UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Designing a search and rescue simulation environment for studying the performance of agent organizations

Ghijsen, M.; Jansweijer, W.; Wielinga, B.

Publication date
2012

Document Version
Final published version

Published in
BNAIC

Link to publication

Citation for published version (APA):
Ghijsen, M., Jansweijer, W., & Wielinga, B. (2012). Designing a search and rescue simulation environment for studying the performance of agent organizations. *BNAIC*, *24*, 115-122. http://www.unimaas.nl/bnaic2012/BNAIC-2012-Proceedings.pdf

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:09 Mar 2023

Designing a Search and Rescue Simulation Environment for Studying the Performance of Agent Organizations

Mattijs Ghijsen [a] Wouter Jansweijer [a] Bob Wielinga [a]

[a] *System and Network Engineering Group, Informatics Institute, University of Amsterdam*

Abstract

In the study on performance of organizations of Multi-Agent Systems there exists a need to understand the effects of the task-environment and organization of the agents on the performance of Multi-Agent Systems. Current simulation environments often lack sufficient control over the environment and lack the ability to systematically vary a number of task-environment and organizational parameters and measure the effect of these changes on performance. For this purpose we have created the Extended Organization Design model which categorizes and describes aspects of Multi-Agent Systems; their organization, their task-environment, and a set of performance metrics. We show how the Extended Organization Design is used as a basis for a parameterized model of the Search and Rescue domain.

1 Introduction

For agents in a Multi-Agent System (MAS) to cooperate effectively and efficiently, organization is required. To study and understand the performance of such organizations, simulation tools can be used. In this paper we address the issue of constructing simulation environments that allow for a systematic analysis of multi-agent organization performance. We have chosen Search and Rescue (S&R) as an application domain due to its challenges for MAS research such as its distributed and cooperative nature and high degree of uncertainty and dynamics.

A number of simulation platforms have already been developed for simulations in the disaster management domain. Well known is the RoboCup Rescue simulation system (RCRSS) [1] which aims at comparing the performance of a number of different MAS organizations in exactly the same setting. However it lacks easy manipulation of the task-environment. Other simulation environments such as the Urban Search And Rescue simulator (USARsim) and the distributed building evacuation simulator [2, 3] also provide realistic simulation environments but lack sufficient control to manipulate the task environment.

An example of a simulation environment that provides more control to the user is the predator-prey pursuit simulation system [4]. In this system, an explicit mathematic model is provided to describe the predator prey domain. Another example of a more controllable environment is a system for simulating software evolution [5]. Although no explicit environment model is presented and the application domain is completely different from ours, their simulator allows for the systematic variation of a number of parameters and the authors use use a clear methodological approach to analyze the results. The latter is a good example of the type of experiments we envision for our simulation environment.

So and Durfee [6, 7] present a more systematic on studying MAS performance. More specifically they present an organization design model in which the performance of a MAS is influenced by the task-environment and the organizational factors of the MAS. Moreover, they recognize that interaction effects exist between the task-environment and MAS organization factors. Their model however does not provide specific task-environment factors and MAS organization factors. Virginia Dignum [8] and Frank Dignum [9] present their approach for the design of a simulation tool for studying MAS reorganization. They first identify the factors that determine the need for organization. Then they explore the different ways of reorganization and finally they identify the different triggers for reorganization. Based on this generic framework for reorganization a simulation environment for reorganization is defined. In our approach we combine the basic framework presented by So and Durfee with the design approach by Dignum et al. to describe a methodology for designing MAS simulation environments that can be used for a systematic analysis of MAS performance.

In this paper we present a methodology that consists of a theoretical framework, the Extended Organization Design (EOD) model, and an approach for using the EOD to design a simulation environment. In Section 2 we discuss the EOD model which is based on the organization design model by So and Durfee. The EOD extends the organization design model with a vocabulary to describe the MAS organization and the task-environment in which the MAS organization operates. Furthermore, we provide a more detailed performance model that distinguishes between effectiveness and efficiency and provide a set of performance metrics. In Section 3 we demonstrate our approach to operationalize the generic factors of the EOD for a Search and Rescue Simulation environment. Next, in Section 4 we show how the simulation environment is used in an experiment to analyze the impact of communication failure on the performance of a MAS organization.

2 Extended Organization Design Model

The model of organization design by So and Durfee [7] explains the interaction between a MAS organization and its task-environment and the effect of this interaction on the performance of a MAS.

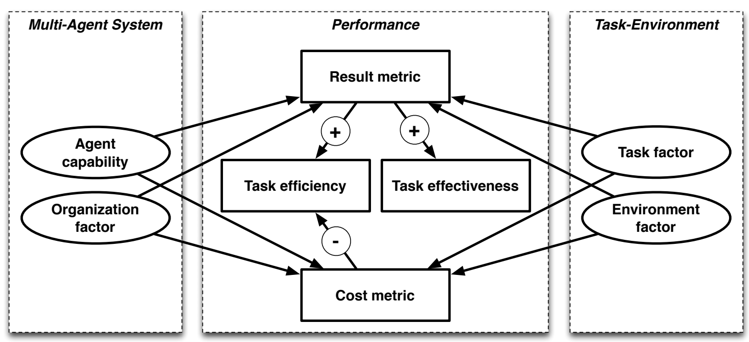


Figure 1: The Extended Organization Design Model.

In this paper we extend the organization design model with a vocabulary for describing a MAS organization, its task-environment and MAS performance metrics. In the latter we identify a number of result oriented and cost oriented metrics. Another extension of the organization design is a more detailed performance model in which we distinguish between effectiveness and efficiency and we couple these concepts to the result and cost oriented performance metrics. Our Extended Organization Design (EOD) model is shown in Figure 1.

2.1 EOD Performance model

```
Performance metric
   |--- Result Metric
           |--- Solution quality
           |--- Time-to-goal-achievement
   |--- Cost Metric
           |--- Resource consumption
           |--- Communication costs
```

Figure 2: Performance metrics

The EOD performance model distinguishes between result oriented and cost oriented metrics. Based on these two types of metrics, we define effectiveness as the ratio between the obtained result while trying to achieve a goal or performing a task and the maximum obtainable result. Efficiency is defined as the ration between the obtained results and the costs that have been made while trying to achieve a goal or performing a task. Figure 2 shows the different performance metrics in the EOD model. We distinguish between the four types of performance metrics described in [10], solution quality, time-to-goal-achievement, resource consumption and communication-costs.

2.2 EOD Task-Environment Model

```
Task factor
   |--- Task size
   |--- Task complexity
           |--- Task decomposability
           |--- Subtask heterogeneity
           |--- Inter-subtask relations
           |--- Subtask distribution
   |--- Task reward
   |--- Task dynamics
```

Figure 3: Task factors

Figure 3 shows the different task factors of the EOD model. We distinguish between the size of the task, factors that determine the complexity of the task, the reward that can be received by performing the task and the task dynamics. Task size relates to the amount of work that needs to be performed. Task complexity describes how easily a task can be composed into subtasks, the heterogeneity of the subtasks and the relations between subtasks.

The reward of a task describes the amount of reward that can be obtained by an agent or its organization if a task is performed. Dynamics in the task size, complexity or reward, require the organization and agents in the organization to constantly adjust their planning and may also lead to more uncertainty in the organization when agents are not able to keep up with dynamics in their tasks.

```
Environment factor
   |--- Communication factor
           |--- Capacity
           |--- Reliability
   |--- Resource factor
           |--- Scarceness
           |--- Distribution
           |--- Types
   |--- Topology factor
           |--- Size
           |--- Accessibility
   |--- Behavior factor
           |--- Observability
           |--- Determinism
           |--- Dynamics
```

Figure 4: Environment factors

The EOD's environment factors are shown in Figure 4. The first factor shown is *communication* and we identify the capacity and reliability of the communication infrastructure as its two main aspects. The next factor is the resource factor. Resources can be described in terms of their scarceness, how they are distributed over the environment and the different types of resources (e.g. consumable or reusable). The third factor is the topology factor of the environment. This factor is defines the size of the environment and the accessibility of the environment. Finally, we define a number of behavior factors of the environment. The observability of the environment – which can be full or partial – is related to whether or not relevant information for decision making can be observed by the agents in the environment. The determinism factor indicates if the outcome of an agent action in a certain state will always result in the same next state or not. Dynamics in the environment determine how the environment changes "spontaneously" without any agent action causing the change.

2.3 EOD Multi-Agent System Model

```
Agent factor
   |--- Physical capability
   |--- Knowledge
           |--- Declarative knowledge
           |--- Procedural knowledge
```

Figure 5: Agent factors

The agent factors, shown in Figure 5, consist of two main aspects: the physical capabilities and the agents knowledge. The physical capabilities determine how the agent interacts with its environment, how (well) the agent observes its environment and which actions is the agent able to perform on the environment. The knowledge of an agent is consists of declarative knowledge and procedural knowledge.

```
Organization factor
   |--- Organization size
   |--- Organization heterogeneity
   |--- Organization structure
           |--- Communication structure
           |--- Normative structure
           |--- Social structure
           |--- Interaction structure
```

Figure 6: Organization factors

The organization factors, shown in Figure6 consists of three main aspects: the size, the agents that form the organization and the structure of the organization. Organization size determines the amount of work that could potentially be done by an organization. We define the heterogeneity of a MAS organization by the heterogeneity of the agents that form the organization. Agents may have different physical capabilities as well as different knowledge. Following [11], we identify the following structural factors of a MAS organization: the communication structure, the normative structure, the social structure and the interaction structure.

2.4 Using the EOD model

The EOD model in this section provides a framework for the designer of a MAS environment. Because the covers a wide range of factors, it helps the designer to be explicit about the design choices that are made. By being explicit about the design choices, the designer provides more insight in the environment to the user.

To implement of the EOD factors in a specific domain we identify two steps. The first is the operationalization step in which an EOD factor is represented by a more specific concept. For example the task size factor can be operationalized as the number of actions that need to be performed in order to complete the task. The second step is the implementation step in which an operationalized factor is parameterized. For example, in the Search and Rescue domain, the number of actions to complete a task can be implemented by two parameters: the number of victims that need to be rescued and the size of the search area.

3 Environment Design

In applying the EOD model to design a search and rescue simulation environment we have to operationalize and implement the EOD factors. To balance the amount of realism and the needed simplicity for a controllable environment we often had to apply two operationalization steps at once. First we introduce search and rescue factors to operationalize the EOD factors. At the same time these operationalized S&R factors are often also a simplification of the real world search and rescue domain. The goal of this simulation environment is to provide a parameterized simulation environment that allows for systematic variation of (mainly) task-environment and (partly) MAS organization parameters. It is not the intention to provide a complete instantiation of the EOD model. A complete implementation of the EOD model, i.e. at least one parameter for each task-environment and organization factor and at least one performance metric for each of the generic performance metrics, is out of the scope of this research.

The S&R simulator is a discrete-time simulator. The main motivation for discrete-time is that this makes it easier to implement a system with reproducible results. The environment consists of a rectangular grid topology on which a number of victims are distributed. We have chosen for a rectangular grid to limit agent movements to just 4 directions and make the speed in which the agents move around more controllable. The victims have a certain health status which may decline over time. The initial health state and the decline of health represents how serious a victim is injured. Victims have a fixed location and cannot move. In order to rescue a victim, agents first have to find the victim and then cooperate to rescue the victim by jointly performing a rescue action in the same time-step. In order to find victims, agents can move around on the grid. In a single time-step, an agent can move either one grid cell up, down, left or right. When an agent is moving around, it is able to observe the grid cells that are within its viewing range. These observations are always accurate. Actions related to rescuing a victim and moving around the search area are deterministic. In order to cooperate, agents may need to communicate with each other. To facilitate communication, the simulator provides the agents with a wireless communication infrastructure. Actions related to communication are non-deterministic due to possible failures in the communication infrastructure. The simulator environment is partly observable, i.e. agents cannot see all relevant information needed for their decision making. For example, agents cannot observe whether a communication tower is operational or not.

The communication infrastructure is a simplified wireless communication network which covers the complete search area. Whenever an agent sends out a message, that message is picked up and sent to the receiver(s). Three types of messages are available to the agent; unicast, multicast and broadcast messages. A broadcast message is sent to all agents on the search area. In the case a directed (unicast or multicast) message is used, the sender has to specify one or more receivers of that message. Each time-step, an agent is allowed to send one message and the message size is limited. The reliability of the network is determined by the uptime of the network. Whether the network is up or down is determined each discrete time step with a probability ranging between 0% and 100%.

The physical capabilities of an agent are defined by its viewing range and the maximum amount of messages the agent can receive per time-step. The simulator does not impose any constraints and does not influence any of the knowledge factors of an agent.

For the Search and Rescue domain, many different types of performance metrics are possible. In our simulator we support three result metrics: the total reward that is received (i.e. the summed health of all victims at the end of a simulation), the amount of victims that are rescued and the amount of time taken to rescue all victims. Furthermore, the simulator supports two cost metrics that both focus on communication-costs: the amount of bytes that are sent and the amount of bytes received by agents.

4 Evaluation

To demonstrate the use of the simulator, we describe a case-study on the performance of a MAS organization. First we describe the MAS organization that coordinates using mutual adjustment. Next, we describe the design, data gathering and analysis of the performance evaluation study on the influence of network reliability and workload on performance.

4.1 Organization Design

The organization that we have designed uses a coordination mechanism that can be characterized as mutual adjustment [12]. This means the agents form a decentralized organization in which agents mutually adjust their actions to each other in order to perform their tasks. The interaction mechanism that is used is similar to a Contract Net [13]. The Contract Net provides a generic mechanism for communicating bids for cooperating on a task, the content of those bids and the offers that other agents can send.

In this organization, an agent can rescue a victim in two ways: the agent can decide to form a coalition for rescuing the victim, or the agent can decide to join a coalition. Forming a coalition consists of the following steps: The agent sends a request for forming a temporary coalition. Other agents can respond to this request by sending an offer to join the coalition. If the coalition accepts the offer, the coalition is formed and the agents will rescue the victim at the agreed time.

The messages that are used in this interaction are `<request>`, `<offer>` and `<accept>` and the content of these messages is shown in Figure 7. A `<request>` message consists of an expiration time which indicates how long the request is valid, an action-window which is the time-window in which the

```
<content>  ::= <request> | <offer> | <accept>;
<request>  ::= exp-time, action-min, action-max, victim-x, victim-y;
<offer>    ::= exp-time, action-min, action-max;
<accept>   ::= rescue-time;
```

Figure 7: Messages types and content.

action should take place and the coordinates of the victim. When the expiration time (`exp-time`) time has expired, the sender and receivers of this message will no longer consider this offer. The action-window allows other agents to decide if they will be able to join the coalition in time at the given location of the victim. If the agent decides to join a coalition, the agent will send an offer with a limited expiration time and the agent indicates it availability by an action-window with is a subset of the action-window in the texttt¡request¿ message. When the requesting agent has received sufficient offers, the agent will then send the accept message to the agents that will form the coalition. This accept message contains the time-step in which the rescue action should take place.

In this communication scheme, `<request>` messages are broadcasted while `<offer>` and `<accept>` messages are directed messages (unicast and multicast respectively). Furthermore, to prevent agents from flooding the communication infrastructure by broadcasting requests, each agent is only allowed to have one valid outstanding request.

4.2 Evaluation Setup

The goal of this evaluation is to study how the reliability of the communication infrastructure affects the solution quality performance of the aforementioned MAS organization. When the network uptime is less than 100%, two types of events can occur in the MAS organization's interaction pattern due to communication failure. First, when the communication network is down for one or only a few time-steps, agents are still able to respond to each others messages before these messages expire. When the communication network is down for longer periods of time, agents will not be able to respond to messages before they expire.

Based on these two delays we hypothesize that when the uptime of the network decreases, the first type of delay will start to occur in the MAS organizations's interaction pattern and performance will drop. Then, when we further decrease the uptime of the network, the second type of delay will also start to occur. This will cause a more severe drop in performance. Once the uptime of the network reaches 0%, performance will also drop to 0.

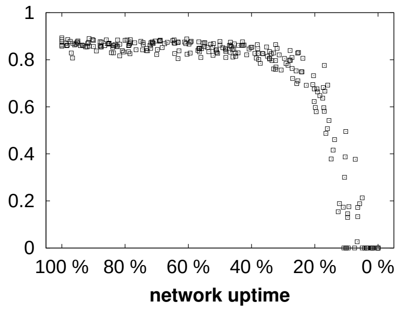
4.3 Results

Data for this evaluation was gathered by varying the network uptime between 0% and 100% with a 2% step size. Each simulation was done on a 30×30 search area. The number of victims was varied from 15, to 60 to 120 and the initial health state of a victim was set to 100 and the health decreased with 0.2 every time-step. Furthermore, the organization consisted of 30 S&R agents, each with the same observability range (5×5 range) and the same receive capacity of 100 messages per time-step. Each simulation is initialized with a different random seed which causes a different distribution of victims, different initial agent positions, and different timing of network failure.

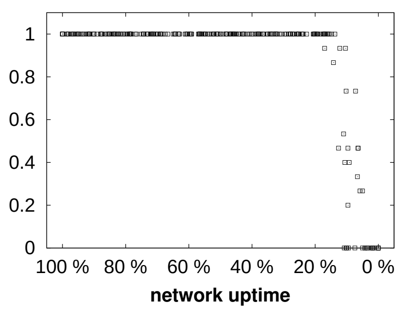
The results of the simulations are shown in Figure 8. We measure the effectiveness by measuring two performance metrics, the total victim health at the end of a simulation and the number of victims rescued during a simulation. For the first performance metric, effectiveness is obtained by dividing the total victim health at the end of the simulation with the total initial health of all victims. For the second performance metric, effectiveness is obtained by dividing the total number of victims that are rescued at the end of the simulation by the total number of victims in the simulation.

The results show the expected decrease in effectiveness for victim health and number of victims rescued when the network uptime decreases.

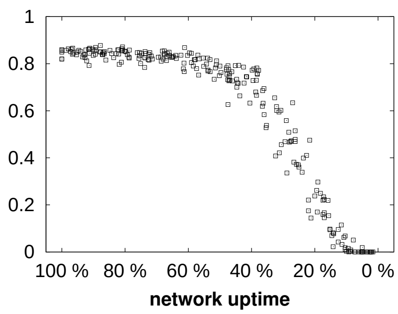
When we look at the total victim health, it shows that the initial decrease in effectiveness is relatively slow. This can be explained because at high uptime values, communication failures mostly cause small delays. At a certain point however, larger delays are caused by larger periods of network downtime. When the workload per agent is relatively low, the agents still manages to rescue a lot of victims despite the network



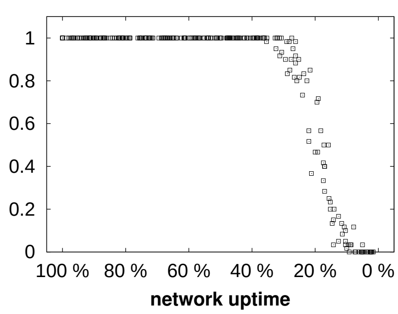
(a) total victim health (15 victims)



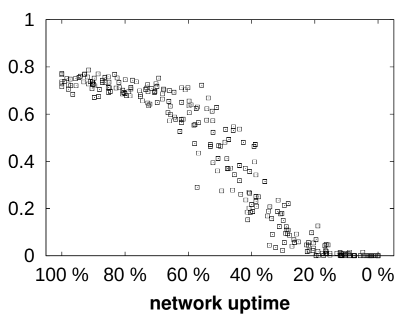
(b) number of victims rescued (15 victims)



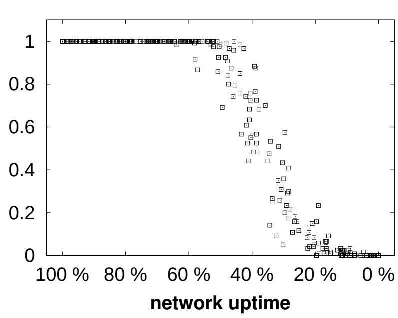
(c) total victim health (60 victims)



(d) number of victims rescued (60 victims)



(e) total victim health (120 victims)



(f) number of victims rescued (120 victims)

Figure 8: Influence of network uptime on effectiveness for total victim health and number of victims rescued.

downtime. However as the workload increases, the delays caused by network downtime prevent the agents from rescuing victims quickly and their total health decreases.

Furthermore, when we look at the number of victims being rescued, it is clear that the uptime of the network influences the maximum number of victims that can be rescued. 15 victims can still be rescued when the uptime is around 15%, 60 victims can be rescued when the uptime is around 30%, while 120 victims can still be rescued when the uptime is around 50%. This indicates a non-linear relation between the uptime and the number of victims that can be rescued.

5 Conclusions

In this paper we present a methodology for the systematic design of simulation environments. Our methodology consists of the Extended Organization Design model which is a domain-independent model to describe organizations of agents, the task-environment in which they operate and how performance is influenced by task-environment and organization factors. The EOD model provides a structure of and vocabulary for task-environment factors, MAS organization factors and performance metrics. Furthermore, we provide an two step approach for implementing the EOD factors as parameters in a simulation environment.

We have used our methodology to design an agent simulation environment for the Search and Rescue domain. The main aim was to create a controllable experimentation environment for conducting experiments on the performance of Multi-Agent Organizations. In an experiment we show how the simulation environment is used to analyze the effect of communication failure with different levels of workload on the performance of a MAS organization. In this analysis we use two complementary measures of effectiveness to understand the agents behavior when communication is failing.

References

[1] Hiroaki Kitano, Satoshi Tadokoro, Itsuki Noda, Hitoshi Matsubara, Tomoichi Takahashi, Atsuhi Shinjou, and Susumu Shimada. RoboCup-Rescue: Search and Rescue for Large Scale Disasters as a Domain for Multi-Agent Research. In *Proceedings of IEEE Conference on Man, Systems, and Cybernetics(SMC-99)*, pages 739–743, 1999.

[2] Nikolaos Dimakis, Avgoustinos Filippoupolitis, and E. Gelenbe. Distributed Building Evacuation Simulator for Smart Emergency Management. *The Computer Journal*, 53(9):1384–1400, February 2010.

[3] Avgoustinos Filippoupolitis and Erol Gelenbe. A distributed decision support system for Building Evacuation. In *2009 2nd Conference on Human System Interactions*, pages 323–330. Ieee, May 2009.

[4] Javier A Alcazar and Epharahim Garcia. Parametric Analysis for Modeling and Simulation of Stochastic Behavior in the Predator Prey Pursuit Domain. *Simulation*, 82(12):827–840, December 2006.

[5] Benjamin Stopford and Steve Counsell. A Framework for the simulation of Structural Software Evolution. *ACM Transactions on Modeling and Computer Simulation*, 18(4):17–45, September 2008.

[6] Young-pa So and Edmund H Durfee. Designing Tree-Structured Organizations for Computational Agents. *Computational and Mathematical Organization Theory*, 2(3):219–246, 1996.

[7] Young-pa So and Edmund H Durfee. Designing Organizations for Computational Agents. In M Prietula, K Carley, and L Gasser, editors, *Simulating Organizations*, pages 47–64. AAAI Press/ MIT Press, 1998.

[8] Virginia Dignum, Frank Dignum, Vasco Furtado, Adriano Melo, and Liz Sonenberg. Towards a Simulation Tool for Evaluating Dynamic Reorganization of Agents Societies. In *Workshop on Socially Inspired Computing. AISB Convention 2005.*, 2005.

[9] Frank Dignum, Virginia Dignum, and Liz Sonenberg. Exploring Congruence Between Organizational Structure and Task Performance: A Simulation Approach. In *AAMAS Workshops*, pages 213–230, 2005.

[10] Mattijs Ghijsen, Wouter N H Jansweijer, and Bob J Wielinga. Agent Decision Making for Dynamic Selection of Coordination Mechanisms. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 87–91, 2008.

[11] Virginia Dignum. *A Model for Organizational Interaction: Based on Agents, Founded in Logic*. PhD thesis, Universiteit Utrecht, 2003.

[12] H Mintzberg. *Structures in fives: designing effective organizations*. Prentice Hall, Englewood Cliffs, N.J., 1993.

[13] R Smith. The contract net protocol: high level communication and control in distributed problem solver. *IEEE Transactions on Computers*, 29:1104–1113, 1980.