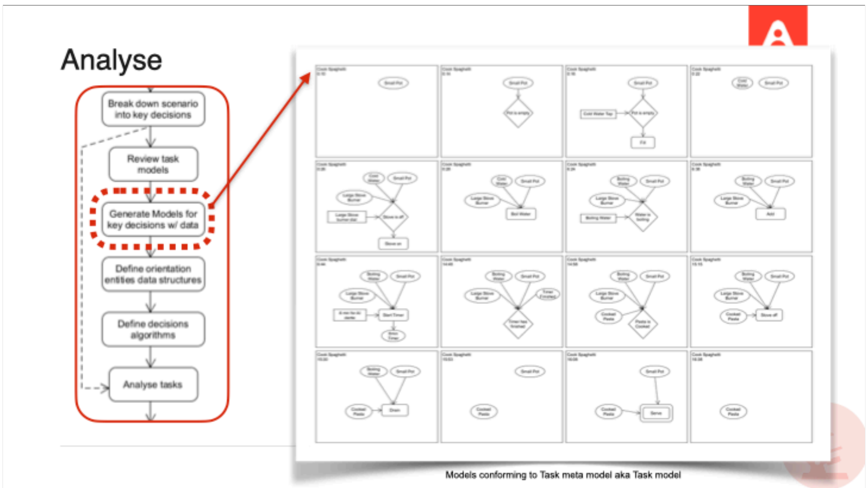
25


Analyse
Describe
Report
Collect
Recommend
Analyse

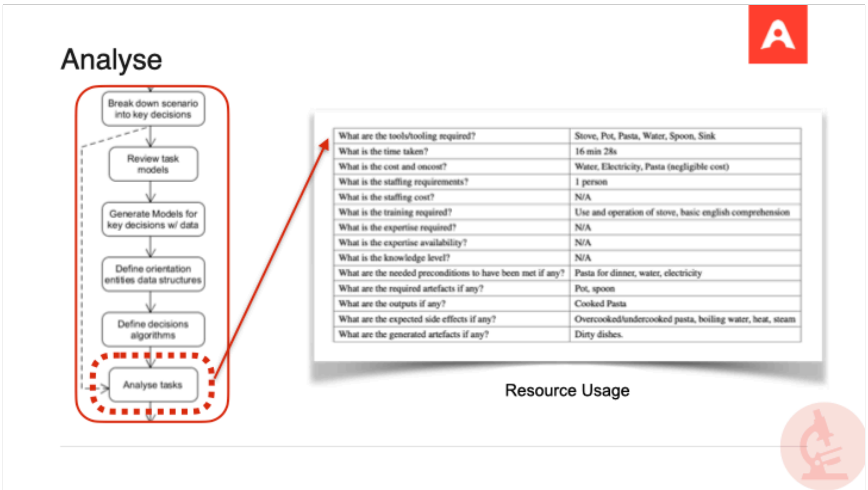
26


Analyse
Break down scenario into key decisions
Review task models
Generate Models for key decisions w/ data
Define orientation entities data structures
Define decisions algorithms
Analyse tasks

27

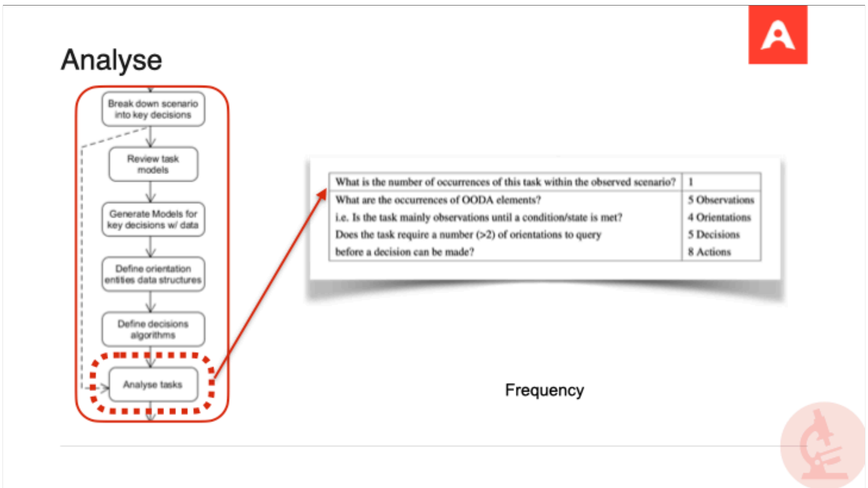

Analyse
Break down scenario into key decisions
Review task models
Generate Models for key decisions w/ data
Define orientation entities data structures
Define decisions algorithms
Analyse tasks
Models conforming to Task meta model aka Task model

28



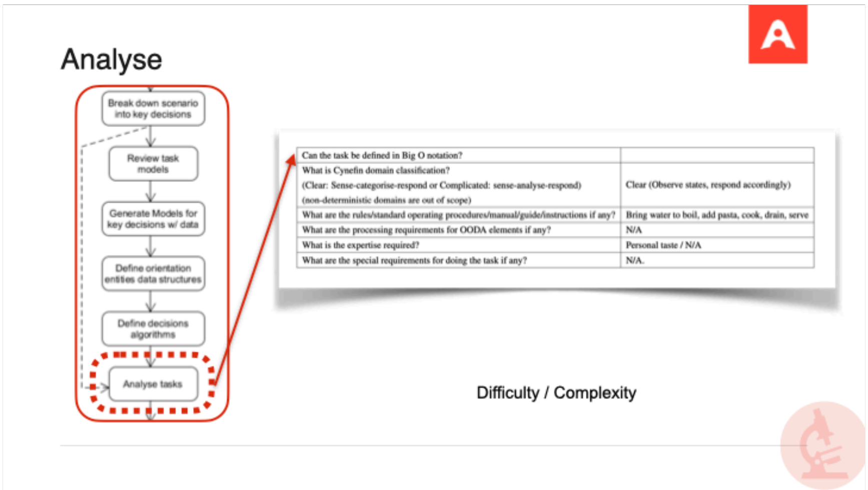
What are the tools/tooling required?	Stove, Pot, Pasta, Water, Spoon, Sink
What is the time taken?	16 min 28s
What is the cost and oncost?	Water, Electricity, Pasta (negligible cost)
What is the staffing requirements?	1 person
What is the staffing cost?	N/A
What is the training required?	Use and operation of stove, basic english comprehension
What is the expertise required?	N/A
What is the expertise availability?	N/A
What is the knowledge level?	N/A
What are the needed preconditions to have been met if any?	Pasta for dinner, water, electricity
What are the required artefacts if any?	Pot, spoon
What are the outputs if any?	Cooked Pasta
What are the expected side effects if any?	Overcooked/undercooked pasta, boiling water, heat, steam
What are the generated artefacts if any?	Dirty dishes.

29

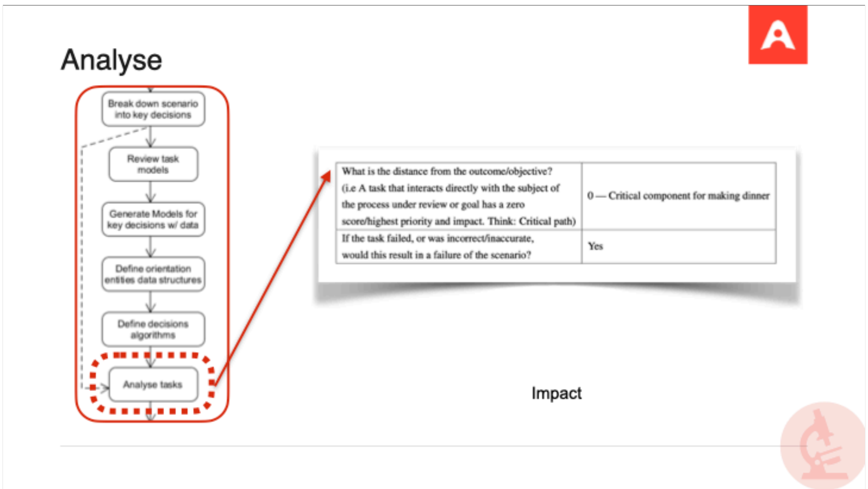


What is the number of occurrences of this task within the observed scenario?	1
What are the occurrences of OODA elements? i.e. Is the task mainly observations until a condition/state is met? Does the task require a number (>2) of orientations to query before a decision can be made?	5 Observations 4 Orientations 5 Decisions 8 Actions

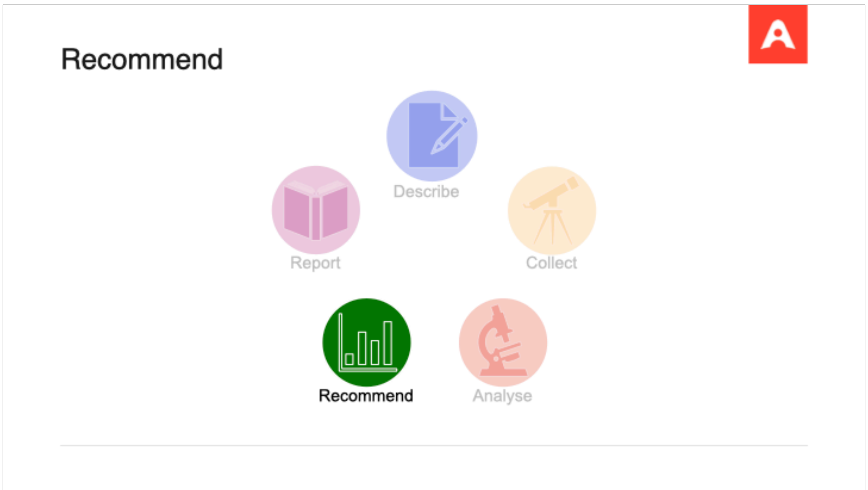
30


Analyse
Break down scenario into key decisions
Review task models
Generate Models for key decisions w/ data
Define orientation entities data structures
Define decisions algorithms
Analyse tasks
Can the task be defined in Big O notation?
What is Cynefin domain classification?
(Clear: Sense-categorise-respond or Complicated: sense-analyse-respond)
(non-deterministic domains are out of scope)
Clear (Observe states, respond accordingly)
What are the rules/standard operating procedures/manual/guide/instructions if any?
Bring water to boil, add pasta, cook, drain, serve
What are the processing requirements for OODA elements if any?
N/A
What is the expertise required?
Personal taste / N/A
What are the special requirements for doing the task if any?
N/A.
Difficulty / Complexity

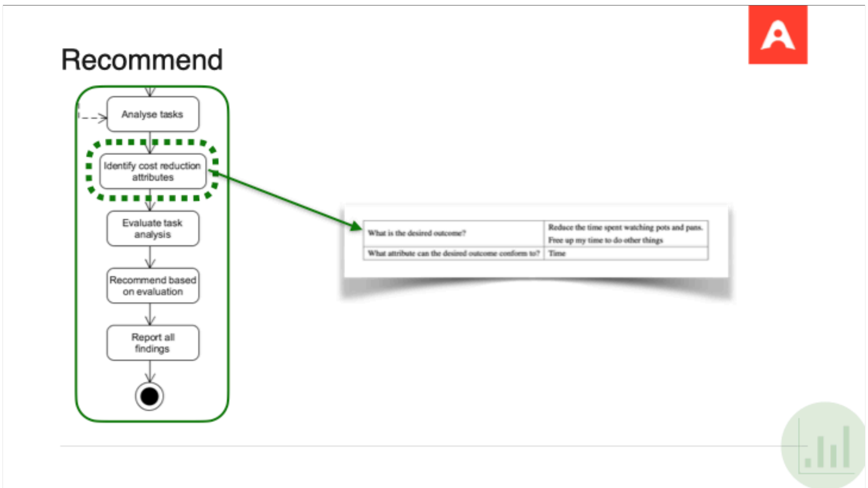
31


Analyse
Break down scenario into key decisions
Review task models
Generate Models for key decisions w/ data
Define orientation entities data structures
Define decisions algorithms
Analyse tasks
What is the distance from the outcome/objective?
(i.e A task that interacts directly with the subject of the process under review or goal has a zero score/highest priority and impact. Think: Critical path)
0 — Critical component for making dinner
If the task failed, or was incorrect/inaccurate, would this result in a failure of the scenario?
Yes
Impact

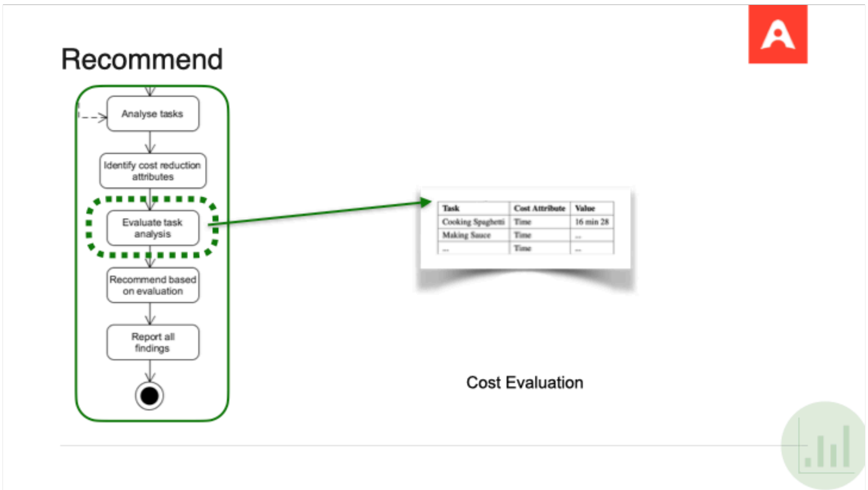
32


Recommend
Describe
Report
Collect
Recommend
Analyse

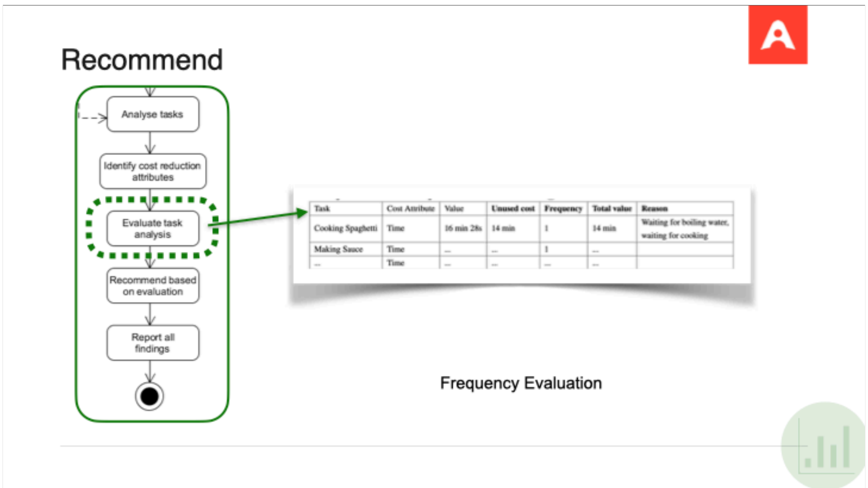
33


Recommend
Analyse tasks
Identify cost reduction attributes
Evaluate task analysis
Recommend based on evaluation
Report all findings
What is the desired outcome?
Reduce the time spent watching pots and pans.
Free up my time to do other things
What attribute can the desired outcome conform to?
Time

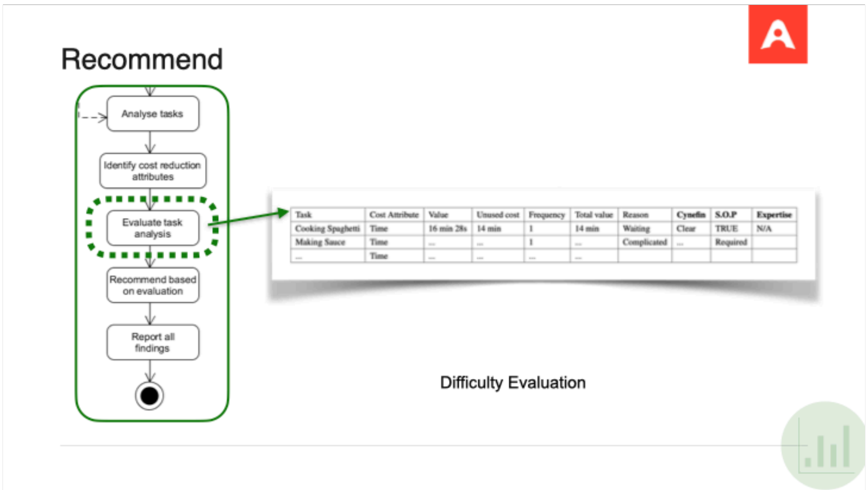
34


Recommend
Analyse tasks
Identify cost reduction attributes
Evaluate task analysis
Recommend based on evaluation
Report all findings
Task | Cost Attribute | Value
Cooking Spaghetti | Time | 16 min 28
Making Sauce | Time | ...
... | Time | ...
Cost Evaluation

35

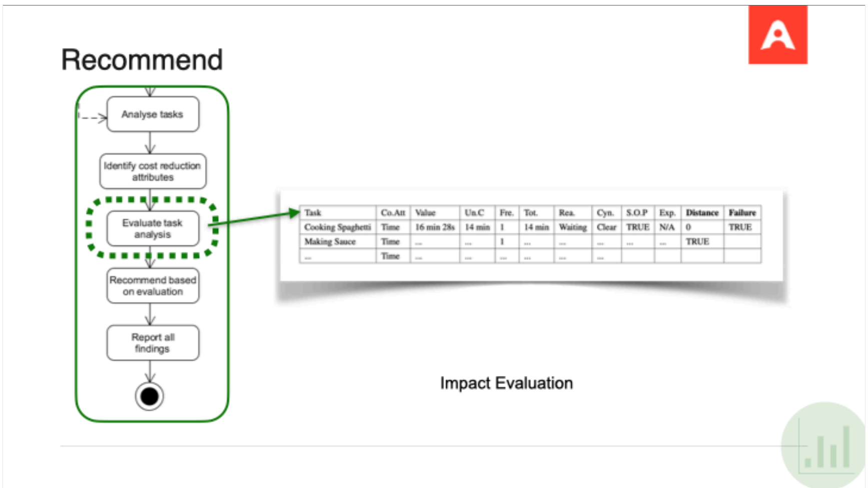

Recommend
Analyse tasks
Identify cost reduction attributes
Evaluate task analysis
Recommend based on evaluation
Report all findings
Task | Cost Attribute | Value | Unused cost | Frequency | Total value | Reason
Cooking Spaghetti | Time | 16 min 28s | 14 min | 1 | 14 min | Waiting for boiling water, waiting for cooking
Making Sauce | Time | ... | ... | 1 | ...
... | Time | ... | ... | ... | ...
Frequency Evaluation

36



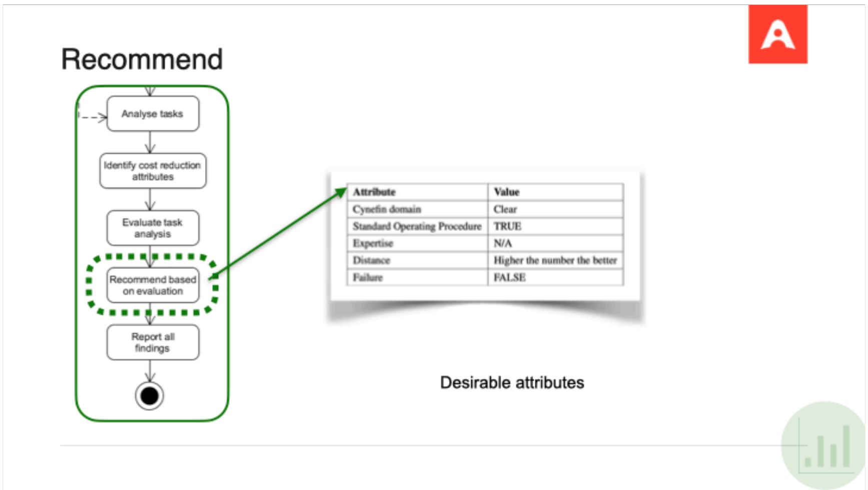
Task	Cost Attribute	Value	Unused cost	Frequency	Total value	Reason	**Cynefin**	**S.O.P**	**Expertise**
Cooking Spaghetti	Time	16 min 28s	14 min	1	14 min	Waiting	Clear	TRUE	N/A
Making Sauce	Time	...	...	1	...	Complicated	...	Required	
...	Time	...	...	...	...				

37



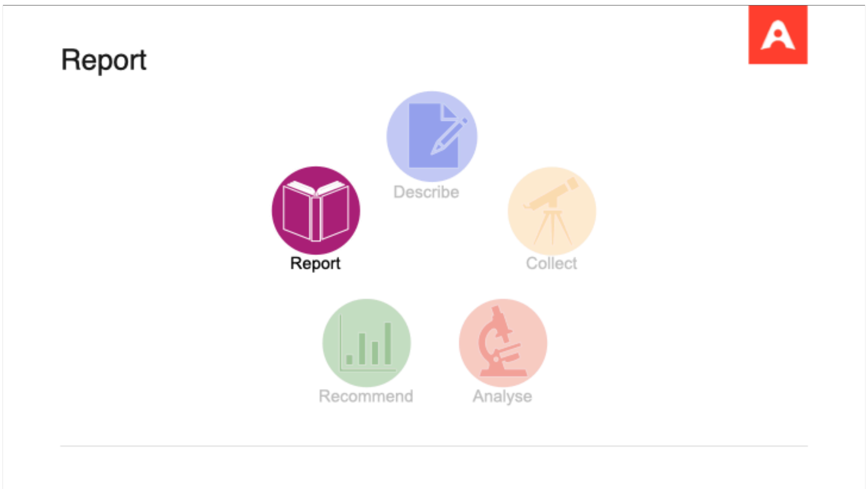
Task	Co.Att	Value	Un.C	Fre.	Tot.	Rea.	Cyn.	S.O.P	Exp.	**Distance**	**Failure**
Cooking Spaghetti	Time	16 min 28s	14 min	1	14 min	Waiting	Clear	TRUE	N/A	0	TRUE
Making Sauce	Time	...	...	1	...	...	...	...	...	TRUE	
...	Time	...	...	...	...	...	...				

38

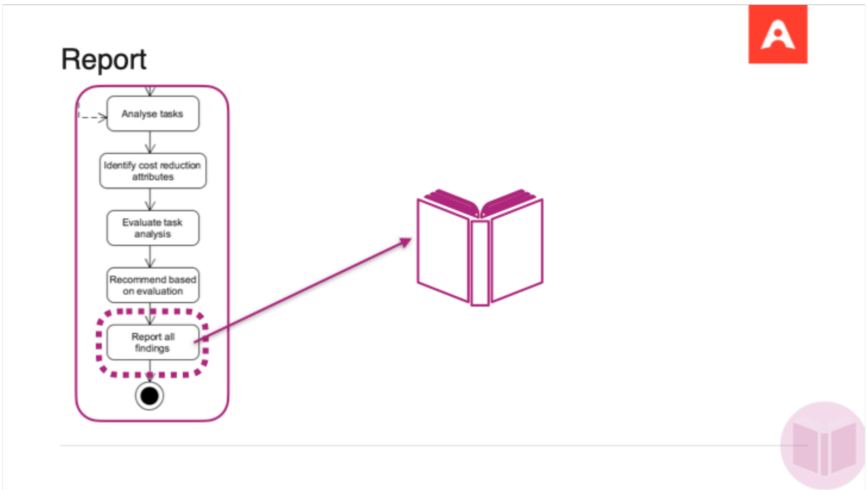


Attribute	Value
Cynefin domain	Clear
Standard Operating Procedure	TRUE
Expertise	N/A
Distance	Higher the number the better
Failure	FALSE

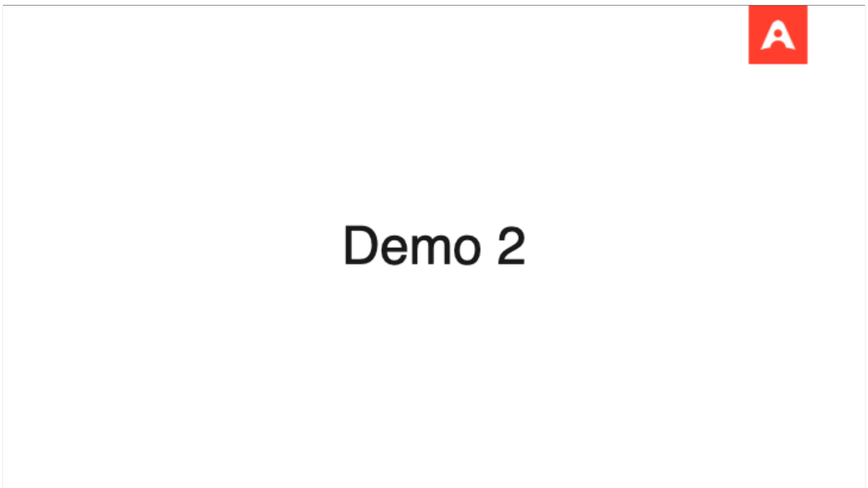
39



40


Report
Analyse tasks
Identify cost reduction attributes
Evaluate task analysis
Recommend based on evaluation
Report all findings

41


Demo 2

42

Task classification

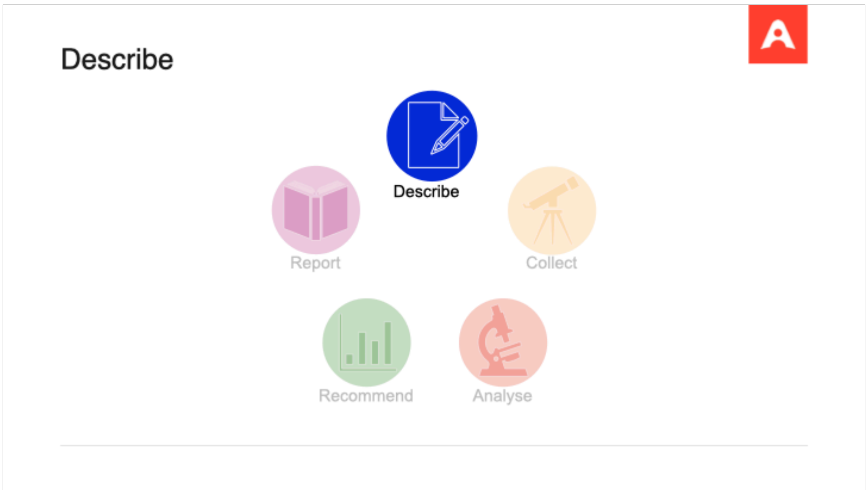
- Control
- Setup

43

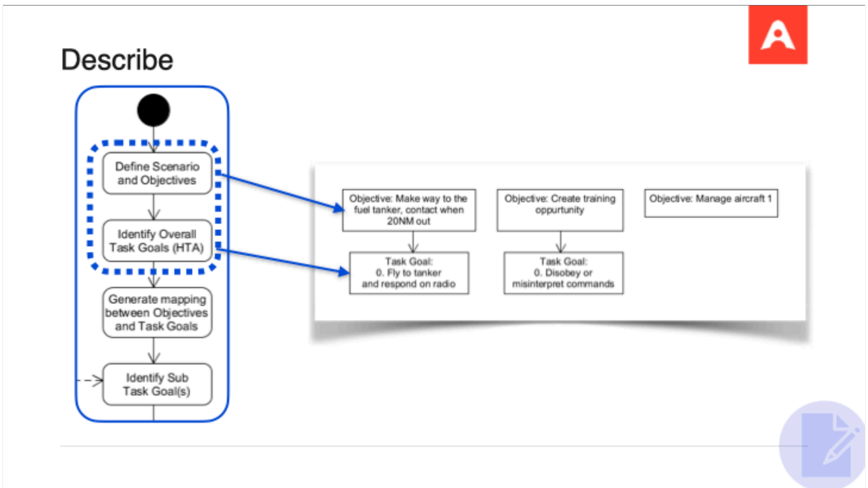
Scenario (Credit: Peter Ross!)

- A simpilot is controlling two aircraft. He/she receives a command over the radio from a trainee in the control agency (wedgetail): “aircraft 2, make your way to the fuel tanker, contact me when you are 20 nautical miles out” (this is paraphrased).

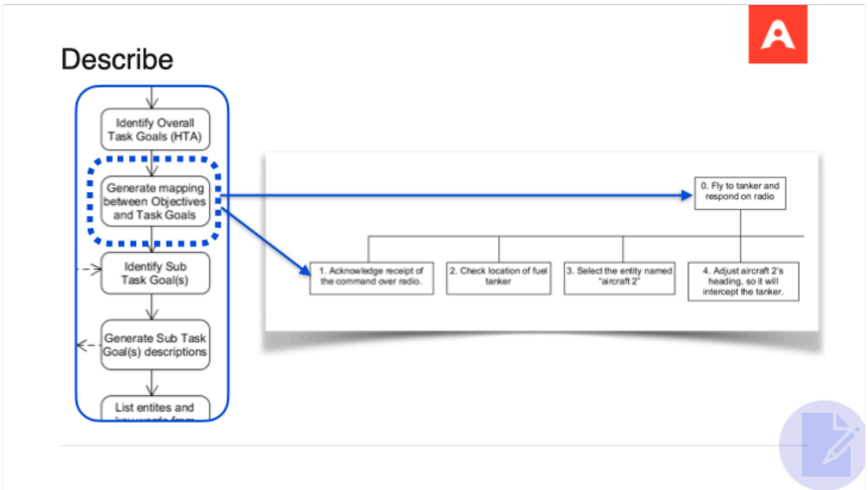
44


Describe
Describe
Report
Collect
Recommend
Analyse

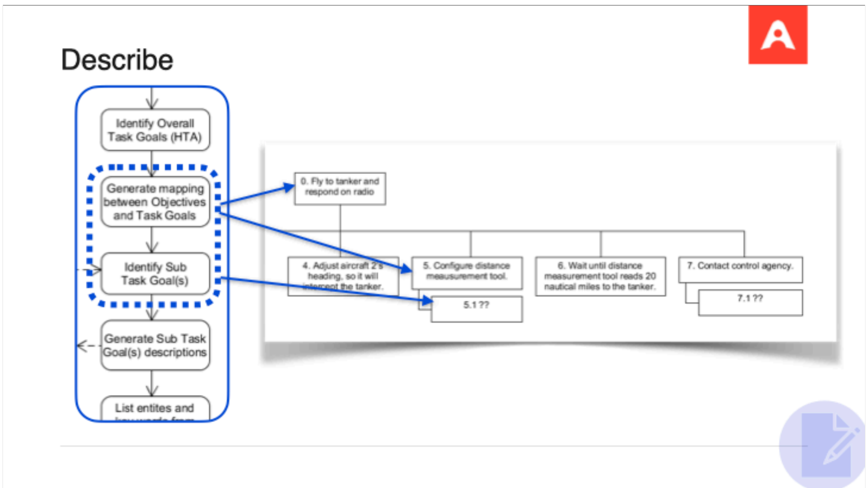
45


Describe
Define Scenario and Objectives
Identify Overall Task Goals (HTA)
Generate mapping between Objectives and Task Goals
Identify Sub Task Goal(s)
Objective: Make way to the fuel tanker, contact when 20NM out
Objective: Create training oppurtunity
Objective: Manage aircraft 1
Task Goal: 0. Fly to tanker and respond on radio
Task Goal: 0. Disobey or misinterpret commands

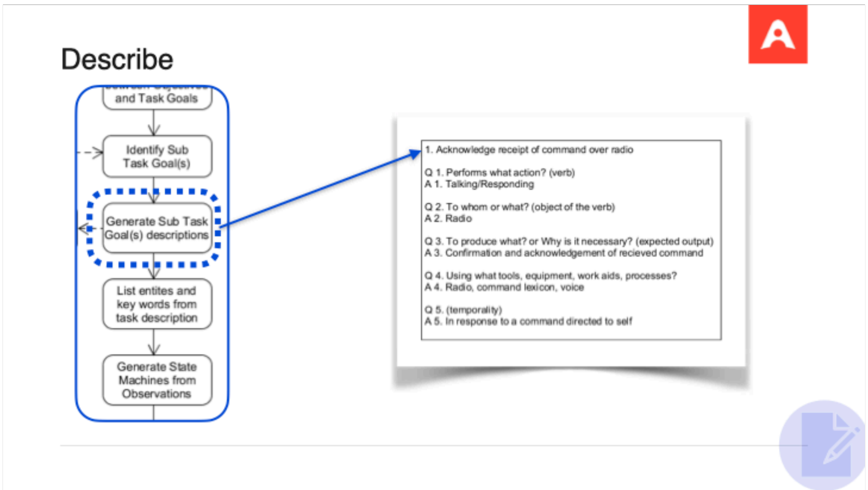
46


Describe
Identify Overall Task Goals (HTA)
Generate mapping between Objectives and Task Goals
Identify Sub Task Goal(s)
Generate Sub Task Goal(s) descriptions
List entites and
0. Fly to tanker and respond on radio
1. Acknowledge receipt of the command over radio.
2. Check location of fuel tanker
3. Select the entity named "aircraft 2"
4. Adjust aircraft 2's heading, so it will intercept the tanker.

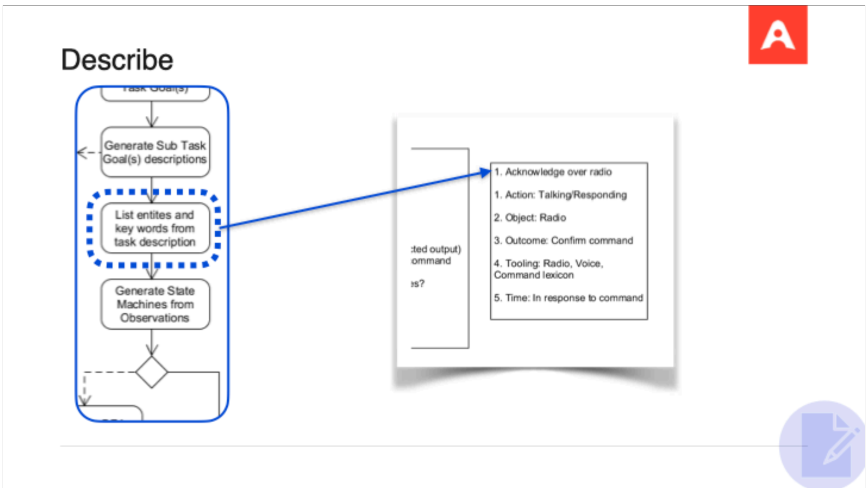
47


Describe
Identify Overall Task Goals (HTA)
Generate mapping between Objectives and Task Goals
Identify Sub Task Goal(s)
Generate Sub Task Goal(s) descriptions
List entites and
0. Fly to tanker and respond on radio
4. Adjust aircraft 2's heading, so it will intercept the tanker.
5. Configure distance meausurement tool.
5.1 ??
6. Wait until distance measurement tool reads 20 nautical miles to the tanker.
7. Contact control agency.
7.1 ??

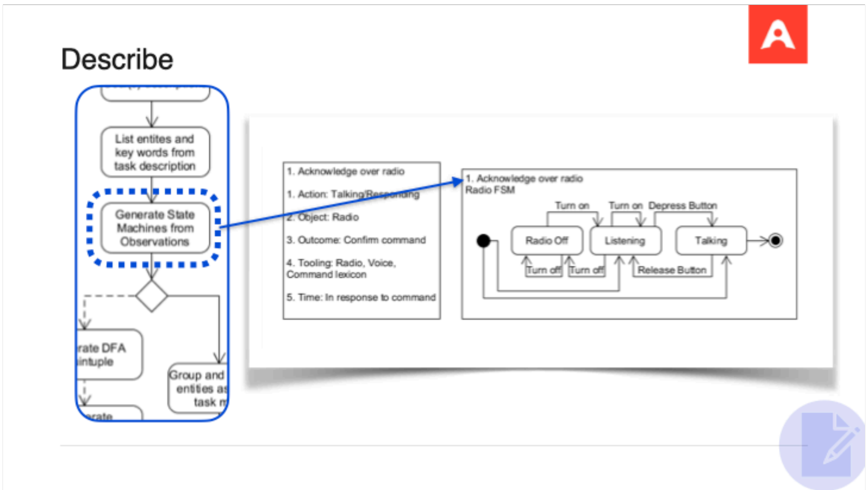
48


Describe
and Task Goals
Identify Sub Task Goal(s)
Generate Sub Task Goal(s) descriptions
List entites and key words from task description
Generate State Machines from Observations
1. Acknowledge receipt of command over radio
Q 1. Performs what action? (verb)
A 1. Talking/Responding
Q 2. To whom or what? (object of the verb)
A 2. Radio
Q 3. To produce what? or Why is it necessary? (expected output)
A 3. Confirmation and acknowledgement of recieved command
Q 4. Using what tools, equipment, work aids, processes?
A 4. Radio, command lexicon, voice
Q 5. (temporality)
A 5. In response to a command directed to self

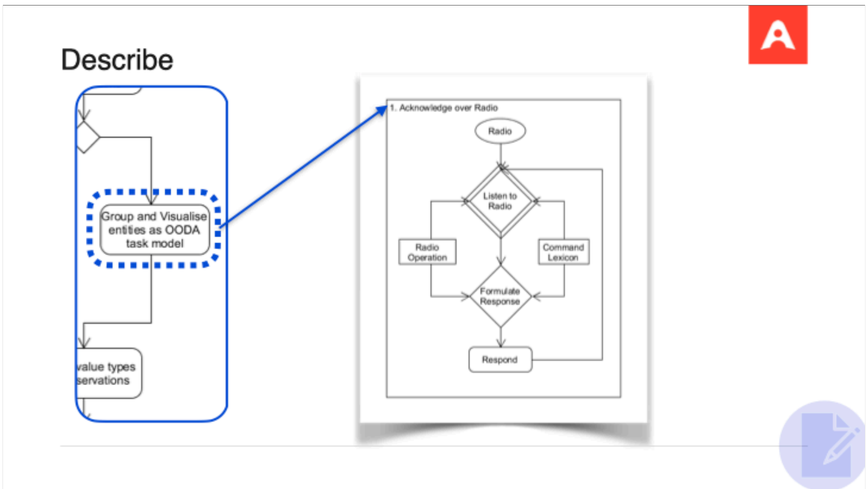
49


Describe
Generate Sub Task Goal(s) descriptions
List entites and key words from task description
Generate State Machines from Observations
1. Acknowledge over radio
1. Action: Talking/Responding
2. Object: Radio
3. Outcome: Confirm command
4. Tooling: Radio, Voice, Command lexicon
5. Time: In response to command

50


Describe
List entites and key words from task description
Generate State Machines from Observations
rate DFA intuple
Group and entities as task m
1. Acknowledge over radio
1. Action: Talking/Responding
2. Object: Radio
3. Outcome: Confirm command
4. Tooling: Radio, Voice, Command lexicon
5. Time: In response to command
1. Acknowledge over radio
Radio FSM
Turn on
Turn on
Depress Button
Radio Off
Listening
Talking
Turn off
Turn off
Release Button

51


Describe
Group and Visualise entities as OODA task model
value types servations
1. Acknowledge over Radio
Radio
Listen to Radio
Radio Operation
Command Lexicon
Formulate Response
Respond

52

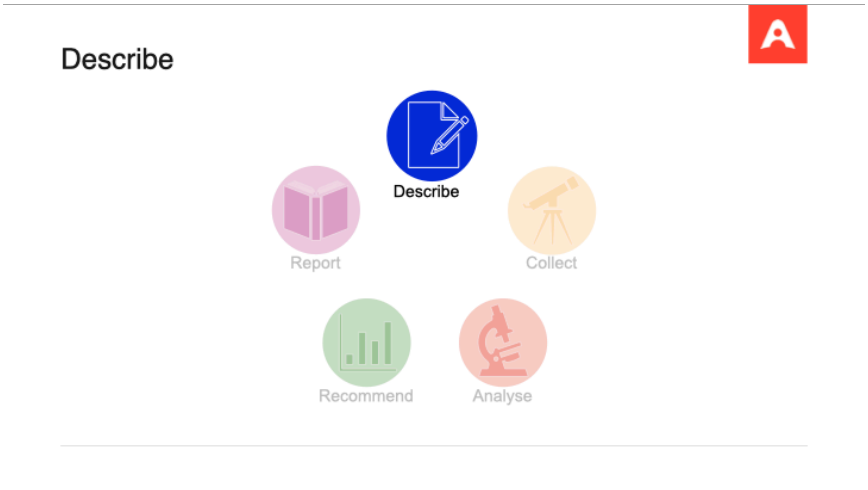
Workshop session

53

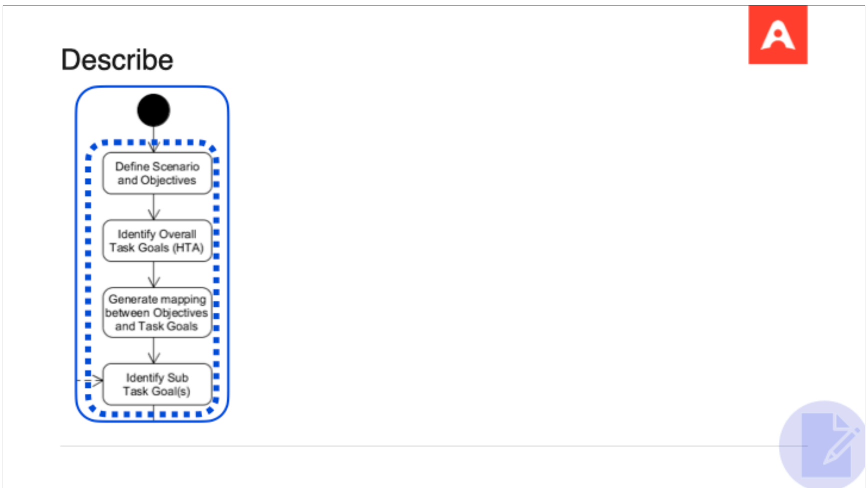
Workshop session

- Goal:
  Feedback and Discussion /
  Validate DCARR
- Tasks:
  - Explain the concepts
  - Walk through examples
  - Try it out together

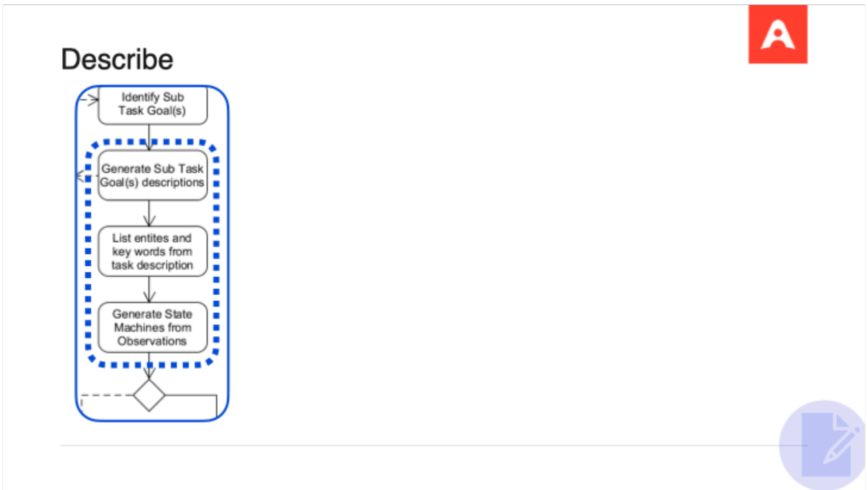
54


Describe
Describe
Report
Collect
Recommend
Analyse

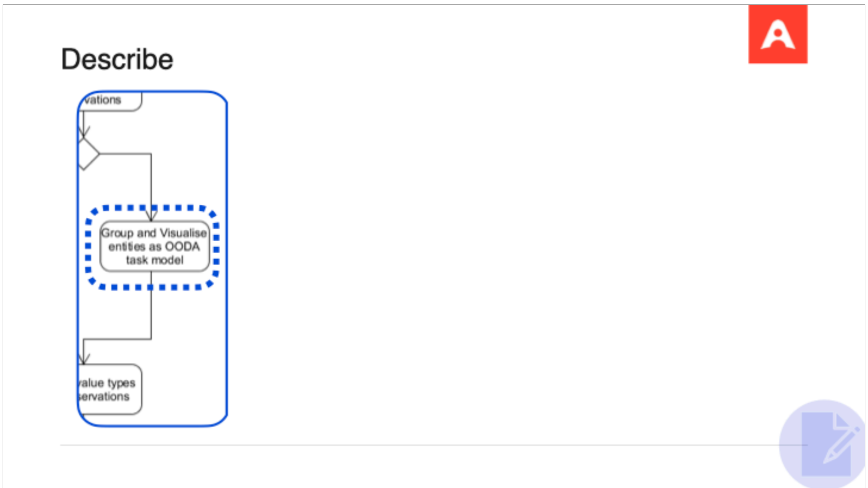
55


Describe
Define Scenario and Objectives
Identify Overall Task Goals (HTA)
Generate mapping between Objectives and Task Goals
Identify Sub Task Goal(s)

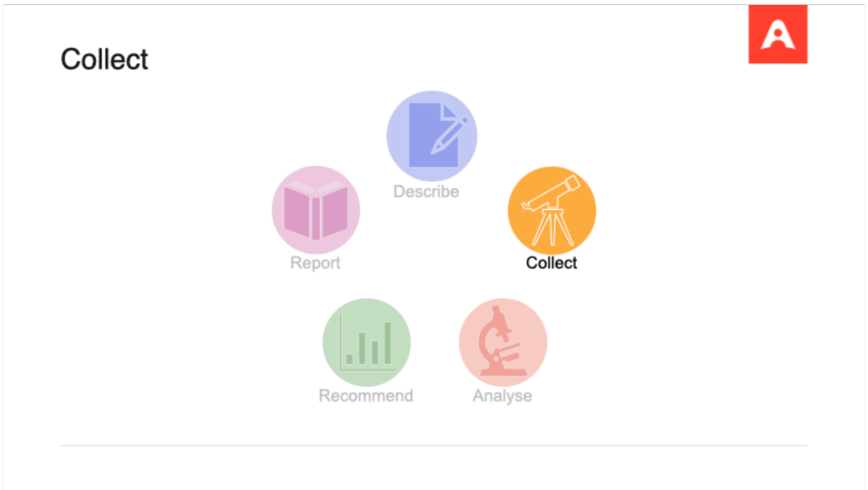
56


Describe
Identify Sub Task Goal(s)
Generate Sub Task Goal(s) descriptions
List entites and key words from task description
Generate State Machines from Observations

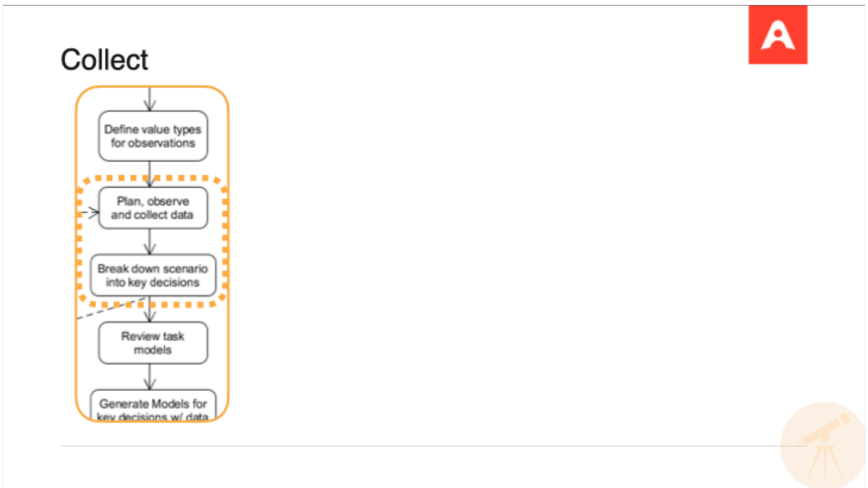
57


Describe
vations
Group and Visualise entities as OODA task model
value types
servations

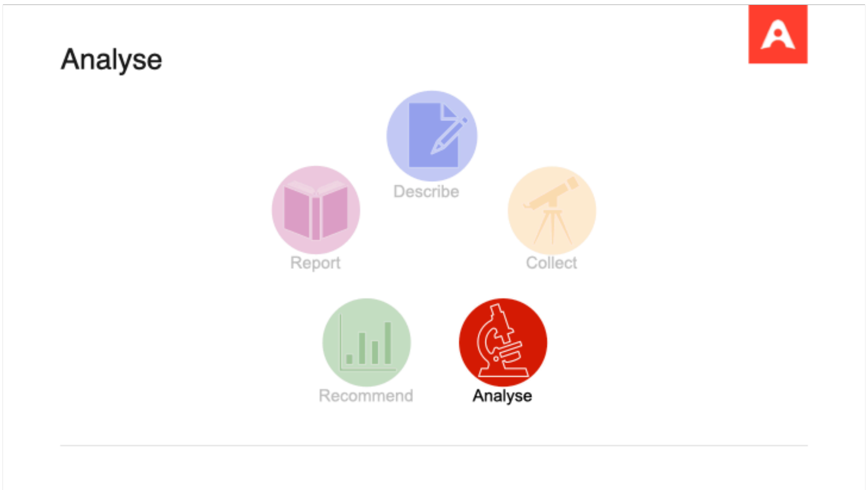
58


Collect
Describe
Report
Collect
Recommend
Analyse

59


Collect
Define value types for observations
Plan, observe and collect data
Break down scenario into key decisions
Review task models
Generate Models for key decisions w/ data

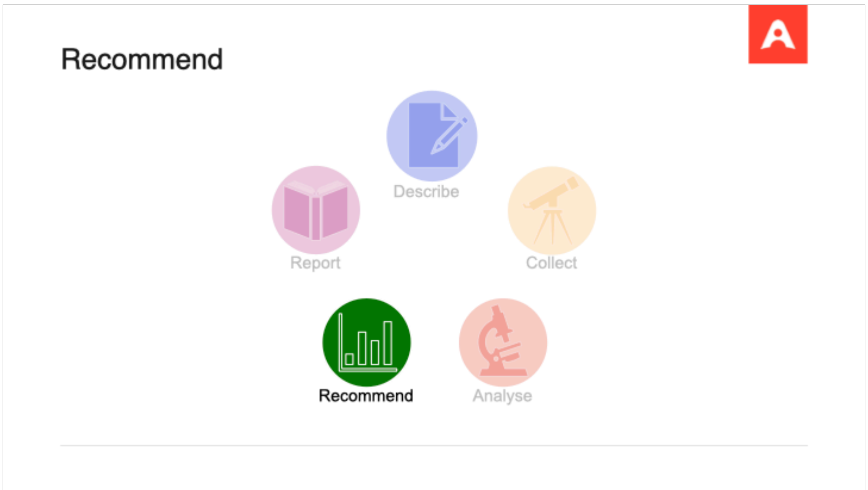
60


Analyse
Describe
Report
Collect
Recommend
Analyse

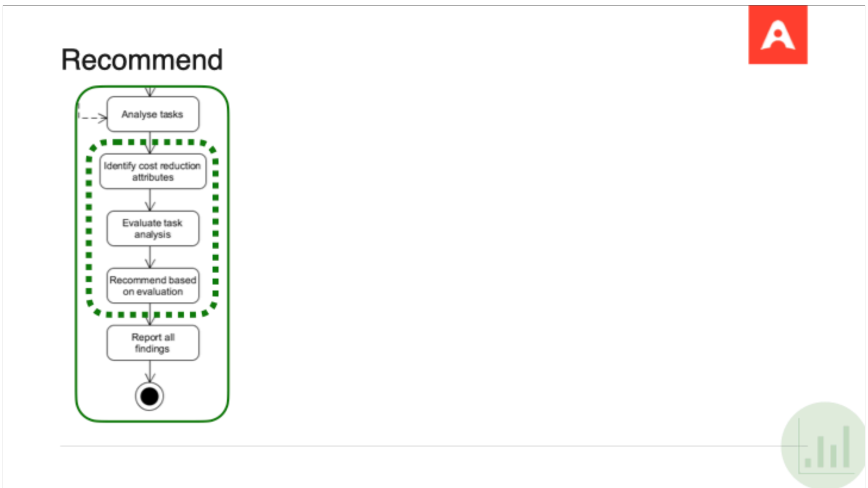
61


Analyse
Break down scenario into key decisions
Review task models
Generate Models for key decisions w/ data
Define orientation entities data structures
Define decisions algorithms
Analyse tasks

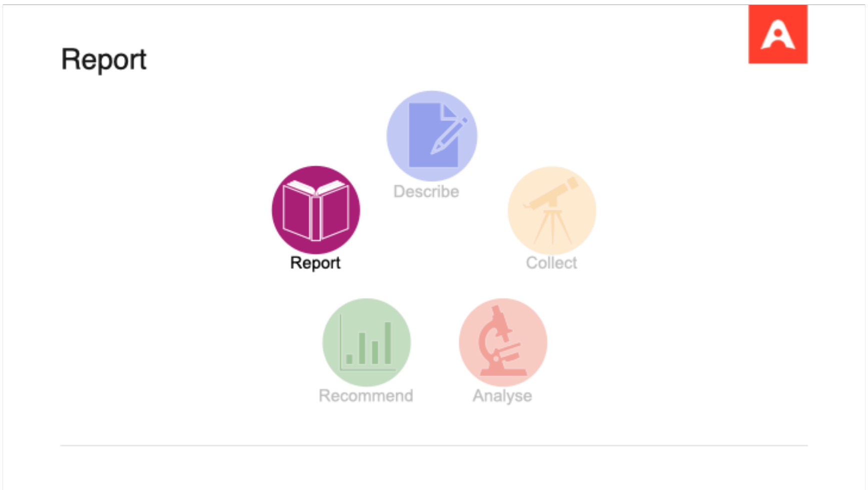
62


Recommend
Describe
Report
Collect
Recommend
Analyse

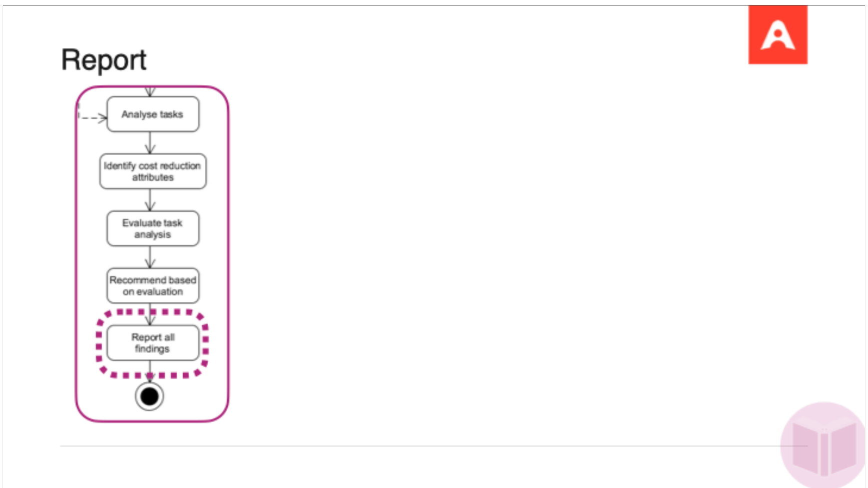
63


Recommend
Analyse tasks
Identify cost reduction attributes
Evaluate task analysis
Recommend based on evaluation
Report all findings

64


Report
Describe
Report
Collect
Recommend
Analyse

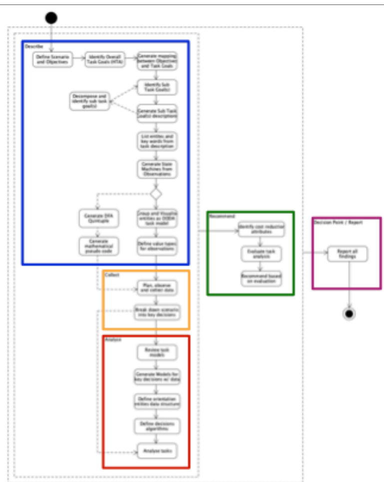
65


Report
Analyse tasks
Identify cost reduction attributes
Evaluate task analysis
Recommend based on evaluation
Report all findings

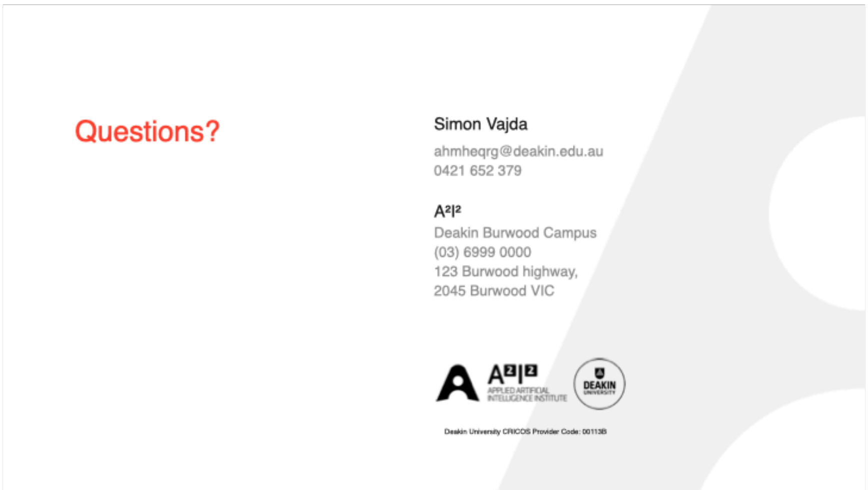
66

Quick Recap

67



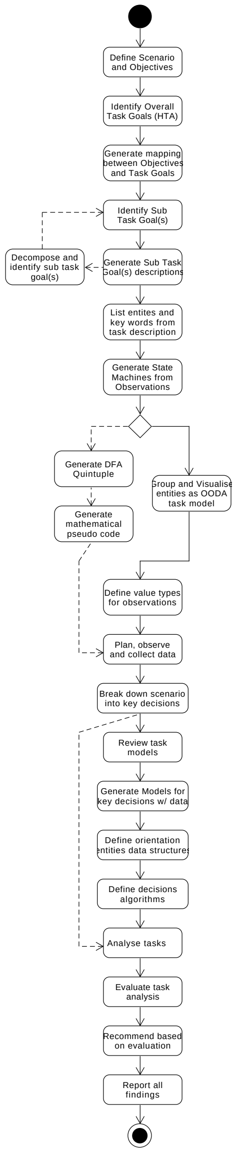
68

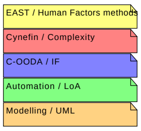

Questions?
Simon Vajda
ahmheqrg@deakin.edu.au
0421 652 379
A²I²
Deakin Burwood Campus
(03) 6999 0000
123 Burwood highway,
2045 Burwood VIC
Deakin University CRICOS Provider Code: 00113B

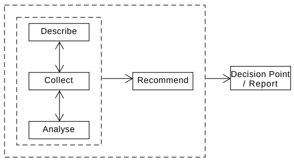
69

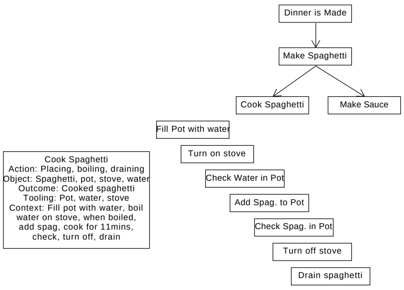
Appendix E

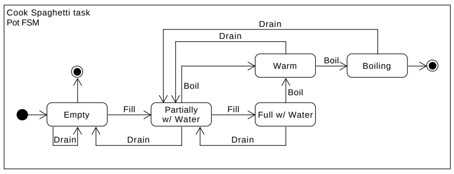
AOSC Presentation — Demo 1

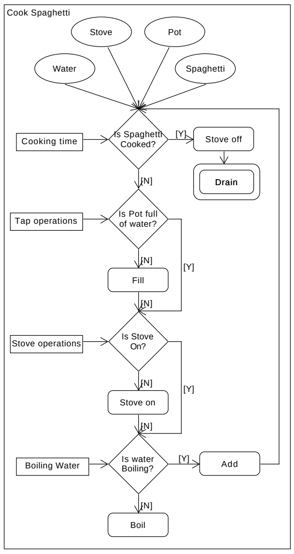

Define Scenario and Objectives
Identify Overall Task Goals (HTA)
Generate mapping between Objectives and Task Goals
Identify Sub Task Goal(s)
Decompose and identify sub task goal(s)
Generate Sub Task Goal(s) descriptions
List entites and key words from task description
Generate State Machines from Observations
Generate DFA Quintuple
Generate mathematical pseudo code
Group and Visualise entities as OODA task model
Define value types for observations
Plan, observe and collect data
Break down scenario into key decisions
Review task models
Generate Models for key decisions w/ data
Define orientation entities data structures
Define decisions algorithms
Analyse tasks
Evaluate task analysis
Recommend based on evaluation
Report all findings

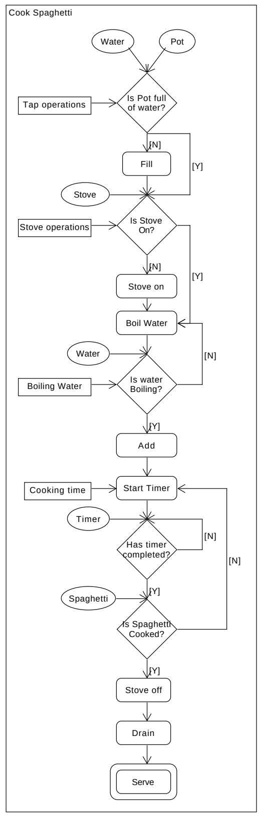

EAST / Human Factors methods
Cynefin / Complexity
C-OODA / IF
Automation / LoA
Modelling / UML

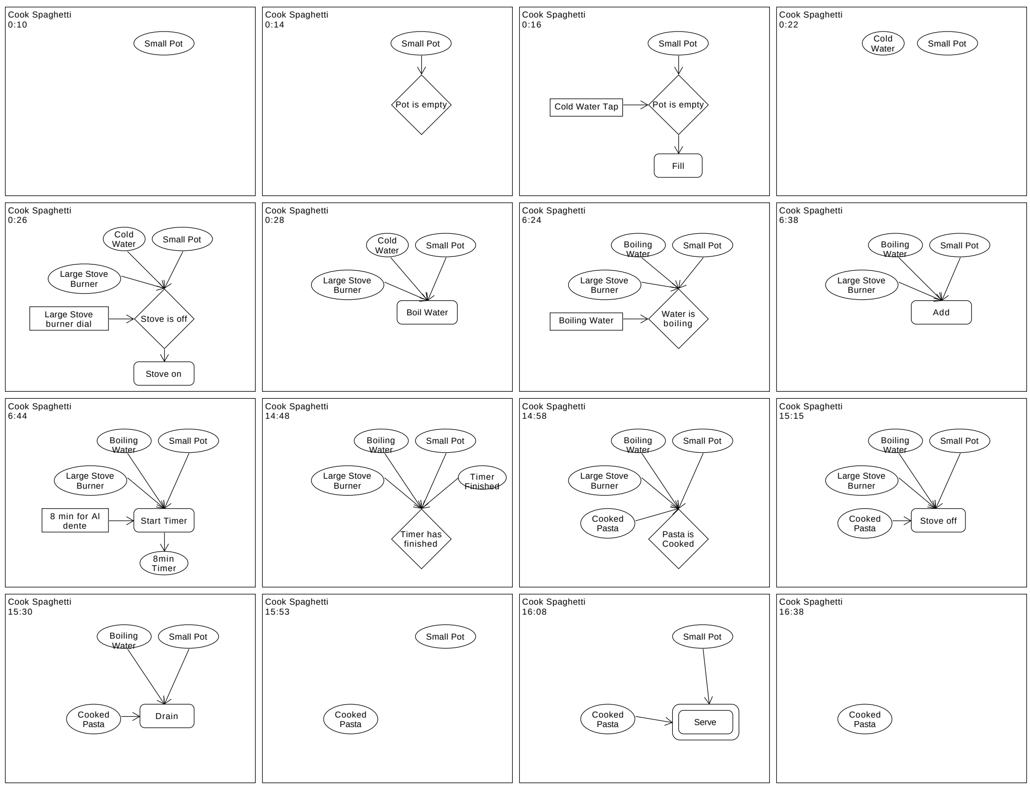

Describe
Collect
Analyse
Recommend
Decision Point / Report


Dinner is Made
Make Spaghetti
Cook Spaghetti
Make Sauce
Fill Pot with water
Turn on stove
Check Water in Pot
Add Spag. to Pot
Check Spag. in Pot
Turn off stove
Drain spaghetti
Cook Spaghetti
Action: Placing, boiling, draining
Object: Spaghetti, pot, stove, water
Outcome: Cooked spaghetti
Tooling: Pot, water, stove
Context: Fill pot with water, boil water on stove, when boiled, add spag, cook for 11mins, check, turn off, drain


Cook Spaghetti task
Pot FSM
Empty
Fill
Partially w/ Water
Fill
Full w/ Water
Boil
Warm
Boil
Boiling
Boil
Drain
Drain
Drain
Drain
Drain


Cook Spaghetti
Stove
Pot
Water
Spaghetti
Cooking time
Is Spaghetti Cooked?
[Y]
Stove off
Drain
[N]
Tap operations
Is Pot full of water?
[N]
Fill
[Y]
[N]
Stove operations
Is Stove On?
[N]
Stove on
[Y]
[N]
Boiling Water
Is water Boiling?
[Y]
Add
[N]
Boil


Cook Spaghetti
Water
Pot
Tap operations
Is Pot full of water?
[N]
[Y]
Fill
Stove
Stove operations
Is Stove On?
[N]
[Y]
Stove on
Boil Water
Water
Boiling Water
Is water Boiling?
[N]
[Y]
Add
Cooking time
Start Timer
Timer
Has timer completed?
[N]
[N]
[Y]
Spaghetti
Is Spaghetti Cooked?
[Y]
Stove off
Drain
Serve


Cook Spaghetti 0:10
Small Pot
Cook Spaghetti 0:14
Small Pot
Pot is empty
Cook Spaghetti 0:16
Small Pot
Cold Water Tap
Pot is empty
Fill
Cook Spaghetti 0:22
Cold Water
Small Pot
Cook Spaghetti 0:26
Cold Water
Small Pot
Large Stove Burner
Large Stove burner dial
Stove is off
Stove on
Cook Spaghetti 0:28
Cold Water
Small Pot
Large Stove Burner
Boil Water
Cook Spaghetti 6:24
Boiling Water
Small Pot
Large Stove Burner
Boiling Water
Water is boiling
Cook Spaghetti 6:38
Boiling Water
Small Pot
Large Stove Burner
Add
Cook Spaghetti 6:44
Boiling Water
Small Pot
Large Stove Burner
8 min for Al dente
Start Timer
8min Timer
Cook Spaghetti 14:48
Boiling Water
Small Pot
Large Stove Burner
Timer Finished
Timer has finished
Cook Spaghetti 14:58
Boiling Water
Small Pot
Large Stove Burner
Cooked Pasta
Pasta is Cooked
Cook Spaghetti 15:15
Boiling Water
Small Pot
Large Stove Burner
Cooked Pasta
Stove off
Cook Spaghetti 15:30
Boiling Water
Small Pot
Cooked Pasta
Drain
Cook Spaghetti 15:53
Small Pot
Cooked Pasta
Cook Spaghetti 16:08
Small Pot
Cooked Pasta
Serve
Cook Spaghetti 16:38
Cooked Pasta

Appendix F

AOSC Presentation — Demo 2

Scenario
A simpilot is controlling two aircraft. He/she receives a comman
over the radio from a trainee in the control agency (wedgetail):
"aircraft 2, make your way to the fuel tanker, contact me when
you are 20 nautical miles out" (this is paraphrased).

From the perspective of the trainee, in a Large Force Engageme
(LFE) exercise this is a 'clear' task, as those trainee already
have proficiency. However for new students undergoing intial
employment training, especially for the first time, this task
may be 'complicated'.

From the perspective of the simpilot, he/she now has a goal to
"fly to tanker and respond on radio". Again, for a real pilot,
or experienced simpilot this is 'clear'. But for anyone else,
I feel that it gets 'complicated', fast!

Tasks sequence in order to satisfy the above goal:

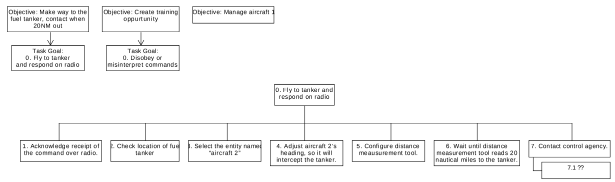
- Acknowledge receipt of the command over radio.
- Check location of fuel tanker (Note the tanker will be orbiting in a circle.)
- Select the entity named "aircraft 2"
- Adjust aircraft 2's heading, so it will intercept the tanker.
- Configure distance meausurement tool.
- Wait until distance measurement tool reads 20 nautical miles to the tanker.
- Contact control agency.

Discussion:
Not all of the tasks above are completed instantaeously (we mig
call that synchronous).
The 'contact control agency' task invokes a whole new task heirachy for operating the radio. (the channel may be jamme the channel may already be in use, etc.). This is where HTA comes in handy.

Tasks may produce errors or execeptions => complexity. Mayb
even chaos when the simpilot has not experienced the error bef

Simpilot may be instructed from white force to deliberately
disobey or misinterpret the commands. View through cynefin
lens, this is to intended create complexity/(chaos?) for the
trainee.

There are two aircraft in the worked example. We haven't
considered what the simpilot is doing with aircraft 1 :-)



1. Acknowledge receipt of command over radio

Q 1. Performs what action? (verb)
A 1. Talking/Responding

Q 2. To whom or what? (object of the verb)
A 2. Radio

Q 3. To produce what? or Why is it necessary? (expected out
A 3. Confirmation and acknowledgement of recieved comma

Q 4. Using what tools, equipment, work aids, processes?
A 4. Radio, command lexicon, voice

Q 5. (temporality)
A 5. In response to a command directed to self

1. Acknowledge over radio

1. Action: Talking/Responding
2. Object: Radio
3. Outcome: Confirm command
4. Tooling: Radio, Voice, Command lexicon
5. Time: In response to comma

2. Check location of fuel tanker

Q 1. Performs what action? (verb)
A 1. Checks, Measures, Locates

Q 2. To whom or what? (object of the verb)
A 2. Fuel tanker

Q 3. To produce what? or Why is it necessary? (expected out
A 3. Identification of correct fuel tanker

Q 4. Using what tools, equipment, work aids, processes?
A 4. HUD, map, named entites, mouse

Q 5. (temporality)
A 5. In response to a command / when fuel is low?

2. Check location of fuel tanker

1. Action: Check/Locates
2. Object: Fuel tanker
3. Outcome: Fuel tanker locate
4. Tooling: HUD, Map
5. Time: Response to command

3.Select entity named aircraft 2

Q 1. Performs what action? (verb)
A 1. Selects, identifies, targets

Q 2. To whom or what? (object of the verb)
A 2. Entity Aircraft 2

Q 3. To produce what? or Why is it necessary? (expected out
A 3. Aircraft 2 is selected

Q 4. Using what tools, equipment, work aids, processes?
A 4. Mouse, Display, map, list of entities

Q 5. (temporality)
A 5. When controlling entity, select it first

3. Select entity named aircraft 2

1. Action: Selects/Targets
2. Object: Aircraft 2
3. Outcome: Aircraft 2 selected
4. Tooling: Mouse
5. Time: When need to control self

4. Adjust Aircraft 2's heading

Q 1. Performs what action? (verb)
A 1. Adjusts/Changes heading

Q 2. To whom or what? (object of the verb)
A 2. Aircraft 2

Q 3. To produce what? or Why is it necessary? (expected out
A 3. Heading adjustment

Q 4. Using what tools, equipment, work aids, processes?
A 4. Mouse, Entity, ??

Q 5. (temporality)
A 5. After unit has been selected

4. Adjust Aircraft 2's heading

1. Action: Adjust/Change
2. Object: Aircraft 2 heading
3. Outcome: Heading changed
4. Tooling: Mouse/Entity
5. Time: After unit selected

5. Configure distance measurement tool

Q 1. Performs what action? (verb)
A 1. Clicking

Q 2. To whom or what? (object of the verb)
A 2. Screen, two points (self and tanker)

Q 3. To produce what? or Why is it necessary? (expected out
A 3. Line between two clicks

Q 4. Using what tools, equipment, work aids, processes?
A 4. Screen, Mouse

Q 5. (temporality)
A 5. After heading change

5. Config tool

1. Action: Clicking
2. Object: Screen, two points
3. Outcome: Line as Distance
4. Tooling: Mouse
5. Time: After changed heading

6. Wait until tool reads 20NM to tanker

Q 1. Performs what action? (verb)
A 1. Observation

Q 2. To whom or what? (object of the verb)
A 2. Distance measurement tool

Q 3. To produce what? or Why is it necessary? (expected out
A 3. Correct distance identified

Q 4. Using what tools, equipment, work aids, processes?
A 4. Screen, HUD

Q 5. (temporality)
A 5. Once a units heading has been adjusted

6. Wait till 20NM

1. Action: Observe
2. Object: Distance tool
3. Outcome: Correct Distance
4. Tooling: HUD/Screen
5. Time: After heading adjustm

7. Contact control Agency

Q 1. Performs what action? (verb)
A 1.

Q 2. To whom or what? (object of the verb)
A 2.

Q 3. To produce what? or Why is it necessary? (expected out
A 3.

Q 4. Using what tools, equipment, work aids, processes?
A 4.

Q 5. (temporality)
A 5.