Event-based Optimization of Air-to-Air Business Processes

Leon Sterling[1] and Kuldar Taveter[2]

[1] The University of Melbourne, Department of Computer Science and Software Engineering, Victoria 3010, Australia
leonss@unimelb.edu.au
[2] Tallinn University of Technology, Department of Informatics, Tallinn 12618, Estonia
kuldar.taveter@ttu.ee

Abstract

This paper describes the results from an industry-oriented research project conducted by the authors at the University of Melbourne. The project was aimed at working out event-based models and 'business process'-oriented simulations of sequential arrivals and departures at airports which we refer to as *air-to-air business processes*. This paper initially describes the problems to be solved that are related to increasingly frequent bottleneck situations at airports. Following, the paper presents a vision for achieving global optimization of air-to-air business processes that would integrate business processes and software systems of different stakeholders. As the first step towards achieving the vision, the paper proposes agent-oriented modelling and simulation of the problem domain. The paper then describes in more detail how event-based models of integrated business processes involving several stakeholders can be created. After that, the paper presents and explains the business process model of arrivals at airports. Finally, the paper proposes an avenue for transforming business process models into the constructs of a simulation environment enabling experimentation with global optimization.

Introduction

It is generally known [1] that in hub airports of the world, overall performance degrades disproportionately in cases of over-demand. A particular problem is punctuality as experienced by passengers, resulting in the following four deficiencies: (1) remaining capacities are not used in full; (2) connectivity is jeopardized; (3) air and ground operations cannot be handled economically; (4) people are exposed to high workload.

In addition to punctuality, other important Quality of Service (QoS) criteria for airports are efficiency of aircraft manoeuvrings, efficiency of airline operations, efficient usage of airport facilities, and optimality of air traffic control clearances, such as start-up and take-off clearances.

The major stakeholders in air-to-air business processes are air traffic control (ATC), airports, and airlines. Each of them is supported by various information systems. The information systems used by the stakeholders perform optimizations according to different criteria. For example, the Departure and Arrival Traffic Managers used by ATC optimize the usage of takeoff and landing runways and off-block and taxi times of aircrafts. The Resource Allocation Manager used by airports optimizes assignment and utilization of stands and gates, and the usage of refuelling, catering, and turnaround resources. In addition, the Environment Manager interacts with both systems to minimize the noise and air quality impact of operations, while the Information Manager provides both real-time and historical operations data for planning, billing, marketing, reporting, and public relations. Airlines have their own internal information systems.

One or more tools helping the stakeholder to achieve maximally efficient usage of its resources thus support each stakeholder. The functioning of the tools is based on in-built criteria for local optimization. We thus have several islands of automation. What would be the criteria for global optimization and how could they be achieved?

One could envisage a kind of 'super system' imposed over various systems that would then achieve a globally optimal solution for each situation at hand. However, because of the dynamicity of the problem domain and the sheer complexity of optimization problems faced by individual stakeholders, this solution is not likely to be efficient.

For this reason, the project team decided to take a novel approach where a global optimization emerges when each stakeholder receives the right information at the right time and immediately acts based on that information. This can be achieved by a combination of a new business model and novel technological solutions.

The proposed solution interconnects existing systems at an airport with the aim to improve airport operations by ensuring that airport partners (for example, airports, airlines, and ATC) receive relevant and accurate information on time and act upon that information. We term this solution *Advanced Collaborative Decision Making* (A-CDM).

The goal of the industry-oriented research project overviewed in this paper was to develop and simulate an integrated event-based business model for achieving A-CDM and to propose technological support for the business

---

model. Because of the safety-critical nature of the problem domain, modelling and simulation are necessary before implementing any solution. The new business model, when implemented, would result in increased throughput of planes through airports and less time on the ground for the airlines. Simulation is also required for demonstrating potential benefits from turning the model into a real socio-technical system. This paper is to describe the event-based business model and its simulation environment.

Agent-Oriented Modelling

Major stakeholders in the problem domain of air traffic management, ATC, airports, and airlines, are both geographically and conceptually distributed. In our previous research projects [2-5] dealing with distributed domains, we have become convinced of the advantages of agent-oriented modelling for designing and implementing systems, including simulation systems, for such domains. In agent-oriented modelling, distributed systems are conceptualised and modelled as consisting of human and man-made agents and their environments. We define an *agent* as an entity that can act in the environment, perceive events, and reason. Events that an agent perceives by, for instance, receiving a message from another agent or sensing a temperature change, are caused by agents or other entities in the environment. Conversely, through acting, which includes sending messages to other agents, agents can affect entities in the environment. By reasoning, we mean drawing inferences appropriate to the situation. We claim that agent-oriented modelling is a more natural way of conceptualizing the world compared to object-oriented modelling, which essentially views the world as consisting of a system to be designed and its users.

Agent-oriented modelling starts with modelling the *goals* set for the system and the *roles* required for achieving the goals [13, 14]. Only after the modelling of goals and roles can we move to the modelling of business processes realizing those goals. In our problem domain, each stakeholder, ATC, an airport, and an airline / ground handler, has specific goals to be obtained. The goal of the highest level is to allocate resources related to a particular flight plan. This goal reflects the overall purpose of the optimization system to be created. The goal to allocate resources is connected to the stakeholder role Passenger whose enacting human agents are the ultimate consumers of the resources. This goal has three attached *quality goals*, namely to achieve maximal safety, maximal airport efficiency and minimal environmental nuisance. The goal to allocate resources has been decomposed into three sub-goals: to allocate ATC resources, airport resources and the resources of the airline / ground handler. The sub-goals can be expanded, as in our experience, hierarchical chunking greatly aids understanding. Because of the focus of this paper, we have chosen not to give the goal models here but rather refer the reader to the source [6].

Event-based Business Process Model

Methodology and Notation

The purpose of the event-based business process model is to figure out *when* the goals are to be achieved and *how* they are to be achieved. As to the first part of the question, we have to model *events* that determine when the goals are to be achieved. With respect to the second part of the question, we model each goal as to be achieved by an *activity* of some type which may recursively consist of sub-activities.

For modelling events and activities, we employ the Radical Agent-Oriented Process / Agent-Object-Relationship (RAP/AOR) methodology of software engineering and simulation that was introduced in [7] and is based on [8] and [9].

For business process modelling, we first map roles to *agent types*. The RAP/AOR methodology understands an agent as any active entity that can perceive events, perform actions, and interact. Agents manipulate *objects*, which serve to represent information. The clues to choosing agent types to be modelled are provided by roles. In general, the decision about modelling a particular entity type as an agent or object type depends on whether entities of this type consume and process information or serve to represent information. For example, in the context of A-CDM we model Aircraft as an object type, although in a different context, for example, within the so-called “free flight”, it could have been modelled as the type of an agent related to a particular aircraft that consumes and processes information about situations at source and destination airports and en route situations.

We distinguish between types of *institutional agents*, such as organizations and organizational units like ATC and airport, *human agents*, and *manmade agents*, such as software agents or robots.

An *AOR diagram* specified by Figure 1 is an agent-oriented extension of a UML activity diagram [10] that enables the representation in a single diagram of the types and instances of institutional, human and manmade agents of a problem domain, together with their internal agent types and instances and their beliefs about instances of private and external (that is, shared with other agents) object types. There may be attributes and/or predicates defined for an object type and relationships (associations) among agent and/or object types. A predicate, which is visualized as depicted in Figure 1, may take parameters. Differently from UML activity diagrams, AOR diagrams make use of rules for starting and sequencing activities that makes AOR diagrams directly executable according to the operational semantics presented in [9]. Additionally, it has been shown in [9] that AOR diagrams can be straightforwardly turned into the constructs of the agent software platform JADE [11], which can be used for simulating business processes and implementing A-CDM.

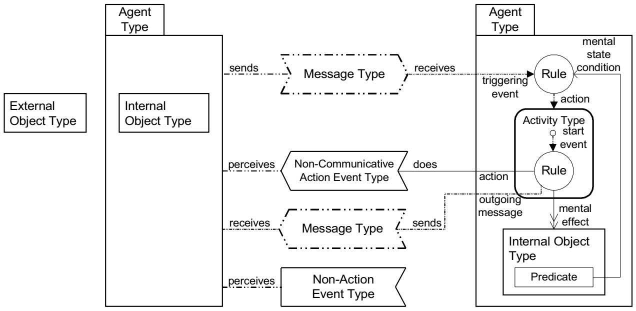

Figure 1. The modelling elements of AOR diagrams.

Figure 1 shows that the graphical notation of AORML distinguishes between an *action event* (an event that is created through the action of an agent, such as a physical move performed by a human) *type* and a *non-action event* (for example, a temporal event or an event created by a natural force) *type*. The graphical notation of AORML further distinguishes between a *communicative* action event (or *message*) type and a *non-communicative* (physical) action event type like providing another agent with a commodity. This distinction is important for adequate reflection of events occurring in the world.

Non-action events also include exogenous events. An *exogenous event* is a kind of event whose creating agent we are not interested in. To model exogenous events, we have to determine the boundary of the system to be simulated or implemented. This is necessary for distinguishing between exogenous events originating outside the system and internal events. For example, since in the context of A-CDM we are not interested in the precise way how approaching aircraft are perceived by radar systems and how the related data is transmitted to the airport's ATC, we summarize perceptions of aircrafts as exogenous events of the types Aircraft is in FIR[1], Aircraft has landed, Aircraft is in block, etc. Each such exogenous event carries along related data about the type of aircraft, its exact position on a given time moment, and so on. In agent-oriented simulation systems, exogenous events are generated by an environment simulator. Similarly, a modeller may also need to represent an action performed by an agent on an environment without indicating a specific agent affected by the action. Examples of exogenous events and actions on an environment will be included in the model of the sequential arrival business process type to be explained in the next section.

The most important elements for modelling agent behaviours by AORML are *rules*. As is shown in Figure 1, a rule is visualized as a circle with incoming and outgoing arrows drawn within the rectangle of the agent type whose behaviour it represents. Each rule has exactly one incoming arrow with a solid arrowhead that specifies the *triggering event* type. In addition, there may be ordinary incoming arrows representing *state conditions* (referring to corresponding instances of object types or to the predicates defined for them). There are two kinds of outgoing arrows for specifying the performance of *epistemic*, *physical,* and *communicative actions*. An outgoing arrow with a double arrowhead denotes an epistemic action (changing the agent's beliefs). An outgoing connector to an action event type denotes the performance of a physical or communicative action, i.e., sending a message to another agent, of that type.

Rules start activities. Each activity belongs to some *activity type*, which we define as a prototypical job function in an organization that specifies a particular way of doing something by performing one or more elementary epistemic, physical, and communicative actions by an agent.

There are activity border events of the *start-of-activity* and *end-of-activity* types implicitly associated with the beginning and end of each activity. As Figure 1 reflects, the *start-of-activity* event type is graphically represented by an empty circle with the outgoing arrow to the symbol of the sub-activity type or internal rule.

The distinguishing feature of AOR diagrams is *integrated modelling*, compared, for example, with diagram types included by UML [10]. Integrated modelling is representing information, interaction, and behaviour models in the same diagram. This facilitates complex event processing where complex events are identified based on event perceptions by agents and their beliefs. An example of representing complex event processing by AOR diagrams will be given in the section "Complex Event Processing".

Arrival Process

We now explain the model of the sequential arrival business process type, which is associated with the inbound leg of a flight plan. This business process type is depicted in Figure 2. A similar model has been created for a departure business process type. We model the types of air-to-air business processes by specifying for each goal represented in a goal model an activity type for realizing that goal, the type of an event determining when the goal is to be realized, and a rule associating the event type with the type of an activity to be started by it.

The order of performing activities in business processes can be traced by the numbers attached to rules. The events that determine when activities are to be started are

---

[1] FIR: Flight Information Region – the area of responsibility of the local ATC

milestone events defined in [12].

Since Figure 2 represents a business process model, as opposed to the model of a software system, activity types represented in it are not fully specified. This means that we model the activities to be performed and when they are to be performed, but for each individual activity type, we do not specify atomic actions that its performing entails. For example, for the activity type Update ELDT[2], EIBT[3] and TOBT[4]" of Local ATC modelled in Figure 2 we do not specify *how* the attributes of a flight plan actually are to be updated. This is done by detailed design which we do not overview in this paper because of space limitations. Note that, as has been shown in [9], even incompletely specified activity diagrams are executable by computers. Taking advantage of this, we can develop a simulation environment by gradually transforming incompletely specified activity types into completely specified ones.

A milestone event of taking off from outstation is represented in the model of an arrival process in Figure 2 as an exogenous event of the type AircraftIsAirborne. This modelling construct abstracts away from the actual mechanism and channel of passing information about the flight being taken off from the outstation to the Airline/GH. An event of the type AircraftIsAirborne starts the first two sequential activities of an arrival process, which are of the types Update ELDT, EIBT and TOBT and Inform ATC. The starting of these activities is modelled in Figure 2 with the help of rule R1. The same rule also models forwarding the information on the status change of the flight plan to the Airport. Within the Airport, starting an activity of the type Allocate gate/stand is modelled by rule R2.

The next milestone event, destination FIR entry, has been modelled as an exogenous event of the type AircraftIsInFIR. Rule R3 models starting an activity of the type Process arrival by the Local ATC in response to registering an event of the type AircraftIsInFIR.

To express that an event of certain type carries new information about attribute values of the corresponding flight plan, notation for the corresponding event type has been complemented with the object type name FlightPlan, as in Figure 2 for the exogenous event types AircraftIsAirborne and AircraftIsInFIR.

An activity of the type Process arrival realizes a goal to allocate ATC resources. An activity of this type consists of a number of sequential sub-activities. The meaning of the rules R6, R7, and R8 located between sub-activity types in Figure 2 is that after the end of an activity of the preceding type, an agent will be waiting for a certain milestone event to occur before proceeding with an activity of the succeeding type. For instance, after completing an activity of the type Determine taxi-in route, signalled by the corresponding *end-of-activity* event, the institutional agent of the type Local ATC will be waiting for a time event of the type inbound.LandingTime – t. This time event corresponds to the milestone event of entering the Final Approach phase at the destination airport, where t is the time parameter defined by ATC. In a similar way, rules R7 and R8 model waiting for and registering the occurrences of the milestone events of landing and arriving in block.

For the institutional agent type Airline/GH, the types of arrival activities realize the goal to allocate the resources of the airline / ground handler. Rule R4 within Airline/GH models the passing of information on the milestone event of destination FIR entry to the Airport. The same rule also models starting of an activity of the type Refine flight servicing plans. Rule R10 within the same agent type models the registering of the milestone event of the start of ground handling and the triggered by it commencing of an activity of the type Record AGHT[5], update TOBT and TTOT[6].

For the institutional agent type Airport, the goal to allocate airport resources is achieved by performing activities of the types Allocate gate/stand and Refine gate/stand allocation. Starting activities of these types is modelled by means of rules R2 and R5.

Complex Event Processing. AOR diagrams also lend themselves to the modelling and simulation of complex events, which consider different atomic events. In Figure 14, rules R11 and R12 specify that if at the time event inbound.InBlockTime + aircraft.MinimumTurnRoundTime + t (that is, t time units after the estimated or actual in-block time of the FlightPlan's inbound leg increased by the minimum turn-around time of the aircraft in question) the current time (which is inbound.InBlockTime + aircraft.MinimumTurnRoundTime + t at that moment) is greater than the estimated off-block-time of the FlightPlan's outbound leg, the Minimum Turn-Around Alarm is raised. This alarm is modelled in Figure 1 as an action on an environment. Note that t is the time parameter defined by the airline / ground handler. This example demonstrates how complex events are identified based on atomic time events and past events that have resulted in beliefs by the agent about the current flight status.

---

[2] ELDT: Estimated Landing Time – the estimated time when an aircraft will touchdown on the runway

[3] EIBT: Estimated In-Block Time – the estimated time when an aircraft will arrive in block

[4] TOBT: Target Off-Block Time – the time that an aircraft operator / ground handler estimates that an aircraft will be ready, all doors closed, boarding bridge removed, push back vehicle present, ready to taxi immediately after receiving ATC clearance

[5] AGHT: Actual Ground Handling Start Time

[6] TTOT: Target Take Off Time

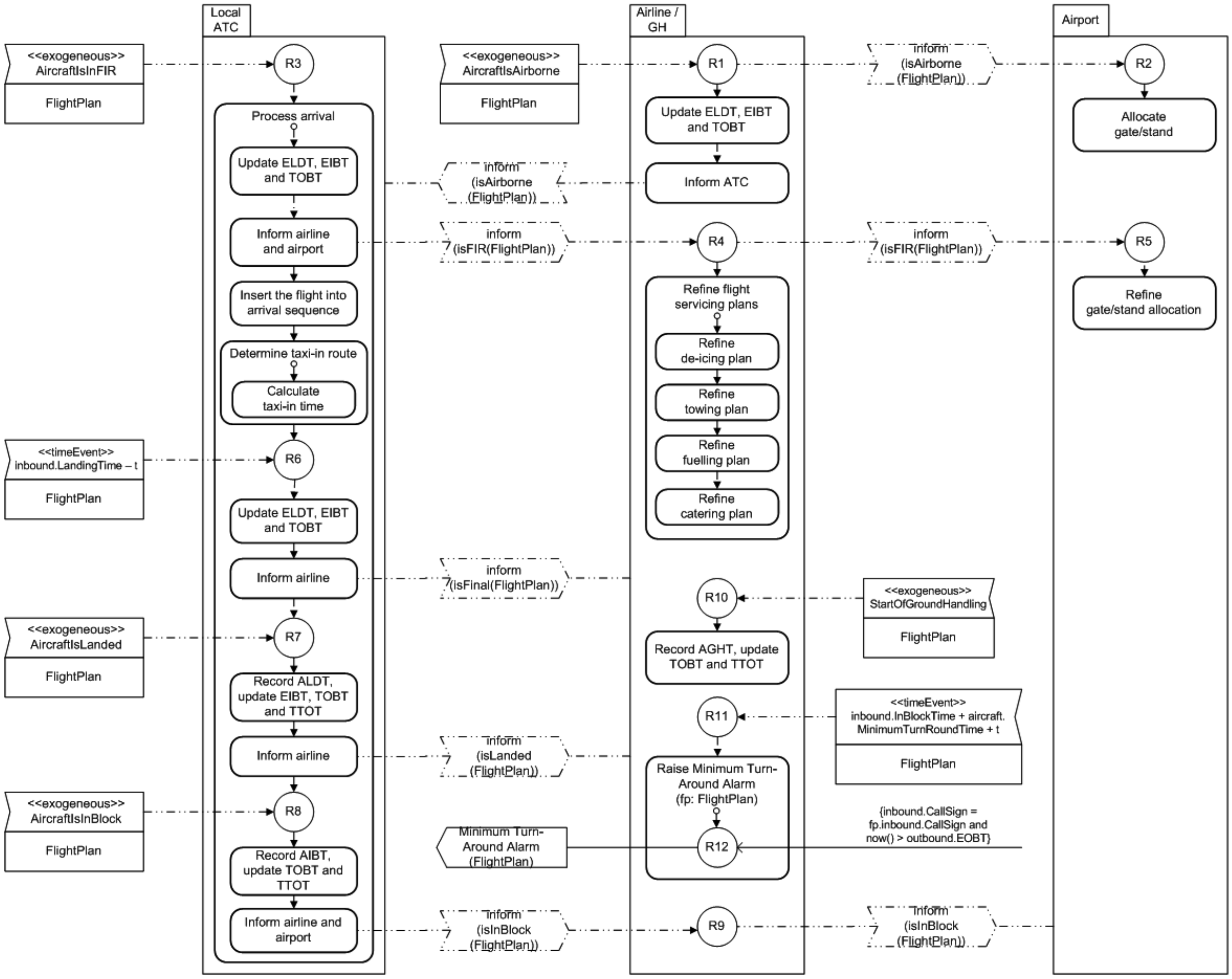

Figure 2. The business process type of the flight plan's inbound leg.

Simulation

Our intention was to perform software simulations with the models of air-to-air business processes. In the simulation environment that we are implementing, the environment simulator generates, from historical event data, exogenous events of the types appearing in the models of air-to-air business processes, such as the model represented in Figure 2. Since the available historical event data consists of records of events of the types AircraftIsLanded and AircraftIsDeparted only, we are not able to simulate events of all the types modelled in Figure 2. However, events of some types, like AircraftIsInFIR, AircraftIsInBlock and AircraftIsOffBlock, and time events can be interpolated from the recorded events of the types AircraftIsLanded and AircraftIsDeparted. Comparing actual landing and take-off times with scheduled arrival and departure times can obtain additional information.

For simulation, the institutional agents of the types Local ATC, Airline/GH, and Airport are turned into software agents implemented by the constructs of the JADE agent software platform [11]. According to [9], this is straightforward. Agents of JADE can then be executed as components of the simulation environment. Each software agent will be equipped with a graphical user interface.

With the functioning simulation environment, we plan to demonstrate: how to get the right information to the right place at the right time to support decision-making; how to assign gates and stands to flight plans in a manner reactive to the changing circumstances; and how to achieve trade-off optimization of arrival and departure traffic flows.

A snapshot of a simulation environment with two agents – for the ATC and Airport – is depicted in Figure 3. At the present stage, the simulation environment notifies of late arrivals and departures and discovers potential arrival and departure conflicts at gates/stands.

In the current follow-up project, we have substituted a simple environment simulator used so far with a sophisticated simulation environment that can simulate, in addition to incoming and outgoing flights at airports, the causality and safety rules of both the ground and airspace environments.

Time allowing, we plan to demonstrate how even better results could be achieved by involving negotiations between agents that represent different stakeholders. For example, agents representing different airlines and ATC could negotiate the most optimal (in the global sense) departure sequence.

Scheduled Arrival ...	Arrival Time	Scheduled Depart...	Departure Time	Arrival Status	Departure Status
Sat Jul 31 23:20:0...	Sun Aug 01 00:30...	Wed Aug 04 05:0...	Wed Aug 04 05:0...	LATE ARRIVAL	LATE DEPARTURE
Sat Jul 31 22:35:0...	Sun Aug 01 00:34...	Sun Aug 01 05:15...	Sun Aug 01 05:16...	LATE ARRIVAL	LATE DEPARTURE
Sun Aug 01 01:20...	Sun Aug 01 00:51...	Sun Aug 01 05:50...	Sun Aug 01 05:49...		
Sun Aug 01 01:30...	Sun Aug 01 00:58...	Sun Aug 01 06:00...	Sun Aug 01 06:03...		LATE DEPARTURE
Sun Aug 01 01:25...	Sun Aug 01 01:09...	Sun Aug 01 06:45...	Sun Aug 01 06:54...		LATE DEPARTURE
Sun Aug 01 02:15...	Sun Aug 01 01:40...	Sun Aug 01 04:45...	Sun Aug 01 04:46...		LATE DEPARTURE
Sun Aug 01 02:20...	Sun Aug 01 01:47...	Sun Aug 01 05:00...	Sun Aug 01 04:55...		
Sun Aug 01 02:20...	Sun Aug 01 01:52...	Sun Aug 01 05:10...	Sun Aug 01 05:20...		LATE DEPARTURE
Sun Aug 01 02:25...	Sun Aug 01 01:53...	Sun Aug 01 04:00...	Sun Aug 01 03:57...		
Sat Jul 31 21:20:0...	Sun Aug 01 01:56...	Sun Aug 01 05:35...	Sun Aug 01 05:37...	LATE ARRIVAL	LATE DEPARTURE
Sat Jul 31 19:45:0...	Sun Aug 01 01:58...	Sun Aug 01 05:00...	Sun Aug 01 04:55...	LATE ARRIVAL	
Sun Aug 01 01:00...	Sun Aug 01 02:16...	Sun Aug 01 05:00...	Sun Aug 01 04:57...	LATE ARRIVAL	
Sat Jul 31 23:50:0...	Sun Aug 01 02:31...	Sun Aug 01 05:00...	Sun Aug 01 05:02...	LATE ARRIVAL	LATE DEPARTURE
Sun Aug 01 03:05...	Sun Aug 01 02:36...	Sun Aug 01 05:55...	Sun Aug 01 05:56...		LATE DEPARTURE
Sun Aug 01 03:20...	Sun Aug 01 02:42...	Sun Aug 01 05:40...	Sun Aug 01 05:47...		LATE DEPARTURE
Sun Aug 01 03:45...	Sun Aug 01 03:16...	Sun Aug 01 08:00...	Sun Aug 01 08:00...		
Sun Aug 01 03:40...	Sun Aug 01 03:20...	Sun Aug 01 04:55...	Sun Aug 01 05:02...		LATE DEPARTURE

Arrival Stand	On-Block Time	Departure Stand	Off-Block Time	Arrival Status	Departur...
06M	Mon Aug 02 13:01:0...	06M	Mon Aug 02 13:43:0...		
33M	Mon Aug 02 13:03:0...	21M	Mon Aug 02 20:16:0...		
56R	Mon Aug 02 13:09:0...	56R	Mon Aug 02 13:29:0...		
02M	Mon Aug 02 13:13:0...	02M	Mon Aug 02 13:50:0...		
61M	Mon Aug 02 13:15:0...	56R	Mon Aug 02 14:37:0...		
60L	Mon Aug 02 13:19:0...	60L	Mon Aug 02 14:43:0...		
62L	Mon Aug 02 13:22:0...	120M	Mon Aug 02 15:27:0...	POTENTIAL ARRIVAL CONFLICT...	
49L	Mon Aug 02 13:25:0...	130M	Tue Aug 03 13:52:0...		
04M	Mon Aug 02 13:27:0...	04M	Mon Aug 02 14:01:0...		
09M	Mon Aug 02 13:30:0...	09M	Mon Aug 02 13:41:0...		
12M	Mon Aug 02 13:33:0...	12M	Mon Aug 02 14:06:0...		
16M	Mon Aug 02 13:35:0...	16M	Mon Aug 02 14:18:0...	POTENTIAL ARRIVAL CONFLICT...	
53L	Mon Aug 02 13:37:0...	53L	Mon Aug 02 14:24:0...		
62R	Mon Aug 02 13:42:0...	62L	Mon Aug 02 15:52:0...		
49R	Mon Aug 02 13:45:0...	49R	Mon Aug 02 14:48:0...		
25M	Mon Aug 02 13:49:0...	25M	Mon Aug 02 14:27:0...		
05M	Mon Aug 02 13:51:0...	05M	Mon Aug 02 14:14:0...		

Figure 3. The simulation environment with two agents.

Conclusions

The usual practice of handling air traffic around an airport is First Come First Served. This practice should be replaced by the practice of On Time Service. This can be achieved by making sure that all stakeholders receive the right information at the right time and immediately act upon that information. In other words, care should be taken that information about any event, once it happens, is communicated to all the relevant stakeholders and stakeholders adjust their actions based on the information received. In this paper, we described how the first step towards global optimization, also known as A-CDM, can be taken by re-modelling business processes and then implementing a simulation environment for demonstrating the benefits to be received by turning the model into a real socio-technical system.

Acknowledgements

The research project described in this paper was supported by the Australian Research Council Industry Linkage Grants LP0348797 and LP0882140.

References

[1] *Joined initiative with focus on cooperative air traffic management*. (2007). A presentation on the 3rd National Aeronautical Research Program of the Federal Ministry of Economics and Labour of the German Government in 2003 – 2007.

[2] Taveter, K. (2004). From business process modeling to business process automation. In J. Cordeiro and J. Filipe (Eds.), Computer Supported Activity Coordination, Proceedings of the 1st International Workshop on Computer Supported Activity Coordination, CSAC 2004. In conjunction with ICEIS 2004, Porto, Portugal, April (198–210). Porto, Portugal: INSTICC Press.

[3] Taveter, K., and Wagner, G. (2006). Agent-oriented modeling and simulation of distributed manufacturing. In J.-P. Rennard (Ed.), *Handbook of Research on Nature Inspired Computing for Economy and Management* (527–540). Hershey, PA: Idea Group.

[4] Sterling, L., Taveter, K., & The Daedalus Team (2006). Building agent-based appliances with complementary methodologies. In E. Tyugu & T. Yamaguchi (Eds.), *Knowledge-Based Software Engineering, Proceedings of the Seventh Joint Conference on Knowledge-Based Software Engineering* (JCKBSE'06), August 28-31, Tallinn, Estonia (pp. 223-232). Amsterdam, The Netherlands: IOS Press.

[5] Luo, Y., Sterling, L., & Taveter, K. (2007). Modeling a smart music player with a hybrid agent-oriented methodology. In *Proceedings of the 15th IEEE International Requirements Engineering Conference*, October 15-19, Delhi, India (pp. 281-286). Washington DC, USA: IEEE Computer Society.

[6] Sterling, L. & Taveter, K. (2009). Event-based optimization of air-to-air business processes. In B. Long & A. MacDonald (Eds.), *Proceedings of the Industry Track of the Australian Software Engineering Conference 2009* (forthcoming).

[7] Taveter, K. & Wagner, G. (2005). Towards radical agent-oriented software engineering processes based on AOR modelling. In B. Henderson-Sellers & P. Giorgini (Eds.), *Agent-oriented methodologies* (pp. 277-316). Hershey, PA: Idea Group.

[8] Wagner, G. (2003). The agent-object-relationship meta-model: Towards a unified view of state and behavior. *Information Systems, 28*(5), 475-504.

[9] Taveter, K. (2004a). *A multi-perspective methodology for agent-oriented business modelling and simulation*. Ph.D. Thesis, Tallinn University of Technology, Estonia. (ISBN 9985-59-439-8).

[10] OMG (2007). Unified Modeling Language: Superstructure. Version 2.1.1, February 2007. Retrieved June 28, 2007, from http://www.omg.org/cgi-bin/doc?formal/07-02-05.

[11] Bellifemine, F., Caire, G, & Greenwood, D. (2005). *Developing multi-agent systems with JADE*. Chichester, UK: John Wiley & Sons.

[12] *Airport CDM Implementation: The Manual*. (2005). Airports' Council International, Eurocontrol & IATA.

[13] Juan, T., Pearce, A. R., & Sterling, L. (2002). ROADMAP: Extending the Gaia methodology for complex open systems. In *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, July 15-19, Bologna, Italy, Proceedings* (pp. 3-10). New York, NY: ACM.

[14] Kuan, P. P., Karunasakera, S., & Sterling, L. (2005). Improving goal and role oriented analysis for agent based systems. In *Proceedings of the 16th Australian Software Engineering Conference* (ASWEC 2005), 31 March - 1 April, Brisbane, Australia (pp. 40-47). Washington DC, USA: IEEE Computer Society.