Agent decision making for dynamic selection of coordination mechanisms

Ghijsen, M.; Jansweijer, W.N.H.; Wielinga, B.J.

Publication date
2008

Published in
Proceedings: 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - workshops: WI-IAT workshops 2008: 9-12 December 2008, University of Technology, Sydney, Australia

2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology

Agent Decision Making for Dynamic Selection of Coordination Mechanisms

Mattijs Ghijsen Wouter N.H. Jansweijer Bob J. Wielinga
University of Amsterdam, Human Computer Studies Laboratory
Kruislaan 419, Matrix 1, 1098 VA Amsterdam, the Netherlands
{m.ghijsen, w.n.h.jansweijer, b.j.wielinga}@uva.nl

Abstract

In this paper we present a decision making framework to enable agents to dynamically select a coordination mechanism. To demonstrate our approach we introduce an abstract task environment in which agents have to cooperate to achieve their goals. These agents are capable of using two coordination mechanisms, a centralized and a decentralized mechanism. We show how the decision making framework is operationalized in this abstract task environment. Furthermore, in an experiment we compare the performance of two static organizations with an organization in which agents have the ability to switch between coordination mechanisms. Results show that the ability to switch improves performance of the MAS.

1 Introduction

The organizational design of a Multi-Agent System (MAS) together with the task-environment in which the MAS is embedded, are two main aspects which determine its performance [8]. In a dynamic environment, a MAS may encounter new situations where its organizational design is no longer the most effective. Possible negative effects caused by dynamics in the task-environment can be mitigated or reduced by continuously adapting the organization of a MAS to changes in the task-environment [1]. Research by Excelente-Toledo and Jennings [2], Martin and Barber [6], and Rosenfeld et al. [7] has shown that enabling agents to dynamically select a coordination mechanism yields better performance compared to agents using a fixed coordination mechanism.

In [2], agents are enabled to dynamically select the coordination mechanism for a collaborative task. However these agents cannot dynamically switch between coordination mechanisms because the agents stick to the selected mechanism for the duration of the task. In [6] this is addressed and agents are given the capability to switch between coordination mechanisms while executing a single task. However the problem of how to determine the costs of the available coordination mechanisms in complex domains remains unsolved in both approaches. In [7] this issue is addressed by using a heuristic approach for determining the expected costs of coordination mechanisms.

Existing research on dynamic selection of coordination mechanisms, as discussed in the previous paragraph, has in common that it does not show the actual decision making process in which agents decide whether to switch, which coordination mechanism to use, and how to switch to a different coordination mechanism. Therefore we address this issue in our paper by presenting such a process and identify the domain independent elements in that process. By separating domain independent and domain dependent knowledge, we increase understanding of the dynamic selection of coordination mechanisms and enable a MAS designer to focus on solving domain specific issues.

In the next section we present our domain independent decision making framework for selecting coordination mechanisms. In order to demonstrate how domain dependent knowledge can be operationalized, Section 3 describes an environment in which the agents have the ability to perform two different coordination mechanisms. In Section 4 we show the operationalization of the domain dependent aspects of the decision making framework in this environment. Section 5 presents an experiment in which we show the effects of applying dynamic selection of coordination mechanisms. We end this paper with our conclusions and directions for future work in Section 6.

2 Decision Making

This section describes a decision making process to enable agents to switch between coordination mechanisms. The process consists of four decisions; (1) whether to initiate the decision making process, (2) which criteria to use to compare different coordination mechanisms, (3) which coordination mechanism is the best, and in the case the current coordination mechanism is no longer the most suitable, (4) how to change the organization of the MAS to another

978-0-7695-3496-1/08 $25.00 © 2008 IEEE
DOI 10.1109/WIIAT.2008.133

coordination mechanism.

The first decision to be made by an agent is a meta level decision about whether to continue the decision making process or not. This depends on the role of the agent in the organization of a MAS, the environment in which the agent is embedded and the agents view on that environment. An example of the organizational context of an agent influencing the outcome of this decision can be seen in centralized organizations. In this case, the decision to switch to another coordination mechanism is usually only made by agents in the top of the hierarchy. Deciding to continue the decision procedure may also be triggered by the occurrence of an event in the agents environment. In general, more accurate information will lead to better decisions and therefore, an agent should only continue the decision procedure when the quality of information is sufficient.

The second decision in the decision making process is the selection of the criteria that are used to compare coordination mechanisms. This step is included to enable agents to be flexible in the way coordination mechanisms are compared. This need for flexibility, also recognized by [2], is motivated by the fact that in some situations we want to select a coordination mechanism that is fast while in other situations we would want a coordination mechanism with a high quality result. To provide more insight in the possible selection criteria, we propose the following domain independent criteria: *time-to-goal-achievement*, *solution-quality*, *communication-costs*, *probability-of-success*, and *resource-consumption*.

The *time-to-goal-achievement* criterion captures performance measures that are related to the amount of time consumed by agents until a goal is achieved. Examples are the time that is needed for the setup of a coordination mechanism, or how a coordination mechanism influences the speed at which agents exchange information or coordinate their actions.

Solution quality criteria focus on the result when a goal is achieved. Such criteria are especially useful in domains where the amount of time is limited and the performance of the MAS is measured in terms of the quality of work that is achieved within that time. A reward received by agents is an often used operationalization since the size of the reward is usually correlated with the quality of the work by the agents.

A *communication-costs* criterion is useful as a selection criteria for minimizing the amount of communication needed by coordination mechanisms when communication resources are scarce or when costs are involved for using a communication resource.

Not all coordination mechanisms provide agents with a good chance of actually reaching a certain goal. This can be the case when a coordination mechanism is suitable only for a specific set of tasks. The *probability-of-success* criterion can be used in a trade-off with other selection criteria such as *solution-quality*, in which the chance of achieving a goal may be low but in the case that goal is achieved, quality of work is high.

In some domains such as robotics, the amount of resources consumed while coordinating determines the suitability of a coordination mechanism. Fuel consumption, CPU cycles and amount of memory needed are all operationalizations of the *resource-consumption* criterion.

Once an agent has decided on the selection criteria to be used, the agent can determine the costs of the coordination strategies. Calculation of the costs of a coordination mechanism can be a difficult problem and it depends on the domain in which this takes place. As mentioned by Lesser [4] "purely symbolic reasoning about costs and benefits" [of coordination mechanisms] "can be extremely complex, particularly in large systems and open environments, or where agents can simultaneously pursue multiple goals." A solution for this problem is to adopt a heuristic approach as done by Rosenfeld et al. [7].

Multiple selection criteria can be combined by using a weighted sum of the criteria or by using a rule based approach, e.g. if criterion α rises above a threshold, use the value of α, else use the value β.

If an agent decides that a switch to a different coordination mechanism should be made, the agent starts a change procedure that describes how the change from one coordination mechanism to the other should take place. In cases where the decision to switch is not directly agreed with by other agents, the change procedure should contain a mechanism to negotiate the new coordination mechanism. When the outcome of the negotiation process is positive, further organizational adaptation may be required. A new coordination mechanism may require changes in agent roles, interaction patterns, or agent relations and the change procedure should include a mechanism to transform the multi-agent organization into the desired state.

3 Abstract Task Environment

Typical coordination problems occur when agents cannot achieve their goals alone because a task requires effort from multiple agents or when resources that are needed for performing tasks are scarce [5].

The abstract task environment (ATE) is a simulation environment with discrete time in which a set of agents A, with size n, have to cooperate to perform a set of tasks B with size m. In the initial stage of the simulation, each agent a_i is randomly assigned a to a subset $\hat{B}_i$. The set of all subsets $\hat{B}_i$ is a partition of B. Agents do not know the tasks of the other agents nor the size of B. The goal of the agents is to complete all tasks in B as fast as possible. A task $b_{j,w}$ (the j^{th} task in B) can only be completed if w agents perform an action $p(i, j)$ (action performed by agent a_i on task

Table 1. ATE Messages

type	formal	example
1	[<t,i>]	[<2,2>,<3,2>,<4,3>,<5,3>,<6,4>,<7,4>,<8,5>,<9,5>]
2	<t,i,[<j,w>]>	<2,3,[<2,2>,<5,3>,<6,4>,<8,3>,<11,3>,<20,2>,<23,5>,<28,3>]>
3	<l_i,[<t,$\hat{A}$,j>]>	<8,[<1,[1,2],2>,<1,[4,5,6],5>,<1,[7,8,9,10],6>,<2,[1,2,3],8>]>
4	<switch,cm>	<switch,decentral>

b_j) at the same time step.

Each agent can only perform one action $p(i, j)$ each time step on any task in B. It is assumed that agents always cooperate and a task always completes when the right number of agents execute that task simultaneously.

3.1 Communication

Agents communicate via a single communication channel. Each time step only one agent can use this communication channel to broadcast a message to all other agents on the channel. We assume agents can "see" all other agents on the communication channel so they know the recipients of their message. Sending and receiving a message occurs in a single time step, so there is no delay in communication.

To enable agents to share information about the world and coordinate their actions we introduce four types of messages that can be exchanged between agents (see Table 1). Message type 1 is a time slot message and it means that agent a_i is allowed to send a message at time t. A task list message (type 2) is the list of uncompleted tasks of agent a_i at time t and for each task the number of agents required to execute the task. Message type 3 is a plan message send by agent a_i and it contains the size of $\hat{B}_i$ and a list of plan elements. The size of $\hat{B}_i$ is the workload l_i of agent i. A plan element is interpreted as; at time t, the agents in the set $\hat{A}$, $\hat{A} \subset A$, have to perform task b_j. Message type 4 is used in the switching process and is explained in Section 4.

The bandwidth of the communication channel is limited. For reasons of convenience the bandwidth d is expressed by the number of plan elements in a message. Since each plan element is a plan for one task, d is the maximum number of tasks for which a plan can be communicated in a single time step. Furthermore, a time slot message may contain $2 \cdot d$ time slots and a task list message may contain at most $2 \cdot d$ tasks. Note the example messages of type 1, 2 and 3 in Table 1 - where the time slot message contains 8 time slots, the task list message contains 8 tasks and the plan message contains 4 plan elements - have about the same length.

Dynamics in the simulator are introduced by breaking the communication channel in one or more places at a random time during a simulation. The occurrence of such a *split* event results in two or more fully separate communication channels and effectively breaks the MAS into two or more smaller groups of agents. The basic characteristics of a communication channel remain unchanged by a split event. The bandwidth will remain the same and the assumption that agents can "see" the other agents on the communication channels still holds. However, after a split event agents can no longer "see" the agents that are on other channels. Furthermore, for each communication channel, only one agent is allowed to send a message per time step.

3.2 Coordination

The ATE poses two coordination challenges; communication coordination and task coordination. Communication resources are scarce (one communication channel with limited access and limited bandwidth) which makes coordination of this resource necessary. This problem is similar to communication coordination problems in other domains such as the RoboCupRescue competition [3] where bandwidth is limited and the number of messages that can be received by agents per time step is also limited. Furthermore in the ATE, each agent can only execute one task each time step and executing a task only succeeds if it is performed by w agents at the same time step. This requires coordination of which task should be performed when and by which agents. Coordinating actions is a problem which occurs in many domains, for example the predator-prey domain where predator agents have to coordinate their movements to capture a prey.

The centralized coordination mechanism is a command driven approach in which one agent acts as a central coordinator and the others act as subordinate agents. The basic task of the coordinator agent is to gather information about the state of the world and issue plans to the subordinate agents. The agent with the lowest id on the communication channel will always be the coordinator agent. The coordinator agent determines who can use the communication channel by allocating time slots to the agents. When a simulation starts, the coordinator agent communicates the time slots that tell the other agents when to send their world view. The coordinator agent starts coordinating tasks when all information is received. Each following time step, the coordinator sends a plan message that prescribes which tasks have to be performed in that time step. The coordinator makes sure that the plan elements do not interfere (i.e. no agents have to execute more than one task in a single time step) and that as much tasks are executed as possible.

In the decentralized coordination mechanism each agent is autonomous in its decision making. Coordination of communication and tasks is achieved by social convention. This convention prescribes that each agent in turn is allowed to communicate its plans for the current time step and to make the decision to switch. The time at which agent a_i is allowed to do so is when $i = t \mod n$. This approach ensures that agents will not simultaneously try to use the communication channel and that their plans will not interfere.

In the case of a split event, each group of agents continues using the same coordination mechanism as they used before the split event.

4 Domain Dependent Decisions

In Section 2 we presented a decision making process for the dynamic selection of coordination mechanisms. In this section we demonstrate how we operationalize these decisions in the ATE.

The first step is for the agent to decide whether to initiate the decision making process. In the ATE, this is different for the two coordination mechanisms. In the centralized case, an agent will only initiate further decision making when it has the central coordinator role. In the decentralized case, an agent will initiate decision making when the agent has an allocated time slot. In order to prevent taking wrong decisions based on insufficient information, the agents will not initate the process until they know how many tasks have to be performed. In the centralized approach, this is the case when the central coordinator has received the complete task lists of the other agents. In the case of decentralized coordination, this is the case when each agent has communicated a plan message containing its workload l_i.

Next, the agent selects the criteria it will use to calculate the costs of a coordination mechanism. In the ATE we use one selection criterion; the *time-to-goal-achievement* criterion which is operationalized as the time to execute all remaining tasks.

As mentioned before, cost calculation of coordination mechanisms in general is a difficult problem to solve. The ATE is fully deterministic (except for the occurrence of split events and the initial distribution of tasks) and the coordination mechanisms are also deterministic when the number of agents needed to perform a task is known a priori (for the experiments we have set $w = 3$ for each $b_{j,w}$). Now, agents are able to compute the costs of the coordination mechanisms for each state of the environment. This also assumes that agents have an accurate view on the world which is ensured by the first step in the decision making process.

In the final step of the decision making process, agents perform the change procedure to switch to the new coordination mechanism. To enable the agents to perform the actual switch between coordination mechanisms the switch message is used (message type 4 in table 1). We assume this message requires only a bandwidth of one, so this message can always be sent in a single time step. Sending a switch message adds extra costs of one time step to the remaining execution times calculated in the previous section. When agents receive such a switch message they know that from the next time step on, the coordination mechanism (`cm`) specified in that message will be used. We further assume all agents are fully cooperative so there is no need in the ATE for a negotiation mechanism to decide whether the other agents will accept the decision to switch to a different coordination mechanism. Furthermore, when agents receive a switch message, each agent will update its own model of the organization of the MAS. If the new coordination mechanism is the centralized mechanism, each agent decides whether it will take the central coordinator role or an operator role (the agent with the lowest id will be the central coordinator). Then, authority relations are created between the central coordinator and the operator agents. The behavior of the coordinator role is prescribed by the centralized coordination mechanism. The subordinate role prescribes that the agent will only act and communicate when ordered by the central coordinator. If the new coordination mechanism is the decentralized mechanism, the agent will remove all authority relations with other agents and it will assume an autonomous role and follow the conventions prescribed by the decentralized coordination mechanism.

5 Experiment

In this section we present an experiment to demonstrate the effect of the decision making process described in the previous sections. In this experiment we compare three versions of a MAS in the ATE. In the first version – the "no-switch-cen" version – the agents only use the centralized coordination mechanism. In the second version – the "no-switch-dec" version – the agents only use the decentralized coordination mechanism. In the third version – the "switch" version – the agents initially use the decentralized mechanism but they are able to switch between the centralized and decentralized coordination mechanisms.

In the experiment, each simulation starts with a group of 60 agents which is split into 3 groups of $n = 20$ at some time during the simulation. The timing of the split event was varied between $t = 10$ and $t = 60$ with a 10 step interval.

Results of the experiment are obtained by running 1000 simulations for each t for each of the three versions and measuring the time it took for the agents to complete all tasks. The median score for each version is shown in Figure 1. Mann-Whitney statistical analyses show significant ($p < 0.01$) differences in scores for each pair of the three

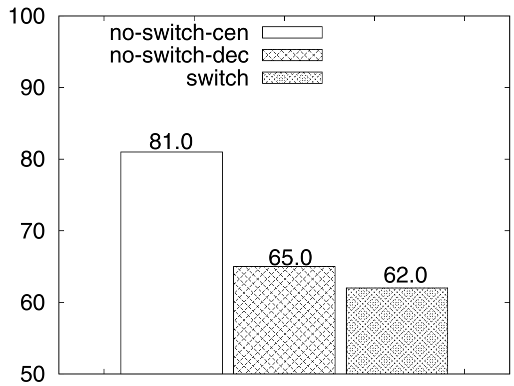

Figure 1. Results

versions. These results indicate that the "no-switch-dec" version outperforms the "no-switch-cen" version. This is because after a split event occurs, the 3 groups using the centralized mechanism have to go through the information gathering stage again before they can start executing tasks. However in some cases, when group size is small (e.g. after a split event has taken place) and the workload is unevenly distributed over the agents in the group, the centralized coordination mechanism performs better than the decentralized mechanism. In these cases, the agents in the switch version will switch from decentralized coordination to centralized coordination which improves performance of the dynamic decentralized version.

6 Conclusions

In this paper we have presented a decision making framework that enables agents to dynamically select the most appropriate coordination mechanism in a given situation. The experiment showed that dynamic selection of coordination mechanisms increases performance of a MAS. The applicability of the framework was demonstrated by a description of its operationalization in the ATE. Because of the distinction between domain dependent and domain independent knowledge in the decision procedure, we provide the designer of a MAS with a framework that supports the designer to focus on domain specific issues.

Using this framework we will continue to study each of the four decisions in more detail. In the operationalization in the ATE, we used the assumption of a deterministic environment and deterministic coordination mechanisms to calculate the costs of coordination mechanisms. Currently we are working on using machine learning to learn cost functions for coordination mechanisms in a more realistic task environment. A direction for future research is to start using incomplete information in the decision making process (the first decision making step). Another direction for further research is in the fourth decision making step where we aim to support more complex organizational dynamics e.g. multiple groups, each using a different coordination mechanism, that have to merge together into one larger group.

7 Acknowledgements

This research is part of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant BSIK03024.

References

[1] K. Carley. Computational and mathematical organization theory: Perspectives and directions. *Journal of Computational and Mathematical Organizational Theory*, 1995.

[2] C. B. Excelente-Toledo and N. R. Jennings. The dynamic selection of coordination mechanisms. *Autonomous Agents and Multi-Agent Systems*, 9(1-2):55–85, July 2004.

[3] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. 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)*, 1999.

[4] V. Lesser. Reflections on the nature of multi-agent coordination and its implications for an agent architecture. *Autonomous Agents and Multi-Agent Systems*, 1:89–111, 1998.

[5] T. W. Malone and K. Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1), 1994.

[6] C. Martin and K. S. Barber. Adaptive decision-making frameworks for dynamic multi-agent organizational change. *Autonomous Agents and Multi-Agent Systems*, 13(3):391–428, November 2006.

[7] A. Rosenfeld, G. A. Kaminka, S. Kraus, and O. Shehory. A study of mechanisms for improving robotic group performance. *Artificial Intelligence*, 172(6-7):633–655, 2008.

[8] Y. So and E. 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.