



## An Approach for Hospital Planning with Multi-Agent Organizations

Larsen, John Bruntse; Villadsen, Jørgen

*Published in:*  
Rough Sets

*Link to article, DOI:*  
[10.1007/978-3-319-60840-2\\_33](https://doi.org/10.1007/978-3-319-60840-2_33)

*Publication date:*  
2017

*Document Version*  
Peer reviewed version

[Link back to DTU Orbit](#)

*Citation (APA):*  
Larsen, J. B., & Villadsen, J. (2017). An Approach for Hospital Planning with Multi-Agent Organizations. In *Rough Sets* (Vol. 10314, pp. 454-465). Springer. Lecture Notes in Computer Science Vol. 10314 [https://doi.org/10.1007/978-3-319-60840-2\\_33](https://doi.org/10.1007/978-3-319-60840-2_33)

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# An Approach for Hospital Planning with Multi-Agent Organizations

John Bruntse Larsen and Jørgen Villadsen

DTU Compute, Technical University of Denmark, 2800 Kongens Lyngby, Denmark  
{jobla, jovi}@dtu.dk

**Abstract.** The background for this paper is a development that the Danish hospitals are undertaking which requires the establishment of a common emergency department. It is uncertain exactly what and how many resources the department needs and so resources are assigned dynamically as seen necessary by the staff. Such dynamic adjustments pose a challenge in predicting what consequences these adjustments may lead to. We propose an approach to deal with this challenge that applies simulation with intelligent agents and logics for organizational reasoning. We present some of the expected obstacles with this approach and potential ways to overcome them.

**Keywords:** multi-agent organizations, logic, simulation, soft computing, process mining

## 1 Introduction

One of the fundamental ideas behind multi-agent systems is that agents act autonomously but in practice the agents are often encoded with rules for coordination that limit their ability to do so. A recent approach to address this issue is by *agent organizations* [5]. The agents in an organization are aware of the norms of the organization but may choose to go against rigorous rules decided by the organization.

Our work is motivated by a recent development in how the Danish hospitals manage acute patients. We consider agent based simulation as a tool for forecasting delayed treatments and expected waiting times. We argue that agent organizations are appropriate for simulating human behavior because of the normative aspect: humans generally act according to the norms of the hospital and may act against rigorous rules that have been decided at a top level.

Inspired by the approach for agent simulation of an emergency department by Taboada et al. [10], we propose an approach in which we distinguish between three types of agents in the hospital organization: those are the *active*, the *passive*, and the *external* agents. For modeling the relationship between the agents we use AORTA, a logical framework for agent organizations developed by Jensen, Dignum and Villadsen [6]. We argue that a formalization of the framework in the proof assistant Isabelle/HOL [15, 16] can be useful in verifying properties of the framework and potential new extensions that we make during the project.

Finally we propose ideas on using KPIs to measure the ‘distance’ between the expected global behavior, as expressed in the organization model, and the actual global state as expressed by staff activity logs. The motivation is to automatically extend the organizational model in the simulation from the data that the hospital produces. In this way, the simulation should adjust itself to changes in the behavior of the people at a given hospital and reduce some of the complexity in the initial hand-crafted model.

## 2 Background

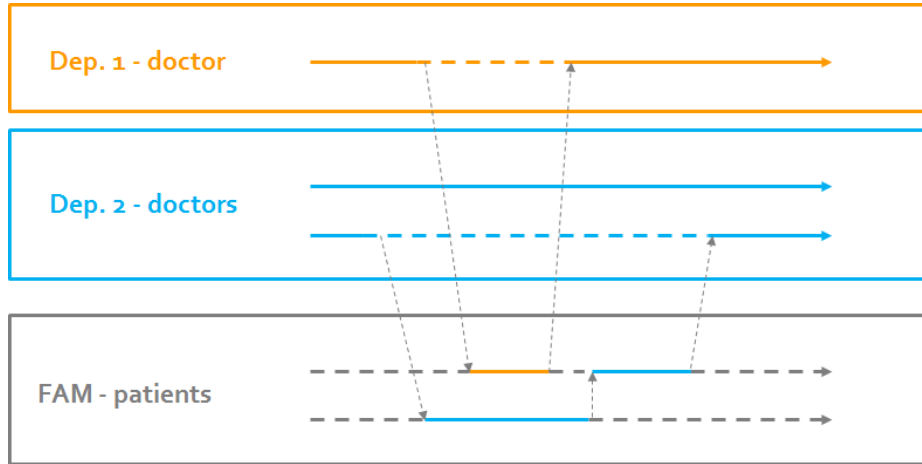
Traditionally, an emergency department takes care of acute patients and acts as an entrance to further treatment in hospital. The acute treatment is taken care of by acute doctors and the further treatment is taken care of by specialists. The Danish hospitals are undergoing a reform in which they establish a common emergency department (FAM, Danish: Fælles AkutModtagelse) where all acute patients can receive treatment from both specialists and acute doctors [1]. The vision is to put the patient in focus and plan the staff for the treatment of the patient. A straightforward way to achieve this vision would be to hire more staff for the FAM but it is not a feasible solution due to the cost and the constrained budget. As a consequence, the FAM draws on staff from the specialized departments that carry out operations, patient status check-ups and other scheduled activities. It also means that the scheduled activities may be delayed because the specialists are called to the FAM for an acute patient or that an acute patient may have to wait for a long time because a specialist is not available. The scenario is illustrated in Figure 1.

### 2.1 Agent-Based Simulation

In our work, we will attempt to simulate the consequences of the actions of the agents in the FAM and forecast likely delayed treatments and expected waiting times for the acute patients. Simulation has been found useful for planning physical resources and staff for the FAM as it produces more accurate results than traditional analytic approaches which tend to oversimplify the processes that go on in the department [1, 2]. Following the arguments presented at the UK Operational Research Societys Simulation Workshop 2010 [4] and in the work of Zhengchun Liu et al. [3], we believe that agent-based simulation is a promising alternative for simulating a complex system with conflicting goals like the FAM scenario.

### 2.2 Modeling Human Behavior

In general, hospital regulations describe best practice in an open manner and it is then up to the individual staff members to determine the exact work processes that also fit within the individual hospital. Thus the regulations are not sufficient for making a process model for an emergency department and a process



**Fig. 1.** A case in the FAM scenario. Two doctors from department 1 were called for FAM to assist with two acute patients. As a result, the treatments that they were scheduled for were delayed.

model that is accurate across different hospitals easily grows highly complicated to maintain manually. With the advent of powerful computers that allow analysis on big data, there has been an increasing focus on systems that learn from human behavior, and soft computing systems that are inspired by human behavior. Process mining is an approach for discovering process models, checking conformance of models and extending models for such systems based on event logs generated by an actual organization. In the context of healthcare, process mining has been studied with the purpose of providing insight into the complex system of a hospital that deals with a lot of human behavior and human values. Typically the hospital model is based on a top-down analysis of the processes that result from the interaction between the individuals in the hospital [11–13]. In our approach, we combine a top-down model that describes the organization with a bottom-up model that describes the interaction between individuals. Considering the high amount of uncertainty in the FAM environment that depends on eventualities and causalities, we expect our approach to provide better insight into the processes of the environment than with a pure top-down approach.

### 3 The Hospital as a Multi-Agent Organization

A multi-agent system is specified at two levels: the agent level and the system level. At agent level, an agent architecture defines the behavior of the agent. At system level, a framework defines the world that the agents are acting within. The framework includes the environment and interaction protocols. Ideally, the agent architectures and the system framework should be loosely coupled such

that new agents can enter the system with only local changes to the system, and so that the agents can act independently no matter what architecture they use.

A key challenge in multi-agent systems is to make the agents able to act in a coordinated fashion without limiting their ability to act autonomously. A recent approach to achieve this is by applying frameworks that model the interaction and dependencies between the agents as an organization. In a multi-agent organization, the agent is aware of its role in the organization and the norms for ‘common good practice’ in the organization. Agents can enter and exit the organization freely, and there is an explicit model of the expected behavior that the agents can choose to go against if deemed necessary [5]. At agent level, the agents distinguish between personal and organizational goals. At system level, there is a framework for defining expected behavior of the agents.

Given the complex flow of information and the independent actors in FAM, we believe that multi-agent organizations can provide insight into the relation between micro-behavior of the agents and the macro-behavior of the system in an agent-based simulation. In this section we propose our approach for modeling the hospital setting as a multi-agent organization. We define three types of agents, and introduce the framework that we later use to model the hospital organization.

### 3.1 Agent Model

As detailed below, we follow the analysis of Taboada et al. [10] that introduces two distinct types of agents for an emergency department. We have *active* agents that represent individuals and *passive* agents that represent services and other reactive systems. We also introduce a third distinct type of agent for the FAM scenario, the *external agent*, that represent an entire specialized department.

**Active agents** Active agents represent individuals that act on their own initiative toward achieving specific goals. A knee specialist that can diagnose the pains that a patient feels is an example of an active agent. For the active agents, we use a Belief-Desire-Intention (BDI) model that allows us inspect the beliefs that the agent has about its current state, the goals that it would like to achieve and the goal that it is currently working towards achieving. In each step of the simulation, the agent takes an input vector of percepts and messages from other agents, update its beliefs, desires and current goal, and then outputs an action toward achieving its current goal.

**Passive agents** Passive agents represent passive entities that only react to the other agents and they do not work toward achieving a goal on their own. An IT-system that the nurses must register data with is an example of a passive agent. For the passive agents we use a rule-based model, in which the agent has a rule base that maps messages to actions. In each step of the simulation, the agent takes an input vector of messages from other agents, selects a corresponding rule for each message and then outputs the vector of actions for all messages.

**External agents** External agents represent an entity that acts towards achieving vague goals. A specialized department that requests assistant nurses for a scheduled treatment is an example of an external agent. The agent is external from the point of view of the FAM. For the external agents we use a BDI model where the goals and actions are generated from a statistical model that corresponds to the average behavior of the agent. The actions that it outputs are in the form of requests for resources. In each step of the simulation, the agent takes an input vector of messages from other agents, updates its beliefs and goals, generates new goals from the statistical model and then outputs the vector of requests.

### 3.2 AORTA

We investigate the logical framework AORTA for modeling organization-aware agents as presented in [7, 8]. In the framework, the agents are assumed to be BDI-agents that each receives an additional module that allows it to include organizational beliefs and goals in its reasoning. The module defines three phases of organizational reasoning that are used in addition to the reasoning that the agent already uses: the *obligation check*, the *option generation*, and the *action execution*.

**Step 1: obligation check** The agent updates the status of its obligation state: it checks if an obligation has been satisfied (objective completed) or violated (deadline reached before objective completed). The agent also checks for new obligations.

**Step 2: option generation** The agent generates options for what it can do regarding the organization. It considers these aspects in the option generation: *role enactment*, *role deactment*, *obligations* (obligation state), *delegation* (based on role dependency relations), *information* (exchange).

**Step 3: action execution** The agent selects a single action to execute based on rules of the form

$$option : context \rightarrow action$$

By separating the organizational reasoning from the reasoning about the personal goals and beliefs, the agents are able to take into account how they are expected to behave, given the role they enact, while also able to reason about personal goals independently. In this fashion, the model of the organization is distributed among the agents so it is possible that the agents have different models of the organization.

The three steps of organizational reasoning in AORTA are based on an *organizational metamodel* defined by the predicates in Table 1. In a later section, we construct a metamodel that describes a simplified version of the situation in the FAM.

Predicate	Informal meaning
$\text{role}(\text{Role}, \text{Objs})$	<i>Role</i> is the name of a role and <i>Objs</i> is a set of <i>main</i> objectives of that role.
$\text{obj}(\text{Obj}, \text{SubObjs})$	<i>Obj</i> is an objective that has <i>SubObjs</i> as a set of sub-objectives.
$\text{dep}(\text{Role}_1, \text{Role}_2, \text{Obj})$	<i>Role</i> <sub>1</sub> depends on <i>Role</i> <sub>2</sub> in order to complete <i>Obj</i> .
$\text{rea}(\text{Ag}, \text{Role})$	Agent <i>Ag</i> enacts <i>Role</i> .
$\text{cond}(\text{Role}, \text{Obj}, \text{Deadline}, \text{Cond})$	When the condition <i>Cond</i> holds, <i>Role</i> is obliged to complete <i>Obj</i> before the objective <i>Deadline</i> .
$\text{obl}(\text{Ag}, \text{Role}, \text{Obj}, \text{Deadline})$	Agent <i>Ag</i> is obliged to enact <i>Role</i> to complete <i>Obj</i> before <i>Deadline</i> .
$\text{viol}(\text{Ag}, \text{Role}, \text{Obj})$	Agent <i>Ag</i> enacting <i>Role</i> has violated the obligation to complete <i>Obj</i> .

**Table 1.** Predicates of the AORTA metamodel and their informal meaning.

### 3.3 Formalization of AORTA in Isabelle/HOL

AORTA can be viewed as a large logical framework. A formalization of the most relevant parts of AORTA in a proof assistant like Isabelle/HOL [15,16] will allow us to verify the logical framework and will also be useful for showing properties of the logical framework. As we work with applying the logical framework for organizational reasoning, we may also find that we want to extend the framework with features that are needed or useful for the FAM scenario. For that purpose, the formalization will allow us to extend the framework in a verified manner. Verification of agents using AORTA has been investigated in [9].

## 4 Modeling a FAM in AORTA

In this section we present our approach to applying the AORTA framework to the FAM scenario. To get started, we make a metamodel of a FAM based on basic assumptions about what processes go on in and around the department with inspiration from the work by Taboada et al. [10] about a conventional emergency department. We will revise this model based on data from interviews and observations from an actual FAM; for this purpose we have a collaboration agreement with the hospitals in the capital area of Denmark.

Based on the descriptions of a FAM in [2] and the scenario described in [10] we assume that the FAM scenario consists of the following stages:

1. *Admission* Arrival of the patient in the department; check in at reception.
2. *Triage* A nurse carries out the triage process on the patient.
3. *Diagnosis and Treatment* A doctor performs a diagnosis and initial treatment on the patient.
4. *Round-up* The patient receives a plan for further treatment and leaves the department.

Additionally, we assume the following norms in the FAM scenario:

- a. Patients arrive in the admission area, either by their own means or by ambulance.
- b. Patients must wait in the admission area until they have been attended to by the reception.
- c. After the admission, patients must wait in a designated room until called by a triage nurse.
- d. The nurse who carries out the triage must fill out a triage form for the patient.
- e. After the triage, patients must wait in a designated room until called by a doctor.
- f. Patients are involved in making their plan for further treatment.
- g. The doctors in the specialized departments take care of scheduled treatments.
- h. The initial treatment of patients may require assistance from doctors from specialized departments.

We translate the informal description of the scenario (1-4) and (a-h) into a formal AORTA metamodel as follows. The complete AORTA metamodel is shown in Table 2. This metamodel is the basis for the AORTA module that allows each agent in the simulation to perform organizational reasoning.

#### 4.1 Roles

The roles in the metamodel are defined by the role-predicate. We use roles to formalize what kinds of actors are involved in the scenario and what their main objectives are. Stages 1-4 mention these roles and their objectives: a *patient* that receives treatment, a *receptionist* that admits patients, a *nurse* that carries out triage, and an *acute doctor* that carries out acute treatment and gives the patient a plan for further treatment. For example we formalize the patient role as such:

$$\text{role}(\textit{patient}, \{\textit{acute\_treatment}(\textit{Patient}), \textit{treatment\_plan}(\textit{Patient}, \textit{Plan})\})$$

Additionally, the norm (g) mentions a *specialized department* that carries out scheduled treatments and *specialized doctors* that work in the department. As mentioned in (h), the doctors are also sometimes needed in the acute department, but we choose to not model it as their main objective.

$$\begin{aligned} &\text{role}(\textit{specialized\_doctor}, \{\textit{scheduled\_treatment}(\textit{Department}, \textit{Patient})\}) \\ &\text{role}(\textit{specialized\_department}, \{\textit{scheduled\_treatment}(\textit{Department}, \textit{Patient})\}) \end{aligned}$$

#### 4.2 Objectives

The objectives in the metamodel are defined by the obj-predicate. We use objectives to formalize what objectives the roles have and what sub-objectives must be



solved with each objective. Stages 1-4 mention 4 objectives that are completed in sequence: (1) *admission*, (2) *triage*, (3) *acute treatment*, and (4) *treatment plan*. We define the sequence in AORTA as objectives that depend on the completion of the previous objective: (4) depends on (3), (3) depends on (2), (2) depends on (1), and (1) does not depend on any sub-objective so that an acute patient can be admitted at any time. For example we formalize the last step in the sequence as such:

$$\text{obj}(\text{treatment\_plan}(\text{Patient}), \{\text{acute\_treatment}(\text{Patient})\})$$

Additionally, the norm (g) mentions that there are also scheduled treatments that should be taken care of but the details about what those treatments involve have been omitted from the description. We model a scheduled treatment as an objective without sub-objectives:

$$\text{obj}(\text{scheduled\_treatment}(\text{Department}, \text{Patient}), \{\})$$

### 4.3 Dependencies

The dependencies in the metamodel are defined by the dep-predicate. We use dependencies to formalize which roles depend on other roles to complete their objectives. Stage 1 mentions that the *patient* depends on the *receptionist* in order to be *admitted*. Stage 2 mentions that the *patient* depends on the *nurse* in order to receive *triage*. Stages 3-4 mention that the *patient* depends on the *acute doctor* in order to receive *acute treatment* and a *treatment plan*. Finally norm (g) mentions that the *specialized department* depends on *specialized doctors* in order to carry out *scheduled treatments*. For example we formalize the dependency between the patient and the receptionist as such:

$$\text{dep}(\text{patient}, \text{receptionist}, \text{admission}(\text{Patient}))$$

### 4.4 Conditions

The conditions in the metamodel are defined by the cond-predicate. We use conditions to formalize the norms about how the roles are expected to complete their main objectives. Norms (a) and (b) mention that *patients* should *wait in the admission area* until they are *admitted* when they have *arrived by themselves or by ambulance*. We formalize this norm as such:

$$\text{cond}(\text{patient}, \text{wait\_in}(\text{Admission\_area}), \text{admission}(\text{Patient}), \\ \text{arrivedBy}(\text{Patient}, \text{Self}) \vee \text{arrivedBy}(\text{Patient}, \text{Ambulance}))$$

Norm (c) mentions that *patients* should *wait in a room* before *triage* when they have been *admitted*:

$$\text{cond}(\text{patient}, \text{wait\_in}(\text{Room}), \text{triage}(\text{Patient}), \text{admission}(\text{Patient}))$$

Norm (d) mentions that *nurses* should *fill in the triage form* before they finish the *triage* when a patient has been *admitted*:

$\text{cond}(\text{nurse}, \text{fill\_form}(\text{Patient}, \text{Nurse}), \text{triage}(\text{Patient}), \text{admission}(\text{Patient}))$

Norm (e) mentions that *patients* should *wait in a room* until they receive *acute treatment* when they have gone through *triage*:

$\text{cond}(\text{patient}, \text{wait\_in}(\text{Room}), \text{acute\_treatment}(\text{Patient}), \text{triage}(\text{Patient}))$

Norm (f) mentions that *acute doctors* should *involve the patient* when they make the *treatment plan* during the *acute treatment*:

$\text{cond}(\text{acute\_doctor}, \text{involve\_patient}(\text{Patient}, \text{Plan}),$   
 $\text{treatment\_plan}(\text{Patient}, \text{Plan}), \text{acute\_treatment}(\text{Patient}))$

Finally norm (h) mentions that *acute doctors* should involve *specialized doctors for specialized treatment* in the *acute treatment* if a *specialist is necessary*:

$\text{cond}(\text{acute\_doctor}, \text{specialized\_treatment}(\text{Patient}, \text{specialized\_doctor}),$   
 $\text{acute\_treatment}(\text{Patient}), \text{specialistNecessary}(\text{Patient}, \text{specialized\_doctor}))$

On top of using the AORTA metamodel we also will investigate KPIs for measuring the distance between the expected global behavior, as expressed by the organization in the model, and the actual global state as expressed by event logs. The goal of this investigation will be to repair the metamodel based on the event logs with the process mining tool *Prom*. In order to get the event logs for the evaluation, we need to analyze the current systems that they use to register activity and assess that the additional necessary activity can be registered in a feasible manner.

## 5 Related Work

Making autonomous agents has been a major focus in the academic *Multi-Agent Programming Contest* that has taken place each year since 2005. Each year the contest organizers adjust the contest to further promote solutions that take advantage of distributed decision making and autonomous agents. The winning team from 2016 used the multi-agent programming framework JaCaMo which combines the multi-agent programming frameworks of Jason and Cartago with the multi-agent organization framework of Moise made by Hübner et al. [14]. JaCaMo is based on the A&A approach which distinguishes between two types of entities: agents and artifacts. An agent is a *goal-oriented pro-active* entity where as an artifact is a *non-autonomous function-oriented* entity. The motivation for this approach is inspired by human organizations that are populated by humans who assume roles and are responsible to obligations and permissions of those roles in the organization, and artifacts that have a designated purpose in

---

role( <i>patient</i> , { <i>acute_treatment</i> ( <i>Patient</i> ), <i>treatment_plan</i> ( <i>Patient</i> , <i>Plan</i> )})	
role( <i>receptionist</i> , { <i>admission</i> ( <i>Patient</i> )})	1
role( <i>nurse</i> , { <i>triage</i> ( <i>Patient</i> )})	2
role( <i>acute_doctor</i> , { <i>acute_treatment</i> ( <i>Patient</i> ), <i>treatment_plan</i> ( <i>Patient</i> )})	3, 4
role( <i>specialized_doctor</i> , { <i>scheduled_treatment</i> ( <i>Department</i> , <i>Patient</i> )})	g
role( <i>specialized_department</i> , { <i>scheduled_treatment</i> ( <i>Department</i> , <i>Patient</i> )})	g
obj( <i>treatment_plan</i> ( <i>Patient</i> ), { <i>acute_treatment</i> ( <i>Patient</i> )})	4
obj( <i>acute_treatment</i> ( <i>Patient</i> ), { <i>triage</i> ( <i>Patient</i> )})	3
obj( <i>triage</i> ( <i>Patient</i> ), { <i>admission</i> ( <i>Patient</i> )})	2
obj( <i>admission</i> ( <i>Patient</i> ), {})	1
obj( <i>scheduled_treatment</i> ( <i>Department</i> , <i>Patient</i> ), {})	g
dep( <i>patient</i> , <i>receptionist</i> , <i>admission</i> ( <i>Patient</i> ))	1
dep( <i>patient</i> , <i>nurse</i> , <i>triage</i> ( <i>Patient</i> ))	2
dep( <i>patient</i> , <i>acute_doctor</i> , <i>acute_treatment</i> ( <i>Patient</i> ))	3
dep( <i>patient</i> , <i>acute_doctor</i> , <i>treatment_plan</i> ( <i>Patient</i> ))	4
dep( <i>specialized_department</i> , <i>specialized_doctor</i> , <i>scheduled_treatment</i> ( <i>Department</i> , <i>Patient</i> ))	g
cond( <i>patient</i> , <i>wait_in</i> ( <i>Admission_area</i> ), <i>admission</i> ( <i>Patient</i> ), <i>arrivedBy</i> ( <i>Patient</i> , <i>Self</i> ) $\vee$ <i>arrivedBy</i> ( <i>Patient</i> , <i>Ambulance</i> ))	a, b
cond( <i>patient</i> , <i>wait_in</i> ( <i>Room</i> ), <i>triage</i> ( <i>Patient</i> ), <i>admission</i> ( <i>Patient</i> ))	c
cond( <i>nurse</i> , <i>fill_form</i> ( <i>Patient</i> , <i>Nurse</i> ), <i>triage</i> ( <i>Patient</i> ), <i>admission</i> ( <i>Patient</i> ))	d
cond( <i>patient</i> , <i>wait_in</i> ( <i>Room</i> ), <i>acute_treatment</i> ( <i>Patient</i> ), <i>triage</i> ( <i>Patient</i> ))	e
cond( <i>acute_doctor</i> , <i>involve_patient</i> ( <i>Patient</i> , <i>Plan</i> ), <i>treatment_plan</i> ( <i>Patient</i> , <i>Plan</i> ), <i>acute_treatment</i> ( <i>Patient</i> ))	f
cond( <i>acute_doctor</i> , <i>specialized_treatment</i> ( <i>Patient</i> , <i>specialized_doctor</i> ), <i>acute_treatment</i> ( <i>Patient</i> ), <i>specialistNecessary</i> ( <i>Patient</i> , <i>specialized_doctor</i> ))	h

---

**Table 2.** Predicates of the AORTA metamodel for the FAM scenario. The “role” predicate defines the roles and the objective of each role. The “obj” predicate defines sub-objectives of each objective. Notice that the first four “obj” predicates form a sequence of objectives. The “dep” predicate defines which other roles a role depends on in order to complete an objective. The “cond” predicate defines conditional objectives that should be fulfilled before a role completes an objective. For example, the first predicate says that a patient should wait in the admission area until admission no matter if they arrive by themselves or by ambulance. The end of each line shows the part of the informal description the predicate corresponds to (1-4 or a-h).

the organization. Typical examples of such artifacts are whiteboards and telephones that the agents can use to coordinate with, and access cards that enables agents with certain permissions and obligations. In comparison, AORTA does not distinguish artifacts from agents at a fundamental level. They are considered primarily reactive agents and, like in JaCaMo, are designated with certain roles.

In 2009 Mans et al. [11] showed initial work on the feasibility of applying the prominent process mining tool *Prom* in the hospital environment with a focus on discovering a process model. In 2013 Kirchner et al. [12] noted the problem of sparse event logs that are common in the hospital environment which increases the importance of clearly defined clinical pathways in the hospital in order to apply process mining successfully. In the works of [11] and [12], the models produced and treated with *Prom* were based on highly procedural modeling languages which are difficult to fit across multiple hospitals with different execution paths. In 2015 Rovani et al. [13] proposed an approach that applied the declarative process modeling language *Declare* which is based on linear temporal logic. In this language, the model only specifies constraints within finite traces on the processes rather than concrete execution paths, which enables a *Declare* model to allow multiple execution paths. They applied a cross validation methodology for automatically creating a repaired model from a manually created model and an event log.

## 6 Conclusion and Future Work

We have introduced and modeled the new kind of emergency department, FAM, that is being implemented at Danish hospitals, as a multi-agent organization in the agent organization framework AORTA. Our goal with this approach is to use the model for simulating the activity that goes on in the department and calculate consequences based on the predicted behavior of the agents in the simulation. We have shown a model of the FAM in the framework. The model is based on previous work on agent simulation for emergency departments and the official descriptions of the FAM and its purposes.

In future work we will use proof assistants to verify properties of the AORTA framework and implement the AORTA model in an agent simulation framework. That way we may get a model that can adjust itself automatically to the soft aspects of human behavior that influence the activity in an actual department.

## Acknowledgements

This work is part of the Industrial PhD project *Hospital Planning with Multi-Agent Goals* between PDC A/S and Technical University of Denmark. We are grateful to Innovation Fund Denmark for funding and the governmental institute Region H, which manages the hospitals in the Danish capital region, for being a collaborator on the project. We would like to thank PDC A/S for providing feedback on the ideas described in this paper. We would also like to thank Virginia Dignum and Anders Schlichtkrull for comments on a draft.

## References

1. Kjeld Møller Pedersen and Niels Christian Petersen. Fremtidens Hospital, Munksgaard, 2014
2. M. Hansen-Nord, J. P. Steensen and S. Holm. Process driven patient tracks in FAM. *Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine*, 18(Suppl. 1):P27, 2010
3. Zhengchun Liu, Eduardo Cabrera, Dolores Rexachs, Francisco Epelde and Emilio Luque. Simulation Modelling in Healthcare: Challenges and Trends, Simulating the Micro-level Behavior of Emergency Department for Macro-level Features Prediction, Winter Simulation Conference, 2015
4. Peer-Olaf Siebers, C. M. Macal, Jeremy Garnett, D. Buxton and Michael Pidd. Discrete-event simulation is dead, long live agent-based simulation! *Journal of Simulation*, Volume 4, 2010
5. Gerhard Weiss, editor. *Multiagent Systems*. MIT Press, 2nd Edition, 2013
6. Andreas Schmidt Jensen, Virginia Dignum and Jørgen Villadsen. A Framework for Organization-Aware Agents. *Autonomous Agents and Multi-Agent Systems*, pp. 1-36, 2016
7. Andreas Schmidt Jensen and Virginia Dignum. AORTA: Adding Organizational Reasoning to Agents. In *International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS*, pp. 1493–1494, 2014
8. Andreas Schmidt Jensen, Virginia Dignum and Jørgen Villadsen. The AORTA Architecture: Integrating Organizational Reasoning in Jason. Volume 8758 of *Lecture Notes in Computer Science*, pp. 127–145. Springer, 2014
9. Andreas Schmidt Jensen. Model Checking AORTA: Verification of Organization-Aware Agents. *CoRR*, 1503.05317, 2015
10. Manel Taboada, Eduardo Cabrera, Ma Luisa Iglesias, Francisco Epelde and Emilio Luque. An Agent-Based Decision Support System for Hospitals Emergency. Volume 4 of *Procedia Computer Science*, pp. 1870–1879, 2011
11. R. S. Mans, M. H. Schonenberg, M. Song, W. M. P. van der Aalst and P. J. M. Bakker. Application of Process Mining in Healthcare – A Case Study in a Dutch Hospital. *Biomedical Engineering Systems and Technologies: International Joint Conference, BIOSTEC 2008 Funchal, Madeira, Portugal*, pp. 425–438, Springer, 2009
12. Kathrin Kirchner, Nico Herzberg, Andreas Rogge-Solti and Mathias Weske. Embedding Conformance Checking in a Process Intelligence System in Hospital Environments. *Process Support and Knowledge Representation in Health Care: BPM 2012 Joint Workshop, ProHealth 2012/KR4HC 2012, Tallinn, Estonia*, pp. 126–139, Springer, 2013
13. Marcella Rovani, Fabrizio M. Maggi, Massimiliano de Leoni and Wil M. P. van der Aalst. Declarative Process Mining in Healthcare. *Expert Systems with Applications*, Volume 42, pp. 9236–9251, 2015
14. Jomi F. Hübner, Olivier Boissier Rosine Kitio and Alessandro Ricci. Instrumenting Multi-Agent Organisations with Organisational Artifacts and Agents. *Autonomous Agents and Multi-Agent Systems*, Volume 20, pp. 369–400, 2010
15. H. Geuvers. Proof Assistants: History, Ideas and Future. *Sadhana*, 34(1):3–25, 2009
16. T. Nipkow, L. C. Paulson and M. Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*. *Lecture Notes in Computer Science* 2283, Springer, 2002