How Can Agents Help in Designing Complex Systems?

Kuldar Taveter, Merik Meriste
Lab of Sociotechnical Systems, Department of Software Science, Tallinn University of Technology, Estonia.
kuldar.taveter@ttu.ee

***Abstract*—Distributed systems for decision support, crisis management, environment monitoring, e-government and smart cities, and for other areas, as well as Internet-of-Things applications are increasingly complex sociotechnical systems. Because of their distributed nature and complexity, such systems are cognitively hard to grasp and design. Therefore, appropriate simplified representations of complex systems – models – are required for designing and analyzing them. Moreover, a complex system needs to be simulated and prototyped before the development process can start to make sure a system with appropriate behavior has been designed. For modeling complex systems, "agent" or "actor" is a useful notion because conceptually sociotechnical systems consist of interacting human agents and man-made agents, such as smart software components and intelligent devices. In simulation and rapid prototyping, software agents can emulate both humans and technical components of a complex sociotechnical system. This article will provide an overview of the Agent-Oriented Modelling (AOM) methodology for modelling, simulation, and prototyping of sociotechnical systems. AOM offers software engineering processes and work products for agile design, simulation, and prototyping of distributed sociotechnical systems. In the center of AOM lies the viewpoint framework within which to design sociotechnical systems. The viewpoint framework supports the modelling, simulation, and prototyping of systems for a given problem domain from three balanced and interrelated viewpoint aspects: information, interaction, and behavior. The article will describe case studies from several industry- and government-related research projects.**

***Index Terms*—Sociotechnical System; Agent; Modelling; Simulation.**

I. INTRODUCTION

Agent is known as an entity that can act in the environment, perceive events occurring in the environment, communicate with other agents, and reason [1]. Agent is by definition reactive, proactive, and social [2]. An agent is *reactive* if it is able to perceive its environment and respond in a timely fashion to changes occurring in it [1]. A *proactive* agent does not simply act in response to its environment but is able to exhibit goal-directed behavior and take the initiative where appropriate [1]. A *social* agent interacts, when appropriate, with other agents in order to complete their own problem solving and to help others with their activities [1].

Two kinds of agents can be distinguished: *human agents* and agents created by them – manmade agents. *Manmade agent* is a kind of agent that has been implemented by humans physically or in software or as a combination of both [1]. *Robot* is a man-made agent that has been implemented by humans physically and is embodied in hardware. *Software agent* is a man-made agent that is implemented as software. There are also *institutional agents*, which are aggregates consisting of internal human and man-made agents, which share collective knowledge, and that act, perceive and communicate through their internal agents. In this article we are foremost concerned with "agent" as a powerful abstraction capable of representing active entities of all kinds – humans, organizations, robots, and software agents. The notion of agent is employed by the authors for understanding, designing, and analyzing complex systems consisting of many interacting agents. Such systems are known as *multi-agent systems*. This article addresses open multi-agent systems of a special kind – *sociotechnical systems* – that include hardware and software, have defined operational processes, and offer interfaces, implemented in software, to human agents [1, 15]. *Openness* means that new agents may join in or opt out at any time.

The Sociotechnical Systems' Lab of Tallinn University of Technology has been and is currently involved in a number of research projects where the abstraction of agent has proved to be highly useful for either understanding a particular problem domain involving complex systems or for designing a sociotechnical system for this kind of problem domain, or for both purposes. This article will describe four of such projects and will generalize from the experience obtained in these projects.

In Chapter II of this article, an overview of the methodology of Agent-Oriented Modelling is provided. Chapter III of this article describes three case studies, where AOM has been applied. Chapter IV presents conclusions.

II. AGENT-ORIENTED MODELLING

Agent-Oriented Modelling (AOM) [1] is an approach for modelling and simulating the behaviors of complex socio-technical systems where a problem domain is first conceptualized in terms of the goals to be achieved by the system, the roles required for achieving them, and the domain entities embodying the required knowledge. The roles are thereafter mapped to the agents playing the roles, the goals – to the activities performed by the agents, and the domain entities – to the items of knowledge held by the agents.

In the center of the AOM methodology lies the *viewpoint framework* [1] depicted in Table 1. It consists of a matrix with three rows representing different abstraction layers and three columns representing the viewpoint aspects of interaction, information, and behaviour. The abstraction layers of the viewpoint framework are ''problem domain analysis,'' ''platform-independent design,'' and ''platform-specific simulation and prototyping.'' In Table 1 these layers are entitled for short as "Analysis," "Design," and "Simulation and prototyping." Each cell in this matrix represents a specific viewpoint, such as ''interaction analysis,'' ‘‘information design,’’ or ‘‘behaviour simulation.’’ The cells of the viewpoint framework represent artefacts – tabular models, graphical models, documents, and program code – that are produced by AOM. Conceptually, we consider artefacts as abstractions reducing the complexity of a sociotechnical system for better understanding of the system’s particular aspects and their impact on its behavior.

We will now provide an overview of artefacts included by the viewpoint framework, proceeding by viewpoints.

From the viewpoint of behavior analysis, a goal model can be considered as a container of three components: goals, quality goals, and roles [1]. A goal is a representation of a functional requirement for the sociotechnical system to be designed. A quality goal, as its name implies, is a non-functional or quality requirement of the sociotechnical system. Goals and quality goals can be further decomposed into smaller related subgoals and subquality goals. The hierarchical structure is to show that the subcomponent is an aspect of the top-level component. Goal models also determine roles that are capacities or positions that agents playing the roles need to contribute to achieving the goals. The notation for representing goals and roles is shown in Table 2. Roles are modelled in detail in the viewpoint of interaction analysis. Goal models go hand in hand with motivational scenarios that describe in an informal and loose narrative manner how goals are to be achieved by agents enacting the corresponding roles [1].

From the viewpoint of interaction analysis, the properties of roles are expressed by role models. A role model describes the role in terms of the responsibilities and constraints pertaining to the agent(s) playing the role. Organization model is a model that represents the relationships between the roles of the sociotechnical system, forming an organization [1].

From the viewpoint of information analysis, domain model represents the knowledge to be handled by the sociotechnical system. Domain model consists of domain entities and relationships between them. Domain entity is a modular unit of knowledge handled by the system [1]. Domain model also represents the environment(s) in which agents of the sociotechnical system are deployed. In AOM environment is a first-class abstraction that should be modelled and implemented on its own.

From the viewpoint of interaction design, agent models transform the abstract constructs from the analysis stage, roles, to design constructs, agent types, which will be realized in the implementation process. Interaction models represent interaction patterns between agents of the given types. They are based on responsibilities defined for the corresponding roles [1]. In AOM, interaction models are represented by means of action events and non-action events [1]. An action event is an event that is caused by the action of an agent, like sending a message or starting a machine. An action event can thus be viewed as a coin with two sides: an action for the performing agent and an event for the perceiving agent. A message is a special type of action event—communicative action event—that is caused by the sending agent and perceived by the receiving agent. On the other hand, there are non-action events that are not caused by actions—for example, the fall of a particular stock value below a certain threshold, the sinking of a ship in a storm, or a timeout in an action. The notation for modelling both kinds of events is represented in Figure 1. Non-action events also include exogenous events. An exogenous event is a kind of event whose creating agent we are not interested in. As has been pointed out in [5], if we intend to simulate or prototype the sociotechnical system to be designed, which corresponds to the lowest abstraction layer of the viewpoint framework, exogenous events need to be generated.

Table 1
The Viewpoint Framework

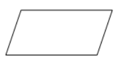
	Viewpoint aspect		
Abstraction layer	Interaction	Information	Behavior
Analysis	Role models and organization model	Domain model	Goal models and motivational scenarios
Design	Agent models and interaction models	Knowledge models	Scenarios and behavior models
Simulation and prototyping	Platform-specific models		

From the viewpoint of *information design*, it is essential to represent both private and shared knowledge by agents. An agent’s *knowledge model* represents knowledge about the agent itself and about the agents and objects in its environment [1].

Finally, from the viewpoint of *behavior design*, we model how agents make decisions and perform activities. There are two kinds of models under this viewpoint. A *scenario* is a behavioral pattern that describes how the goals set for the sociotechnical system can be achieved by agents of the system. *Behavior models* describe the behavioral patterns of individual agents [1].

At the *abstraction layer of simulation and prototyping*, agent-oriented models are turned into dynamic platform-specific models that show the effects of the behavioral patterns of individual agents as well as provide information on the complex feedback dynamics required for understanding the behavior of the sociotechnical system as a whole. Appropriate simulations relying on scenarios can help to understand and exploit the emergent behavior of an individual agent or an entire sociotechnical system over time. The abstraction layer of simulation and prototyping can in turn be split into several layers, each of which is geared towards simulating and prototyping at a different granularity level the system under design.

Table 2
Notation for modelling goals and roles

Symbol	Meaning
	Goal
	Quality goal
	Role
	Relationship between goals
	Relationship between goals and quality goals

III. CASE STUDIES

In this section, we provide an overview of three case studies, where AOM has been applied. The first case study deals with designing a sociotechnical system for advanced collaborative decision-making at airports. The second case study addresses providing decision support in crisis management by *human-in-the-loop* simulations. The third case study is concerned with designing proactive services based on registries and information systems connected across the borders in the European Union.

A. Advanced Collaborative Decision-Making at Airports

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

The major stakeholders in aviation industry are air traffic control (ATC), airports, ground handlers, 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 Traffic Manager (DMAN) and Arrival Traffic Manager (AMAN) are used by ATC optimize the usage of takeoff and landing runways and minimize taxi times of aircraft. The Resource Allocation Manager (RMAN) used by airports optimizes assignment and utilization of stands and gates, and the usage of refueling, catering, and turnaround resources. In addition, the Environment Manager (EMAN) interacts with both systems to minimize the noise and air quality impact of operations, while the Information Manager (IMAN) provides both real-time and historical operations data for planning, billing, marketing, reporting, and public relations. In addition, airlines and ground handlers working on behalf of them at airports have their own internal information systems.

Resulting from the lack of global optimization, when several aircraft are ready for push back from their stands, ATC normally makes its decision on who will move first purely on air traffic considerations. Therefore, long queues of aircraft can often occur at the runway holding points, which can cause taxiway congestion, additional fuel consumption, and environmental problems.

The problems described above have prompted the project team to take a novel approach where different stakeholders made decisions based on global situation awareness. This can be achieved by the combination of a new business model and novel sociotechnical solution. The proposed business model and solution would interconnect existing systems managed by different stakeholders. The aim of the solution is to offer situation-aware decision-support for the stakeholders – airports, airlines, and ATC – to ensure that they receive on time relevant and accurate information and enable them to act upon that information. This business model and solution is termed as Advanced Collaborative Decision Making (A-CDM).

A-CDM is a complex sociotechnical system. Figure 1 represents the highest level of functional and non-functional requirements for A-CDM in the form of a hierarchical goal model. As is reflected by Figure 1, the purpose of A-CDM is to allocate resources related to any particular flight. The goal to allocate resources is associated with the stakeholder role Passenger, whose enacting human agents are the ultimate consumers of the resources to be allocated. The goal to allocate resources has three attached quality goals, namely to achieve maximal safety, maximal airport efficiency and minimal environmental nuisance. The main goal has been decomposed into the subgoals to allocate ATC resources, airport resources and the resources of the airline / ground handler (GH). Each of these sub-goals is associated with a respective role, namely Local ATC, Airport, and Airline/GH. The sub-goals can be expanded, as in our experience, hierarchical chunking greatly aids understanding. However, because of the focus of this paper, refinements of the goal models are not presented here.

The A-CDM aimed at by the requirements represented in Figure 1 is represented in Figure 2, which originates in [4].

Because of the safety-critical nature of the problem domain, modelling and simulation are necessary before implementing any solution. Simulation is also required for demonstrating potential benefits of implementing and applying A-CDM, such as the increased throughput of planes through airports and less time on the ground for the airlines. The purpose of the intended simulation system is to emulate the decisions made by different stakeholders in A-CDM. The simulation system also enables to try out decisions by an individual stakeholder in the "human-in-the-loop" manner where the decisions by other stakeholders are simulated.

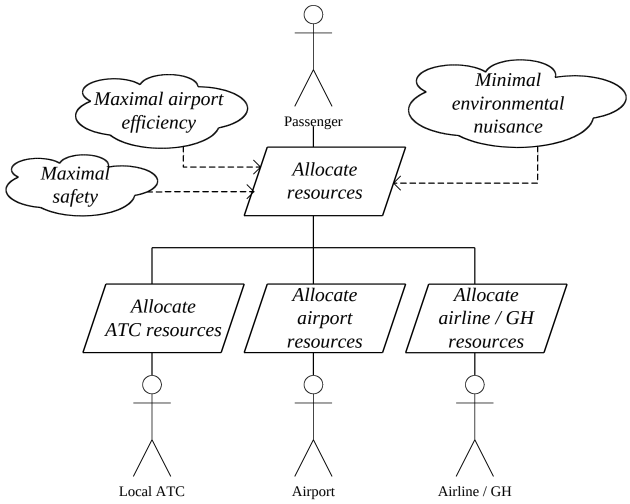


Figure 1: The goal model of resource allocation

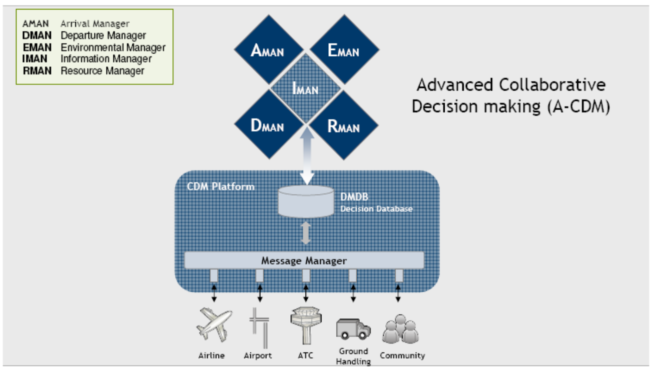


Figure 2: An overview of A-CDM [4]

In the project, an agent-based simulation system depicted in Figure 3 was designed by AOM. In the course of the design process, other types of AOM models included by Table 1 were created, such as role models, organization model, acquaintance model, interaction models, domain model, knowledge model, scenarios, and agent behavior models. An interested reader can find the descriptions of these models in [5-7].

The simulation system was implemented on the JADE platform for agent-based systems [8]. In the simulation environment, the "Environment simulator" generates from historical event data exogenous events of the types AircraftIsLanded and AircraftIsInBlock, denoting the time of landing of a particular aircraft and the time of its arrival at the assigned gate or stand. Based on the data about the incoming flights, decisions have to be made by ATC, airport authorities, airlines, and ground handlers for efficient servicing of aircraft and passengers, to achieve maximally fast turnaround of flights. As it is impossible to simulate individually the behaviors by information systems operated by different stakeholders, such as AMAN, DMAN, RMAN, EMAN, and IMAN, in the simulation system common situation awareness is achieved by means of the Total Airspace and Airport Modeler (TAAM, https://taam.jeppesen.com/), which allows modelling and simulation of airports and the surrounding airspace. TAAM is a flagship product of our project partner Jeppesen, which is a subsidiary of Boeing. As a result, an agent-based system for simulation of decision-making represented in Figure 3 was achieved. In the simulation system, the influence of A-CDM in case of each given airport can be evaluated by comparing the decisions made with the help of A-CDM with the historical off-block and departure times of aircraft, corresponding to the historical arrival and in-block times mentioned above. A screenshot of the A-CDM simulation system is shown in Figure 4.

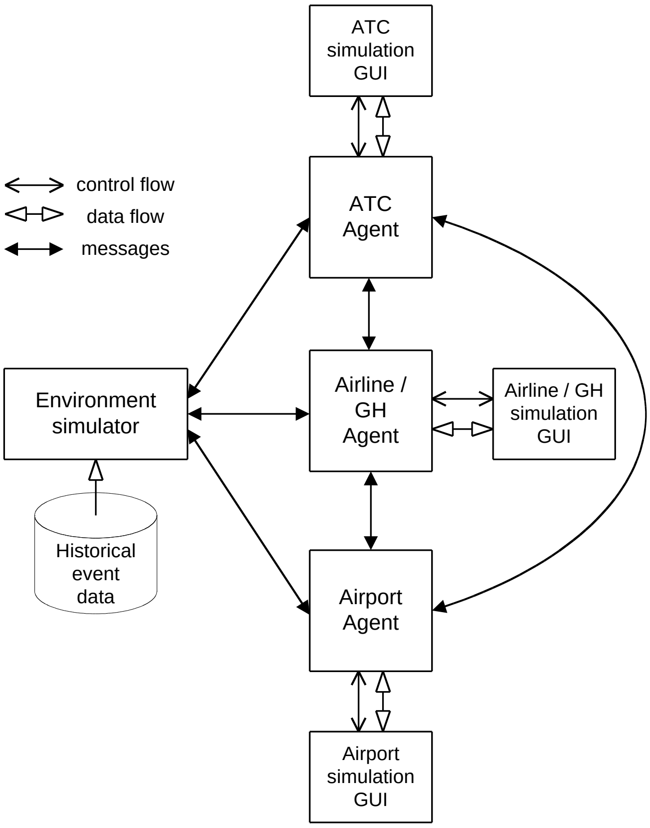


Figure 3: The simulation system of A-CDM

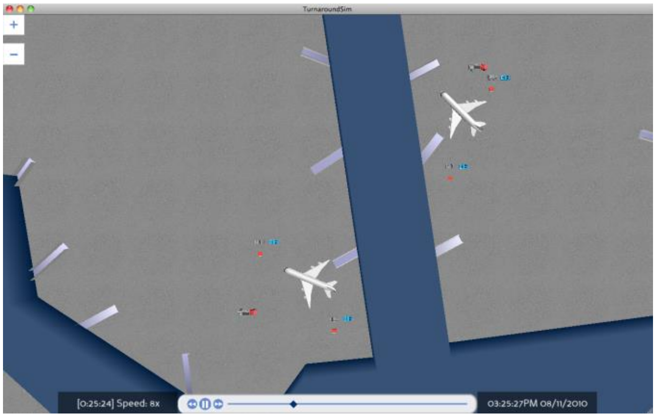


Figure 4: Screenshot of the A-CDM simulation system

In the A-CDM case study, AOM facilitates obtaining a holistic understanding of the problem domain. Moreover, AOM supports simulation-based design, where a multi-agent simulation of *human-in-the-loop* type is used to emulate the intended system before setting out to design and implement it. *Human-in-the-loop* means that different decisions by various stakeholders can be tried out while the decisions by other stakeholders are simulated.

B. Agents for crisis management

The CRISMA project (http://www.crismaproject.eu) focused on simulation of large-scale crisis management scenarios with multi-dimensional effects on the society. The workflow pattern of a CRISMA application is presented in Figure 5. The applications allow the decision-maker to visualize the state of a crisis, and compare such state with possible "alternative" states that may result from certain decisions and events of the simulated scenario. CRISMA applications do not impose decisions, but allow comparing effects of decisions by mean of relevant indicators [4].

CRISMA applies a hybrid model of crisis management comprising agent-based simulation models, other kinds of computational simulation models, spatial models and *human-in-the-loop* simulations for typical crisis scenarios. In the CRISMA architecture [10], the virtual world of crisis management is modelled in terms of virtual world states. The situation of the world at a given time during a crisis management scenario is represented as a world state, which consists of a structured collection of data records – Objects of Interest (OOI). OOIs are used in CRISMA to designate entities that are of interest to crisis management practitioners and therefore need to be represented and handled by a CRISMA application. Examples of OOI types are Emergency Vehicle, Field Rescue Worker, and Patient. In a CRISMA application, changes in the world occurring in time are represented as the virtual world state transitions. World state evolution in CRISMA applications is represented in Figure 6.

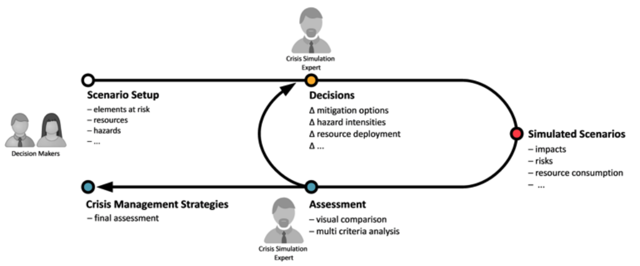


Figure 5: CRISMA workflow pattern [9]

The initial world state is specified as an initial collection of OOIs representing the inventory data, weather data, situation maps, vulnerability classes of buildings, and so on. All this data is provided by the CRISMA user and/or is retrieved through dedicated applications and web services that are interfaced with the CRISMA application.

In CRISMA applications, a simulated scenario evolves through transitions from one world state to the next world state, while the content of OOIs is updated by simulations and services applied to the transitions. Depending on the application, new states may be generated (a) at regular time intervals; (b) following the decisions by the users (e.g. "evacuate"); (c) as reactions to certain events, resulting from the execution of a particular simulation system (e.g. "ambulance has arrived at the scene"). Conceptually, the state-based behavior of this kind is represented in Figure 6.

Starting from the world state of interest, the user of a CRISMA application can simulate the evolvement of a crisis and the response either by executing predefined simulations or directly manipulating the current world state data. A state transition is conducted by one or several simulation threads – corresponding to different simulation systems attached to the transition – that take selected world state data as input and produce new world state data as output. Multiple traces are acceptable as the world state can also be updated and changed by a CRISMA user to define alternative scenarios. The "joystick" drawn within the "Manipulation" tag of Figure 6 represents the interactions by the user with a CRISMA application either by directly manipulating the current state data, or by adjusting the simulation control parameters.

In CRISMA applications, indicators and criteria for meeting them can be set and costs can be calculated at any state to provide adequate information for the analysis and assessment of the current situation in the simulated scenario. The analysis of the current situation in the virtual world supports decisions for selection of further actions including the initiation of alternative scenarios starting from the current transition. The user can test the effects of alternative decisions, whereby states and threads can be compared and assessed using relevant indicators and criteria. The main principle of CRISMA decision-support is to help the user of a CRISMA application in evaluating the results of their decisions, while not suggesting a course of action or imposing any particular decisions.

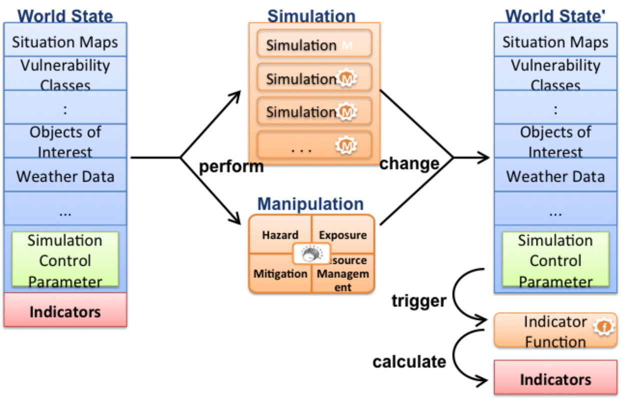


Figure 6: World state evolution in CRISMA applications [11]

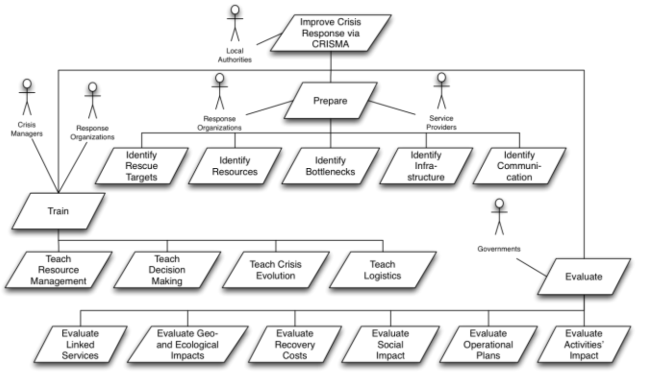


Figure 7: Generic goal model for CRISMA pilots

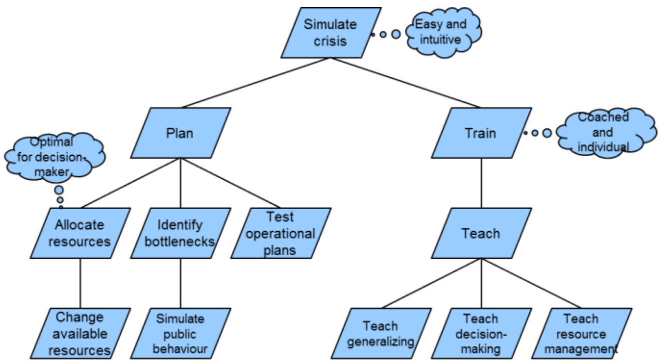


Figure 8: An elaboration of the generic goal model

In the CRISMA project, the purpose of applying AOM was twofold: (a) systematic specification of requirements for the simulated CRISMA world by means of the types of artefacts included by the viewpoint framework represented as Table 1; (b) creating a proper conceptual problem domain analysis and system design foundation for implementing agent-based simulations for particular pilots. Figure 7 represents the generic goal model for CRISMA pilots. As Figure 7 reflects, the purpose of CRISMA is to improve crisis response. This is done through training, preparing, and evaluating.

Based on the generic goal model for CRISMA, goal models for CRISMA pilots were created. The pilots deal with planning and training for typical crisis events, such as a coastal flood, winter storm with blackouts, chemical contamination, and an earthquake. Figure 8 represents a refinement of the generic goal model for the pilot dealing with the response to an industrial accident that has resulted in a chemical plum.

Goal models and other types of models created in the project were turned into agent-based simulations, according to the viewpoint framework of the AOM methodology represented as Table 1. For example, the goal model depicted in Figure 8 was further refined, complemented by models of other types of AOM, and turned into agent-based simulations for teaching resource management and decision-making in pilots devoted to a mass casualty incident [12] and an accidental contamination [13].

We will now consider some aspects of the process of constructing an agent-based simulation system for teaching resource management and decision-making. During a crisis, commanders in crisis management organizations have to decide which resources to deploy, where to send them, and what should be their tasks. The scheme of resource management in accidental contamination is presented in Figure 9. The crisis scenario deals with accidental spilling of chemicals in a major port. The resulting plume poses a threat to a large number of inhabitants in the neighboring city and the severity of the impact depends on the decisions made by the crisis managers. The main purpose of the application is to assure the impact of the decisions taken by the trainees is realistic, in the sense that the impact is guided by the natural laws and the training setup reflects a real-life scene. *Three crucial factors* in achieving the main purpose are as follows. *First*, it should be possible to actively deploy resources and their arrival at the scene should be dependent on their distance from the scene, type of the road, and weather conditions. *Second*, the condition of a patient should deteriorate based on the patient's previous medical history, lapsed time, intensity of exposure to the contamination, and the help received (e.g. decontamination). *Third*, the state of a resource should also change as a result of the interactions between the resource and its environment. This also means that at some point, any simulated resource should become exhausted. A screenshot of the agent-based simulation of teaching resource management and decision-making is represented in Figure 10.

In the agent-based simulation system under discussion, there are two kinds of OOIs – those implemented as active agents and OOIs implemented as passive objects. Agents conform to the definition of "agent" presented at the beginning of Section 1. Objects are passive entities in the sense that they are invoked by agents in predefined ways. The properties and behavioral patterns of agents corresponding to OOI types are specified by means of AOM as requirements for the agent-based simulation. The resulting agent types, such as Patient and Ambulance Vehicle, are intentionally designed to be as generic as possible. The agent-based simulation system obtains the initial world state from the OOI repository and stores in the OOI repository intermediate states of the OOIs that result from running the simulations. The environment for the agents is made up by external simulations, such as simulations of weather and dynamics of a poisonous plum. The agents communicate with the external models lying in their environment through a fixed protocol.

CRISMA applications allow trainees, decision-makers and other stakeholders to compare the results of different decision paths with the help of performance indicators, criteria, and multi-criteria ranking. Users of a CRISMA application can learn from own "mistakes" in a simulated reality where experimenting and making mistakes is encouraged rather than sanctioned. Simulations can thus substantially improve the capability to understand the potential impact of various decisions in situations of interest.

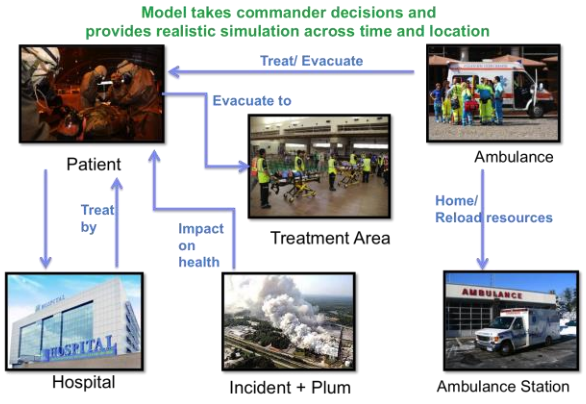


Figure 9: Resource management in accidental contamination [13]

According to the experience from the CRISMA project [13], defining an appropriate level of abstraction for agent-based simulations as well as understanding the limitations of the simulations is extremely important. The role of AOM here is substantial because the "right" choice of simulations for decision support depends on the intended usage context.

In addition to training, agent-based simulations help to explore and forecast more generic behavioral patterns in crisis management.

The results from the CRISMA project can be applied for designing operative decision-support systems of crisis management, where simulations facilitate decision-making in a nearly real-time context.

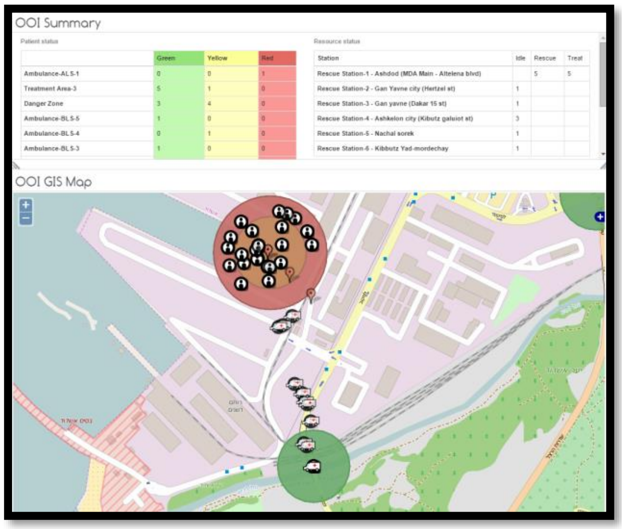


Figure 10: Agent-based simulation for teaching resource management and decision-making

C. *Interconnecting Information Systems of EU Countries*

The "Once-Only" Principle Project (TOOP) funded by the Horizon 2020 program of the European Union (http://www.toop.eu/) has the ambition to connect 60 registries and information systems from 22 countries. This new project is concerned with the interchange of company data between the registries and information systems. Information about a company is stored in the business registry of the country where the company is registered. However, the same information or a part of it is also stored by registries managed by public administrations and other stakeholders in other countries. Keeping this information up to date is a real challenge, especially when it is related to companies that have daughter companies in other countries.

The purpose of the project is to reuse across EU and beyond information on companies available in business registries and other governmental information systems. Conceptually, information systems and registries in different countries can and should be treated as autonomous agents. In particular, this article is concerned with information systems and registries that exhibit the characteristics of proactive agent behavior. Such information systems and registries monitor changes in a company profile or in organizational relationships in which the company is involved and notify relevant parties across borders about the changes related to them. For example, when a company registered in one country has a daughter company in another country, the business registry in the country of origin can proactively notify relevant parties related to the daughter company in the destination country about the changes potentially affecting them.

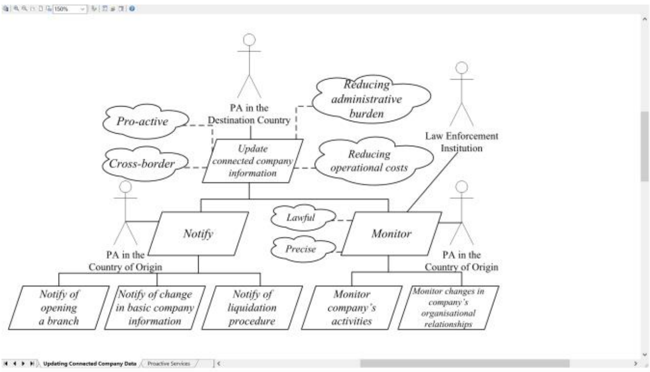


Figure 11. Goal model for updating connected company information

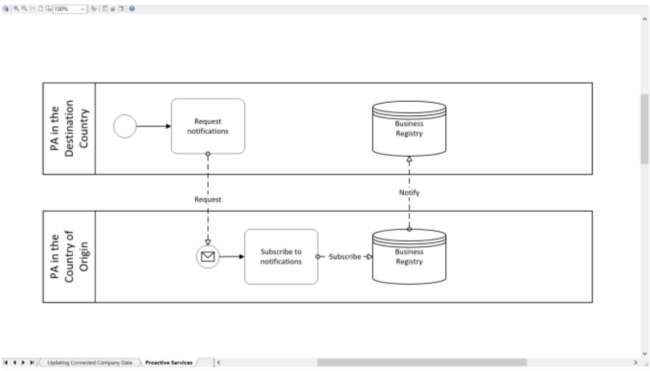


Figure 12. Process model of the cross-border "push" service for updating connected company information

Figure 11 represents a sketch of functional and non-functional requirements for the corresponding pilot of the TOOP project. The purpose of the pilot is to update connected company information across the borders. Such updating has to be proactive and is supposed to reduce administrative burden and operational costs. There are two aspects of updating connected company information – monitoring the information about the company in its home business registry managed by the Public Administration of the Country of Origin and notifying about the changes the Public Administration of the Destination Country where, for example, operates a daughter company of the given company. Monitoring should be lawful – should occur according to the legislations by the countries involved and EU – and should be precise. Different aspects of monitoring are monitoring company's activities and monitoring changes in company's organizational relationships. Various aspects of notifying are notifying of opening a branch of the given company, notifying of change in the basic company information, and notifying of the liquidation procedure.

Figure 12 depicts a sketch of the process model of the cross-border proactive or "push" service for updating connected company information. As is reflected by the process model, the Public Administration in the Destination Country has to subscribe from the Public Administration of the Country of Origin to monitoring and notifying information about a particular company. The Public Administration of the Country of Origin, in turn, subscribes to the monitoring of the relevant information by its Business Registry and to notifying the Business Registry of the Destination Country about the relevant changes.

The previous paragraph contained a description of a simple "push" or proactive service run by the Public Administration of the Country of Origin. In real life, there are many cases where the assumption does not hold that all information needs in a dynamic setting can be defined in advance [14]. For example, if the purpose of monitoring companies is to prevent money laundering and other kinds of crime, many complicated organizational relationships in which the given company might be involved need to be monitored, which means that it is usually not possible to know precisely in advance what information changes should be subscribed to. Another danger in this kind of situation is cognitive overload due to "pushing" too much information, which makes intelligent information "pushing" desirable [14]. Some techniques describing how to achieve this kind of intelligent information "pushing" are described in [14]. Design of proactive services that are based on intelligent information "pushing" is also addressed by the research work that is being conducted at the Lab of Sociotechnical Systems [19-20]. This work is based on our earlier work in information systems' integration projects [16-18].

It can be stated that in the TOOP project, the application of the conceptual abstraction of "agent" and other related abstractions through employing the AOM methodology facilitates the design and implementation of truly intelligent "push" services. Such services enable to monitor the relevant information and notify about the changes in a much broader sense than simply by subscribing to the relevant information.

IV. Conclusion

This article demonstrated the usefulness of the notion of "agent" or "actor" in designing, prototyping, and simulating sociotechnical systems. We used the examples of three case studies to emphasize different aspects of this notion. In all three case studies, the AOM methodology was applied to problem domain modelling for a sociotechnical system and for designing the sociotechnical system for the given problem domain. A sociotechnical system in the first case study was a multi-agent simulation of *human-in-the-loop* type for emulating the intended system of collaborative decision-making at airports. Because of the complexity of the problem domain, designing and implementing a simulation system before setting out to design and implement the real system was an approach taken in the first case study. A sociotechnical system in the second case study was a training system for planning and training decisions of resource allocation in natural crises. In the training system, different evolvements of incident scenarios were emulated by agent-based simulations. In the future, the second case study can be extended to designing operative decision-support systems of crisis management, where simulations facilitate decision-making in a nearly real-time context. A sociotechnical system in the third case study was a proactive service for monitoring and notifying the relevant company information, where the relevant information is automatically "pushed" from the relevant registry of one country to the relevant registry of another country. In the future, conceptual usage of the notion of "agent" in the third case study can be extended by applying agent technologies that enable intelligent monitoring of company data instead of subscribing to the relevant data in advance.

Acknowledgment

The research work reported in this article has been supported by Lochard, Inc. and Jeppesen, Inc., Australian Research Council, and the FP7 and Horizon 2020 programs for research and innovation by the European Commission.

REFERENCES

[1] Sterling, L., and Taveter, K. (2009). *The art of agent-oriented modelling*. Cambridge, MA and London, England: MIT Press.

[2] Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.

[3] *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.

[4] *CDM: From Challenge and Philosophy to Implementation. A Holistic View*. Presentation for EUROCONTROL CDM TF/11 by Ned Macesic, President EMEA, Lochard. Brussels, 29 Nov 05.

[5] Miller, T., Lu, B., Sterling, L., Beydoun, G., Taveter, K. (2014). Requirements Elicitation and Specification Using the Agent Paradigm: The Case Study of an Aircraft Turnaround Simulator. *IEEE Transactions on Software Engineering*, *40*, 1007−1024.

[6] Sterling, L., Taveter, K. (2009). Event-based Optimization of Air-to-Air Business Processes. *Intelligent Complex Event Processing: AAAI Spring Symposium on Intelligent Event Processing, March 23-25, 2009, Stanford University, Stanford, California, USA*. Ed. Nenad Stojanovic; Andreas Abecker; Opher Etzion; Adrian Paschke. California: AAAI Press, 80−85.

[7] Sterling, L., Taveter, K. (2009). Event-based optimization of air-to-air business processes (longer version). Proceedings: The Industry Track of the 20th Australian Software Engineering Conference (ASWEC 2009), 14-17 April. Gold Coast, Australia.

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

[9] Havlik, D., Dihé, P., Schlobinski, S., Heikkilä, A. M., Polese, M., & Taveter, K. (2015). Modelling crisis management for improved action and preparedness. *Espoo, Finland: VTT Technology*, *228*.

[10] Dihe, P., Denzer, R., Polese, M., Heikkilä, A.-M., Havlik, D., Sautter, J., Hell, Th., Schlobinski, S.: An architecture for integrated crisis management simulation. In: 20th International Congress on Modelling and Simulation, Adelaide, Australia (December 2013).

[11] Taveter, K., Meriste, M., Rannat, K., Jähi, M., Dihé. P., Scholl, M., Guarino, S., (2015). Validated simulation tool for crisis management strategies and planned actions, Deliverable D44.4, CRISMA Project, http://www.crismaproject.eu/deliverables/CRISMA_D444_public.pdf

[12] Sautter, J. Havlik, D., et al (2015) Simulation and Analysis of Mass Casualty Mission Tactics: Context of Use, Interaction Concept, Agent-Based Model and Evaluation. IJISCRAM 7(3), 16-39.

[13] Havlik, D., Deri, O., Rannat, K., Warum, M., Rafalowski, C., Taveter, K., Kutschera, P., Meriste, M. (2015), Training Support for Crisis Managers with Elements of Serious Gaming. In: R. Denzer, R. M. Argent, et al (Eds.). Environmental Software Systems. Infrastructures, Services and Applications (217–225). Springer.

[14] Fan, X., Yen, J., & Volz, R. A. (2005). A theoretical framework on proactive information exchange in agent teamwork. Artificial Intelligence, 169(1), 23-97.

[15] Sommerville, I. (2011). *Software engineering*. London, England: Pearson Education.

[16] Taveter, K., and Wagner, G. (2002). A multi-perspective methodology for modelling inter-enterprise business processes. In: *H. Arisawa, Y. Kambayashi, V. Kumar, H. C. Mayr, and I. Hunt (Eds.), Conceptual Modelling for New Information Systems Technologies, ER 2001 Workshops HUMACS, DASWIS, ECOMO, and DAMA. Revised Papers* (403-416). Springer.

[17] Taveter, K. (2004). *A multi-perspective methodology for agent-oriented business modelling and simulation*. PhD thesis, Tallinn University of Technology, Estonia. Tallinn University of Technology Press (ISBN 9985-59-439-8).

[18] Norta, A., Dua, Y., Ma, L., Rull, A., Kolvart, M., and Taveter, K. (2015). eContractual Choreography-Language Properties Towards Cross-Organizational Business Collaboration. *Journal of Internet Services and Applications, 6* (8), 1-23.

[19] Sirendi, R., and Taveter, K. (2016). Bringing service design thinking into the public sector to create proactive and user-friendly public services. In: *F. F.-H. Nah, C.-H. Tan (Eds.), HCI in Business, Government, and Organizations: Information Systems, Third International Conference, HCIBGO 2016, Held as Part of HCI International 2016, Toronto, Canada, July 17-22, Proceedings, Part II* (221−230). Springer.

[20] Sirendi, R. (2016). Designing proactive public services as sociotechnical systems by using agent-oriented modelling. In: *M. Decman, and T. Jukic (Eds.), Proceedings of the 16th European Conference on e-Government (ECEG): Ljubljana, Slovenia, 16-17 June* (308−316). Reading, UK: Academic Conferences and Publishing International Limited.