TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technology
Department of Software Science

IT70LT

Uku-Rasmus Lind 121874IAPM

AGENT-BASED MODELLING AND SIMULATION OF PERSONALITIES BASED ON THE FIVE FACTOR PERSONALITY MODEL

Master's thesis

Supervisor: Professor Kuldar Taveter
Consultant: Nico Zimmer,
Jeppesen GmbH

Tallinn 2017

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

IT70LT

Uku-Rasmus Lind 121874IAPM

AGENDIPÕHINE ISIKSUSTE MODELLEERIMINE JA SIMULATSIOON KASUTADES VIIE FAKTORI ISIKSUSEMUDELIT

magistritöö

Juhendaja: Professor Kuldar Taveter

Konsultant: Nico Zimmer,
Jeppesen GmbH

Tallinn 2017

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Uku-Rasmus Lind

10.05.2017

Abstract

This thesis has been written as part of a university collaboration project between Jeppesen GmbH and Tallinn University of Technology. Jeppesen's interest in this work was to find new and innovative ways how to improve the performance of airline's operations control. Similar work has been done previously by Balašova [1] and her work was also taken as a starting point for this work.

This work is focused on modelling control organization of integrated operations of a generic European airline of the hub and spoke type with detailed attention given on the problem domain of disruption handling.

The problem domain analysis and modelling was performed using the method of Agent Oriented Modelling (AOM) [8].

Four different agent-based design architectures were compared to find the most suitable one for this work. After the architecture had been chosen and the problem domain had been modelled, an agent-based simulation system corresponding to the models of the problem domain was designed that considered the Five Factor Personality Model (FFM). The simulation system was designed by means of the renewed ODD protocol [10][11].

The resulting simulations provided means to evaluate the impact of the trait of Agreeableness on teamwork performance. Finally, a quantitative assessment was performed based on the simulation runs.

This thesis is written in English and is 62 pages long, including 6 chapters, 21 figures and 10 tables.

Sisukokkuvõte

Agendipõhine isiksuste modelleerimine ja simulatsioon kasutades viie faktori isiksusemudelit

Käesolev lõputöö on valminud osana Tallinna Tehnikaülikooli ja Jeppesen GmbH vahelisest koostööprojektist. Jeppesen on Boeingu tütarfirma, mis tegeleb navigatsiooni, logistika ja monitooringu toodete arendusega erinevatele lennundusalastele- ja muudele transpordiettevõtetele. Jeppeseni huvi käesoleva töö puhul oli leida uusi ja innovatiivseid suundi lennundusetevõtete juhtimiskeskuste töö optimeerimiseks ja parendamiseks. Sarnasel teemal on varem lõputöö kirjutanud Balašova [1], kelle töö oligi lähtepunktiks käesolevale tööle.

Käesolev töö keskendud Euroopa geneerilise *hub and spoke* tüüpi lennundusettevõtte juhtimiskeskusele. Kõrgendatud tähelepanu all on lennuplaani häirete haldamise probleemvaldkond.

Probleemvaldkonna analüüsi ja modelleerimise metoodikaks on agentorienteeritud modelleerimise (AOM) metoodika [8].

Töö käigus uuriti ja võrreldi nelja erinevat agendipõhist arhitektuuri ning valiti nende seast antud töö jaoks sobivaim. Seejärel kavandati lähtuvalt valitud arhitektuurist, probleemvaldkonna mudelitest ning viie faktori (FFM) isiksusemudelist agendipõhine simulatsioonisüsteem. Simulatsioonisüsteem kavandati uuendatud ODD protokolli [10][11] metoodika kohaselt ja teostati agendiplatvormil NetLogo.

Simulatsioonide tulemuste põhjal anti kvantitatiivne hinnang isikuomadusele *koostöövalmidus.*

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 62 leheküljel, 6 peatükki, 21 joonist, 10 tabelit.

List of abbreviations

ACMI	Aircraft, Crew, Maintenance and Insurance
AOC	Airline Operations Control Center
AOM	Agent-Oriented Modelling
BDI	Belief-Desire-Intention
BOID	Belief-Obligation-Intention-Desire
FFM	Five Factor Personality Model
MBTI	Myers-Briggs Type Indicator
OCC	Operations Control Center
FFM	Openness, Conscientiousness, Extraversion, Agreeableness and Neuroticism
ODD	Overview, Design concepts, Details
SRA	Simple Reactive Agents
QDT	Qualitative Decision Theory

Table of Contents

List of Figures

List of Tables

1 Introduction

In Chapter 1 will be given a brief introduction to the previous work in the field and an understanding of how this thesis defines the research problem by formalising the main research questions. In sub-chapter 1.2 will be described the chosen methodology, in sub-chapter 1.3 the main goals of this work are presented, and finally in sub-chapter 1.4 an overview of the structure of the thesis is presented

1.1 Background and problem definition

In the recent years, the agent-research community has found more interest in the incorporation of human personality traits into agent reasoning algorithms. The Five-Factor Personality Model (FFM for short) is widely accepted in psychology researchers and has, therefore, found its way to the agent-research too. FFM as its name suggests consists of five orthogonal personality traits which describe the human personality in whole.

Previous work in the field has created conceptual models of the problem domain of airlines and has studied in general what kind of impact the FFM personality trait of agreeableness can have on decision-making in airlines [1]. Some previous research has also been reported on agent-based simulation of FFM personality traits [2]. One rather noteworthy implementation of the personality traits is crowd simulation proposed by Durupinar et al [3]. However, further modelling and simulation work is required for studying the impact of FFM personality traits and particularly the impact of agreeableness and its subordinate trait of helpfulness on decision-making in airlines.

This thesis is written as a part of a university collaboration project between Jeppesen GmbH and Tallinn University of Technology. Jeppesen is a company providing innovative products that integrate navigation, operations, and logistics information into one end-to-end solution [4].

Three main research questions have been formalised to study the problem area:

1. How to best represent the FFM in agent-based simulations?

2. How to adequately reflect an organisation by an agent-based simulation system where humans with different personalities perform various roles?

3. How big is the impact of the personality trait of agreeableness and its sub-trait of helpfulness on the performance of individual employees and teams of an airline?

1.2 Methodology

The chosen research methodology for the M.Sc. work is a combination of design science [5] and agent-based modelling [6][7], which is the newest and increasingly popular method for studying complex socio-technical systems. For modelling the problem domain the methodology of Agent-Oriented Modelling (AOM) proposed by Sterling and Taveter in [8] will be used.

To answer the research questions, it is necessary to complement the modelling aspect with designing and implementing an agent-based simulation as described in [9]. The simulation will be described according to the refined ODD protocol [10][11].

1.3 Main goals of the thesis

Three main research questions were formalised in Chapter 1.1. To answer those questions the thesis aims to achieve the following goals:

1. Obtain an overview of different agent-based representations of the FFM and choose one as the foundation for achieving the remaining goals 2-4.

2. Model the problem domain of airlines resulting from the previous work [1] by means of the Agent-Oriented Modelling (AOM) methodology [8].

3. Based on the problem domain analysis elaborated under goal 2 and using a combination of the AOM [8][12] and ODD [10][11] methodologies, design an

agent-based simulation system of an organisation with roles performed by agents having different personality traits.

4. Implement on the agent-based simulation environment NetLogo [13] a prototype of an agent-based simulation system of personality traits that was designed under goal 3 and analyse the impact of agreeableness and helpfulness on problem-solving activities from a teamwork perspective.

1.4 Overview of the thesis

The second chapter gives a brief overview of previous attempts to model personality traits by agent-based simulations. Thereafter, four different agent-architectures are presented and compared, and finally, the chapter concludes with the choice of architecture to be used for the simulation system design and implementation.

Chapter three gives the reader an overview of the industry partner's problem domain by presenting the AOM diagrams that were modelled based on the state of the art literature and industry partner's expertise.

Chapter four presents the design of the proof-of-concept simulation system derived from previous work and the requirements coming from problem domain specific needs. First, the connection between the simulation system and problem domain is explained, then the incorporation of the FFM is explained, and finally, a comprehensive simulation design description is presented according to the ODD protocol.

Chapter five shows the results of the simulation runs done on the implemented system, evaluates different quality aspects of the simulation system and finally, presents the findings from the analysis of the simulation results.

2 Overview of previous work and potential alternatives

In this chapter, we will give an overview of previous work done in the field of agent-based simulation incorporating the Belief-Desire-Intention(BDI) agent architecture and the Big Five Personality Traits model also known as OCEAN personality traits model or Five Factor Personality traits Model(FFM). We will also analyse the suitability of BDI as the base architecture for such simulation systems and highlight some possible alternatives to BDI.

2.1 Existing solutions and previous work based on BDI

The BDI architecture for intelligent agents was formalised in the beginning of 1990's by Anand S. Rao and Michael P. Georgeff. According to their foundation-laying work [14] the decision-making in BDI depends equally on three agent-specific concepts:

1) Beliefs – representing the knowledge state of the agent, such as signals from other agents, changes in environment and so on;

2) Desires – representing the motivational state of the agent or simply put what the agent would like to accomplish and

3) Intentions – representing the deliberative state of the agent. Intentions are a subset of desires to which the agent has chosen to commit to.

BDI agents have a repetitive life-cycle which consists of four phases [15], namely *the Belief Revision, the Option Generation, the Filter Process,* and *the Actuation* or *Execution* phase.

In *Belief Revision*, the agent reviews all new environment perceptions and messages from other agents to update its beliefs. Once beliefs are revisited, the agent generates options based on its beliefs, desires and previous intentions. Options are potential intentions that the agent can commit to. In the filtering phase, the agent considers all generated options and selects the ones most suitable with current beliefs and agent's desires as its intentions. In the actuation phase, all selected intentions are translated into actions and are executed. The four phases incorporation into the BDI execution loop is best described by pseudocode presented in Figure 1:

```
while true
  set percepts = percept(Env, Msg)
  set bels = beliefRevision(bels, percepts)
  set opts = options(bels, intents, desires)
  set intents = filter(opts, bels, desires, intents)
  foreach intent in intents:
    actuate(intent)
end
```

Figure 1. BDI execution loop in pseudocode

BDI as an agent architecture has been widely accepted by the research community and, hence, is found also in several implementations trying to incorporate personality traits into agents' decision-making algorithms. For instance, BDI agents have been used in the AntMe! simulation by Ahrndt et. al described in [2] and in the emergency evacuation simulation framework by Zoumpoulaki et. al [16].

2.2 Strengths and shortcomings of BDI

As every other architectural framework, also BDI comes with its strengths and weaknesses. Let us first examine some of the more important positive aspects of BDI.

Since BDI derives from Bratman's theory of human practical reasoning [17], it provides the designer with a very human-like decision-making framework making it easy to both model and implement agent based simulations.

BDI provides the designer with a good separation between the activity of choosing a plan to execute and the activity of executing the currently chosen plans. This in turn makes BDI highly scalable and usable in complex multi-agent systems as shown in [14].

Although widely used, BDI is often criticised for its limitations in learning, forward planning and agent interactions. Since the simulations presented in this paper require no forward planning, we will only have a quick look at learning or adaptability to environmental changes and agent interactions.

Phung et. al showed in [18] that learning proves crucial in situations where environment can change and render the pre-defined behaviour to no longer be optimal. Unfortunately for BDI, to incorporate the learning aspect, rather drastic additions to the architecture are needed.

BDI defines no formalism for agent interactions but, unlike learning, it is achievable with reasonable extension of the BDI framework. As described by Kinny et. al in [19], at the time when BDI was formalised there was no clear view on how agents should interact, thus, all efforts were made to make implementing any kind of communication protocol in BDI as simple as possible.

2.3 Analysis of alternative architectures

Although BDI is clearly among the most popular architectural frameworks used in the agent research community, we will also consider some other alternatives. For the purposes of this thesis, we will do a comparison of the alternatives using BDI as a base benchmark.

2.3.1 Simple Reactive Agents

Reactive agents are simple entities that involve relatively simple algorithms for decision making. They make decisions based on current situation with no regard to past experiences. One good example of a reactive agent is a car thermostat: it opens or closes depending on whether the coolant temperature has reached the thermostat's threshold.

Simple reactive agents can prove useful in solving different map exploration and object collecting tasks as presented by Scheutz and Schermerhorn in [20]. With simple knowledge of where all other agents currently reside on the map, the simple reactive agent population proved to be almost as efficient as the more advanced deliberate population.

Our own experimentations with the architecture of a simple reactive agent confirmed the findings of Scheutz et. al [20], but also revealed that compared to BDI the simple architecture of reactive agents comes with significantly lower overhead in designing and programming agent-based simulations.

On the other hand, we also found that the complexity of the simple reactive agent architecture keeps growing with the growth of the simulation complexity, thus, demonstrating that this kind of simple architecture does not scale up well for more extensive and complex simulation environments.

It is hard to provide a precise evaluation on what is reasonable and what is not to be implemented using only simple reactive agents, but the initial results of our original tests of trying this approach on NetLogo showed that having many instances of up to 2 different types of agents, this architecture still performed well in both resource usage and code efficiency. Significant increase was evident in the number of code lines necessary to implement more than 3 different kinds of agents, but no significant performance issues occurred.

In the tests were implemented the following scenarios in both the BDI architecture and Simple Reactive Agents approach:

1) Ant colony with all ants belonging to the same agent type and having the same tasks ("explore" and "gather food");
2) The same ant colony as in the first scenario (added tasks "fight bug" and "run away") and bug colony with its own agent type and execution cycle;
3) The same bug colony as in the second scenario, but the ant colony consisting of agents of two types with different tasks (agents of one type being "explorers" and agents of the other type being "food gatherers", sharing fighting and running);
4) The same ant colony as in the third scenario but two different bug colonies (the first fighting ants on way, the others around food sources);

5) The same bug colonies as in the fourth scenario but the ant colony consisting of three different types of ants (“explorers”, “food gatherers” and “fighters”).

For those five scenarios following the characteristics were measured:

1) average running time to complete 5000 simulation ticks;

2) amount of code in the implementation (number of lines);

3) average CPU consumption;

4) average memory consumption.

The running time was measured in milliseconds, but for the sake of readability time values are presented as rounded values to 0.1 second precision. The results of the tests are presented in Table 1.

Table 1. BDI vs Simple Reactive Agents test results.

Scenario	1		2		3		4		5	
Architecture	BDI	SRA	BDI	SRA	BDI	SRA	BDI	SRA	BDI	SRA
Runtime(s)	19.3	18.6	35.4	34.9	52.6	51.7	69.5	69.0	86.1	85.8
Lines of Code	557	382	635	486	692	591	759	687	812	794
Avg. RAM(MB)	487.3	481.2	488.8	482.5	490.1	483.9	491.4	485.6	493.2	487.1

2.3.2 BOID – a derivative of BDI

Rao and Georgeff explained in [14] that there are tendencies to question whether the three concepts of BDI are perhaps too many or too few; whether intentions are covered by beliefs and desires or if there should be something more emphasizing the social aspects of agents. One attempt to bring the social aspect into the BDI architecture is the Belief Obligation Intention Desire architecture [21] or, BOID for short.

As the name suggests, BOID expands on BDI and adds the concept of social obligation to the decision-making algorithms. The addition of a social concept makes it possible to define four types of agents with different behaviours: *the realistic agent, the simple-minded agent, the selfish agent and the social agent*. From the perspective of this thesis, this is of low significance as the types explained refer to specific personality profiles, which are much better defined by the Five Factor Personality model. Thus, combining BOID with the FFM would most likely result in either losing clear separation between the reasoning algorithms and personality traits' model or colliding social aspects in the two. For example, let us consider that we have set up a BOID selfish agent. If we now complement this agent with FFM and from the personality traits model perspective make it agreeable and open, then the agent no longer can be selfish. Similarly, the same agent could be made selfish by both the BOID architecture and FFM traits. Also noteworthy is the fact that BOID has extra steps in its reasoning algorithm which, in turn, according to [21] at least on paper, should make it slower than BDI.

2.3.3 Qualitative Decision Theory

Qualitative Decision Theory (QDT) [22] is a qualitative extension to the classical decision theory. It criticizes the classical decision theory and proposes a non-quantitative reasoning logic that could be applied in uncertain conditions.

We have not repeated the work of Dastani et. al [23] to compare QDT and BDI. Their results were more than conclusive for our needs – QDT is as comprehensive in logical reasoning as BDI, but lacks the layer of abstraction to make the decision-making algorithms at least seem more intuitive and understandable for humans. Since the decision-making algorithms are very mechanical, it is also significantly harder to incorporate any personality traits' models. Additionally, QDT does not have anything like intentions to make the agent behaviour stable over time. For the purposes of this thesis, QDT does not present any added value, but complicates the implementation of uniform and driven by personality traits reasoning architecture for simulation agents.

2.3.4 Comparison summary

This final sub-chapter will present the results of comparing the alternatives as an easy to review table which uses BDI as baseline. The table is based on the BDI to QDT comparison table presented by Dastani et. al in [23]. Here, the table is complemented with simple reactive agents and BOID. Also, three new evaluation categories are proposed:

1) Effort for FFM – effort needed to incorporate the personality traits into an agent's execution cycle;

2) Code efficiency – the amount of code needed for similar implementations, less being better and more being worse and

3) Value add – what kind of added value the architecture brings compared to BDI, or what it lacks.

The comparison table is shown as Table 2.

Table 2. Architectures comparison.

	BDI	SRA	BOID	QDT
Area	Software engineering	Simulation environments	Software engineering	Artificial intelligence
Focus	Application-oriented	Application-oriented	Application-oriented	Theory-oriented
Criticism	Resource bounded	Reactive to perceptions	Resource bounded	No quantitative evaluation
Intentions	yes	no	yes	no
Rules based	no	yes	no	yes
Knowledge	yes	no	yes	yes
Desires	yes	no	yes	yes
Norms	Possible extension	no	yes	yes
Effort for FFM	baseline	Easier for small scale / harder for big scale	Equal or potentially worse	Not evaluated
Code efficiency	baseline	Better for small scale / worse for big scale	Not evaluated	Not evaluated
Value add	baseline	Less, no human-like reasoning incorporated	Social norms and personalities by means of obligations	Not evaluated

2.4 Architecture choice

This chapter provided a quick overview of the most popular agent architecture used in agent-based simulations – the Belief Desire Intention architecture framework – and some of the more important previous attempts to combine it with the personality traits of FFM.

Then the focus shifted towards briefly considering possible alternatives that are also used in agent-based simulations, but have little to no literature on attempts to combine them with FFM. The brief analysis revealed that none of the considered alternatives outperformed BDI in the categories most important for this work. Perhaps the most noteworthy is the finding that Simple Reactive Agents could be a feasible option when working with a small number of unique agent types. Regardless of that, BDI is still a very clear first choice for combining a reasoning architecture with the models of personality traits.

3 Problem domain analysis – Handling disruptions in airline operations control centres

On a high level this work is a continuation of the work by Balašova [1] but focusing on one sub-domain and trying to elaborate it in a more detailed manner. This chapter presents the changes and additions that were necessary to be introduced into the original models [1] due to both new findings from the literature and new insight and contributions by the industry partner. The work by Balašova [1] was in many ways found exceptionally good as a starting point and much of it could be re-used. Regardless, this paper introduces further improvements which allow to better align with both the industry partner Jeppesen's descriptions of the problem domain and previous work describing the AOC workflows [24] and disruption handling scenarios [25].

The viewpoint framework originating in the AOM methodology [8] was used for modelling the problem domain. Table 3 illustrates the different horizontal abstraction layers and vertical perspectives of the viewpoint framework.

Table 3. Viewpoint framework.

Viewpoint models	Viewpoint aspect		
Abstraction layer	Interaction	Information	Behaviour
Conceptual domain modelling	Role models and organization model	Domain model	Goal models
Platform-independent computational design	Agent, acquaintance and interaction models	Knowledge model	Behaviour models
Platform specific design and implementation	Agent interaction specifications	UML class diagrams	Agent messaging diagram

In this thesis are presented all the models of the conceptual domain modelling layer. From the layer of platform-independent computational design, only agent and interaction models are presented. No models of the platform-specific design and implementation layer are presented. This is because the aim is to only analyse the problem domain and not to go any further.

3.1 Conceptual domain modelling layer

3.1.1 Role models and organization model

The role models proposed by Balašova [1] were considered mostly satisfying for describing the disruption handling sub-domain. One new role was added and another one was expanded after discussions with industry partner Jeppesen:

1) new role: Duty Manager – as a central governing role of the Operations Control Centre, it is the main airline stakeholder in the Airline Operations Control Centre;

2) expanded role: Operations Controller – Previously the role of Operations Controller had the responsibility of monitoring flights and stepping into action in case of disruptions. Now the same role is is complemented also with the responsibility of fleet management.

The new responsibility of fleet management by the modelled Operations Controller role incorporates the scheduling of airline's own aircrafts and making sure that there is an aircraft available on time for every flight. This is even more important when regular operations are disrupted and schedule breaking changes are needed. Sometimes it is possible that the disruption at hand cannot be solved by using airline's own fleet. For instance, if an aircraft needs to be replaced but there are no other aircrafts of the same airline available in the airport affected by the disruption, the Operations Controller would be the one responsible for finding an external aircraft and initiating ACMI[1] wet-lease negotiations. One could argue that fleet management should be modelled as an entirely new role, but for the sake of not overly complicating the models in this work, it was decided to model the roles corresponding to a generic mid-sized airline's integrated Operations Control Centre of the hub-and-spoke type.

Since the organization model proposed by Balašova [1] was not done by means of the AOM methodology, a new organization model was prepared. A sub-organization of the Operations Control Centre comprising the new and changed roles is presented in Figure 2. The roles in the Operations Control Centre have further relations with their

[1] ACMI wet-lease stands for Aircraft Crew Maintenance and Insurance leasing option where the whole package would come from an external supplier.

respective sub-organization counterparts, such as the connection between MX Controller and Line Foreman from the Maintenance team. All these organizational relations are depicted in other models of the Operations Control Centre organization that can be found in Appendix 1.

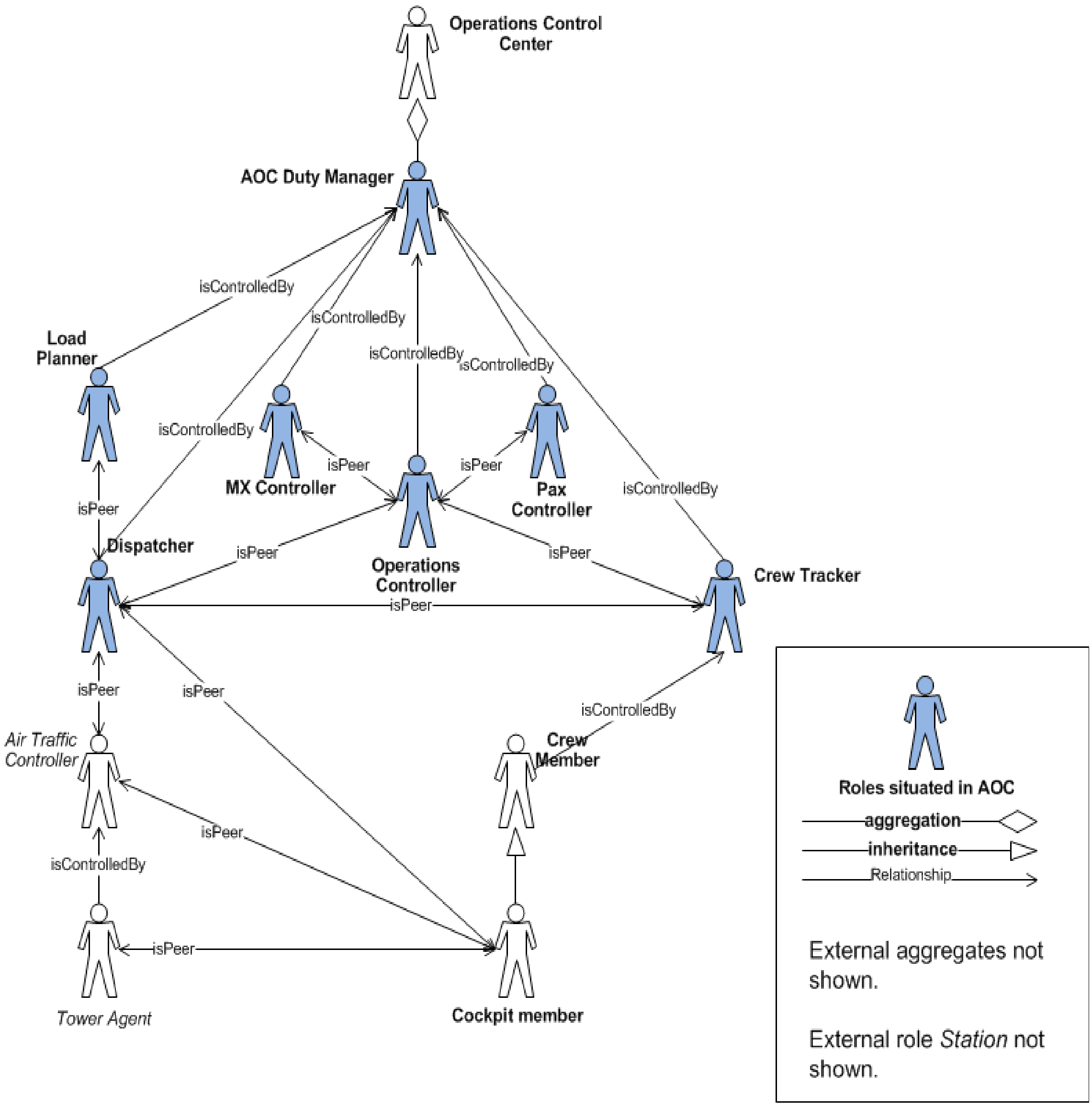

Figure 2. AOC Organization model.

3.1.2 Domain model

The domain model proposed by Balašova [1] did not include the modular knowledge units necessary to support disruption handling. Therefore, all the necessary roles and entities were added to the original model to make it compliant with the needs of the sub-domain. Additionally, some discouraged and obsolete modelling techniques were replaced with more standardized notations. The resulting final domain model depicting domain entities, their connections to different roles and relations with each other is presented in Figure 3.

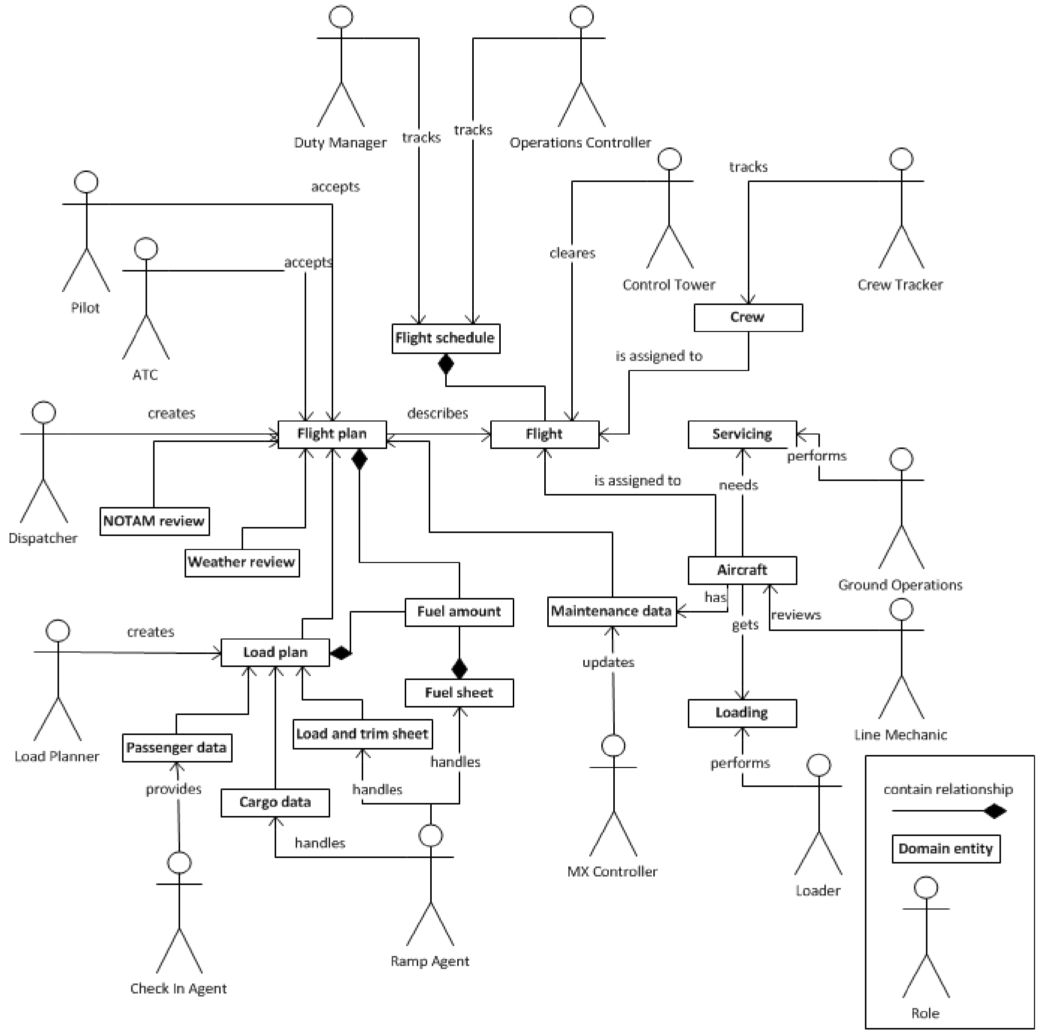

Figure 3. AOC Domain model.

3.1.3 Goal models

Since the motivational models of an airline by Balašova [1] had a different focus, it was decided to identify a good point in her models from where to branch out and go in-depth with describing the sub-domain of disruption handling. In goal models, such a point was found to be the *Maintain Operational Control* goal. From there downwards new goal models were created. Figure 4 depicts the new and improved version of the *Maintain Operational Control* goal model that is more accurate in terms of both disruption handling and sharing responsibilities within the AOC. The goal *Maintain Operational Control* is on a high level the responsibility of *Duty Manager*, but the responsibility for its sub-goals is delegated to either the *Operations Controller* or the whole AOC team in case of disruptions.

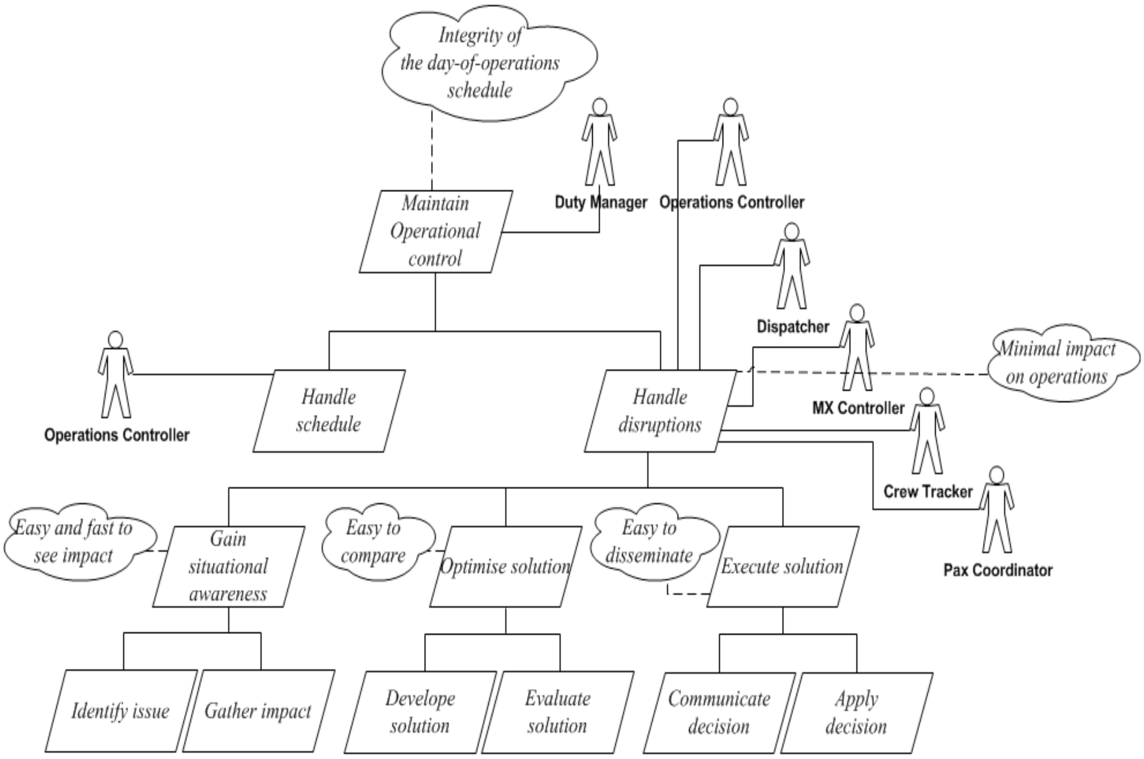

Figure 4. AOC Goal model - Maintain Operational Control.

In addition, it was identified that the *Maintain Operational Control* goal as a whole had to be brought up to a higher level. In Balašova's work [1], the same goal was presented as a sub-goal of *Prepare flight*, which, based on both the insight by the industry partner and literature, is not entirely correct due to the fact that AOC is responsible for the flight also en-route until the flight has been completed.

Therefore, we also present the modified airline goal model as Figure Error: Reference source not found depicting the AOC goal of *Maintain Operational Control* as a direct sub-goal to *Provide air transportation services*.

From these two goal models we can observe two quality goals – *Optimize cost* and *Minimal impact on operations* – which in the disruption scenarios presented later in the text have a tendency to conflict with eachother.

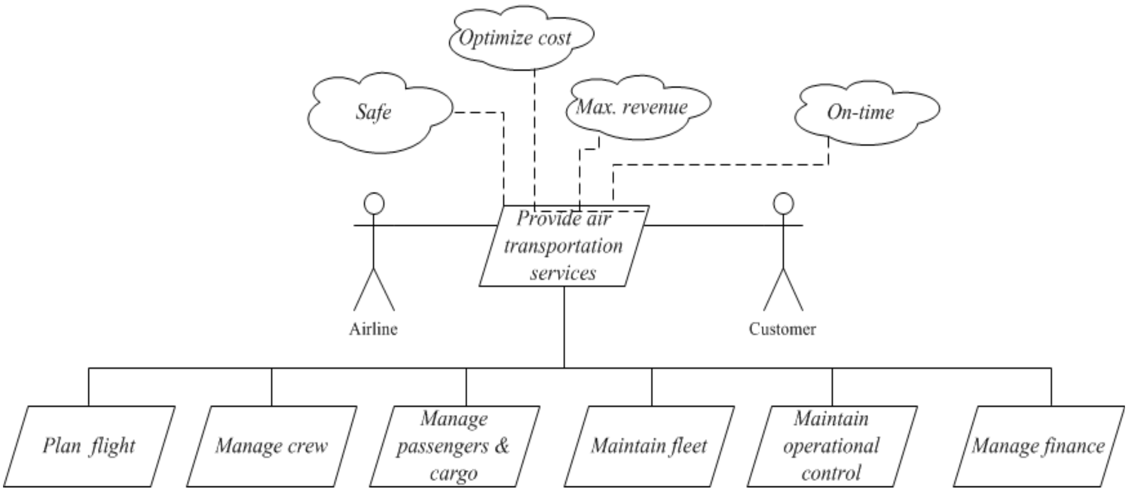

Figure 5. Airline Goal model – Provide air transportation services.

The rest of the modified or newly created goal models are presented in Appendix 2.

3.2 Platform-independent computational design

The abstraction layer of platform-independent computational design takes the models of the conceptual domain modelling layer and proposes based on them a potential design for the planned socio-technical system, which in the given case is a simulation system. This sub-chapter is not crucial in terms of problem domain modelling for the purposes of the current thesis but provides a starting point for the future work planned by the industry partner Jeppesen. In addition, the interaction diagrams give the reader a better understanding of the complexity of disruption handling scenarios.

Sterling and Taveter have proposed in their book [8] a merged agent and acquaintance model to depict different agent types and the interaction pathways between them. The agent types depicted in Figure 6 were defined based on the roles defined at the stage of conceptual domain modelling and how the industry partner wanted to group them into specific agent types that are to be implemented in their future simulations.

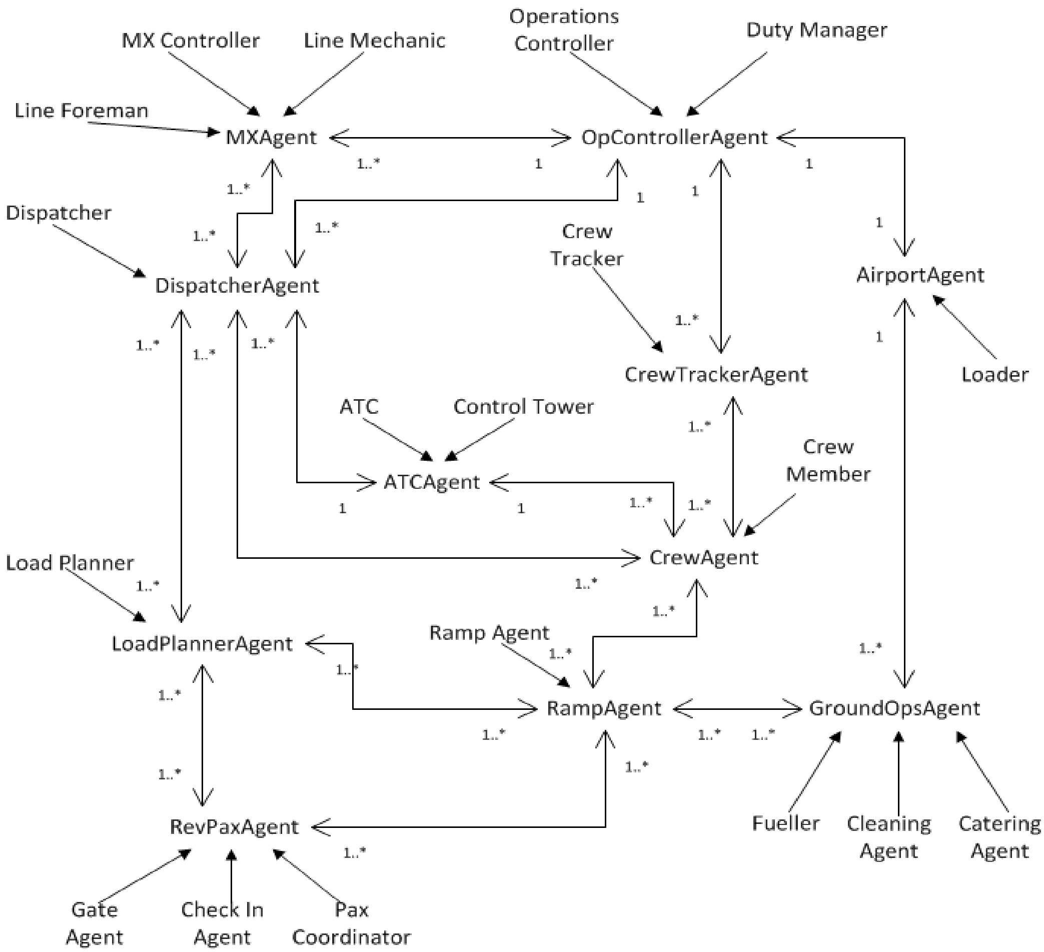

Figure 6. AOC Agent acquaintance model.

In addition, the interaction diagrams were drawn for two different disruption scenarios to further improve the conceptual understanding of the problem domain and prepare for implementing future simulations by the industry partner. Since the main purpose of the interaction diagrams drawn was to contribute to the conceptual domain modelling layer, the lifelines represent roles rather than agents. Interactions for an en-route emergency disruption are depicted in Figure 7. In this scenario, first the need for immediate landing needs to be identified. Once the aircraft has been landed and emergency been taken care of, crew legality needs to be confirmed or a replacement crew assigned if needed before a new flight plan can be filed and flight can be continued.

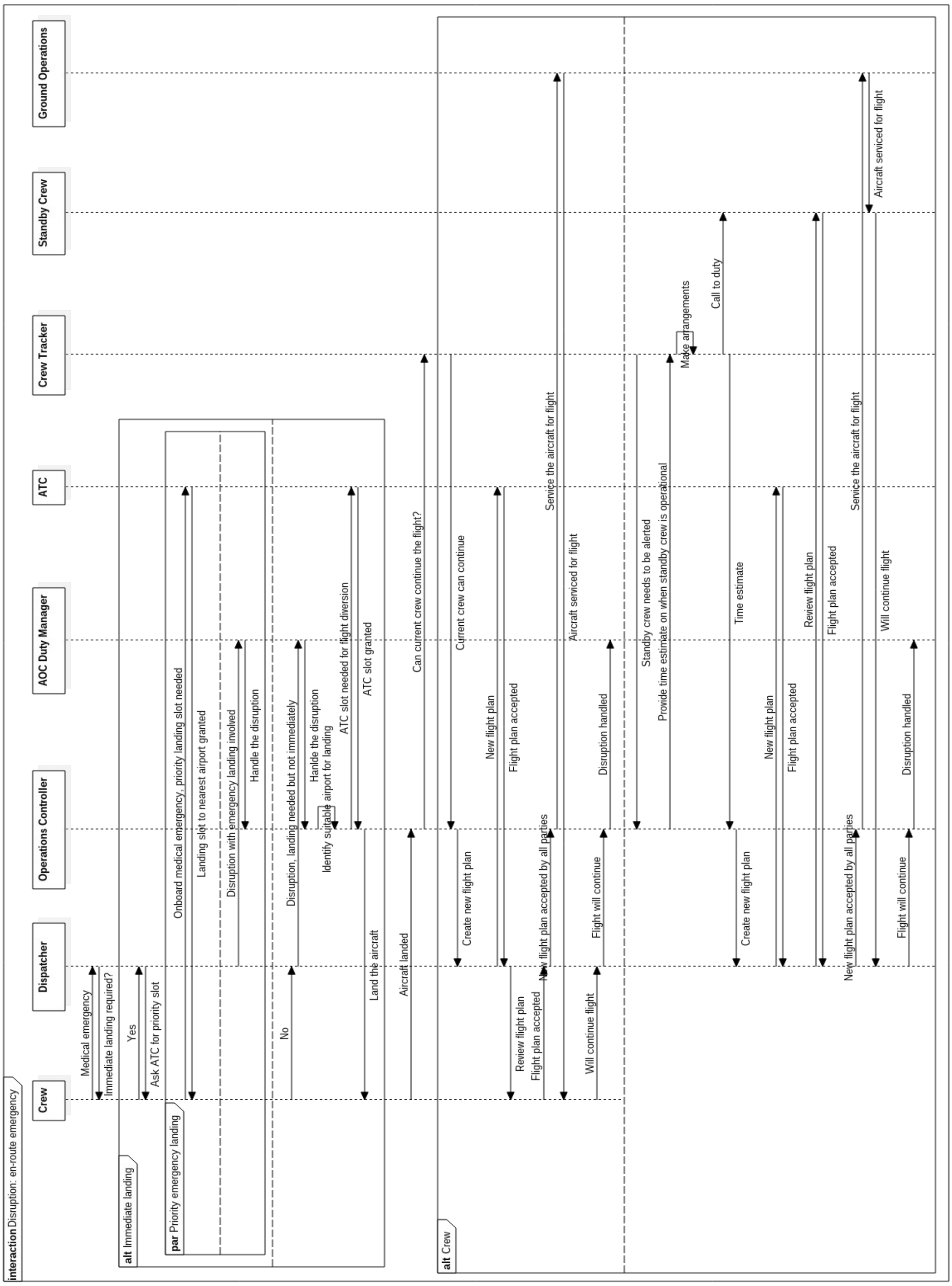

Figure 7. Interaction diagram - En-route medical emergency disruption.

The disruption handling scenario depicted in Figure 7 is not entirely complete. Handling such diversion situations in many cases means that even if the same crew can continue, the flight would still be delayed and the next leg scheduled for that aircraft would face a disruption of the “aircraft not ready” type. A solution for that disruption could in turn cause yet another disruption elsewhere. The disruption handling in real life, therefore, weighs also the impact of the potential solution on the entire schedule and in many cases, the only option is to cancel the disrupted flight and/or consider ACMI wet-lease for some other flight so that the day's schedule would remain as intact as possible. Another, more comprehensive interaction diagram was created that accommodates more of the different solution options and describes more accurately real-life disruption handling. However, because of being more comprehensive, it is also in a format not suitable to be presented in between text. Therefore, the reader can find this more comprehensive interaction diagram modelling disruption handling in Appendix 3.

3.3 Analysis summary and conclusions

The work by Balašova [1] was thoroughly analysed together with the industry partner and found to be not entirely correct but also not entirely incorrect. Therefore, it was decided to choose one sub-domain – the disruption handling – and re-model it by incorporating both modern literature on AOC workflows and the industry partner's renewed expertise in the field.

The problem domain analysis conducted in Chapter 3 of this thesis results in a more comprehensive and accurate conceptual representation of the disruption handling sub-domain. The conceptual understanding is complemented by the AOC agent acquaintance model and interaction diagrams, which further illustrate the complexity of AOC workflows.

To conclude Chapter 3, the problem domain is rather diverse and to be able to provide qualitative assessments on AOC workflows, factors such as the complexity of the workflows, personalities of the people involved, and accurate teamwork descriptions need to be considered in a potential simulation environment. Although many of the previous agent-based simulations address either the aspect of personalities, complexity of workflows, or specialization in teamwork, none of them addresses two or more of these aspects together. Therefore, this thesis aims to propose a proof-of-concept simulation system that could be used to evaluate the impact of personality traits in a teamwork situation, where handling disruptions and coping with conflicting goals is of high significance. Such simulation is described in the next chapter – Chapter 4.

4 Implementation of the Simulation

This chapter describes the design and implementation of the proof-of-concept simulation created to study the impact of FFM trait Agreeableness in a team-work environment corresponding to the chosen problem domain of airline operations control. The simulation system is in many ways an elaboration of the work described in [2] with some additional teamwork aspects considered.

Sub-chapter 4.1 explains the connection between the problem domain described in Chapter 3 and the proof-of-concept simulation environment proposed in Chapter 4. A description of the exact incorporation of the personality traits into the BDI execution cycle algorithm is presented in sub-chapter 4.2. In sub-chapter 4.3 the final proof-of-concept simulation system is described according to the refined ODD protocol [10][11]. Finally, in sub-chapter 4.4 the quality goals of the simulation system are discussed.

4.1 Design considerations originating in the problem domain

From the results of the problem domain analysis conducted in Chapter 3, it became clear that in most disruption handling scenarios at least three specialized AOC members play the key roles in the handling of disruptions. The exact members can vary to some extent depending on the nature of the disruption. The simulation proposed in [2] clearly fails to satisfy the needs of the problem domain described in Chapter 3 due to the specialized teamwork nature of the problem domain. This work intends to propose a solution that would correspond better to the problem domain by setting a clear division between the tasks performed by different agents in the simulation and by providing means to define how different kinds of agents can specialize on specific tasks. Additionally, new tasks are proposed to better evaluate the impact of the trait of Agreeableness in collaborative teamwork. The following list of new features and tasks compared to [2] need to be incorporated:

1) New ant agent types to make the specialization possible: *Explorer ants*, *Gatherer ants* and *Fighter ants*;

2) New tasks related to fighting with bugs: *Call for help* and *Help with fight*.

3) Specialization parameter. A new parameter that determines how specialized different agents are to their type-specific tasks. The lowest value represents no specialization at all and high values represent higher levels of specialization.

With the above additions to the simulation proposed in [2], the correlation between the AOC problem domain modelled in Chapter 3 and the ants' simulation has been improved. The correlation is illustrated in Table 4.

Table 4. Correlation between problem domain and simulation.

AOC	Description	Simulation
1) Operations Controller, 2) Dispatcher and 3) Crew Tracker	Three specialized roles participating in achieving of both the main goal and its sub-goal	1) Explorer ant, 2) Food gatherer ant and 3) Fighter ant
Goal: Provide air transportaion services	Main goal of the team	Gather food
Goal: Handle disruptions	Sub-goal under the main goal	Fight off aggressive bugs
Quality goal: Minimal impact on operations	First quality goal attached to the sub-goal	Minimal amount of food gathered less
Quality goal: Optimize cost	A conflicting quality goal attached to the main goal	Minimal number of ants deceased

4.2 Requirements coming from the Five Factor Personality Model

Since 1980's there have been numerous attempts to incorporate personality traits into both artificial intelligence and agent based systems. For that purpose, researchers have cooperated with scientists studying human behaviour and personalities. The two most significant models adopted for that purpose by researchers in multi-agent systems and artificial intelligence are the Five Factor Personality model or FFM for short, and the Myers-Briggs Type Indicator [26] (MBTI) model. A comprehensive comparison between the two has already been accomplished with a conclusion that the FFM is more suitable for agent-based simulations [27]. Therefore, this thesis only focuses on the FFM.

The FFM consists of five high-level personality traits which according to [28] are described as follows:

1) Openness to experience. Openness reflects the person's intellectual curiosity, creativity and a preference for novelty and variety. A person with very low openness is considered dogmatic and close-minded;

2) Conscientiousness shows the person's tendency towards being organized, having high self-discipline, acting dutifully, aiming for achievement, and preference for planned behaviour. Low conscientiousness is associated with spontaneity, but can also reflect the lack of reliability;

3) Extraversion reflects how outgoing and assertive the person is when it comes to social activities. Low extraversion is a feature descibing more self-absorbed and closed persons;

4) Agreeableness shows the person's tendency to be cooperative and compassionate. Low agreeableness will often lead to a very analytic personality that likes to argue and often ends up in disagreement with others;

5) Neuroticism describes the tendency to experience unpleasant emotions. Persons with high neuroticism are more likely to experience anger, anxiety and depression, whereas low neuroticism provides more emotional stability and impulse control.

Since this work in many ways continues the work by Ahrndt et. al [2], but with more focus at reflecting team-work and introducing conflicting goals, the personality traits have been incorporated into the BDI execution cycle described in chapter 2 in the following manner similar to [2]:

1) Perception is not influenced by personality traits;

2) Belief revision is influenced by Openness and Agreeableness – the more open and agreeable the agent is, the more likely it is to allow for different perceptions to influence its beliefs;

3) Option generation is influenced by Conscientiousness, Agreeableness and Neuroticism. The more conscientious the agent is, the more likely it is to include its previous intentions as options. The more agreeable the agent is, the more likely it is to accept intentions from other agents. The more neurotic the agent is, the more likely it is to generate irrational options;

4) Filtering process is influenced by all five personality traits. The agent will decide between otherwise equal options depending on the personality profile;

5) Actuation or execution phase is influenced by Openness, Conscientiousness and Extraversion traits.

Figure 8 represents the abstract architecture of the designed BDI agent. The figure depicts the classical components of the BDI architecture, communications between them and the incorporation of personality traits into the components.

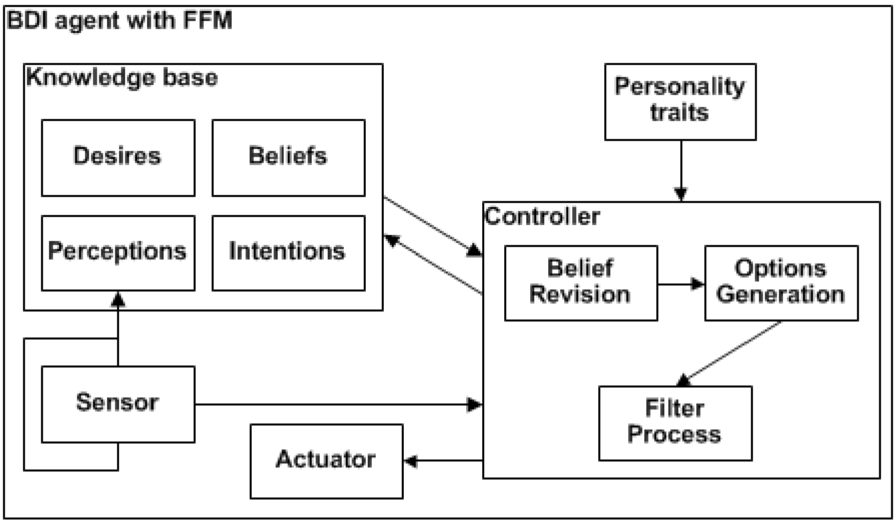

Figure 8. Abstract BDI agent architecture with personality traits incorporated.

4.3 Design overview

The detailed design of the implemented agent-based simulation system is presented using the ODD protocol. The different parts of the ODD protocol are presented as sub-chapters 4.3.1 to 4.3.7.

4.3.1 Purpose

The purpose of the implemented agent-based simulation of personality traits is to understand how personality traits influence the performance of a multi-agent team of specialized individuals with varying levels of task distribution. In the centre of the simulation map resides an ant nest with food sources and hostile bugs. There are three different types of ants specialized for different tasks. The main goal of the ants is to gather as much food as possible within a given time, at the same time keeping the mortality rate of the population as low as possible.

4.3.2 Entities, state variables and scales

The simulation model comprises of the following four types of entities: 1) ant nest; 2) food source; 3) ant agent; 4) bug agent. There is one instance of the ant nest, four instances of the food source, configurable number of instances of ants and configurable number of instances of bugs.

The simulation model has four hierarchical levels: *environment, map, ant population* and *individual*. At the *environment level* are defined the values for food source refill rate and age to die for all agents. The *map level* is characterised by the size and location of food sources, the ant nest, and the number of bug spawns. At the level of *ant population* are determined the population size, levels of specialization, locations for known food sources, amount of food gathered, values of personality traits for all three types of ant agents, and initial energy level for ant agents. At the level of *individual agents* are determined variables for agent state, current amount of energy, amount of food carried, damage inflicted to opponent in one attack action, and location of the agent. All the variables and their default values are presented in Table 5. The table includes an additional column to show which variables are configurable from the user interface of the simulation and which variables are hard-coded into the simulation.

Note that in Table 5 the default values of personality traits and the levels of specialization are presented as probabilities. Other default values presented in Table 5 are either scalar values or lists of scalar values.

Table 5. Overview of simulation variables and default values.

Variable	Default value	Configurable
	Environment level	
Food-refill-rate(ticks)	1000	yes
Age-to-die(ticks)	1000	yes
	Map level	
Map-size-x(pixels)	700	no
Map-size-y(pixels)	700	no
Food*-x	{-(0.8*Map-size-x); 0.8*Map-size-x}	no
Food*-y	{-(0.8*Map-size-y); 0.8*Map-size-y}	no
Nest-x	0	no
Nest-y	0	no
Number-bug-spawns	3	yes
	Ant population level	
Ant-population-size	99	yes
Level-specialization	0.0	yes
Known-food-sources	{}	no
Amount-food-gathered	0	no
Personality-explorer	{0.5; 0.5; 0.5; 0.5; 0.5}	yes
Personality-gatherer	{0.5; 0.5; 0.5; 0.5; 0.5}	yes
Personality-fighter	{0.5; 0.5; 0.5; 0.5; 0.5}	yes
Initial-ant-energy	1000	yes
	Individual level	
Agent-state	null	no
Amount-energy	500	yes
Amount-food	0	no
Attack-damage-inflicted	100	yes
Location-x	0	no
Location-y	0	no

The screenshot in Figure 9 shows the layout of the map with default parameters being used. The nest is situated in the centre of the map at coordinates {0; 0}. The food sources are located at the corners of the map and three bugs are chasing the ants between the nest and food sources. All the ants started their exploration journeys from the centre of the map. All the ants visually look the same, but in fact one third of them belong to the explorer type, which can travel at double speed, another third belong to the gatherer type, which can carry double amount of food, and the last third belong to fighter type which can inflict double damage when fighting bugs.

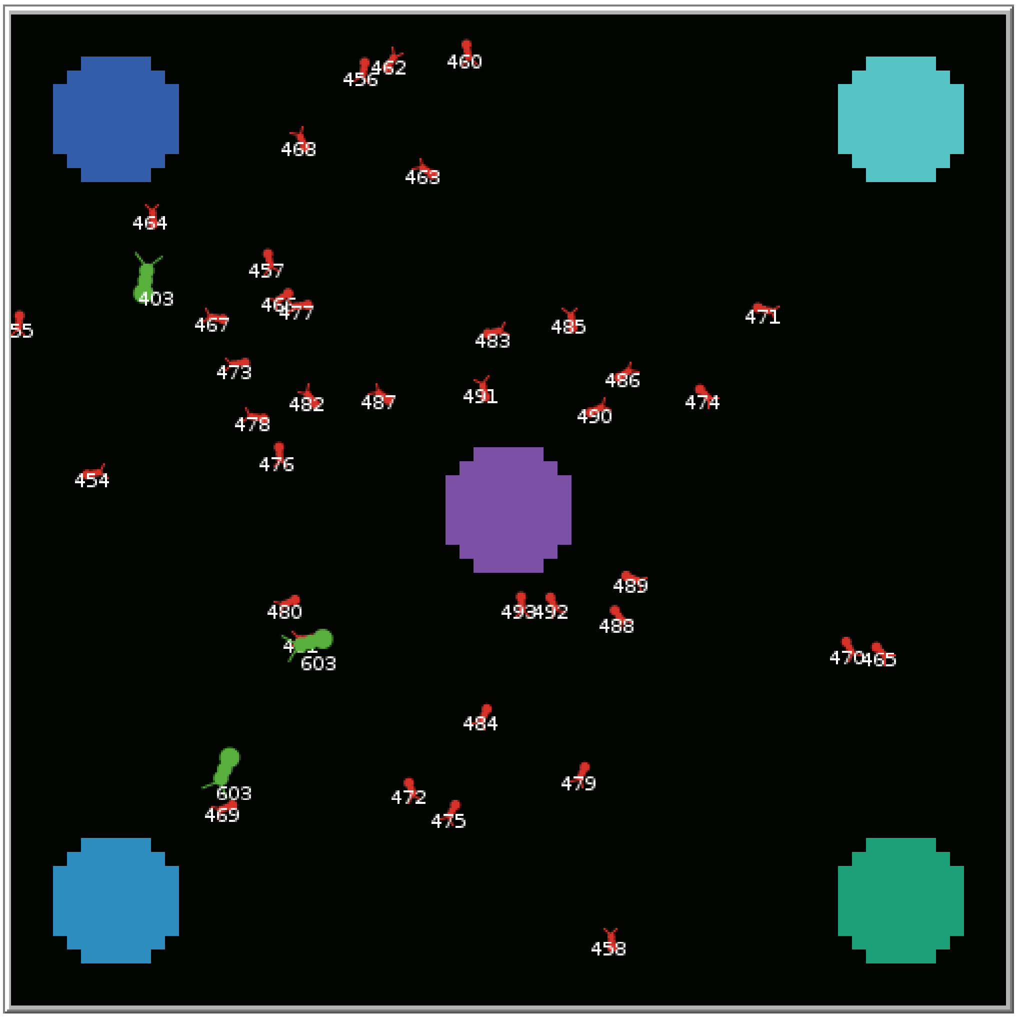

Figure 9. Simulation map after 50 ticks.

4.3.3 Process overview and scheduling

The model proceeds in discreet time which is measured in ticks. On every tick the ant agents run the BDI execution cycle that is an improved implementation of algorithm 1 described in [2].

The belief revision depends on previous beliefs, new perceptions, and on the personality traits of Agreeableness and Openness. The personality traits are implemented as probabilities. Agreeableness is important for signals obtained from other agents, such as a cry for help or reporting the location of a food source. The higher agreeableness is, the more likely the agent is to accept perception as a belief. Openness plays an important role for perceptions that conflict with the agent's previous beliefs. For instance, if the agent believed the food source to exist at some location and then it senses from the environment that the food source is no longer at that location, the less open the agent is, the more likely it is too keep looking for the food source around the location it used to exist. At the same time, an agent with the 1.0 probability for openness would immediately accept this new perception as a belief.

Once the beliefs have been revised, options to accomplish different desires are generated based on the new beliefs, previous intentions, and personality traits. Here the traits of Conscientiousness, Agreeableness and Neuroticism are considered. For instance, the more conscientious the agent is, the more likely it is to keep its previous intention as one of the options. Agreeableness is an important factor for options that involve helping other agents. For example, if another ant is under attack and has asked for help, – the more agreeable the responding agent is, the more likely it is to generate an option to go and help the ant in trouble. Neuroticism plays a key role in stressful situations. For instance, if the agent is under attack, the more neurotic ant will generate an option to run away to avoid bad emotions arising from the fight, whereas a less neurotic ant will generate different options for fighting.

Once the options have been generated, one of them is chosen by means of the filtering process. This is achieved by calculating the scores of all possible options by aggregating the impacts of positive and negative personality traits. Additionally, the priorities of the desires for the particular agent type are used as a component in the scoring of options. The highest-scored option will be chosen as an intention for the agent.

Once the intention has been selected, the action to fulfil it will be scheduled for the current execution cycle. For instance, if an ant under attack decides to fight then depending on the level of its extraversion, it might also consider not just fight on the current execution cycle alone against the bug, but also call for help from other ants. To do that, it will add two actions to its plan – Cry-for-help and Fight-bug. We note here that all the agents in the simulation can plan for an unlimited number of messaging actions for one execution cycle, but only for one movement action or fighting action.

After the planning has been done, the agent enters into execution phase and carries out all the actions it had planned for that execution cycle.

Bug agents, on the other hand, are much more simple and run around the map randomly, but once an ant is close to them, they will start chasing it, and will try to engage in a fight with the ant.

4.3.4 Design concepts

Although the ODD protocol defines 11 concepts, here we will present only the ones relevant for this simulation system. For instance, the concepts of learning and prediction are neglected as the agents in this simulation do not learn or predict future events.

Basic principles: The main observable in this simulation is the ant population representing a team of specialized individuals, which belong to three different agent types, each representing a team member of a specific type. All ant agents have been implemented using the BDI architecture, into which the Five Factor Personality traits' model was incorporated as is described in Chapter 4.1.

Emergence: The dynamics of the ant population emerge from the behaviours of agents belonging to the three sub-types that behave as three different swarm, but where the execution cycle and behaviour of each individual agent is entirely defined by its desires, personality, and perception of the environment.

Objectives: Agents of all three different types have different hierarchies of objectives, which in the BDI architecture are termed as differently prioritised desires.

Table 6 lists objectives for all ant types and attaches priorities to each objective. Priorities are provided as values ranging from 1 to 4; 1 being the highest and 4 being the lowest.

Table 6. Objectives priorities by ant agent type.

	Explorer type	Gatherer type	Fighter type
Find food sources	1	3	4
Gather food	3	1	3
Stay alive	2	2	2
Fight bugs	4	4	1

Sensing: Agents can sense their location on the map, possible threats or victims in the near proximity, which is hard-coded as the radius of 10 pixels. In addition to sensing the environment, the ant agents can sense whether there is an ongoing fight in the near proximity.

Interaction: Three interactions occur during the simulation: 1) fighting between ants and bugs; 2) distress call from one ant to another; 3) sharing of food source locations between all the ants.

Collectives: The ant population in general acts like a swarm with no collective objectives. But at the same time, they show up clear signs of teamwork efforts and, as has been indicated in Table 4, they "unconsciously" strive towards the same common high-level goals – collect as much food as possible and keep the mortality rate low. Therefore, ants should be considered as one collective. Bug population, on the other hand, does not strive towards achieving any higher-level goals other than what the individual bugs do and, therefore, the bugs should not be considered as forming a collective.

Observation: Throughout the simulation the following metrics are monitored:

1) *First food* – time required to gather food from the first food source;

2) *Energy* – total amount of food gathered;

3) *Starved* – number of ants deceased because of starvation;

4) *Killed* – number of ants died in fights;

5) *No bugs* – time span over which all bugs on the map have been killed.

4.3.5 Initialization

Before the simulation starts, the user can change the values for all the configurable variables listed in Table 5. Once all the variables have been set, the environment will be generated by pressing the Setup button. After the initial setup a rectangular map is generated with an ant nest in the middle and four food sources near the corners. All the ants will be placed in the centre of the nest and bugs evenly around the ant nest. All the values of the observation metrics will be set to initial values and a simulation can be started by pressing the Go button.

4.3.6 Input data

Although it is possible to read input data from external sources into NetLogo simulations, this simulation does not include the feature of reading in external input data. Instead, all the significant input values should be set up manually before initializing the simulation.

4.3.7 Sub-models

The algorithmic sub-models of this simulation have been elaborated from the ones defined in [2].

In the belief revision phase, perceptions are categorised into the following two types: 1) conflicting with previous beliefs and 2) non-conflicting.

The decision to accept a conflicting perception as belief is made according to following logic:

```
if randomNumber(between 0 and 1) greaterThan Openness
then discard(perception) else acceptBelief(perception)
```

Similarly, a non-conflicting perception will be discarded or accepted according to the following logic:

```
if randomNumber(between 0 and 1) greaterThan Agreeableness
then discard(perception) else acceptBelief(perception)
```

Personality traits are incorporated into the phase of options generation in the same manner.

Slight differences are observable in the filtering phase where the scoring is performed according to following scheme:

1) Generate random values from 0.0 to 1.0 for all personality traits;

2) Calculate positive scores for all personality traits by subtracting the personality trait value from the corresponding random value generated in 1);

3) For every option, add up the calculated positive scores of significant personality traits and divide the result with the number of participating significant personality traits. The result is the scoring component based on personality traits;

4) For every option, transpose the corresponding objective priority to 0.0-1.0 scale by subtracting the objective priority from the maximum priority value and dividing the result with the number of all unique priority values;

5) For every transposed objective priority factor in the level of specialisation by multiplying the transposed objective priority with the level of specialisation.

For example, the scoring of the option to run from a fight is described by the following pseudo code:

```
set devOpenness = Openness - randomNumber(between 0 and 1)
…
set devNeuroticism = Neuroticism - randomNumber(between 0 and 1)
set score-run-from-fight = (- devOpenness + devConsicentiousness
    + devExtraversion + devNeuroticism) / 4
    + ((4 – prio-fight-bug) / 4) * level-specialization
```

Note that in the pseudo code above, the divisions by 4 could be mathematically grouped, but are kept for the sake of uniformity. Personality trait Openness has a negative impact on the option to run, therefore, its positive score is negated. Other significant personality traits for this option are Conscientiousness, Extraversion and Neuroticism, which all have a positive impact on the option, thus, positive personality trait scores are used in the aggregation. The trait of Agreeableness has no impact on the option to run from fight, hence, it does not participate in the scoring. The division by the number of participating significant personality traits is necessary to ensure that the scoring component value for all options is in same scale regardless of the number of personality traits contributing to the result. The transposing of agent desire priorities from 1 to 4 scale to 0 to 1 scale is necessary to ensure that neither the desire component or the personality traits component fully overrides the other component value.

After all options are scored in a similar manner, the option with highest score is chosen.

4.4 Simulation quality goals

The quality aspects of the simulation are as follows presented in the order of importance: 1) accuracy of the personality traits implementation and 2) accuracy of the implementation of specialization level.

To evaluate the accuracy of the personality traits' implementation, the experiments put forward in [2] will be repeated. However, due to new features with respect to engagement of ants in fighting, differences are to be expected in concerning the traits of extroversion and agreeableness.

To evaluate the accuracy of the implementation of specialization level, the following three metrics are being monitored: 1) *First food,* 2) *Ants killed and* 3) *Time without bugs.*

By increasing the level of specialization, it is expected that: a) the ants will find food sources faster and b) ants of the fighter type will be more engaged in fighting.

5 Simulation results

This chapter presents the results from the simulation experiments, evaluates how well the simulation quality goals were achieved and, highlights the findings with respect to the personality trait of Agreeableness.

5.1 Results

The simulation was executed for 15 different personality profiles at three different specialization levels, resulting in 45 different configurations. For each configuration, 10 repeated simulation runs were executed. Based on the results of all those 450 simulation runs averages were calculated along with standard deviations.

The following personality profiles were chosen:

1) the 9 significant profiles identified in [2];
2) Operations Control Centre (OCC for short) and norm population average profiles from [29] and
3) four OCC profiles with the value of the trait of Agreeableness manipulated.

Table 7 presents the exact personality traits values for OCC, norm and the manipulated OCC profiles.

Table 7. Initial profiles from [29] and the profiles with manipulated agreeableness

Profile	O	C	E	A	N
Norm	0.5	0.5	0.5	0.6	0.4
OCC	0.5	0.6	0.5	0.5	0.3
OCC+A	0.5	0.6	0.5	0.6	0.3
OCC++A	0.5	0.6	0.5	1.0	0.3
OCC-A	0.5	0.6	0.5	0.4	0.3
OCC--A	0.5	0.6	0.5	0.0	0.3

For specialization parameter were chosen three values – 0, 0.5 and 1 – zero value causing the ants to have no effective specialization and behave as if they all were of the same type; one causing the ants to have maximum specialization possible as described in chapter 4.3.2. This allows the later assessment of the quality goal defined in chapter 4.4 regarding the implementation of the specialization level.

Table 8 presents the average values for all the used metrics over all simulation executions. The full summary table with standard deviations is available in Appendix 4.

Table 8. Simulation results extract.

OCEAN	Energy			First food			Starved			Killed			Total died			No bugs		
Specialization	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0
(0, 0, 0, 0, 0)	748266	617511	424501	85	56	47	65	70	92	541	552	560	606	623	652	899	1217	1389
(0, 0, 0, 1, 0)	612070	583508	395053	107	89	55	41	52	79	32	34	38	73	87	117	4502	4604	4635
(0, 1, 0, 0, 1)	867167	720375	509583	90	57	47	35	40	55	256	377	507	290	418	561	2	2	2
(0, 1, 1, 0, 1)	839667	656625	450500	90	57	46	31	43	68	281	379	443	312	422	510	370	434	199
(0, 1, 1, 1, 0)	601500	600875	419417	105	72	60	50	60	89	158	161	160	209	221	249	3978	3843	3845
(1, 0, 0, 0, 0)	672901	586269	399855	107	60	50	18	20	40	326	298	298	344	318	338	2614	1810	2180
(1, 0, 1, 0, 0)	692116	594649	397396	95	61	49	18	32	41	217	221	217	235	253	258	2909	2810	2778
(.5, .5, .5, .5, .5)	669659	596442	476277	91	73	50	35	44	50	49	45	49	84	89	99	4148	4289	4140
(1, 1, 1, 1, 1)	651654	603263	388436	95	68	70	42	41	77	23	27	32	65	68	109	4575	4540	4594
OCC	632413	619262	466309	99	79	51	46	44	67	51	53	51	97	97	118	4361	4270	4291
Norm	585771	604101	457851	104	66	60	49	37	55	37	48	44	86	84	100	4002	4491	4389
OCC++A	567828	544456	359441	103	74	59	38	50	81	41	40	43	79	90	124	4531	4495	4549
OCC—A	782681	714458	434507	90	55	48	43	42	53	304	307	307	346	350	360	2033	1763	2388
OCC+A	616054	634878	475711	99	66	50	47	41	73	49	49	45	96	91	118	4351	4419	4490
OCC-A	679930	626033	475897	96	64	52	39	46	56	69	73	67	108	119	123	3712	3497	4045
Average	681312	620180	435382	97	67	53	40	44	65	162	178	191	202	222	256	3132	3099	3194

5.2 Evaluation of the implementation

In sub-chapter 4.4 were defined two quality goals for the simulation. For the first goal – accuracy of personality traits – the results shown in upper part of Table 8, in rows with OCEAN values from (0, 0, 0, 0, 0) to (1, 1, 1, 1, 1), and with Specialization value equal to zero were compared to the results achieved in [2]. Please note that in [2] the number of ants killed in fights was not measured separately but was rather included in the number of ants starved. On the other hand, in this work different food sources were not measured separately as they were in [2]. Therefore, the number of ants starved in [2] should be compared with the numbers in the column *Total died* of Table 8. Similarly, the column *Energy* of Table 8 needs to be compared with the sum of all different food sources measured in [2]. Taking these considerations into account, the results obtained by us match well with the ones reported in [2], with only minor differences which can be explained by deviation in simulation executions. Therefore, it is fair to conclude, that the first quality goal defined in sub-chapter 4.4 was achieved successfully.

For evaluating the specialization aspect, the whole of Table 8 needs to be considered and comparisons should be made between the same personality profiles in simulations with different specialization levels. It is easy to see from Table 8 that as the specialization value was increased, the *First food* value has decreased regardless of the personality profile, and the *Killed* value varies depending on the personality profile but has increased on average. The changes observed in *First food* and *Killed* values confirm that also the second quality goal defined in sub-chapter 4.4 was achieved.

Considering all the above, the evaluation can be concluded as successful and proving that the simulation has achieved all the quality goals defined in sub-chapter 4.4.

5.3 Findings and Interpretation of the results

According to the fourth goal of this thesis described in sub-chapter 1.3, this research explores the impact of the traits of agreeableness and helpfulness on problem-solving activities from a teamwork perspective. The impact of the agreeableness trait and its sub-trait of helpfulness were measured in the simulation configurations shown in the lower part of Table 8, in the rows starting with OCC. The exact personality traits values for all those profiles were given in Table 7.

From the data recorded for those five profiles at three different specialization levels, clear indications about the impact of the trait of agreeableness on the quality indicators defined in Table 4 were observed. The impact is illustrated in Table 9, where the OCC profile was considered as a comparison baseline with the other four profiles. The *Energy* and *Total died* values in Table 9 are presented as differences from baseline values in percentages. In addition, another calculated column has been added to Table 9 to show the ratio between the increase in the *Energy* gain and the increase in the *Total died*. This ratio expresses how well the team managed to handle conflicting goals compared to the baseline OCC profile team.

Table 9. The impact of the trait of agreeableness on the OCC personality profile.

Personality	Energy	Total died	Gain/cost ratio
Specialization		0	
OCC	baseline	baseline	
OCC++A	-10.2%	-18.6%	1.10
OCC+A	8.5%	22.0%	0.89
OCC-A	10.4%	12.8%	0.98
OCC—A	15.1%	219.7%	0.36
Specialization		0.5	
OCC	baseline	baseline	
OCC++A	-12.1%	-7.2%	0.95
OCC+A	16.6%	0.7%	1.16
OCC-A	-1.4%	30.9%	0.75
OCC—A	14.1%	194.7%	0.39
Specialization		1	
OCC	baseline	baseline	
OCC++A	-22.9%	4.8%	0.74
OCC+A	32.3%	-4.9%	1.39
OCC-A	0.0%	4.2%	0.96
OCC—A	-8.7%	193.2%	0.31

There is only one personality profile for which decisive conclusions can be made based on the simulation results obtained until now: the OCC+A (minimally increased Agreeableness trait) profile. Although the OCC+A profile was outperformed by the OCC base profile where there was no specialization among the ant agents (section with Specialization 0 in Table 9), we can observe a significant performance increase for the same profile in a more specialized setup where different types of ants behave more according to their type-specific expectations. For instance, in the runs with the maximum specialization (section with Specialization 1 in Table 9), we can witness a 32.3% increase compared to the OCC base profile in the total energy gained and a 4.9% decrease in the number of ants deceased. This in total means a 39% increase in the overall performance of the team.

In the context of the AOC problem domain modelled in Chapter 3, we saw a very clear distinction between different roles, their responsibilities and conflicting quality goals. This corresponds to the scenario with maximum specialization in the simulation. Therefore, we can conclude that also for the AOC, similar small changes in the

Agreeableness trait of the personality profile can improve the overall performance of the team and should be considered when hiring new staff.

6 Conclusions

The first goal of this thesis was to obtain an overview of different agent-based representations of the FFM personality model and to choose one as the foundation for the rest of this work. First, the widely adopted architecture in the agent-community – BDI – was investigated. Thereafter, three alternatives were considered to make sure that nothing more suitable has emerged. Altogether, four alternative architectures were compared and it was concluded that none of the three alternatives had significant benefits compared to the well-adopted BDI architecture.

The second goal of this work was to further elaborate the problem domain of airlines modelled in the previous work [1] by means of the Agent-Oriented Modelling (AOM) methodology [8]. For that purpose, the models created by Balašova [1] were first further validated and enhanced. Following, the sub-domain of disruption handling in AOC was chosen for a more comprehensive analysis and modelling. The problem domain analysis provided the work with further details on what kind of requirements and quality goals the new simulation had to satisfy. The first finding was that the simulation of personality traits had to be consistent with that described in [2]. Secondly, the simulation proposed in [2] had to be complemented with the features that were better aligned with the AOC problem domain.

Based on the previous work [1][2] and on the problem domain analysis carried out in the context of this thesis, an agent-based simulation system of an organisation with roles performed by agents having different personality traits was designed and described according to the ODD protocol.

The final goal was to create a proof-of-concept implementation of the newly designed simulation system on the agent-based simulation environment NetLogo [13] to analyse the impact of the psychological traits of agreeableness and helpfulness on problem-solving activities from a teamwork perspective. The simulation system was implemented, evaluated and applied for studying the impact of the trait of agreeableness using the average personality profile of the airline Operations Control Centre [29].

Simulation executions carried out with the created simulation system revealed that the trait of agreeableness has noticeable impact on the performance of the agents in a teamwork environment. This outcome was illustrated by calculating a gain-cost ratio for all the profiles in simulations with different levels of specialization. The ratio showed that the more specialized the team members were, the more benefit a small increase in Agreeableness provided. In the specialized to the highest degree environment, which best corresponds to the modelled AOC problem domain, we witnessed a 39% increase in the overall team performance.

The most important directions of the future work are as follows:

1. Using the created simulation system for a greater number of simulation executions with more variations in the values of configurable parameters resulting in a much bigger dataset to analyse;

2. Creating a problem-domain specific simulation that would precisely model and simulate the roles and workflows of airline Operations Control Centres and verify if the improvement possibilities identified in this work also remain valid in that simulation environment.

The first direction for future work would involve more theoretical research work with the aim to study the trait of agreeableness more deeply, attempting to find all the correlations and patterns on how the trait of agreeableness impacts the team performance.

The second direction for future work would involve more of socio-technical system analysis with the attention mostly focused on creating a simulation system that would accurately reflect the exact working environment of an actual airline Operations Control Centre.

References

[1] Balašova, J. (2015). Agent Oriented Modelling and Simulation of Airline Workflows. Master's Thesis. Tallinn, Estonia: Tallinn University of Technology.

[2] Ahrndt, S., Fähndrich, J., Albayrak, S. (2015). Modelling of Personality in Agents: From Psychology to Implementation. In: E. Bordini & Y. Weiss (Eds.), Proceedings of the Fourth International Workshop on Human-Agent Interaction Design and Models (HAIDM 2015), Istanbul, Turkey, 4 May, http://www.smart-society-project.eu/publications/proceedings/haidm15/, last accessed on 16 January 2017, 1-16.

[3] Durupinar, F., Allbeck, J., Pelechano, N., Badler, N. (2008). Creating Crowd Variation with FFM Personality Model (Short Paper). In: Padgham, Parkes, Müller and Parsons (Eds.), Proceedings of 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, 12-16 May, last accessed on 14 January 2017, 1217-1220.

[4] About Jeppesen. [WWW] http://ww1.jeppesen.com/company/about/what-we-do.jsp (10.01.2016).

[5] Von Alan, R. H., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS quarterly, 28(1), 75-105.

[6] Axelrod, R. (2006). Agent-based modeling as a bridge between disciplines. In: L. Tesfatsion and K. L. Judd (Eds.), Handbook of Computational Economics 2. Amsterdam, The Netherlands: North Holland, 1565–1584.

[7] Borrill, P. L. & Tesfatsion, L. (2011). Agent-based Modeling: The Right Mathematics for the Social Sciences? In: J. B. Davis and D. W. Hands (Eds.), The Elgar Companion to Recent Economic Methodology. Cheltenham Glos, UK, and Northampton, MA: Edward Elgar Publishing, 228-258.

[8] Sterling, L., Taveter, K. (2009). The Art of Agent-Oriented Modeling. Cambridge, MA, and London, England: MIT Press.

[9] Gilbert, N. & Troitzsch, K. G. (2005). Simulation for the Social Scientist. 2nd Ed. Berkshire, UK: McGraw-Hill International, Open University Press.

[10] Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., ... & DeAngelis, D. L.(2006). A standard protocol for describing individual-based and agent-based models. Ecologi-cal modelling, 198(1), 115-126.

[11] Grimm, V., Berger, U., DeAngelis, D. L., Polhill, J. G., Giske, J. & Railsback, S. F. (2010). The ODD protocol: a review and first update. Ecological modelling, 221(23), 2760-2768.

[12] Taveter, K., Du, H., & Huhns, M. N. (2012). Engineering societal information systems by agent-oriented modeling. Journal of Ambient Intelligence and Smart Environments, 4(3), 227-252.

[13] Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/, last accessed on 22 November 2016. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University.

[14] Rao, A. S., Georgeff, M., P. (1991). Modeling Rational Agents within a BDI-Architecture. In: Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR&R-91), San Mateo, USA, April 1991, 473-484

[15] Wooldridge, M. (2000). Reasoning about Rational Agents. Intelligent Robotics and Autonomous Agents, The MIT Press.

[16] Zoumpoulaki A, Avradinis N, Vosinakis S. (2010). Multi-agent simulation framework for emergency evacuations incorporating personality and emotions. In Artificial Intelligence: Theories, Models and Applications, Lecture Notes in Computer Science, vol. 6040. Springer: Berlin Heidelberg; 423–428.

[17] Bratman, M. E. (1987). Intentions, Plans, and Practical Reason. Harvard University Press, Cambridge, MA

[18] Phung, T., Winkoff, M., Padgham, L. (2005). Learning Within the BDI Framework: An Empirical Analysis. In: Proceedings of the 9th international conference on knowledge-based intelligent information and engineering systems (KES 2005), Melbourne, Australia, September 14-16. Springer: Berlin Heidelberg, 282–288

[19] Kinny D, Georgeff M, Rao A (1996) A methodology and modelling technique for systems of bdi agents. In: Agents breaking away, Springer, pp 56–71

[20] Matthias Scheutz and Paul Schermerhorn. Many is more: The utility of simple reactive agents with predictive mechanisms in multiagent object collection tasks. In: Web Intelligence and Agent Systems, 3(2), 2005.

[21] Jan Broersen, Mehdi Dastani, Joris Hulstijn, Zisheng Huang, Leendert van der Torre (2001). The BOID Architecture - Conflicts Between Beliefs, Obligations, Intentions and Desires. In: AGENTS'01, Montreal, Canada, May 28-June 1. ACM

[22] Pearl, J. (1993). From conditional ought to qualitative decision theory. In Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (UAI'93), John Wiley and Sons, 12-20

[23] Dastani, M., Hulstijn, J. & van der Torre, L. (2001). BDI and QDT: a comparison based on classical decision theory. In: Game Theoretic and Decision Theoretic Agents: Papers from the AAAI Spring Symposium (GTDT'01). 16-26, AAAI Press.

[24] Machado, N. G. (2014). Impact of the organizational structure on operations management: the airline operations control centre case study. Master's thesis. Porto, Portugal: University of Porto.

[25] Castro, A. M. (2013). A Distributed Approach to Integrated and Dynamic Disruption Management in Airline Operations Control. PhD Thesis. Porto, Portugal: University of Porto.

[26] Myers, I.B., Byers, P.B. (1995). Gifts Differing: Understanding Personality Type. Nicholas Brealey Publishing, 2 edn.

[27] Ahrndt, S., Aria, A., Fahndrich, J., Albayrak, S. (2014). Ants in the FFM: Modulating agents with personality for planning with humans. In: Bulling, N. (ed.) Multi-Agent Systems (EUMAS 2014), pp. 1–16. Lecture Notes in Artificial Intelligence, Springer (December 2014)

[28] McCrea, R.R., John, O.P. (1992). An introduction to the five-factor model and its applications. Journal of Personality 60(2), pp 175–215

[29] Peters, T. E. (2015). Personality Traits Modeling of Operations Control Center (OCC) Personnel and Vocational Implications. Master’s Thesis. Berlin, Germany: Technische Universität Berlin.

Appendix 1 – AOC Organization Sub-models

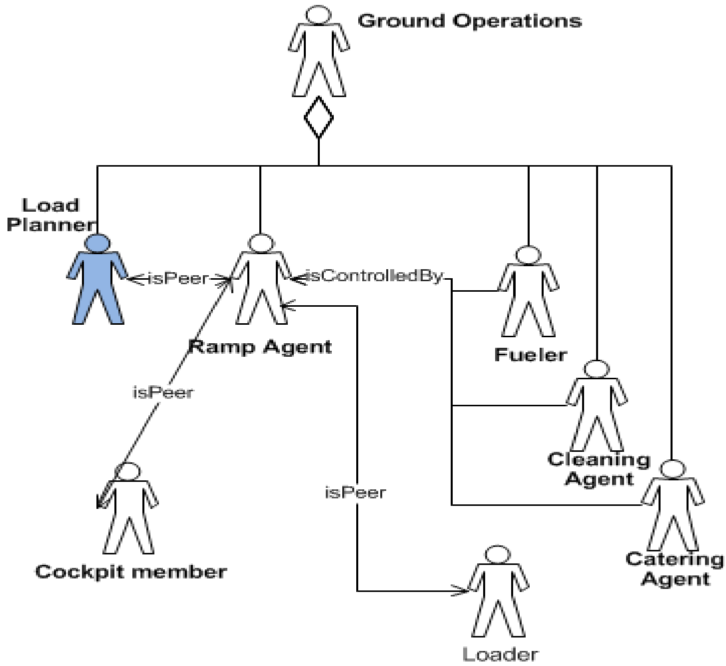

Figure 10. AOC Ground Operations Organization sub-model.

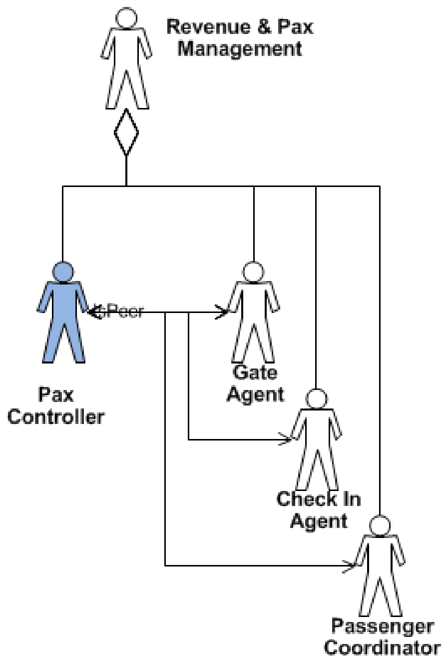

Figure 11. AOC Passenger & Revenue Organization sub-model.

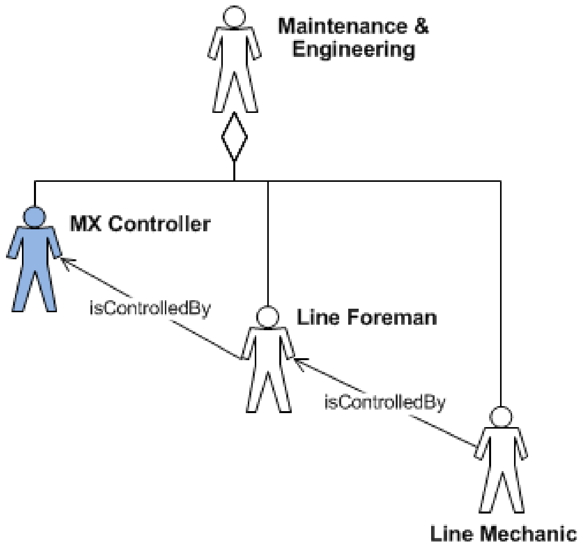

Figure 12. AOC Maintenance Organization sub-model.

Appendix 2 – AOC Goal models

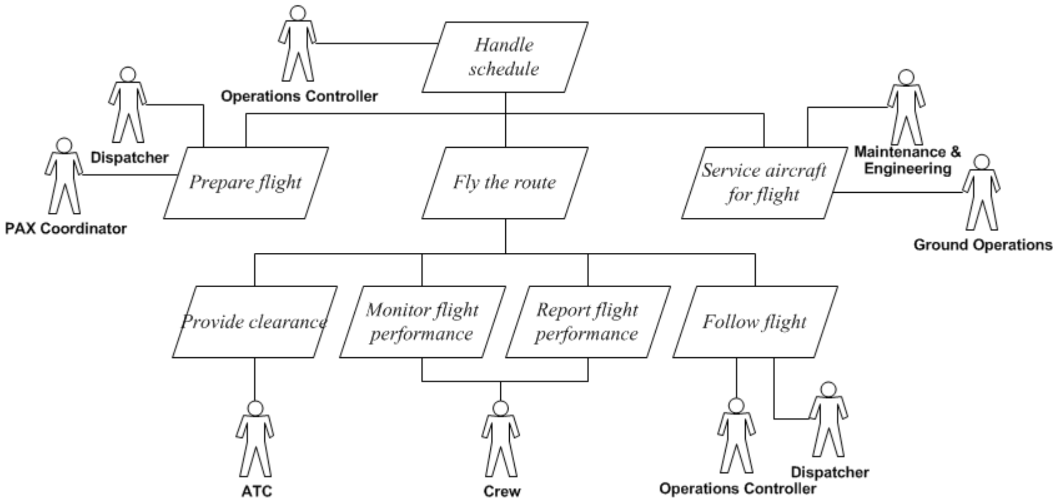

Figure 13. AOC Goal model: Handle schedule.

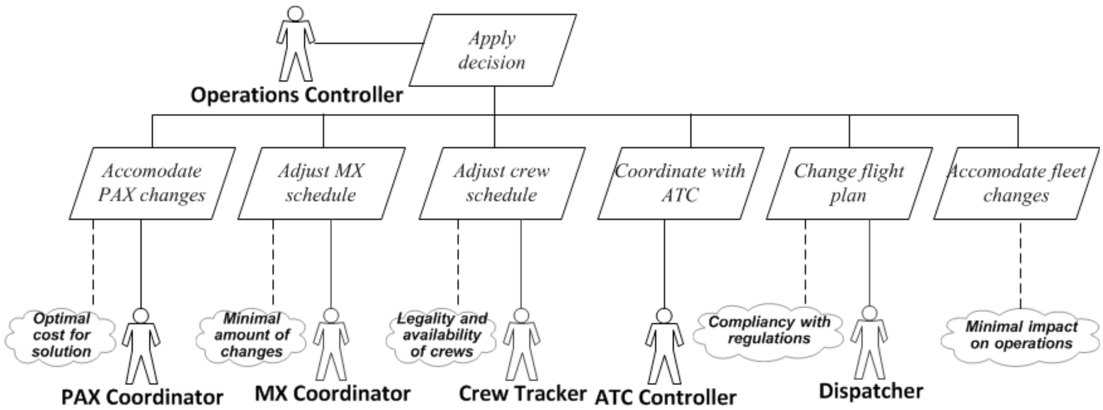

Figure 14. AOC Goal model: Apply decision.

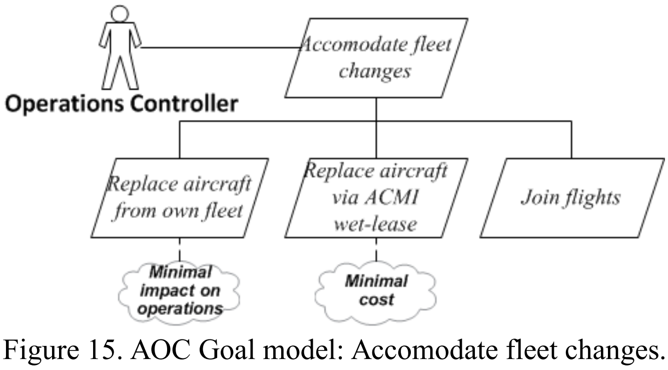

Figure 15. AOC Goal model: Accomodate fleet changes.

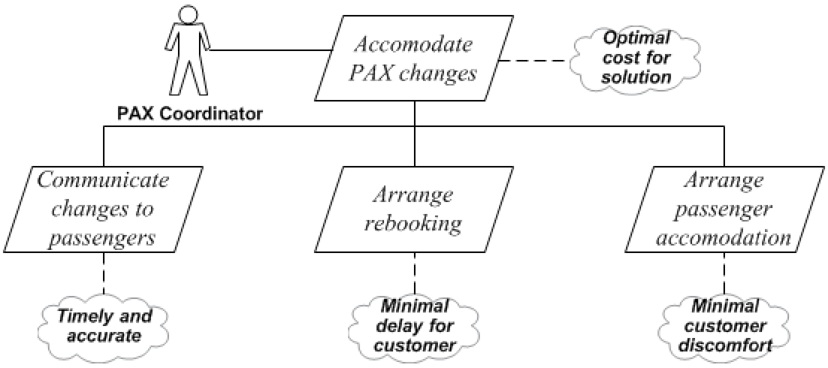

Figure 16. AOC Goal model: Accomodate PAX changes.

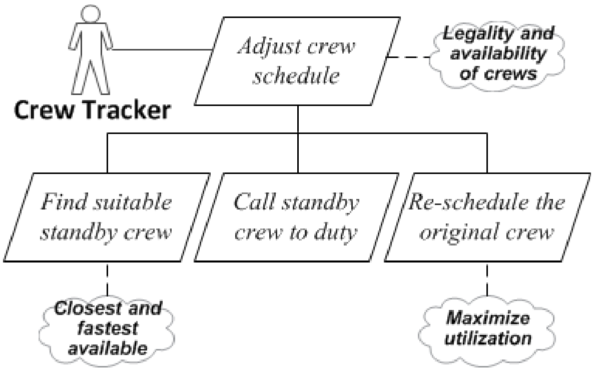

Figure 17. AOC Goal model: Adjust crew schedule.

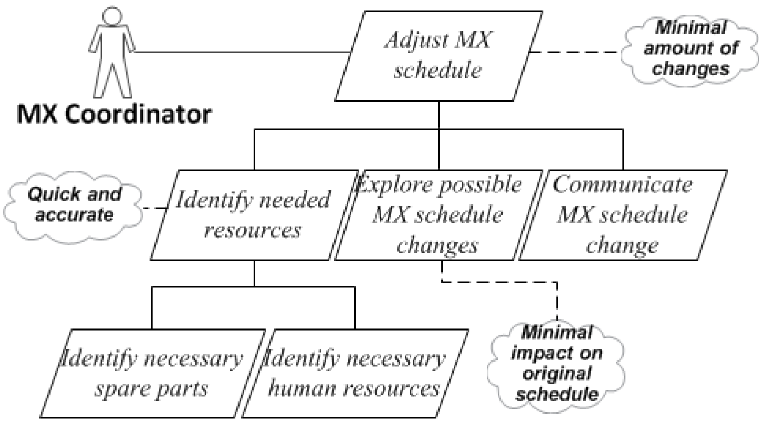

Figure 18. AOC Goal model: Adjust MX schedule.

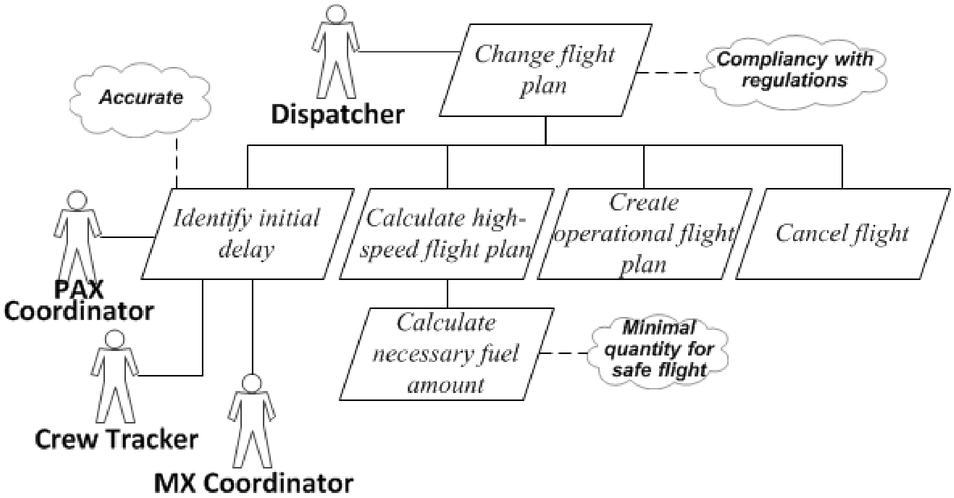

Figure 19. AOC Goal model: Change flight plan.

Appendix 3 – Interaction diagrams

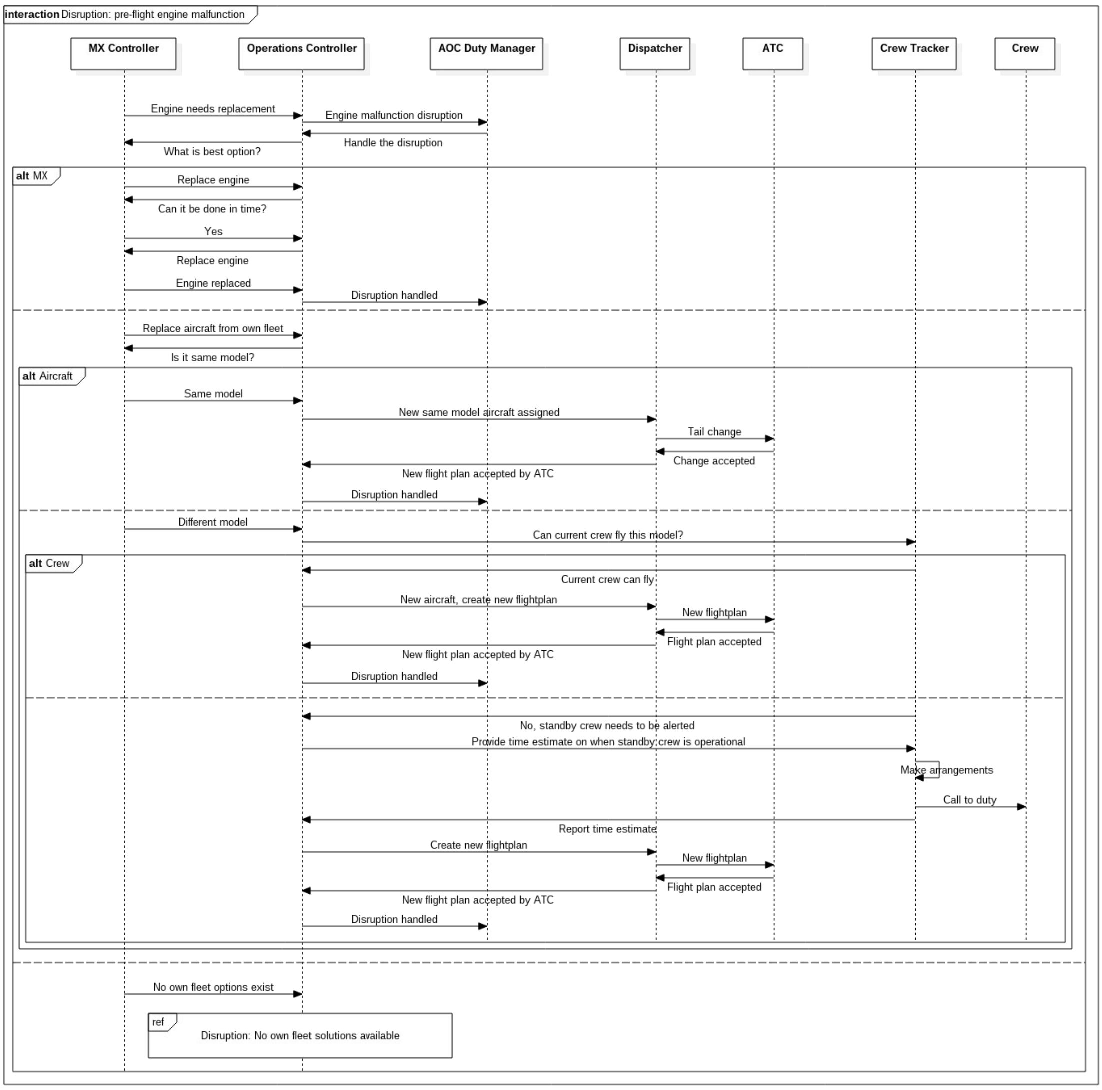

Figure 20. Interaction diagram - Pre-flight engine malfunction disruption.

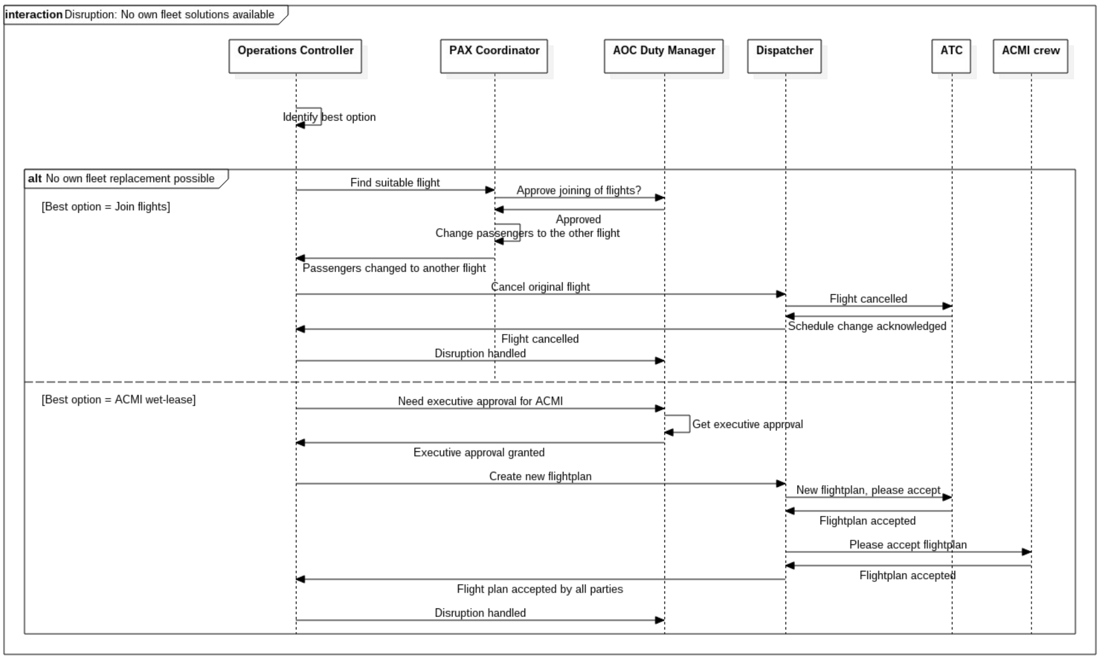

Figure 21. Interaction diagram - No own fleet solutions available.

Appendix 4 – Simulation results

Table 10. Simulation results full table.

OCEAN	Energy			First food			Starved			Killed			Total died			No bugs			sdEnergy			sdFirstFood			sdStarved			sdKilled			sdNobugs		
Specialization	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	1.0
(0, 0, 0, 0, 0)	748266	617511	424501	85	56	47	65	70	92	541	552	560	606	623	652	899	1217	1389	40321	56786	24541	7	5	4	8	10	9	12	40	36	86	136	162
(0, 0, 0, 1, 0)	612070	583508	395053	107	89	55	41	52	79	32	34	38	73	87	117	4502	4604	4635	9018	30948	24490	5	13	5	3	3	7	3	1	8	148	27	93
(0, 1, 0, 0, 1)	867167	720375	509583	90	57	47	35	40	55	256	377	507	290	418	561	2	2	2	35250	44453	8661	10	3	3	4	9	5	46	71	63	0	0	0
(0, 1, 1, 0, 1)	839667	656625	450500	90	57	46	31	43	68	281	379	443	312	422	510	370	434	199	35105	6506	16876	9	6	2	4	4	8	40	43	27	42	53	12
(0, 1, 1, 1, 0)	601500	600875	419417	105	72	60	50	60	89	158	161	160	209	221	249	3978	3843	3845	27825	13917	24002	9	1	8	8	2	9	7	21	25	56	320	245
(1, 0, 0, 0, 0)	672901	586269	399855	107	60	50	18	20	40	326	298	298	344	318	338	2614	1810	2180	47474	41108	33591	11	9	4	1	2	5	17	11	47	144	465	203
(1, 0, 1, 0, 0)	692116	594649	397396	95	61	49	18	32	41	217	221	217	235	253	258	2909	2810	2778	51604	44631	31453	5	3	7	2	5	5	18	41	7	312	357	189
(.5, .5, .5, .5, .5)	669659	596442	476277	91	73	50	35	44	50	49	45	49	84	89	99	4148	4289	4140	23225	42286	10572	5	10	8	2	2	2	4	3	1	361	159	237
(1, 1, 1, 1, 1)	651654	603263	388436	95	68	70	42	41	77	23	27	32	65	68	109	4575	4540	4594	14513	18876	13888	8	8	7	3	7	5	1	3	6	49	78	33
OCC	632413	619262	466309	99	79	51	46	44	67	51	53	51	97	97	118	4361	4270	4291	53161	8395	28882	10	7	2	7	3	8	12	11	4	280	275	58
Norm	585771	604101	457851	104	66	60	49	37	55	37	48	44	86	84	100	4002	4491	4389	19549	49044	13544	10	2	10	6	8	9	4	5	3	423	88	49
OCC++A	567828	544456	359441	103	74	59	38	50	81	41	40	43	79	90	124	4531	4495	4549	17246	21637	11367	8	9	9	6	7	2	7	10	1	110	121	36
OCC—A	782681	714458	434507	90	55	48	43	42	53	304	307	307	346	350	360	2033	1763	2388	56558	30553	16287	8	4	9	5	9	6	30	38	45	603	206	182
OCC+A	616054	634878	475711	99	66	50	47	41	73	49	49	45	96	91	118	4351	4419	4490	24677	45106	14241	1	9	4	4	6	5	3	5	5	246	162	90
OCC-A	679930	626033	475897	96	64	52	39	46	56	69	73	67	108	119	123	3712	3497	4045	50101	44798	28191	7	9	7	5	8	6	10	13	9	383	345	296
Average	681312	620180	435382	97	67	53	40	44	65	162	178	191	202	222	256	3132	3099	3194	33708	33270	20039												