Developing Agent-based Organizational Models for Crisis Management

Thomas B. Quillinan, Frances Brazier
Vrije Universiteit Amsterdam/Thales Nederlands
thomasq, frances@few.vu.nl

Huib Aldewereld, Frank Dignum, Virginia Dignum, Loris Penserini
Universiteit Utrecht
huib, dignum, virginia, loris@cs.uu.nl

Niek Wijngaards
D-CIS Lab, Thales Nederlands
niek.wijngaards@icis.decis.nl

ABSTRACT

Simulations of crisis scenarios have the potential to increase insight in the organizational structures needed as crises escalate. Real-life simulations involving personnel and figurants are expensive and time-consuming. Multi-agent system models allow for cost-effective simulations of changing organizational structures, enabling analysis of the implications for enactment during crisis escalation with respect to roles and communication structures. This paper presents both an organization-based model for crisis management that supports simulation of the dynamics of crisis management and a proof of concept implementation.

Categories and Subject Descriptors

I.6.8 [**Simulation and Modeling**]: Types of Simulation—*Distributed, Gaming*

General Terms

Management, Measurement, Design, Reliability, Experimentation, Human Factors.

Keywords

ALIVE, AgentScape, Multiagent Systems, Crisis Management, Simulation

1. INTRODUCTION

Crisis management is a challenge, especially when crises escalate. The numbers of organizations involves increases, communication lines change, roles change. Simulations provide a means to study the consequences of escalation of crises with respect to structures involved. As a crisis escalates, organizational structures are systematically updated to reflect the changes in the nature of the crisis and the number of parties involved. In the real world, simulations are enacted using active personnel [6], real people. However, such simulations are expensive, both in terms of the cost of execution and the cost of the time required for the emergency service personnel involved. Computer models of escalation of crisis, provides a more cost effective means to study the potential of different organizational structures in very many different scenarios.

The domain of crisis management, in particular the study of crisis response scenarios, is an active area of research in the field of Multi-Agent Systems (MAS) [4, 7]. In this domain agents, roles and groups can be clearly identified, the interaction patterns between (groups of) agents can be coordination, and the interaction between (groups of) agents with their changing environment can be modelled. Most current systems provide coordination and planning capabilities for teams of agents, often assuming the emergence of group behaviour. Highly formalized organization processes defined by governments and aid agencies define a strict frame for action.

Organizations have been defined as instruments of purpose [8], that is, organizations have objectives to be realized. Objectives of an organization are achieved through coordinated actions of agents efficiently, requiring distribution and coordination of activities such that 'the right agent is doing the right thing' [3].

In this paper, formal processes and requirements are the basis for the modelling of a MAS that regulates the activities of the different agents. Such an approach can be seen to be a form of adjustable autonomy A formal organizational model determines the range of autonomy of the participating agents, that in fact is adjustable [14]. The organizational model presented in this paper enables the explicit representation of structural and strategic concerns, and their adaptation to environmental changes in a way that is independent from the behaviour of the agents.

A recent exercise in the Netherlands to increase the country's preparedness in the event of a large-scale flood took one week and involved many hundreds of emergency personnel and figurants[1]. Lessons learned from this exercise showed that during emergencies there is typically no time to discuss the "right" approach, and that the Netherlands will need to adopt a military type command structure. This paper presents the first results of an agent-organization simulation of adaptation of command communication structures and its consequences for information and control relations on the field.

The paper is organized as follows. Section 2 describes the formal crisis management process in force in the Nether-

Cite as: Developing Agent-based Organizational Models for Crisis Management, T.B. Quillinan, et. al., *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. XXX-XXX.

[1]See BBC's news item: http://nl.youtube.com/watch?v=MxD2mJ3oVYA.

lands. Section 3 introduces the ALIVE project. Section 4 presents the model for crisis management organization. Section 5 discusses ongoing work on simulation of crisis organizations. Finally, Section 6, presents our conclusions and directions for future work.

2. CRISIS MANAGEMENT

The Netherlands has an extensive crisis management structure to respond to disasters. This structure is based on the severity of the disaster, and allows local, regional and national authorities to take action when necessary. Most incidents are relatively minor, and do not require much cross-organizational interactions. However, when a more complex crisis does arise, this structure defines how first responders within a safety region, that is, the police, fire and medical services, municipal governments and other services, are to cooperate.

The Netherlands is divided into 25 safety regions. Each safety region is managed by the Governing Board of the region. The members of the Governing Board are the mayors of all of the municipalities within the region. The Governing Board is responsible for all crises within the region. If a crisis occurs within the borders of a single municipality, the mayor of that specific municipality assumes control, If a crisis extends past the borders of a single municipality, responsibility shifts to the President of the Governing Board of the safety region. The President also acts as the head of the Regional Policy Team (Regionaal Beleidsteam – RBT). The Governing Board is always supported by the chief of police, the regional medical official, the chief of the fire service and the regional safety coordinator. When and where appropriate the municipal governments and other organizations such as the water management authority (waterschap) and specialist military units may become involved. Initially, when a crisis takes place, operational assistance is the first concern. The regional safety coordinator, however, is also tasked with providing post-crisis relief and compensation for damage caused by the disaster.

2.1 Sample Scenario

A simple scenario can be used to illustrate organizational change during crisis management. This scenario, describing the rescue procedures in the case of a flooding situation, contains sufficient detail to discuss the organizational transitions needed.

This scenario assumes that a major storm is emerging above the North Sea after a long period of dry weather. The storm is expected to peak during high tide above the West coast of the Netherlands, raising the water levels on the coasts to dangerous heights. Advance notice allows crisis management teams to determine the risks and their consequences, in particular the risk of dike breaches. Crisis teams are mobilized and informed of potential problems.

Initially, the storm reaches the Port of Rotterdam, Europoort. One of Shell's refineries is at risk: large quantities of petroleum products and associated dangerous chemicals may be dispersed at any moment. As the disaster progresses, several dikes are breached, flooding northern Rotterdam. Due to recent weather conditions several inland dikes outside of Rotterdam have dried out completely. Sudden inundation of water causes them to weaken and breach as well. Water has reached other suburbs of Rotterdam.

2.2 GRIP Levels

Procedures for coordinated regional incident handling in the Netherlands, GRIP (Gecoordineerde Regionale Incidentenbestrijdings Procedure) have been standardized[2] by the Dutch Ministry of Internal Affairs. GRIP defines levels for emergency handling, specifying the tasks, authorizations and responsibilities of all of the parties involved: operational, tactical and strategic.

Five levels of disaster severity are distinguished in GRIP:

GRIP 0 Routine accident: no coordination.

GRIP 1 Incident: multi-disciplinary operational coordination is needed.

GRIP 2 Large scale incident: more advanced multidisciplinary coordination is needed. At the tactical level the Regional Operational Team is activated, and at the strategic level, a municipal policy team is created, headed by the mayor.

GRIP 3 Disaster concerning multiple regions: strategic coordination is transferred to a regional policy team, headed by a separate regional coordinator, called the LOCC (Landelijk Operationeel Coordinatie Centrum).

GRIP 4 Large scale disaster: strategic scaling to provincial or national level is required.

GRIP levels provide a mechanism to describe the severity of a problem. As a crisis grows and more responders become involved, more organization between these responders is required. At a regional and national level, the LOCC acts as a dedicated crisis management coordinator with links to all of the emergency response services/organizations.

In the scenario described in Section 2.1, the GRIP level would be initially set to GRIP 2, but as the crisis grows and more dikes are breached, the level would increase to GRIP 3. Increasing the GRIP level implies change in the organizational structure of the crisis management team. Responsibilities previously handled locally are now managed by a coordination team. Modeling these organizational changes is described in Section 4.

3. ALIVE

The research presented in this paper is part of the ALIVE project. ALIVE aims to apply organizational theory to the design and implementation of software systems. The main focus of the project is to create complex systems based on the composition of (existing) services, through the addition of levels of abstraction. The advantage of added levels of abstraction to the design process of systems is two-fold: 1) it is often more intuitive to think in organizational structures and interactions while designing complex interactions for services, and the addition of the layers of abstraction allows for a gradual (fluent) transition from the system as foreseen to the actual implementation; 2) when changes happen in the environment (for example, specific services become unavailable) the added levels of abstraction act as an explicit representation of the conceptual steps made at design, thus giving additional information on why certain interactions are as they are, that enables the system to dynamically cope with the changes. The layers of abstraction introduced by the project are the following (see Figure 1).

[2]http://www.handboekrampbestrijding.nl

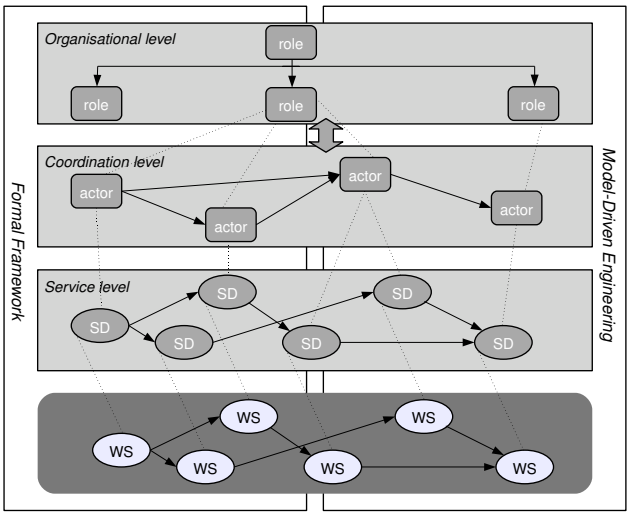

Figure 1: The Alive framework for software and service engineering.

- The *Service Layer* augments and extends existing service models with semantic descriptions (SD) to make components aware of their social context and of the rules of engagement with other services, in particular Web Services (WS).
- The *Coordination Layer* provides the means to specify, at a high level, the patterns of interaction between services, using a variety of powerful coordination techniques from recent research in the area.
- The *Organization Layer* provides context for the other levels – specifying the organizational rules that govern interaction and using recent developments in organizational dynamics to allow the structural adaptation of distributed systems over time.

This paper show how the use of such levels of abstraction can aid the creation of complex simulations, such as simulations of the crisis management domain. The focus on the organizational level of the ALIVE project shows how organizational specification on a high level of abstraction helps the design of systems that simulate crisis management processes.

The organizational level provides grips for the lower levels of abstraction, detailing why the system is being designed as such. The organizational specification itself, however, also has to fulfill general abstract purposes that are taken from the domain. These abstract purposes are an abstract specification of what *any* organization designed for this domain has to adhere to *at the least*. There are two different types of organizational purposes; those expressing liveness and those expressing safety of the organization.

Liveness purposes are an abstract specification of the overall aim of the organization; they define the objective for which the organization is created. Similar to liveness properties in, for instance, program verification, a liveness purpose defines an objective that should be reached at the end of the organizational execution (one can think of these as abstract achievement goals). In the crisis management domain, the liveness purpose of organizations is "solve crisis".

Safety purposes, on the other hand, express invariants of the organization; that is, abstract specification of objectives that should be maintained over the entire execution of the organization. Again, similar to safety properties or invariants as used in, for instance, program verification, the safety purpose should be uphold in every step of the execution (one can think of these as abstract maintenance goals). In the crisis management domain, possible safety purposes might be "minimize casualties", "minimize structural damage", and "minimize cost".

These abstract purposes describe the minimal requirements for the organization, but more information can be added to, for instance, improve the efficiency, robustness or flexibility of the organization. The next section describes how the crisis management organization is represented.

4. ORGANIZATION MODEL

Based on the domain description given in Section 2, this section presents the organizational model for the crisis management domain. The representation of the organizational model is done in the OPERA framework (see [2]), and contains three types of models:

1. the role dependency graph, to denote the relations between roles in the domain;
2. the interaction structure diagram, to represent the order of important interactions at a high level of abstraction, and
3. landmark patterns, that denote the important states in the organizational interaction, thus specifying how specific interactions should be achieved.

In the following, these models are applied to the crisis management scenario.

The need for changes in the simulation of the crisis management domain is motivated by the fact that maintenance costs of crises teams are high, that requires a tradeoff between local (and thus more rapid) availability of resources and the high costs of sustaining them when no emergency is visible on the horizon [5]. This implies that the organizational model presented in this section tries to specify the necessary components of the crisis management domain at a level of abstraction that allows the organization to remain stable over the different GRIP levels, whilst providing enough meaningful information for the simulation design.

As such, the role dependency graph for the crisis management scenario, presented in Figure 2, provides a level of abstraction that is stable at all GRIP levels.

A role in organizations is the abstract representation of a policy, service or function. Roles abstract from specific actors in the simulation and instead describe groups of actors that have similar functionality, rights and capabilities from the perspective of the organization. For instance, the role of *Information Provider* in a flooding situation can be played by a weather forecaster service, or dike watchers[3], but in a fire situation, this role is played by, e.g., smoke detectors.

Each of the roles has its individual objectives within the organization. In the organization specified in Figure 2 these role objectives are the following.

[3]Dike watchers are (Dutch) people hired to walk along the dikes in stormy or hazardous situations to check the conditions of the dikes and to make sure that potential breaches are noted as early as possible.

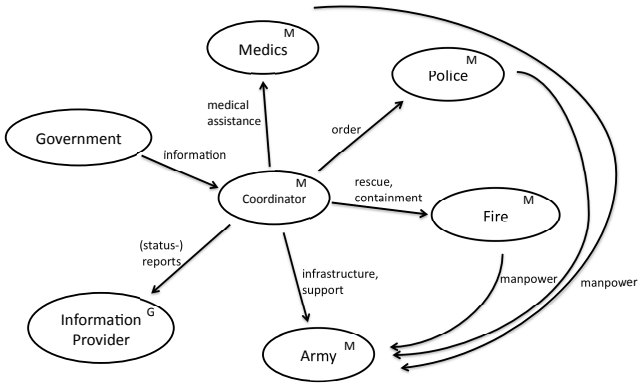

Figure 2: Organizational Role Dependency Graph.

- *Coordinator*: Solve the crisis.
- *Government*: Look after the citizens' well-being.
- *Medics*: Save all (or, get everyone healthy).
- *Police*: Maintain order.
- *Fire*: Save/rescue people.
- *Army*: Support (the other incident handling forces and/or the government).
- *Information Providers*: Reliable and/or fast information provision.

Next to specifying the different roles in the domain, the role dependency graph also details the dependencies between these roles. The direction of the dependency arrow in the role dependency graph specifies the direction of the dependency relation; that is, an arrow pointing from role A to role B means that role A depends on role B for the specified (sub)objective. As shown in Figure 2, the coordinator is the most important role in the organization as it depends on most of the other roles to help in solving the crisis.

The G in the *Information Provider* role denotes that this role is a role group. Role groups are used to refer collectively to a set of roles and to specify shared norms for the roles in the group. The number of agents playing a role can change from one GRIP to another. For example, at GRIP-1, the *Police* role is typically enacted by a single agent (i.e., there is only one Police Officer involved in the crisis handling). At GRIP-2, however, the *police* role might be enacted by several units. These police units, while their overall interaction with the coordinator remains the same, now have an internal structure (a hierarchy in this case) and *Police*-specific coordination and control mechanisms (e.g., the **order** objective can be divided in sub-objectives, for example, regulate traffic, patrol sector B5, etc., that are assigned to different units). These internal structures are unimportant for the crisis management organization, and can safely be abstracted from. Roles that can have such internal structure (at a higher GRIP) are denoted with an M; in essence, the role is enacted by a multi-agent system (or, sometimes even, by a multi-agent organization).

Note that some of the roles will not be enacted at the lower GRIP levels (e.g., the roles of Government or Army will not be available in GRIPs below GRIP-2 and GRIP-3, respectively).

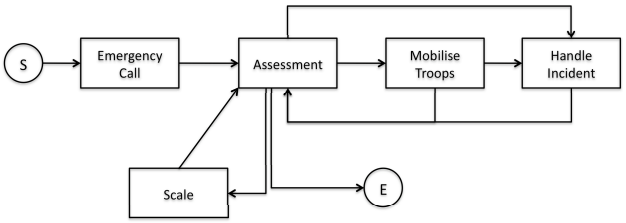

Figure 3: Organizational Interaction Pattern.

The interaction structure, as depicted in Figure 3 defines the order of the interactions between the roles in the organization. Interactions are grouped into scenes, where a scene reflects a meaningful (on its own standing) subset of organizational interaction related to the achievement of a (set of) (sub)objectives. The interaction pattern for the crisis management domain is shown in Figure 3.

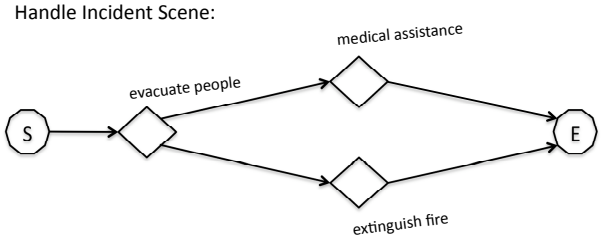

Figure 4: Landmark pattern of the Handle Incident scene.

Each scene in the interaction structure can be further detailed. An ordered pattern of important states in the achievement of the scene is grouped into a landmark pattern. Figure 4 shows the landmark pattern for the *Handle Incident* scene. Landmarks denote the important states that should be reached in the achievement of the scene, and the landmark pattern imposes an ordering over these landmarks to denote the order that these important states should be reached. The landmark pattern of the *Handle Incident* scene shown in Figure 4 has 3 landmarks: *evacuate people*, *medical assistance*, and *extinguish fire*. It denotes the interactions that need to take place to achieve the handle incident scene, in this case, meaning that people should first be evacuated, after which (possibly in parallel) those people can be medically assisted and the fire can be extinguished.

5. ORGANIZATION SIMULATION

As a crisis evolves, organizational structures are systematically updated to reflect the scale of the crisis. Simulation provides a means to study and evaluate the effect of these organizational changes. In the real world, simulations are enacted using active personnel [6]. However, such simulations are expensive, both in terms of the cost of execution and time of the people involved.

Computational simulations, utilizing computer models of the organizations involved and the procedures defined, provides a means to study and evaluate possible implications.

However, such organizations are often too complex to be analyzed by conventional techniques but require integrated models of people, tasks, goals, and dynamic environments [12], and the creation of models that combine formal analysis and empirical simulation results [13]. Not only many, highly interconnected, and unforeseen factors that influence the performance of the organization, but also require a quick evaluation of change and adequate decision making. This makes it hard to evaluate a priori a design or strategy and foresee its implications under different circumstances. Providing decision-makers with models and tools that enable them to effectively identify, assess and respond to environmental changes is of utmost importance for the survival of the organization. Organization simulations for such domains are based on two main assumptions: dynamic adaptation to a changing environment is necessary, and agents are autonomous and have local control. This indicates the following requirements for simulation models [2]:

- **internal autonomy**: interaction and structure of the society must be represented independently from the internal design of the agents
- **collaboration autonomy**: activity and interaction in the society must be specified without completely fixing in advance the interaction structures.

This section presents the design and implementation of the first stage of crisis management simulation environment.

5.1 Crisis Simulations

Simulation of a crisis involves modeling the crisis management structure. As described in Section 2, this structure includes the major emergency service personnel as well as specialized crisis management coordinators. As it would be impractical to model every single unit of the emergency services, particularly for a GRIP 3 or GRIP 4 crisis, a representative number of these services are modeled. The internal logic of these services represents the interactions within the lower levels of these organizations. This model focuses on *organization* and *coordination* of the crisis coordinators and the units with whom they directly communicate. These map to the organizational and coordination layers of the ALIVE project, described in Section 3.

Crises happen in *locations*. These locations have attributes, such as the population, facilities, as well as the status of the crisis taking place. This status information can include information such as the depth of the water, in a flooding scenario, or the extent of a fire, in a fire scenario. Other information, such as the weather forecast, information from the public (using the emergency services telephone number) or information from external organizations are modeled as *services*. These services act as information repositories.

Services are defined using ontologies and are accessed using the AgentScape Web Services gateway. During a crisis simulation, matchmaking [11] is utilized to select appropriate services when necessary. For example, during an evacuation scenario, matchmaking is utilized to select the form of transport for discrete sets of people. In particular, different types of people, for example, prisoners, have different transportation requirements. In the case of a prison evacuation, army trucks or prison buses would be selected, rather than school buses.

5.2 Organizational Change

When a crisis is in a stable state, that is, when the organizational structure is in place, information flow is also stable. In particular, each actor knows to whom they must report, receive orders and information, and to whom they direct orders. However, if the crisis is in flux, the structure of the organization and the information flow may change. The roles and responsibilities of the actors change. For example, when a crisis is elevated from GRIP 1 to GRIP 2, the role of the policeman at the scene, who has been acting as the coordinator changes. She will report to the new coordinator as defined by GRIP 2, the municipal crisis management team.

One of the major challenges involved in changing roles or responsibilities from one actor to another is transfer of knowledge: knowledge concerning both information and decisions. The *information* critical to decisions in the future, such as the status of each of the crisis management agencies, as well as the status of the crisis. Secondly, *decisions* that have been made, and orders that have been issued. This knowledge must be transferred to the actor who now holds the role.

For example, the scenario outlined in Section 2 is initially a GRIP 2 level crisis. At this level, the municipal crisis management team is the *co-ordinator*, lead by the mayor of the municipality. Decisions made by this team are communicated to the heads of the emergency services of the municipality and information regarding the crisis is routed to this team.

If the crisis is upgraded to GRIP 3, the organizational structure changes. The role of coordinator is now assigned to the LOCC. The municipality now acts in the role of *government* that receives information from the LOCC coordinator. Decisions are now routed through the centralized heads of the respective services directly to the personnel involved.

5.3 Implementation

AgentScape is an agent platform that provides middleware infrastructure needed to support mobility, security, fault tolerance, distributed resource and service management, and service access, to heterogeneous agent applications. The multi-level AgentScape middleware infrastructure [9] has been designed to be extensible and scalable.

Within AgentScape, *agents* are active entities that reside within *locations*, and *services* are third-party software systems accessed by agents hosted by the AgentScape middleware. Agents in AgentScape can communicate with other agents and migrate from one location to another.

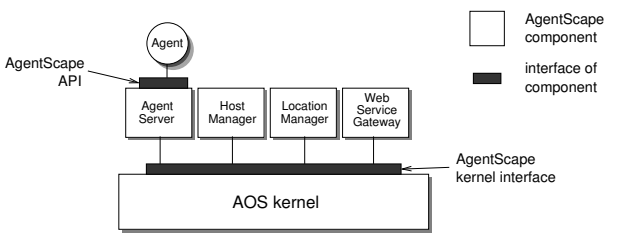

Figure 5: An AgentScape host operating as a location manager.

AgentScape, shown in Figure 5, has been designed as a multi-layered architecture with (1) a small *middleware ker-*

nel, called the AgentScape Operating System (AOS) kernel [15], that implements basic mechanisms and (2) high-level *middleware services* that implement agent platform specific functionality and policies. The current set of middleware services includes agent servers, host managers, location managers, directory services, anonymity services, and configuration services. AgentScape supports a number of communication structures for large scale agent simulations.

AgentScape is used to model the crisis management structure in the Netherlands. Each actor is modelled as an agent, with static agents running on specific AgentScape locations representing physical locations. One of the additional services provided by AgentScape is a *Web Services* gateway module. The Web Service Gateway module enables agents to communicate with web services using the SOAP/XML protocol [10]. This allows the direct invocation of web services by agents. This module provides the means to access services, such as a mapping and routing service and weather services.

5.4 Analysis and Implementation

Modelling the entire crisis management infrastructure is a complex task that requires detailed knowledge of the organizational and coordination interactions of each of the stakeholders. This information is currently being assembled and represents the future direction of this work. However, a proof of concept implementation of a scenario has been developed using a GRIP-1 level crisis. In this scenario, a fire has started in a building. This building contains a number of injured people who need evacuation to a hospital as well as a fire that needs to be extinguished. An agent representing the building holds this knowledge. Other agents represent a unit of the fire service, a police unit and an ambulance. The ambulance agent has a location, and a capacity. The fire service agent has a fire fighting capability. The police agent has a number of responsibilities, including looking for emergencies, and coordinating any emergencies that it finds. Buildings, such as hospitals, fire stations and police stations, are all represented by *static* agents, that is agents that do not migrate from the location where they are started. These static agents act as parent agents to unit agents from their respective service, such as a fire service agent.

When the fire breaks out, the police agent is informed either through polling or through notification. The notification comes from an external service reporting the incident. The police agent migrates to the location where the incident is taking place. This police agent either determines what other services are required and issues a request for these services, or the notification contained enough specific details to require some additional services immediately. For example, when a notification occurs about a fire, fire service personnel are immediately dispatched to the incident. In this case, an ambulance agent and a fire service agent are requested. These requests are sent to the fire station agent and the hospital agent. The fire service agent and ambulance agent are instructed by their parent agents to migrate to the incident location. At the incident, the police agent acts as the coordinator and instructs the other agents to perform any tasks required. For example, the ambulance agent is instructed to evacuate casualties from the location to the hospital. This proof of concept implementation has been developed using AgentScape.

Each agent has a set of objectives. For example, the police agent has the objective "maintain order" and the ambulance agent the objective "save lives". These objectives are codified with specific tasks. In the case of the ambulance agent, one task is "evacuate people". When the ambulance agent arrives at the location, it is ordered to begin evacuation (by the police agent), it removes a number of people from the building, depending on the capacity of the ambulance, and migrates to the hospital location. When an agent migrates, the distance between the physical locations and the speed of the agent, defined in the agent are used to determine when the agent will arrive at the remote location.

Time is critical in any crisis simulation. In this case, time is broken into units representing one minute. Each action takes one or more units and is defined by the programmer. For example, evacuating a person takes a single time unit. Time is synchronized across all AgentScape locations by the coordinator. Similarly, the fire service agent can extinguish a specific amount of fire per unit of time. This capacity is used to determine how many fire service and ambulance agents are required by the police agent. This analysis is performed periodically by the coordinator agent.

5.5 Results

The results of the ALIVE process are used to implement the police agent and the overall scenario. The instructions are interpreted by the police and, in this case, building agents, and are executed. The landmark patterns determine sequence of actions that the police agent will order other agents to perform. For example, if a landmark pattern determines that all people must be evacuated before the fire can be extinguished, the fire service agent will wait for this condition to exist before putting out the fire. These patterns are utilized to simulate different strategies to achieve a set of goals.

This methodology supports rapid development of new scenarios while maximising the reuse of existing agents. Scenarios depend on a set of predefined agent types and a set of landmarks. The coordinator interprets these landmarks. Work is ongoing towards developing a complete set of agents that will allow the representation of more complex scenarios, such as the flooding scenario introduced in Section 2. This also involves the creation of a set of additional agent types to represent all of the actors involved in such scenarios. However, the proof of concept illustrates both the suitability of the agent paradigm to such problems and how to address these issues. In particular, the concepts of locations and migration directly model the real world actions that they represent.

6. CONCLUSIONS AND FUTURE WORK

This paper presents a model for an organization and an application that uses this model to simulate the response to crisis in the Netherlands. The organizational models allow the scenarios to be defined in a structured way. These scenario structures are then taken and implemented in the AgentScape system. Organization modelling provides the ability to determine where the relationships between stakeholders exist and how these relationships influence the results of a crisis. The organization model presented in this paper enables the explicit representation of both structural and strategic concerns and their adaptation to changes in the environment.

The implementation described in Section 5 is intended as

a proof of concept for simulation of a simple scenario. Ongoing development of the simulation environment, extending this scenario, will result in a generic structure for adaptation of communication structures supporting rapid construction of new scenarios. This generic structure will also allow the results of a reconfiguration to be automatically reflected in a running system. Furthermore, the integration of results from the co-ordination, presented in Section 3, will allow reconfiguration of organizations to be automatically simulated. The results of these simulations have the potential to inform change of the crisis management structures in the Netherlands.

One of the major issues with such simulations is with respect to timing and how time can be synchronized over a distributed system. While much work exists [1, 6] in this area, our own future work will address this issue in detail. Other areas of future work include the development of a visual representation of the scenario, including the use of web services to simulate, for example, routing of personnel and equipment to locations.

Acknowledgements

This work is a result of support provided by the ALIVE project (FP7-IST-215890) and the NLnet Foundation (`http://www.nlnet.nl`). The authors would like to acknowledge the contributions of their colleagues from ALIVE Consortium (http://www.ist-alive.eu). The authors are also grateful to Michel Oey for implementing the proof of concept.

7. ADDITIONAL AUTHORS

8. REFERENCES

[1] P. M. J. Christie and R. R. Levary. The use of simulation in planning the transportation of patients to hospitals following a disaster. *Journal of Medical Systems*, 22(5):289–300, October 1998. doi: 10.1023/A:1020521909778.

[2] V. Dignum. *A Model for Organizational Interaction: based on Agents, founded in Logic*. SIKS Dissertation Series 2004-1. Utrecht University, 2004. PhD Thesis.

[3] V. Dignum. Ontology support for agent-based simulation of organizational change. *International JournalMultiagent and Grid Systems*, to appear.

[4] I. Garcia-Magarino, C. Gutierrez, and R. Fuentes-Fernandez. Organizing multi-agent systems for crisis management. In L. Antunes, L. Moniz, and J. Pavao, editors, *7th Ibero-American Workshop in Multi-Agent*, pages 69–80, Lisbon (Portugal), 2008. Springer.

[5] A. Howitt and H. Leonard. Beyond katrina: Improving disaster response capabilities. *Taubman Center Policy Briefs*, PB-2006, 2006.

[6] J. Jenvald and M. Morin. Simulation-supported live training for emergency response in hazardous environments. *Simul. Gaming*, 35(3):363–377, 2004.

[7] H. Kitano. Robocup rescue: a grand challenge for multi-agent systems. In *Proc. ICMAS00*, pages 5–12, 2000.

[8] J. March and R. Sutton. Organizational performance as a dependent variable. *Organization Science*, 8(6):698–706, Nov. Dec. 1997.

[9] B. J. Overeinder and F. M. T. Brazier. Scalable middleware environment for agent-based internet applications. In *Proceedings of the Workshop on State-of-the-Art in Scientific Computing (PARA'04)*, volume 3732 of *LNCS*, pages 675–679, Copenhagen, Denmark, 2004. Springer.

[10] B. J. Overeinder, P. D. Verkaik, and F. M. T. Brazier. Web service access management for integration with agent systems. In *Proceedings of the 23rd Annual ACM Symposium on Applied Computing, Mobile Agents and Systems Track*, 2008.

[11] J. A. Padget, S. A. Ludwig, W. Naylor, and O. F. Rana. Matchmaking framework for mathematical web services. *Journal of Grid Computing*, 4 (1):33–48, Mar. 2006. 1570-7873.

[12] M. Sierhuis. *Modeling and Simulating Work Practice*. SIKS Dissertation Series 2001-10. University of Amsterdam, 2001. PhD Thesis.

[13] M. Tsvetovat and K. Carley. Simulation of human systems requires multiple levels of complexity. In *Proc. 1st World Congress on Social Simulation*, 2006.

[14] B. van der Vecht, F. Dignum, J.-J. C. Meyer, and M. Neef. A dynamic coordination mechanism using adjustable autonomy. In *Proc. COIN@MALLOW07*, Durham (UK), 2007.

[15] G. van 't Noordende, B. J. Overeinder, R. J. Timmer, F. M. T. Brazier, and A. S. Tanenbaum. A common base for building secure mobile agent middleware. In *Proceedings of the 2nd International Multiconference on Computer Science and Information Technology (IMCSIT)*, volume 2, pages 13–25, Wisła, Poland, Oct. 2007.