Intelligent Environments 2019
A. Muñoz et al. (Eds.)

doi:10.3233/AISE190068

A Multi-Agent Negotiation Approach for Airline Operation Control

Soufiane Bouarfa
Aerospace Engineering, Abu Dhabi Polytechnic, Al Ain, United Arab Emirates
Reyhan Aydoğan
Department of Computer Science, Özyeğin University, İstanbul, Turkey
Alexei Sharpanskykh
Faculty of Aerospace Engineering, Delft University of Technology, Delft, the Netherlands

Abstract - This paper proposes and evaluates a new airline disruption management policy using agent-based modelling, simulation, and verification. The new policy is based on a multi-agent negotiation protocol and is compared with three airline policies based on established industry practices. The application concerns Airline Operations Control whose core functionality is disruption management. In order to evaluate the new policy, a rule-based agent-based model of the AOC and crew processes has been developed. This model is used to assess the effects of multi-agent negotiation on airline performance in the context of a challenging disruption scenario. For the specific scenario considered, the multi-agent negotiation policy outperforms the established policies when the agents involved in the negotiation are experts. Another important contribution is that the paper presents a logic-based ontology used for formal modelling and analysis of AOC workflows.

Keywords: Workflow modelling, Rule-based modelling, Formal modelling, Multi-agent negotiation, model checking, Airline operations control.

1. Introduction

Airlines cope with many disruptions of different nature that implicitly or explicitly test their resilience on a regular basis. These disruptions may interact with each other, potentially creating a cascade of other disturbances that may span over different spatial as well as time scales, ranging from affecting only one aircraft or crew, up to a group of aircraft. In current airline operations, disruptions are managed by Airline Operations Control (AOC), and may impact the economic performance of the airline and customer service. E.g., some flights are rerouted, some aircraft are leased, and some flights are re-booked. Consideration of the aircraft routings, crew, maintenance, weather, customer needs, security and turnaround processes complicate AOC. Current AOC practice consists of a coordination process between many human operators, each of which plays an essential role in disruption management. With the ever-growing complexity and various types of interdependencies between airlines, airports, and ATC centres, maintaining airline resilience to expected and unexpected disruptions becomes a challenging task. In order to manage disruptions in a resilient way, advanced forms of coordination between human operators and automation is required. This paper aims at evaluating a new coordination approach based on multi-agent negotiation and comparing it with existing strategies in the context of a realistic operational scenario.

Multi-agent negotiation, negotiation among more than two agents, has taken the attention of the AI research community in recent years. In the AOC domain, this has only been explored by one researcher [1]. There is a variety of negotiation protocols proposed in the literature. For instance, the **Stacked Alternating Offers Protocol (SAOP)** [2] governs the interaction among agents in a turn-taking fashion. One of the agents initiates the negotiation by making an offer. The next agent in line can accept this offer or make a counteroffer by overriding the previous offer, or end the negotiation. This process continues until reaching a mutual consensus or reaching a deadline. In the **Single Text Mediated Protocol** [3], there is an unbiased mediator searching for an agreement without knowing each agent's preferences. The mediator initiates the negotiations by making a random bid and asks each agent to vote to accept or reject this offer. When all agents accept the given offer, the mediator keeps this offer as "most recently accepted bid". In the next round, it only changes the value of one of the issues and asks agents to vote accept or reject this modified offer. Other protocols include the **Feedback based Protocol** [4], and the **Intra-team negotiation protocol** [5]. While some of these protocols involve an unbiased mediator, which aims to help negotiating agents to find a consensus; others focus on the interaction among only negotiating agents. In order to model negotiation in AOC, the authors developed a new approach similar to the single text mediated protocol. In the proposed policy, a team representative acts like a mediator to reach a unanimous agreement by making offers according to his preferences and asking other agents to vote for or against the given offers. This protocol is compatible with AOC in which the supervisor makes the final decision upon feedback from other experts.

This paper proposes developing and evaluating a new multi-agent negotiation policy for airline disruption management. This was motivated by the need to improve coordination processes in AOC. The paper is organized as follows. Section 2 presents the simulated policies. Section 3 explains the development of the agent-based model and the case study. Section 4 explains how the model is verified. Section 5 provides the simulation results, and finally section 6 provides key conclusions of this work. Appendix A includes the ontology used for developing the agent-based model.

2. AOC disruption management policies

2.1. Current AOC Policies P1-P3

In order to select representative AOC policies and make a clear distinction between them, a critical element is the understanding of how AOC operators make their decisions in relation to various aspects during disruption management. Bruce [6] has systematically studied the decision-making processes of 52 controllers in six AOC centers and found out that airline controllers use policies with three different levels of performance. These policies are shown in table 1.

Table 1. Overview of the three AOC policies P1-P3 in relation to various disruption management aspects.

Aspect	AOC Policy P1	AOC Policy P2	AOC Policy P3
Maintenance Information	Accept information source and content and act on information given about a maintenance situation	Challenge/ query information about a maintenance situation	Seek alternative information and recheck source and reliability.
Crewing	Await crew from inbound aircraft	Challenge crew limits/ Seek extensions to crew duty time	Seek alternative crew (e.g. from nearby base or other aircraft)
Curfews	Curfews are not taken into account	Identify curfews and work within them	Seek curfew dispension
Aircraft	Seek first available aircraft	Request high speed cruise	Combine flights to free up aircraft

2.2. Multi-agent Negotiation Policy P4

We developed a new negotiation approach based on the Single Text Mediated Protocol. In this context, the AOC supervisor has the power of making the final decision based on the feedback given by other agents and it also needs other agents' expertise to generate potential solutions for the underlying problem. For example, if the problem is related to aircraft, it is required that the aircraft controllers inform the AOC supervisor about all possible aircraft solutions. Since the given solutions may have an influence on other agents' processes, it is required to find a consensus among all agents. Accordingly, the proposed negotiation approach works as follows:

Pre-negotiation phase:

- Upon identification of a problem, the AOC supervisor asks the specialist agents to provide all possible solutions corresponding to their problem dimension within a certain deadline.
- All specialist agents provide their solution to the AOC supervisor. The specialist agents include the aircraft controller agent ACo, the crew controller agent CCo, and Passengers Services agent PS.
- If the AOC supervisor does not receive solutions from all three specialist agents, the disruption cannot be managed.

Negotiation phase:

- The AOC supervisor evaluates all proposals received from the specialist agents and selects one of the solutions according to his bidding strategy. The AOC supervisor announces his chosen solution to the specialist agents.
- The specialist agents vote for or against the announced solution by the AOC supervisor. Note that the specialist agents may use different criteria to evaluate the offer (e.g., cost, safety, crew satisfaction, etc.)
- If all three agents agree about the solution, the negotiation ends with the current solution successfully.
- If no consensus is received, the AOC supervisor makes a new offer for the three agents to vote on. In the meantime, it keeps the offer which was accepted by the majority and updates this offer over time. Note that this process is repeated until reaching an agreement or the deadline.

- If the agreement is not reached before the deadline, the AOC supervisor ends the negotiation with the compatible offer that has the most favourable votes.

The AOC supervisor can employ a time-based concession strategy in order to decide which offer to announce. I.e., he first offers the best solution according to his own evaluation function and concedes over time. That is, in the next round he offers his second best solution, then his third best and so forth

Specialist Agents' Acceptance Strategy:

- When the AOC supervisor makes an offer, the utility of this offer, U(o) is calculated by each specialist agent. If the utility of the given offer is greater than the threshold value, the specialist agent accepts the offer; otherwise, it votes to reject it. Note that agents determine their threshold value before the negotiation starts.
- Before negotiation starts, each specialist agent determines the importance of their evaluation criteria by setting weights directly, or using the Analytic Hierarchy Process [7] to estimate the weights. By using pairwise comparison such as "Key Performance Area (KPA) 1 is two times as important as KPA2", agents can estimate the weight values of each criterion (e.g., w_{KPA1}=0.7). Note that the sum of those weight values is equal to one.
- The preferences of each agent can be modelled by an additive utility function where the utility of an offer is estimated by the weighted sum of evaluation values for each criterion. These evaluation values can be estimated by using domain knowledge. For example, if the agent estimates $EV\_{KPA1}(o)$= 0.8 and $EV\_{KPA2}(o)$=0.6 and the criterion weights are w_{KPA1}=0.7 and w_{KPA2}=0.3, then the utility of the given offer is calculated as U(o)= 0.7*0.8+0.3*0.6.

3. Agent-Based Modelling

Developing the AOC agent-based model is performed in four major steps. In **step 1**, the agents are identified. Since the purpose of the simulation model is to compare different coordination strategies, the main agents are those human operators involved in managing the disruption and the decision support systems they use. Thus, all actors were modelled as proactive agents. The complete list of agent types is provided in Appendix A. Once the key agents have been identified, their behaviour in the context of the considered scenario is accurately specified in the **step 2** based on data and interviews with experts. Subsequently, interactions between the agents are implemented in the modelling environment in **step 3** and the model is verified and executed in **step 4**. To represent the agents and their behaviour we employ a generic temporal-causal modelling approach [8]. The ontology used can be found in Appendix A.

3.1. Case Study

In order to assess the impact of the four policies (P1-P4) we will consider a challenging AOC scenario that is well described and evaluated in [7] and includes an overview of the flights being monitored by the airline controller at the time of disruption [9]. The scenario concerns a mechanical problem with an aircraft at Charles de Gaulle (CDG) airport, aiming for a long-haul flight to a fictitious airport in the Pacific, which is indicated by the code PCF. In [7], this scenario was considered by a panel of AOC experts. They developed several alternatives, and subsequently identified the best solution, which was to re-route the flight from CDG to PCF and to include a stop-over in Mumbai (BOM). In

parallel, a replacement flight crew was flown in as passengers on a scheduled flight from PCF to BOM in order to replace the delayed crew on the flight part from CDG to PCF. The central question therefore is how well the outcome of the agent-based modelling and simulation of the AOC center compared to the expert panel in finding a best solution?
Having built a good understanding of the roles and responsibilities of each agent in the considered scenario, various rules were assigned to different agents. These rules are either based on established airline policies P1-P3 [10] or the proposed multi-agent negotiation policy P4. Complementary to this, operational workflows from a European airline were used [11] to identify the different kinds of technical systems being used in case of a mechanical breakdown, the interactions of agents with these systems, and decision-making rules for each agent. Figure 1 shows how the workflow would look if agents follow policy P4.

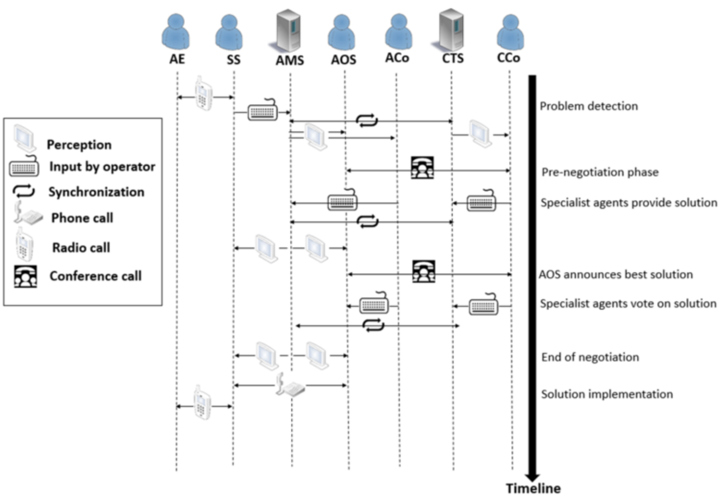

Figure 1. Operational Workflow Corresponding to the two phase multi-agent negotiation protocol P4.

3.2. Agent-based modelling environment LEADSTO

To implement agent interaction rules we made use of the LEADSTO simulation environment [12,13] using the formal ontology presented in Appendix A. LEADSTO proved its value in a number of projects in multi-agent systems research (e.g. in the areas of emergency response, organizational modelling, and behavioural dynamics [14-16]. In LEADSTO, one can specify both qualitative and quantitative aspects of complex socio-technical systems using the Temporal Trace Language (TTL). TTL has the semantics of order-sorted predicate logic (Manzano 1996) that is defined by a rich ontological base including sorts, predicates, and variables. Relationships between system components can be expressed in a straightforward way. This provides wide means for the conceptualization of the airline disruption management domain. In addition, TTL is an extension of the standard multi-sorted predicate logic in the sense that it has explicit facilities to represent dynamic (temporal) properties of systems. Such a temporal expressivity is particularly important for the representation and analysis of processes over time.

The LEADSTO format is defined as follows: let α and β be predicates, and e, f, g, h be non-negative real numbers. Then $\alpha \rightarrow_{e,f,g,h} \beta$ means: If predicate α holds for a certain time interval with duration g , then after some delay (between e and f) predicate β will hold for a certain time interval of length h . An example of a dynamic property in the LEADSTO format is $\alpha \rightarrow_{0.25,1,1,1.5} \beta$ where α represents the predicate *Communication_from_to(external_world, AE, observe, leak)* and β represents the predicate *Communication_from_to(AE,SS,inform,pump_change_required)*. This property expresses the fact that, if the airport engineer AE observes that there is a hydraulic leak during 1 time unit, then after a delay between 0.25 and 1 time unit, AE will inform the station supervisor SS about the problem during 1.5 time units. By executing this rule, a trace of predicates holding true or false can be generated and visualized. The time units in this case study are in minutes. For the temporal parameters, the following assumptions were made: Solving an aircraft/crew problem takes 8 minutes; synchronization between IT systems takes 0.1 minutes following an update; an observation-belief-action cycle takes 1.5 minutes. These assumptions were based on observations made at two AOC centres in Europe.

4. Model Verification

For the identified policies P1-P4, it is important to ensure that some required dynamic properties hold. Such properties may for example represent system requirements, desired performance characteristics, absence of deadlocks and other forbidden states. To verify the identified policies in the context of the case study, automated model verification tools can be used, such as TTL Checker [17]. The dynamic properties in TTL Checker need to be specified in Temporal Trace Language (TTL) [18]. LEADSTO is an executable sublanguage of TTL. TTL is also a variant of order-sorted predicate logic with the possibility to specify and reason about time. By using TTL Checker, dynamic properties in TTL could be checked automatically on simulation traces automatically generated by LEADSTO software based on agent-based model specifications.

- **Policy P1 - Property 1:** If the Station Supervisor (SS) believes that there is a mechanical failure, then within 5 minutes the Airline Operations Supervisor (AOS) also believes there is a mechanical failure. Formally:

∀t at(Belief(SS, Disruption(mechanical_failure, LHB, CDG)),t) &
∃t' ∃c:MSG_TYPE t'<t &
at(Communicate_from_to(AE, SS, c, Disruption(mechanical_failure, LHB, CDG)), t')
⇒ ∃t'' t'' > t & t'' ≤ t+10 & at(Observation (CCo, Disruption(mechanical_failure, LHB, CDG)), t'')

It is important for this property to hold because under policy P1, AOS must accept maintenance information content and act on it without challenging the information.

- **Policy P1 - Property 2:** If SS believes that maintenance information reported to him by the Airport Engineer (AE) is true, then this information should be noticed by the Crew Controller (CCo) within 10 minutes. Formally:

∀t ∀x:AIRCRAFT ∀y:AIRPORT at(Belief(SS, Disruption(mechanical_failure, x, y)),t) ⇒ ∃t' t' > t & t' ≤ t+5
& at(Belief(AOS, Disruption(mechanical_failure, x, y)),t')

It is important that this property holds to ensure proper synchronization between the Aircraft Movement System (used by the SS) the Crew Tracking System (monitored by the CCo).

- **Policy P2 - Property 1:** If SS believes that there is a mechanical failure, then AOS should call the AE within 5 minutes the to verify the information. Formally:

∀t ∀x:AIRCRAFT ∀y:AIRPORT at(Belief(SS, Disruption(mechanical_failure, x, y)),t) ⇒ ∃t' t' > t & t' ≤ t+5 & at(Communicate_from_to(AOS, AE, ask, Check_disruption(mechanical_failure, x, y)), t')

It is important for this property to hold because under policy P2, the AOS must challenge information about a maitnenance situation, and query the information source.

- **Policy P2 - Property 2:** if CCO believes there is a crew problem, then, within 2 minutes, CCO should ask the Flight Crew (FC) to extend their crew duty time. Formally:

∀t ∀x:CREW_PROBLEM at(Belief(CCO, Crew_problem(x)),t) ⇒ ∃t' t' > t & t' ≤ t+2 & at(Communicate_from_to(CCO, FC, ask, extend_crew_hours), t')

It is important for this property to hold because under policy P2, when the CCo is facing with a crew problem, he must challenge crew limits and seek extensions to crew duty time, for instance through negotiating with the Flight Crew (FC).

- **Policy P3 - Property 1:** If SS believes that there is a mechanical failure, then within 5 minutes, AOS should organize a conference call with AE and Maintenance Watch Engineer to recheck information. Formally:

∀t ∀x:AIRCRAFT ∀y:AIRPORT at(Belief(SS, Disruption(mechanical_failure, x, y)),t) ⇒ ∃t' t' > t & t' ≤ t+5 & at(Start_conf_call(AOS, Disruption(mechanical_failure, x, y), AE, MWE), t')

It is important for this property to hold because under policy P3, the AOS must seek alternative information and recheck information source and reliability, e.g., through seeking a second opinion from the MWE.

- **Policy P3 - Property 2:** if MWE believes there is a mechanical failure, then within 5 minutes, the CCo should notice the aircraft solution on the CTS. Formally:

∀t ∀x:AIRCRAFT ∀y:AIRPORT at(Belief(MWE, Disruption(mechanical_failure, x, y)),t) ⇒ ∃t' t' > t & t' ≤ t+5 & ∃s:AIRCRAFT_SOLUTION at(Observation(CCo, Aircraft_problem(s))), t')

This property is checked to verify a proper synchronization between the CTS (monitored by the CCo) and the AMS (used by the ACo). After hearing the confirmation from MWE in the conference call, the ACo directly reports the aircraft solution through AMS.

- **Policy P4 - Property 1:** Before announcing an integrated disruption management solution to ACo and CCo, the AOS should have noticed the solutions to the aircraft problem and crew problem reported on the AMS by the ACo and CCo respectively. Formally:

$\forall$t1,t2 $\forall$x:AIRCRAFT_PROBLEM $\forall$y:CREW_PROBLEM at(Observation(AOS, Aircraft_problem(x))), t1) & at(Observation(AOS, Crew_problem(y))), t2) & $\exists$t3,t4 t3 < t1 & t4 < t2 & at(Communicate_from_to(AMS, ACo, inform, Aircraft_problem(x)), t3) & at(Communicate_from_to(AMS, CCo, inform, Crew_problem(y)), t4) $\Rightarrow$ $\exists$t',t'' t' > t1 & t' > t2 & t'' > t1 & t'' > t2 $\exists$s: INTEGRATED_SOLUTION & at(Communicate_from_to(AOS, ACo, inform, integrated_solution(s)),t') & at(Communicate_from_to(AOS, CCo, inform, integrated_solution(s)), t'')

This property is checked to ensure that the specialist agents provide solutions to the AOS before he announces offers to solve the problem.

- **Policy P4 - Property 2:** If AOS announces an integrated disruption management solution, he should obtain within 5 minutes the vote results (approval/rejection) from both ACo and CCo on the AMS. Formally:

$\forall$t1,t2 $\forall$s: INTEGRATED_SOLUTION & at(Communicate_from_to(AOS, ACo, inform, integrated_solution(s)),t1) & at(Communicate_from_to(AOS, CCo, inform, integrated_solution(s)), t2) $\Rightarrow$ $\exists$t', t'', t3, t4 t' < t1+5 & t'' < t2+5 & t3 < t1+5 & t4 < t2+5 $\exists$z1,z2: VOTE_RESULT & at(Communicate_from_to(AOS, ACo, reply, vote_for(s, z1)), t') & at(Communicate_from_to (AOS, CCo, reply, vote_for(s, z2)), t'') & at(Observation(AOS, vote_for(s, z2)), t3) & at(Observation(AOS, vote_for(s, z2)), t4)

This property is checked to ensure that the AOS obtain the vote results after he announces a solution to solve the problem. All the identified properties were verified as true for the developed agent-based model.

5. Simulation Results

The four AOC policies introduced in Section 3 have been implemented and simulated in the presented agent-based model. For each of these four policies various results have been collected such as related to aircraft, crew, passengers, and the minimum time needed to manage the disruption. Table 2 presents the simulation results obtained for the four AOC policies.
The outcome of policy P3 concurs with the best solution identified by the expert panel. However, the outcomes of P1 and P2 are significantly worse, and the outcome of P4 even outperforms the expert panel result. In order to understand the background of these differences, the agent-based simulation results have carefully been analyzed. Under policies P1 and P2, AOC operators make decisions based on limited coordination, as a result of which the disruption considered is not efficiently managed. The aircraft mechanical problem was eventually fixed, however the flight was cancelled. As a result, the 420 passengers were accommodated in hotels (i.e. greatly inconvenienced). This unfavorable outcome can be explained as a result of the possible actions identified by the crew controller i.e. “await crew from inbound aircraft” and “seek extensions to crew duty time.” Crew controllers mainly considered crew sign-on time and duty time limitations and tried to work within these constraints. In this scenario, none of the possible actions solves the crew problem.

Under policy P3, AOC controllers consider complex crewing alternatives and can either choose to deadhead replacement crew from another airport or use crew from other aircraft. Therefore, under P3 the decision was made to reroute the flight via BOM and

fly-in a replacement crew from PCF into BOM. Here, both the delayed crew and replacement crew were able to operate in one tour of crew duty time. This solution was chosen instead of using crew from other aircraft based on the transcript data from the expert panel simulations in [7]. In comparison to policies P1 and P2, policy P3 is much better from both the airline and the passenger's perspectives. Regarding the minimum time required for managing the disruption policy, P3 takes more time than P1 and P2.
Under policy P4, it was assumed that AOC agents make level 3 decisions similar to P3. Under P3, the crew controller agent can either consider various crew deadheading possibilities or user alternative crew from other aircraft. If the latter policy is followed, policy P4 is able to identify a possibility that had not been identified by the expert panel. The flight crew that had landed the aircraft at CDG had received sufficient rest to fly the delayed aircraft directly to PCF instead of enjoying their scheduled day-off in Paris. Passengers had a minimum delay compared to the previous policies (P1-P3) as they only had to wait for the aircraft to be fixed. If the assumption regarding AOC agents under policy P4 was changed to decision level 1 or 2 similar to P1 and P2, the crew problem would not have been resolved.

Table 2. Simulation Results. **P:** Policy; **FL:** Flight; **MP:** Mechanical Problem; **CP:** Crew Problem; **PAX:** Passengers Problem; **MDT:** Minimum Disruption Management Time; **OC:** Operation Costs; **TL:** Time Lost for passengers.

P	FL	MP	CP	PAX	MDT	OC	PAXC	TL
P1	Cancelled	Fixed	Not resolved	Distressed	26 min	326kEUR	168kEUR	24
P2	Cancelled	Fixed	Not resolved	Distressed	30 min	326kEUR	168kEUR	24
P3	Diverted	Fixed	Resolved	Delayed	33 min	360kEUR	126kEUR	8
P4	Delayed	Fixed	Resolved	Delayed	29 min	326kEUR	0kEUR	3

6. Conclusion

Efficient handling of disruptions by airlines requires advanced coordination and communication means employed by socio-technical teams, in which human operators are supported by intelligent technology. Human operators often demonstrate ingenuity and creativity in problem solving, particularly necessary for handing previously unknown disruptions. By combining these human abilities with the computational power and analysis capabilities of machines, diverse disruptions could be handled efficiently. It is worth noting that the final decision will be made by a human decision maker. The proposed system is supposed to support human controllers by recommending mutually agreeable solutions.

In this paper, we investigated four policies for handling disruptions by a socio-technical team of the AOC, based on agent-based coordination and negotiation models. The policies varied in the level of performance in terms of reasoning and coordination capabilities of the involved agents. The effects of the policies were studied by simulation in the context of a realistic scenario involving a mechanical failure disruption. The results demonstrated that the effectiveness and efficiency of the policies were in direct relation to the capabilities of the agents: richer reasoning and coordination abilities resulted in more efficient and sophisticated solutions, generated within limited time.

Another important contribution of the paper is the formal specification of the policies in an agent-based model using LEADSTO and TTL languages, which enabled simulation and automated verification. Using TTL Checker, a set of formalised TTL properties was verified on the model simulation traces, which were required to hold for the operational scenario under consideration.

In the future work, the identified policies will be applied and evaluated in other operational scenarios, including ones with cascading disturbances. Furthermore, the properties of the policies and the associated coordination protocols will be analysed more extensively for their efficiency and robustness.

Based on the obtained preliminary results, we can conclude that the proposed approach could be a promising way forward for modelling, designing, and analysing collaborative decision-making mechanisms for handling disruptions by socio-technical teams in the air transportation system. As a future work, we are planning to consider more sophisticated decision models such as “Markov decision process”. Agents can learn what to vote (accept/reject) based on their previous experience and feedback given by the framework.

References

1. Castro, A. J. M. (2013). A Distributed Approach to Integrated and Dynamic Disruption Management in Airline Operations Control. PhD thesis, University of Porto.
2. Aydoğan, R., Festen, D., Hindriks, K. V., & Jonker, C. M. (2017). Alternating offers protocols for multilateral negotiation. In Modern Approaches to Agent-based Complex Automated Negotiation, pp. 153-167. Springer, Cham.
3. Klein, M., Faratin, P., Sayama, H. , and Bar-Yam, Y. (2003). Protocols for negotiating complex contracts. IEEE Intelligent Systems. Vol. 18, No. 6, pp. 32-38.
4. Aydoğan, R., Hindriks, K. V., & Jonker, C. M. (2014). Multilateral mediated negotiation protocols with feedback. In Novel Insights in Agent-based Complex Automated Negotiation (pp. 43-59). Springer, Tokyo.
5. Sanchez-Anguix, V., Aydoğan, R., Julian, V., & Jonker, C. M. (2014). Unanimously acceptable agreements for negotiation teams in unpredictable domains. Electronic Commerce Research and Applications, 13(4), 243-265.
6. Bruce, P. J. (2011-a). Understanding Decision-Making Processes in Airline Operations Control, Ashgate Publishing Company, Farnham, UK.
7. Bruce, P. J. (2011-a). Understanding Decision-Making Processes in Airline Operations Control, Ashgate Publishing Company, Farnham, UK.
8. Treur, J. (2016). Dynamic modeling based on a temporal-causal network modeling approach. Biol.Inspired Cogn. Archit.16, 131–168
9. Bouarfa, S., Blom, H.A.P., Curran, R. (2016). Agent-Based Modelling and Simulation of Coordination by Airline Operations Control. IEEE Transactions on Emerging Topics in Computing, Volume:PP, Issue:99, February.DOI 10.1109/TETC.2015.2439633.
10. Bruce, P. J. (2011-b, January). Decision-making in airline operations: the importance of identifying decision considerations. Internal Journal of Aviation Management. Vol. 1, Nos. 1/2. pp 89-104. Available: http://inderscience.metapress.com/content/m34750h347u85401/
11. Machado, N. (2010). Impact of the Organizational Structure on Operations Management. Msc thesis, University of Porto.
12. LEADSTO software (2019). Available for download at: http://www.cs.vu.nl/~wai/TTL/
13. Bosse, T. Jonker, C. M. van der Meij, L. Treur, J. (2007, June). A language and environment for analysis of dynamics by simulation. International Journal on Artificial Intelligence Tools. Vol. 16, issue 03, pp. 435-464. Available http://www.worldscientific.com/doi/abs/10.1142/S0218213007003357
14. van den Broek, E.L., Jonker, C. M., Sharpanskykh, A., Treur, J., Yolum, P., 2006. Formal modelling and analysis of organizations,” in Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems, volume 3913, O. Boissier, J. Padget, V. Dignum, G. Lindermann, E. Matson, S. Ossowski, J. S. Sichman, J. V. Salceda, Eds. Springer Berlin Heidelberg, pp18-34.
15. Bosse, T., Jonker, C. M., Treur, J., 2007b. On the use of organisation modelling techniques to address biological organisation. Multi-agent and Grid Systems. Vol. 3, Nr. 2, pp 199-223, June.

16. Sharpanskykh, A., Treur, J., 2006. Modeling of agent behavior using behavioral specifications. Vrije Universiteit Amsterdam. The Netherlands. Technical Report 06-02ASRAI, February.
17. Bosse, T., Jonker, C. M., Van der Meij, L., Sharpanskykh, A., & Treur, J. (2009). Specification and verification of dynamics in agent models. International Journal of Cooperative Information Systems, 18(01), 167-193.
18. Sharpanskykh, A., & Treur, J. (2010). A temporal trace language for formal modelling and analysis of agent systems. In Specification and verification of multi-agent systems (pp. 317-352). Springer, Boston, MA.

Appendix A – Ontology

Agents captured in the agent-based model

- **AOS:** Airline Operations Supervisor
- **ACo:** Aircraft Controller
- **CCo:** Crew Controller
- **MS:** Maintenance Services
- **AE:** Airport Engineer
- **SS:** Station Supervisor
- **AMS:** Aircraft Movement System
- **CTS:** Crew Tracking System
- **FC:** Flight Crew

Domain Ontology – Logical Predicates: Internal states and communication activities of the agents

- *Observation (A,I):* Agent A observes information I from the world
- *Belief (A, I):* Agent A believes that information element I is true in the world
- *Incoming_communication(A, C, I):* Agent A receives message type C with content I
- *Communicate_from_to(A, B, C, I):* Agent A communicates to agent B message type C with content I

Other predicates used in the considered scenario

- *Disruption(DT,AC,AP):* Describes a disruption of type DT, concerning aircraft with registration code AC, at airport AP
- *Query(A, B, I):* Query by agent A to agent B about Information I
- *Query_disruption(DT,AC,AP):* Query about disruption (DT,AC,AP)
- *Flight_crew(AC):* Flight crew of aircraft with registration code AC
- *Reserve_aircraft(amount):* To denote the number of reserve aircraft available
- *Aircraft_available_for_swap(amount):* Number of aircraft within the same type available for swap
- *Crew_inbound_aircraft(amount):* To denote the number of crew available from inbound flights
- *Aircraft_problem(AS):* Proposed solution to the aircraft problem
- *Crew_problem(CS):* Proposed solution to the crew problem
- *extend_crew_hours(y/n):* Possibility to extend crew hours (yes/no)
- *Check_disruption(DT,AC,AP):* Checking information reliability about a disruption of type DT, concerning aircraft with registration code AC, at airport AP
- *Disruption(t/f):* Confirmation whether there is a disruption or not by local agents
- *Conf_call(O,D,A,B,…,N):* Conference call organized by agent O about a certain disruption D with N+1 participants in alphabetical order.
- *Early_serviceability(AC,DT,AP):* Request for earlier serviceability for aircraft AC with problem DT at airport AP
- *early_serviceability(AC,DT,AP,y/n):* Possibility for earlier serviceability of aircraft AC with problem DT at airport AP (yes/no)
- *Start_conf_call(O,D,A,B,…,N):* Start of conference call
- *End_conf_call(O,D,A,B,…,N):* End of conference call
- *Transmit_construal(DT,AC,AP,RT,F):* Transmitting construal of the meaning of the signal back to the sender
- *Construal(DT,AC,AP,RT,F):* Content of a signal being sent
- *Exit_reporting(DT,AC,AP,RT,F):* Signal of exiting a coordination phase (reporting) about a certain type of disruption with various attributes
- *Start_aircraft_problem_solving(DT,AC,AP,RT,F):* Signal of starting a new coordination phase (solving crew problem) for a certain type of disruption with various attributes
- *Renew_compact(AS):* Renewing the basic compact about a particular information element.
- *crew_day_off(AP,y/n):* Possibility to use crew with day off at airport AP
- *Verify_disruption(DT,AC,AP,RT,F):* Verifiying a certain disruption with different attributes
- *integrated_solution(s):* Integrated disruption management solution s
- *vote_for(s, z):* Vote result for s (approval or rejection)

Domain Ontology – Sorts and elements

- DISRUPTION_TYPE {mechanical_failure}
- AGENT {AE,SS,AMS,AOS,ACo,CTS,CCo,FC,MWE}
- MESSAGE{fix_aircraft,hydaulikc_leak,_no_reserve_aircraft,no_crew_available,delayed_crew, crew_hours,extend_crew_hours,no,yes,true,none, exit_AOC_disruption_management, start_crew_problem_solving}
- MESSAGE_TYPE{inform,request,permit,ask,declare,report,synchronize,confirm,answer,negotiate, check,consult,transmit,verify,announce}
- AIRPORT{CDG, BOM, PCF}
- AIRCRAFT{LHB}
- AIRCRAFT_SOLUTION{cancel_flight, fix_aircraft, no_reserve_aircraft,no_aircraft_available,pump_change}
- CREW_SOLUTION {no_crew_available, reroute_via_BOM, use_day_off_crew}
- INTEGRATED_SOLUTION {aircraft_solution, crew_solution, pax_solution}
- VOTE_RESULT{approval, rejection}
- REPAIR_TIME{three_hours}