PRE-PROCEEDINGS

of

COIN@AAMAS 2016

The 21st International Workshop on Coordination, Organizations, Institutions and Norms in Agent Systems

and

CARE@AAMAS 2016

The Seventh International Workshop on Collaborative Agents Research & Development:
CARE for Digital Education

Edited by Stephen Cranefield and Samhar Mahmoud (COIN);
Fernando Koch, Christian Guttmann, Andrew Koster and Tiago Thompsen Primo (CARE)

Contents

COIN

Communication and Shared Mental Models for Teams Performing Interdependent Tasks

Ronal Singh, Liz Sonenberg, Tim Miller

University of Melbourne, Melbourne, Australia.
ronals@student.unimelb.edu.au, {l.sonenberg, tmiller}@unimelb.edu.au

Abstract. Research shows that performance of human teams improves when members have a shared understanding of their task; that is, when teams develop and use a shared mental model (SMM). An SMM can contain different types of information or components and this paper investigates the influence on team performance of sharing different components. We consider two components of an SMM: intentions (e.g goals) and world knowledge (e.g beliefs) and investigate which component(s) contribute most to team performance across different forms of interdependent tasks. We performed experiments using a Blocks World for Team (BW4T) testbed for artificial agent teams and our results show that with high levels of interdependence in tasks, communicating intentions contributes most to team performance, while for low levels of interdependence, communicating world knowledge contributes more. Additionally, as is the case with human teams, higher sharedness correlated with improved team performance for the artificial agent teams. These insights can assist in the design of communication protocols that improves team performance when team members are engaged in interdependent tasks and help design artificial agents that can communicate effectively when working with humans as teammates.

Keywords: Task interdependence, shared mental models, joint action

1 Introduction

Agents perform tasks that range from independent tasks that does not require interactions with others to highly *interdependent tasks* requiring close and continuous interactions [14]. When faced with interdependent tasks, effective coordination and collaboration of team members become crucial. One of the key foundations of effective coordination and collaboration is having *shared mental models (SMM)*. Shared mental model has been defined as [1]: "knowledge structures held by members of a team that enable them to form accurate explanations and expectations for the task, and, in turn, coordinate their actions and adapt their behaviour to demands of the task and other team members".

More than a decade of research has correlated SMMs with improved team performance in human teams [12]. The basic assumption is that SMMs allow team members to anticipate the needs and actions of other members, thereby increasing team performance. Recent studies in human-agent and artificial agent teams have also found similar correlations [3, 5]. SMMs can be broadly classified as either task work model or team work model. Task work concerns the task or job that the team is to perform, while team work concerns what has to be done in order to complete a task as a team [9]. SMMs can also be viewed as having different components [5, 9], such as world knowledge and intentions. World knowledge includes knowledge of the current state of the environment and the team while intentions represent what the agents intend to do [4].

Four types of task interdependence have been identified for human activities: pooled, sequential, reciprocal, and team [14, 15]. In sequential task interdependence, tasks are performed in a sequential order. For example, in a relay race each runner has to wait for the previous team member to pass on the baton. In reciprocal task interdependence, participants take their turn in completing part of the task. A key property associated with reciprocal task interdependence is interleaved execution: for example, surgical teams often work reciprocally. In team task interdependence, participants execute their individual tasks concurrently and may include *joint actions*. By "action", we mean the atomic actions that make up a task. In joint action, multiple participants execute a particular action concurrently, for example when two people lift a heavy object together. In pooled task interdependence, the participants can successfully execute tasks without any interaction with each other. Due to the simple nature of these tasks, we do not study such tasks in this paper. The four types of task interdependence forms a hierarchy of pooled-sequential-reciprocal-team, with this hierarchy representing increasing levels of dependence between team members as well as increasing needs for coordination [14].

While *sharedness* has been linked with better team performance, central to the notion of SMM is how much and what to share. There has been recent work investigating this question in multi-agent systems research, such as [5, 11, 17]. However, as far as the authors are aware, with the exception of Li et al. [10], studies in the related work only consider sequentially-interdependent tasks, rather than more tightly linked team and reciprocal tasks. A recent report [16] highlights the need for studies considering other types of interdependence, notably *intensive* task interdependence – a type that we characterise as a *joint action.*

The subject of this paper is the communication content, specifically *what* to share when team members engage in interdependent tasks. We investigate the influence of the two components of the SMM (world knowledge and intentions) on the team performance across different forms of interdependent tasks. We used search and rescue like scenarios for a team of artificial agents for the experiments. The scenarios were generated using a Blocks World for Teams (BW4T) testbed [8]. In BW4T, which is an extension of the classical blocks world domain, the teams' joint task is to find and deliver coloured blocks in a particular order. Using the testbed, we designed and executed two sets of experiments. The first set studies the influence of sharing the two components – world knowledge and intentions – on the team performance for each form of task interdependence. The second set introduces joint actions within sequential and reciprocal tasks and studies the influence of sharing the two components on team performance. Introduction of joint actions allows for a shift from sequential or reciprocal to team task interdependence where members execute individual actions concurrently.

The outline of the paper is as follows. Section 2 introduces SMM, along with related work. Section 3 describes the task and the testbed and provides the details of the artificial agents that we implemented. Section 4 details the experimental setup while Section 5 discusses the results. Sections 6 and 7 conclude the paper with a discussion.

2 Background and Related Work

Mental models are simplified representations used by individuals to explain and predict their surroundings [13]. These models comprise content and structure or relationships between the content. In addition, individuals can simultaneously hold multiple mental models. In a team setting, when team members interact, their mental models converge resulting in shared mental models.

To extend the concepts of SMM that has been well studied for human teams [12] to human-agent teams, Jonker et al. [9] proposed mental model ontologies. They view a team as a system. A team performs team activities and has physical components, e.g. team members. A team member is an agent with a mind comprising many mental models: all but one of which represent the mental models of others in the team. Based on this conceptualisation, they proposed a measure that could be used to assess the similarity or the overlap of agents' mental models. We discuss this measure in the next section.

2.1 Measuring SMM

While several methods exist for measuring SMMs for human teams [2], one for teams comprising artificial agents is Jonker et al. [9]. Harbers et al. [5] extended Jonker et al's similarity measure so that it could be applied to teams of agents and performed experiments to show that their similarity measure can be used to predict team performance. We discuss the extended version of the measure next. In the following discussions, similarity refers to the overlap of the mental model contents of the agents. We consider the SMM to be made of two components – world knowledge and intentions.

Figure 1 shows an example of SMM. Assume Bot 1 and Bot 2 are two agents engaged in a joint task. Each has its mental model. While engaged in their task, the agents may communicate their beliefs and goals, making their own beliefs and goals known to others. For example, notice that each agent has it's own as well as others' beliefs and goals, which are shown in italics. The SMM is a theoretical construct that can be used to represent the overlapping content of the mental models of the two agents. In the example, the SMM is composed of the components - world knowledge (beliefs) and intentions (goals).

Jonker et al. [9] and Harbers et al. [5] proposed a compositional measure of sharedness. We reproduce their definitions here with some simplifications. They view SMMs as having components, which can include sub-components.

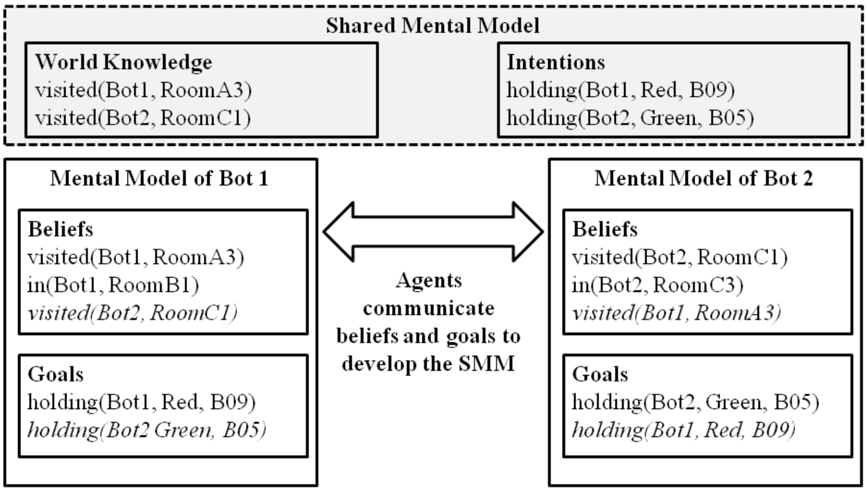


Fig. 1: Example SMM. The beliefs and goals of other agents are shown in italics. An agent has certain beliefs and goals that it is not required to communicate, e.g. *in(agent, room)*, and these may not part of the SMM.

For example, Figure 1 shows an SMM with two components. Examples of sub-components can be found in Section 3.3. The (sub)components can be queried by posing questions that all team members should be able to answer. The answers are used to compute the model agreements, which is a measure of the similarity of the answers provided by each agent for each question. Formally, let M be the set of all mental models, Q be the set of all questions, and $ans(m, q)$ be the answer of model $m \in M$ with respect to question $q \in Q$. The agreement between models M for questions Q is:

$$Ag(M, Q) = \frac{1}{|Q|} \sum_{q \in Q} \frac{| \cap_{m \in M} ans(m, q)|}{| \cup_{m \in M} ans(m, q)|} \tag{1}$$

If $| \cup_{m \in M} ans(m, q)| = 0$ then the agreement for question q is 0. Given a set of agents A, a set of mental models M_A (a model for each agent), and questions Q, we say that the model m is shared *to the extent* θ, denoted by $Sh(M, A, Q, \theta)$, with respect Q, iff $Ag(M_A \cup \{m\}, Q) >= \theta$. The compositional measure CS is:

$$\begin{aligned} CS(M, A, Q) &= max\{\theta \mid Sh(M, A, Q, \theta)\}, \text{ if M is not composed} \\ CS(M, A, Q) &= c(\{CS(m, A, Q) \mid m \in M\}), \text{ if M is composed} \end{aligned} \tag{2}$$

Where m is a component of M and c is composition function, for example: $\sum_{m \in M} w_m CS(m, A, Q)$. Each component and sub-component can be weighted to model the relevance of each (sub)component. The weight of each (sub)component is $w_m \in [0, 1]$ and CS can be normalised to $[0, 1]$ by setting $\sum_{m \in M} w_m = 1$.

2.2 SMM and Task Interdependence

Interdependence is the central organising principle of *Coactive Design Method*, from Johnson et al. [7], which is a method aimed at designing systems in which humans and agents collaborate as teammates. They define interdependence as relationships between members of a team, and argue that these relationships determine what information is relevant for the team to complete (interdependent) tasks, and in that sense, the interdependent relationships define the *common ground* that is necessary. A number of studies have considered some of the different forms of task interdependence [5, 17, 10], and some have also measured sharedness [9, 5]. Generally, higher sharedness of mental models produces better team performance. For example, Harbers et al. [5] found higher sharedness correlated with better team performance. In their work, SMM were composed of world knowledge and intentions, which is how we view SMM in this work. Similarly, task interdependence has naturally been part of these studies. However, almost all involve sequentially interdependent tasks. The exception is Li et al. [10], who introduced joint action in sequentially interdependent tasks and Wei et al. [17], who studied tasks that were not very strongly sequential. They did this by creating subtasks that multiple agents could complete simultaneously. None of these have explicitly employed reciprocally interdependent tasks.

Mixed results have been reported for studies involving sequentially interdependent tasks in terms of which type of information or component contributes more to team performance, that is task completion times. Harbers et al. [5] reported that when agents communicated their intentions with others, the team performance improved more than if they shared world knowledge. However, Wei et al. [17] reported that beliefs contributed more to team performance than goals. While [17] did not measure sharedness, they view the agents mental models to comprise of two components, goals (intentions) and beliefs (world knowledge). We perform further experiments involving sequentially interdependent tasks and may help explain the difference between the two studies.

In a separate study, Li et al. [10] introduced joint action in sequentially interdependent tasks. They studied search and retrieval tasks using the BW4T testbed. In one setup, agents collaborated on a task in which some blocks were heavier, and required two agents to collect. The agents exchanged goals, beliefs, and both. Their experiments revealed that with joint actions, exchanging goals improved team performance, measured as completion time, more than sharing beliefs only. When agents shared their goals that fulfil the current team sub-goal with others, the other team members could start on a new task. This allowed the team to finish the team task more quickly.

These works show that sequentially interdependent tasks have been investigated, but other forms of task interdependence have not. This work aims to fill that gap.

3 Scenario: Blocks World for Teams

We used a BW4T testbed [8] for our experiments. As explained next, we modified the testbed to be able to setup tasks with joint actions.

Basic BW4T: In BW4T, teams find and deliver coloured blocks in a particular order. The environment has a set of rooms, each containing coloured blocks, and a drop zone. The agents search the rooms, find the required blocks and drop these in the drop zone. Agents have a map of area but do not know the location of the required blocks. Agents have to go to each room to perceive the blocks that are present in it. Agents cannot see each other but can communicate with others. A simplified map is shown in Figure 2. Each room has one door. The teams' joint task, i.e. the sequence of colours, is displayed at the bottom left. A black triangle appears on top of a colour if the colour is dropped off. The room above the joint task is the drop zone. The agents are represented by either black squares or the colour the agent is holding, and their names are displayed in red. The basic version is well suited to perform experiments for sequential and reciprocal tasks. However, it does not explicitly support joint actions.

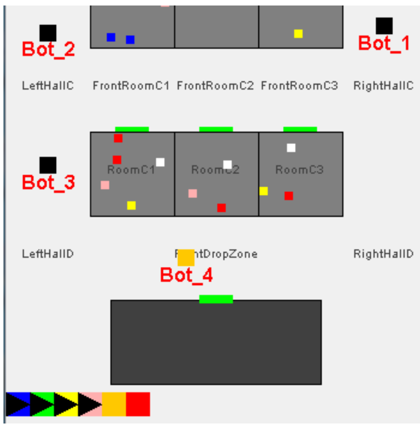


Fig. 2: Sample BW4T environment.

Modified BW4T: To design joint tasks that would be a fair representation of the different forms of task interdependence, we modified the testbed. In the original version, only one agent could be in a room at any one time. To implement joint actions, we follow Li et al. [10] and introduce "heavy blocks", which required two agents to carry to the drop zone. This means in our version, two agents can carry the same block simultaneous, and therefore can be in the same room at the same time. Secondly, for team task interdependence, the blocks could be delivered in any order, that is, we removed the sequential delivery requirement.

3.1 Task Design

We designed tasks to be able to test the effects of communication content on the team performance for each type of the task interdependence as well as later include joint action within other forms of task interdependence and test the effect of communication content on the team performance for each combination. Variations of two basic joint tasks (Figure 3) has been used to realise the different forms of task interdependence.

(a) Task 1 (b) Task 2

Fig. 3: Basic joint tasks used to simulate different types of task interdependence.

Team Task: In team tasks, agents execute their actions concurrently. The joint task had some *heavy* blocks. The heavy blocks required one agent to help the other lift it, and afterwards the first agent delivers it to the drop zone. The act of lifting the heavy block *together* is the joint action. Additionally, the agents could lift any colour. Consider the task shown in Figure 3a. In this task, agents can lift both colours. The red blocks are heavy blocks. In order to remove the underlining sequential interdependence from this task, the agents could deliver the blocks in any order, for example, the second (red) block can be delivered before the first (yellow) block. Green, pink and red are heavy blocks in Task 2.

Reciprocal Task: In a reciprocal task, each agent takes it's turn in completing part of the task. In this task, the agents deliver a sequence of alternating colour sets in the order the colours appear in the task. Furthermore, each agent can lift colours from only one of the two distinct colour sets. Consider the task shown in Figure 3a. For this task, one agent would be delivering yellow blocks while the other red ones. The blocks must be delivered in the order they appear. This means that agent delivering the red block now depends on the agent delivering the yellow blocks and vice-versa, making them reciprocally interdependent.

Sequential Task: In sequential task, the first three colours are delivered by one agent while the remaining three by another agent. The blocks must be delivered in the specified order, but the second agent is free to search for its coloured blocks while the first agent is delivering.

3.2 Agent Teams and Agent Behaviours

We had two team compositions; 1) 2-agent team and 2) 4-agent team. The 4-agent team was a 2x2-agent team, i.e. 2 sub-teams of 2 agents each. This composition was required for certain tasks, such as reciprocal tasks in which we needed to have at least one agent for each of the two colour sets.

Agents were programmed in GOAL [6]. The BW4T testbed provides interfaces that enable GOAL agents to interact with it. Using these interfaces, the agents can perceive specific details of the environment, such as the blocks present in rooms, and can perform actions, such as picking up a block. The abstract decision cycle of an agent is shown in Figure 4. The basic steps each agent takes are: *1*) decide the colour to search for; *2*) choose a room; *3*) go to and search room; *4*) if required block is found and is not heavy, pick it up; *5*) if required block is found and is heavy, ask for help and wait. When help arrives, pick up the block; *6*) deliver the block to the drop zone; *7*) if help is requested, go to the particular room and help lift the heavy block.

Initially, agents start searching for the first undelivered colour. However, agents use a two-block look-ahead protocol to determine which colour to deliver. If an agent knows the location of the first undelivered colour and has the intention of collecting it, remaining agents search for the second undelivered colour. If the

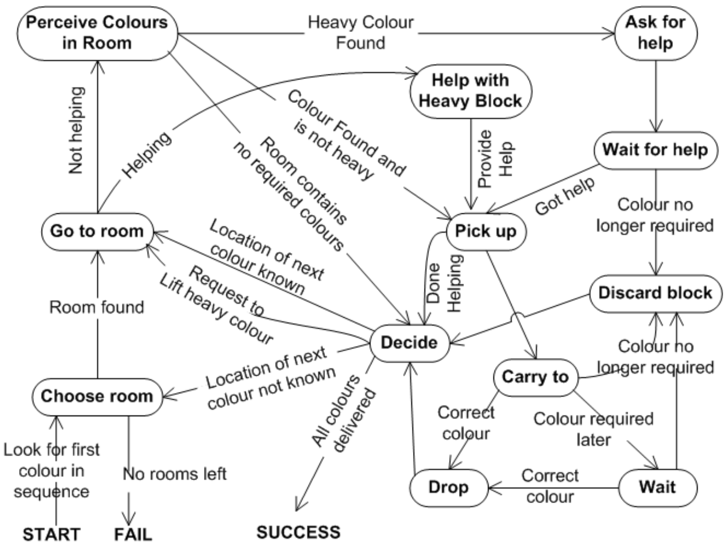


Fig. 4: Abstract decision cycle of an agent.

one or more of the remaining agents know the location of the next required colour, they go to that room. However, only one will be able to collect the block. When the first colour is picked up, one agent collects the second colour while others start searching for the third colour. The aim of this is to ensure that sufficient time is dedicated to search. When required to lift a heavy block, an agent only asks for help when it is physically present at the heavy block. Other (helper) agents could potentially infer that help will be required soon and go to the location of the heavy block before the agent actually asks for help because the agent may tell others that it has the goal of going to the (heavy) block. However, our agents do not perform this level of reasoning and only go to help when asked. Furthermore, if one agent asks for help, all agents that are waiting to drop a block at the drop zone or those that are currently searching for their block will go to help. If the agent knows that the colour that it is searching for, has the intention of holding or is holding is no longer required, then it will discard the colour and go on to deciding what it will do next. Rooms are chosen randomly and the agents avoid visiting a room more than once unless the room contains multiple required blocks.

While the basic behaviours of agents are almost the same across the different forms of task interdependence, there are differences in the way agents reason about which colour to search for:

1) Sequential and reciprocal tasks: Agents choose the first undelivered colour. If another agent has the goal of holding this colour, the agent chooses the next undelivered colour.

2) Team Task: Blocks can be delivered in any order. Therefore, agents do not reason about when the block has to be delivered. Instead agents have to determine whether the block is heavy and ask for help.

While certain aspects of agent behaviours are different because of task interdependence, there are differences because of what the agents share with each other. Therefore, while the basic decision cycle shown in Figure 4 is used by all agents, there are some variations in their implementation. The implementation has been guided by what the agents actually do with the information they receive and has been described later in Section 3.4. Therefore, if only one component is exchanged, the agent performs reasoning described for that component only.

3.3 Communication and SMM

Agents exchange messages that are indicative of the world knowledge and the intentions. To develop the shared mental model, agents communicate as soon as they have the required information. Agents exchange six sub-components, three each of goals and world knowledge. These sub-components were selected based on prior research work [5, 10] and preliminary experiments revealed that each sub-component had the potential to improve team performance. The sub-components are communicated as messages, which are discussed next. The keyword *imp* stands for imperative and indicates what the agent intends to do.

The messages indicative of intentions are:

1) imp(in(Sender, Room)): Sender intends to visit Room.
2) imp(holding(Sender, Colour, Block)): Sender intends to collect Block of Colour.
3) delivered(Sender, Colour, Block): Sender has delivered Block of Colour - implies agent has dropped current goal and may have a new goal.

The messages indicative of world knowledge are:

1) blockLoc(Sender, Block, Colour, Room): Sender has perceived Block of required Colour in Room.
2) pickedUp(Sender, Colour, Block): Sender has picked up Block of Colour.
3) visited(Sender, Room): Sender has visited Room. This message is sent irrespective of whether room contains required blocks.

3.4 Using Shared Mental Models

Agents employ the following policies to SMM to choose their activities such that it prevents potential conflicts with the activities of others. The following outlines how the agents use the components of the shared mental model.We chose a straightforward use of each intention and world knowledge, which was sufficient to test the effect of the component on the team's performance and avoids side-effects that would have been introduced because of using more complex mechanisms. The intentions are used as follows:

1) An agent will not adopt a goal to go to a particular room if another agent has the goal of going to that room. For reciprocal task, this logic applies when both agents are delivering blocks from the same colour set, that is in a 4-agent team and not in a 2-agent team.

2) An agent will not adopt a goal to hold a block that has been delivered.
3) An agent will not adopt a goal to hold a block/colour that another agent has the goal of holding — *unless* the block is heavy (both agents need to lift it together). For reciprocal tasks, this logic is applicable in a 4-agent team.

World knowledge is used as follows:

1) An agent will not search for a colour if this been found by another agent.
2) An agent will search for the next colour if the currently required colour has been picked up.
3) An agent will not search a room that another agent has already searched.

Agents employ the above policies to SMM to reduce interference and duplication of effort. However, the agents have their own decision processes and may make decisions simultaneously. This may result in instances where the agents may adopt similar goals, for example to look for the same colour. Like Wei et al. [17], we simply implement a "first-come first-served" policy instead of implementing detailed negotiation mechanisms to assist agents resolve these issues.

4 Experiment Design

We ran a series of simulation experiments, measuring the following:

1) Completion Time: Time it takes the team to complete the task. We used this measure as a proxy for team performance.
2) Number of messages: We measured the total number of messages exchanged by the agents. We also counted the number of messages per component. These measures are indicative of the communication cost.
3) Sharedness: We measured the sharedness of the agents' mental models. This is a compositional measure (see Section 2.1) and was calculated at the time any block was delivered to the drop zone. When one agent drops off a correct colour in drop zone, all agents log their belief and goal bases. These logs are then analysed to find the overlapping content, which is used to compute the sharedness values. The two components had a weight of 0.5 and each of the three sub-components had a weight of 0.33. In experiments where only one component was measured, the weight of the component was set to 1, and only questions related to that component were asked.

In case of sub-teams, we also measured the number of messages and sharedness of the agents with each sub-team.

Independent Variable The independent variable is the component of the SMM. This variable has three values (see Section 3.3): *1*) World Knowledge (WK); *2*) Intentions (INT); and *3*) World Knowledge and Intentions (ALL).

Setup We used two different maps, one for each task outlined in Section 3.1. Variations of each task gave us three different task interdependence types. We

	Set 1				Set 2	
Team size	2 agents		4 agents		4 agents	
Map	1	2	1	2	1	2
Setup	S1	S2	S3	S4	S5	S6

Table 1: Experimental setups (S1 – S6) for each type of task interdependence.

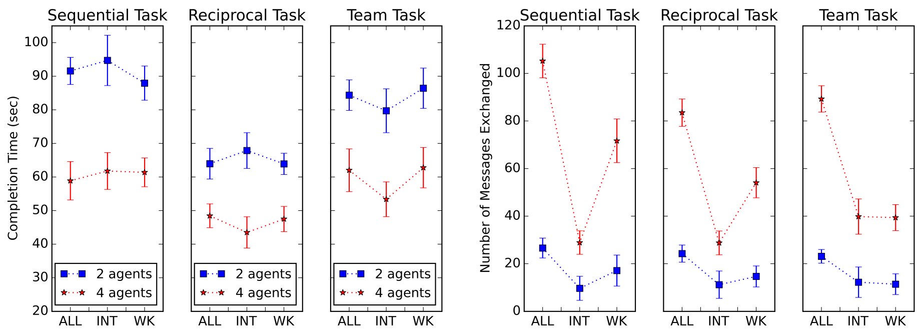
refer to Task 1 (Figure 3a) as Map 1 and Task 2 as Map 2. The setups are as shown in Table 1. We had two sets. In set 1, we had four setups (S1-S4) (both maps combined with two team compositions) for each of the three types of task interdependence giving us 12 combinations.

Set 2 has two setups, S5 and S6, representing reciprocal and sequential tasks with joint actions respectively. Here the sub-teams were reciprocally or sequentially interdependent and were required to lift heavy blocks. We tested the effect of the SMM components on completion times by employing three communication strategies: *1*) ALL-ALL: where agents exchanged the two components with every other agent. *2*) WK-Within: where agents shared world knowledge within each sub-team but shared intentions with all agents. *3*) INT-Within: where agents shared intentions within each sub-team but the world knowledge with all agents. For these two setups, we only used a 4-agent team because a 2-agent team would not have enabled us to fully test the effects of the two components. For example, we needed to have at least 2 agents in each sub-team to be able to test the effect of sharing a component within the sub-team. Combining S5 and S6 with the two types of task interdependence (sequential and reciprocal) gave us further 4 combinations, and a total of 16 combinations.

Combining each of the 12 combinations from Set 1 with the three components of the SMM (ALL, INT, WK) and the 4 combinations from Set 2 with the three communication strategies (ALL-ALL, WK-Within, INT-Within) resulted in 48 combinations in total. Each combination was run 30 times resulting in 1440 runs. Each map had 25 blocks pre-allocated to rooms and further 10 blocks were randomly generated giving a total of 35 blocks for each run. Each map had 9 rooms, 1 drop zone and 6 blocks in the joint task. Statistical significance tests were conducted using Wilcoxon rank-sum (WRS) and Kruskal-Wallis (KW) tests.

5 Results

This study was aimed at identifying the components that contributed most to team performance across different forms of task interdependence. Recall that going from sequential to team tasks represents increasing levels of dependence between agents as well as coordination requirements. For simplicity, we collapse the results of the two tasks (shown in Figure 3) and report the averages.



(a) Performance (b) Communication Cost

Fig. 5: Performance and communication cost for different forms of task interdependence. The communication cost is expressed as the average number of messages exchanged by all team members. Error bars represent on standard deviation.

5.1 SMM Components and Team Performance

Figure 5a shows the average task completion times for the 2-agent and 4-agent teams performing different tasks. These results are for experiments resulting from setups S1-S4. Recall that a 4-agent team comprises 2 sub-teams of 2 agents each. For team tasks, the intentions contributed more to team performance than world knowledge. This finding is significant at 5% for all except two combinations and consistent for both team compositions. In the team task, some blocks were heavy and the agents could pick any colour. In such scenarios, knowing the intentions of team members allows agents to avoid duplicating their activities, therefore reducing interference. These results are in line with Li et al. [10], who reported that with joint actions, exchanging goals results in improved completion times.

However, for sequential and reciprocal tasks, different trends have been observed between 2-agent and 4-agent teams. For sequential tasks and 2-agent team, the world knowledge contributed significantly more ($p < 0.05$) than intentions in terms of task completion times. In this task setting, the first agent delivered first three blocks while the remaining three by the other agent. Because agents had separate sub-tasks, exchanging world knowledge helped the other agent find it's required blocks faster. However, for reciprocal tasks, this difference was less pronounced. We discuss this more later.

In 4-agent teams performing sequential tasks, no significant difference in terms of completion times were noted between the two components. However, it is worth noting that moving from 2-agent to 4-agent team, the importance of intentions increases. A similar trend occurs for reciprocal and team tasks. In these team settings, the agents within each sub-team could choose conflicting goals, for example, choosing the same block to deliver. By exchanging intentions, agents within sub-teams avoided duplicating their activities, therefore improving the completion times.

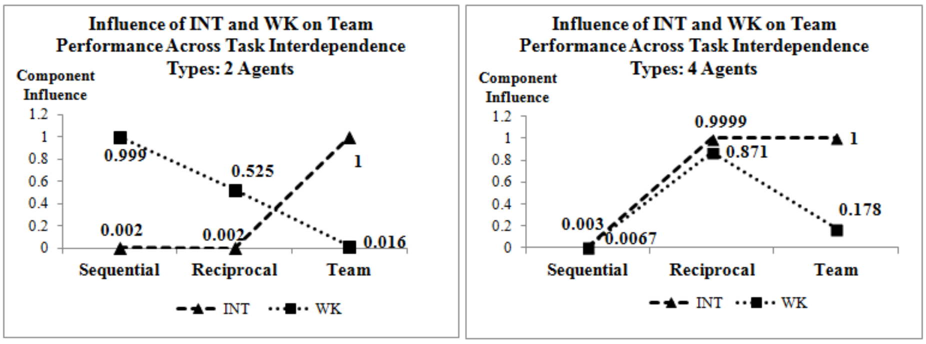


Fig. 6: The two graphs (2 agents and 4 agents) show that intentions become more important more as the level of interdependence increases and as the number of agents in each sub-team increases.

To make these trends clearer, we computed *component influence* (CI) for each task. CI is computed based on the difference between the completion times achieved when communicating both components and any one of the two components. To normalise the difference between completion times across different experiments, we used the *tanh* function. The CI for component c is:

$$CI_c = tanh(CompletionTime_{all} - CompletionTime_c)$$

The resulting values were normalised to between 0 and 1 using $CI_{normalised} = (CI - min(CI))/(max(CI) - min(CI))$. Figure 6 for 2-agent teams show that with increasing dependence between agents, that is, going from sequential to team interdependence, the importance of intentions increases while the importance of world knowledge decreases. For 4-agent team, the intentions were almost always more important than world knowledge.

SMM and Joint Actions Results of experiments relating to setups S5 and S6 indicated that the difference between completion times of WK-Within and INT-Within is significant (p-value = 0.009) in favour of INT-Within. This indicates that sharing intentions within sub-teams and world knowledge with everyone achieves the best team performance. This is consistent with our earlier findings that intentions and team tasks are positively correlated. Also, world knowledge and sequential and reciprocal tasks are positively correlated.

5.2 Communication Performance

Figure 5b shows the communication cost (average number of messages exchanged). The number of intentions exchanged was significantly lower ($p < 0.05$) than world knowledge for about two-thirds of the combinations. This indicates that agents generally have more information to communicate about the world than their intentions. There was no correlation between the number of messages and team performance. More communication resulted in worst performance in some cases, particularly for larger teams. This is due to the two-block look-ahead policy. When agents exchange information about possible blocks, in larger teams this often results is agents trying to collect the same block/colour and this increases the completion time. When agents only exchange intentions, all agents are required to find the blocks themselves, and so search randomly, thus reducing the number of unnecessary runs for the same block/colour. While it is clear that a mechanism could be designed to improve this by using a different look-ahead policy, we believe our policy is reasonable. Importantly though, this result shows that simply throwing more information towards agents can result in worse performance if significant thought is not given to how that information is used.

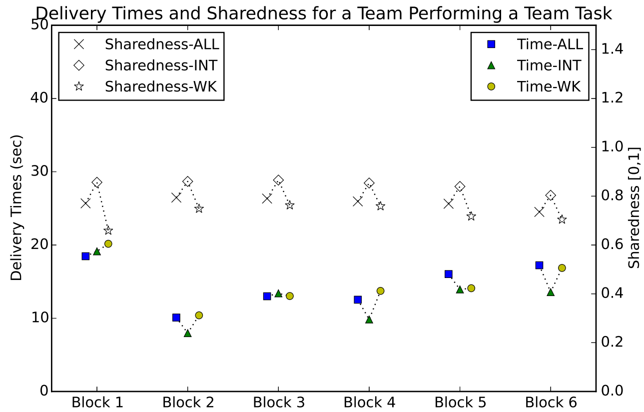


Fig. 7: Sharedness and delivery times for a 2-agent team engaged in a team task.

5.3 Analysis of Sharedness

We computed the sharedness in relation to each component at the time a block was delivered to the dropzone. Generally, higher sharedness correlates with improved completion times. For simplicity we show the data for team task and note that the results for sequential and reciprocal tasks are similar. For example, Figure 7 shows the sharedness at the time each correct block is dropped off for team tasks. The plotted delivery times are the time differences between block deliveries. For team tasks, exchanging intentions achieved the best completion times and the sharedness was highest for this component. Notice that in Figure 7, sharedness of intentions is highest across all six blocks and the delivery times when teams exchange intentions are fastest across most of the six blocks.

Sharedness and Sub Teams We measured the sharedness of members within each sub-team for tasks solved by 4-agent teams. Sharing intentions resulted in the best completion times and the sharedness of intentions was highest for reciprocal and team tasks. For sequential tasks, we noted a significant increase in the importance of intentions compared to 2-agent team. This supports the finding that higher sharedness results in better completion times. The other consistent finding is that in situations where we may have members of sub-teams potentially duplicating their efforts, sharing intentions with each other helps avoid such conflicting actions and therefore, improves the completion times.

6 Discussion

We intended to identify the components contributing most to team performance across the different forms of task interdependence. Our results show that as the interdependence increases, the importance of intentions to team performance also increases. These results are in line with [5, 10] who found that when team members exchanged intentions, the team performance improves. In [5], teams were engaged in sequential tasks and their team composition was similar to our 4-agent team while in [10], the authors introduced joint actions in sequentially interdependent tasks.

While our results are in line with the above works, we have observed that when team members can perform their sub-tasks independently, e.g. in 2-agent teams, exchanging world knowledge contributes more to team performance for sequential and reciprocal tasks. This makes sense intuitively: if other members provide potentially useful information, such as location of blocks that one is required to deliver, the team performance improves. This is a form of *soft interdependence* [7] where one team/member 'helps' another voluntarily. In case of 4-agent teams, we found that intentions contributed more to team performance across all forms of task interdependence. This indicates that team composition plays a role in which component is important to team performance.

Our findings that are partially consistent with [17] who found that for sequential tasks, beliefs contributed more to team performance. While this is consistent with the results of our 2-agent team, we noted a marked increase in the importance of intentions when 4-agent team was concerned. These differences may hinge on other factors, such as how effectively the agents use the information that it receives. This is an area of future work.

Finally, our findings are consistent with others (e.g. [5]) in terms the role SMM plays in improving team performance. Across all tasks and both team compositions, higher sharedness of SMM resulted in improved team performance.

7 Conclusions and Future work

The four types of task interdependence form a hierarchy, from pooled to team, representing increasing levels of dependence between team members as well as increasing needs for coordination. We found that with increasing levels of interdependence, the importance of intentions increases as well. Team composition also plays a role in which component contributes more to team performance. In team compositions, where agents can perform their tasks independently, e.g. in sequential and reciprocal tasks, world knowledge contributed more to team performance. When multiple team members may be engaged in a single sub-task, the potential of interference increases and so does the importance of knowing the intentions of others.

A factor to investigate further is the reasoning capability of the agents; that is, how the agents reason with information that they receive from others. We also have not explicitly analysed the behavioural changes in the agents when agents switch from one task interdependence type to another, making this another opportunity for future investigation.

References

1. J. A. Cannon-Bowers, E. Salas, and S. Converse. Shared mental models in expert team decision making. In J. Castellan, N. John, editor, *Individual and group decision making: Current issues*, pages 221–246. Hillsdale, NJ:Lawrence Erlbaum., 1993.
2. L. A. DeChurch and J. R. Mesmer-Magnus. Measuring shared team mental models: A meta-analysis. *Group Dynamics: Theory, Research, and Practice*, 14(1):1, 2010.
3. X. Fan and J. Yen. Modeling cognitive loads for evolving shared mental models in human–agent collaboration. *Systems, Man, and Cybernetics, IEEE Transactions on*, 41(2):354–367, 2011.
4. M. Harbers, C. M. Jonker, and M. B. Van Riemsdijk. Context-sensitive sharedness criteria for teamwork. In *Proceedings of AAMAS'14*, pages 1507–1508. ACM, 2014.
5. M. Harbers, M. v. Riemsdijk, and C. Jonker. Measuring sharedness of mental models and its relation to team performance. In *Proceedings of 14th Workshop on COIN*, pages 106–120. Springer Berlin Verlag, 2012.
6. K. Hindriks. Programming rational agents in GOAL. In A. El Fallah Seghrouchni, J. Dix, M. Dastani, and R. H. Bordini, editors, *Multi-Agent Programming*, pages 119–157. Springer US, 2009.
7. M. Johnson, J. Bradshaw, P. Feltovich, C. Jonker, and M. B. V. Riemsdijk. Coactive design : Designing support for interdependence in joint activity. *Journal of Human-Robot Interaction*, 3(1):43–69, 2014.
8. M. Johnson, C. Jonker, B. van Riemsdijk, P. J. Feltovich, and J. M. Bradshaw. Joint activity testbed: Blocks World for Teams (BW4T). In *Engineering Societies in the Agents World X*, pages 254–256. Springer, 2009.
9. C. Jonker, M. van Riemsdijk, and B. Vermeulen. Shared mental models. In *Proceedings of 9th Workshop on COIN*, volume 6541 of *LNCS*, pages 132–151. Springer Berlin Heidelberg, 2011.
10. S. Li, W. Sun, and T. Miller. Communication in human-agent teams for tasks with joint action. In *Proceedings of 9th Workshop on COIN*, pages 111–126, 2015.
11. M. Manner and M. Gini. Improving agent team performance through helper agents. In *Cognitive Agents for Virtual Environments*, volume 7764 of *LNCS*, pages 89–105. Springer Berlin Heidelberg, 2013.
12. S. Mohammed, L. Ferzandi, and K. Hamilton. Metaphor no more: A 15-year review of the team mental model construct. *Journal of Management*, 36(4):876–910, 2010.
13. W. B. Rouse and N. M. Morris. On looking into the black box: Prospects and limits in the search for mental models. *Psychological bulletin*, 100(3):349, 1986.
14. R. Saavedra, P. C. Earley, and L. Van Dyne. Complex interdependence in task-performing groups. *Journal of Applied Psychology*, 78(1):61–72, 1993.
15. R. Singh, T. Miller, and L. Sonenberg. A preliminary analysis of interdependence in multiagent systems. In *PRIMA 2014*, volume 8861 of *LNCS*, pages 381–389. 2014.
16. K. A. Smith-Jentsch. On shifting from autonomous to interdependent work. Technical Report February, NASA, Houston, Texas, 2015.
17. C. Wei, K. Hindriks, and C. M. Jonker. The role of communication in coordination protocols for cooperative robot teams. In *Proceedings of International Conference on Agents and Artificial Intelligence*, pages 28–39, 2014.

An empirical approach for relating environmental patterns with agent team compositions

Mariana Ramos Franco[1], Gustavo A. L. Campos[2], and Jaime Simão Sichman[1]

[1] Laboratório de Técnicas Inteligentes
Escola Politécnica
Universidade de São Paulo
ramos.franco@gmail.com, jaime.sichman@usp.br
[2] Mestrado Acadêmico em Ciência da Computação
Universidade Estadual do Ceará
gustavo@larces.uece.br

Abstract. The design of a rational organization composed of a team of agents is a challenging problem in domains such as collective robotics, cyber warfare, war games and military missions. In these domains, the team is designed to confront an opponent team with technical and numerical equivalence, and aiming to conquer areas where there are scarce resources of high economic value, that are distributed in locations within a territory whose topology is unknown. In these scenarios, it is hard for the agents to do the right thing. In addition to being competitive, the task environment is unknown, partially observable and dynamic. The challenge is how to design a rational team whose members are not ideal rational agents. This work argues that one approach is to implement a suitable organizational specification that fits the task environment, according to some previously defined environmental patterns that include both domain-specific and topological characteristics. In this work, we present an experimental evaluation of these patterns' influence on the performance of teams of agents evolving on the Agents on Mars scenario, a well-known agent programming testbed. The results of the evaluation show that organizational specifications that exploit this information perform better than others that don't.

Keywords: Organizations, Team Formation, Engineering Multi-Agent Systems

1 Introduction

The design of rational agents is a nontrivial task, especially in hard task environments. The hardest case corresponds to a multi-agent, unknown, partially observable, nondeterministic, sequential, continuous and dynamic environment [1]. The designer must be satisfied with a non-ideal rational agent to evolve in these domains. In a multi-agent domain, the challenge that arises is how to design the organization of the multi-agent system; more specifically, in competitive environments, the problem to solve is how to design a rational team of agents, each of them not ideally rational [2, 3].

In our context, a MAS organization is "… a supra-agent pattern of emergent or predefined cooperation of the agents in the system, that could be defined by the designer or by the agents themselves, in order to achieve a purpose." [2]. The notion of team identifies an organizational paradigm where a group of agents has to work together in order to achieve a common goal in a task environment. Each agent assumes a role and commits himself to attain some goals that are necessary to achieve the team's overall objective. The team maintains an explicit representation of its organization, called Organizational Specification (OS) [3].

This work is a contribution for the sub problem of designing the teams' organizational structure, what is called in organization theory as the "synthesis problem", namely: "which structures are best suited to solve optimally certain types of problems?" [4] or "given a certain set of conditions to be satisfied, how to find the network which is best?" [5]. In particular, the problem considers that the designer knows in which kind of environment the team should evolve and his task is to determine a suitable organizational structure that fits the task environment.

We consider that a suitable team's organizational structure must fit the task environment in each domain. We focus on the relation between some environmental patterns in the task environment and the number of groups/squads of agents in the team structure, specially to assess the influence of this relation in the performance of a team designed to confront an opponent team with technical and numerical equivalence, and aiming to conquer areas where there are scarce resources of high economic value (clusters), that are distributed in locations within a territory whose topology is unknown.

We consider that the same approach can be applied in other domains in which the environment can be represented by a weighted graph, where the vertexes denote resources and possible locations for the agents, and the edges indicate the possibility of crossing from one vertex to another with a cost for the agent. In our work, a cluster is a "valuable area" represented by a subgraph of vertexes with high resource values. The notion of environmental pattern is defined based on some clusters' spatial attributes, whose values can be perceived by the teams when they explore the environment.

The main contribution of this paper is a methodological one – viz. an empirical approach for relating features of task environments to successful agent team compositions. The team composition problem considers that the designer has worked previously on the team's OS; his task is to refine the initial structural specification, determining which roles the agents can play in the groups, and how many groups and their cardinality are necessary to maximize the team's performance evaluation measure in an environment with previously known environment patterns.

The agents' team performance evaluation is domain-dependent. In our case, it is based on the *Agents on Mars* scenario, a testbed provided by the Multi-agent Programming Contest [6]. The results of our evaluation show that organizational specifications that exploit this information perform better than others that don't. For example, the number of clusters leads, in some cases, to situations where it may be better for the whole team to occupy a single cluster, while in other cases it may be better to divide the team into smaller squads to try to gain control over multiple clusters in the environment.

2 Evaluation of the Organizational Design

Previous researches in the field of organization theory has already focused on the evaluation of the quantitative effect of the organizational design on the SMA's performance.

Horling and Lesser [7] focus on the comparison of organizational paradigms. They present a survey of the major organizational paradigms used in multi-agent systems. These include teams and others human organizational patterns. They provided the descriptions of these patterns, their advantages and disadvantages, and examples of use. They argue that their work allows the designers to recognize a large set of structural possibilities, who can realize comparative evaluation of organizational structures and then select an appropriate organizational design for a particular domain and environment.

van der Broekwork et al. [8] proposed an approach for the analysis and the formal modeling of agent-based organizations. The approach addresses both the organization structure and its dynamics. The environment is considered as a special component of the organization model. It serves as a source of events for the organization. The environment is populated by agents that under certain conditions may be allocated to organizational roles. By performing simulations and verification, the approach provides formal techniques and tools for different types of analysis of organization models.

Hodgson et al. [9] developed a framework that supports hierarchical modeling of teams of agents. A team (group) is a composite component that is characterized by a number of roles, that are enacted by agents and other teams. The framework focuses on the technical side of programming and implementation of SMA. The authors introduced a formal language for specifying the dynamics of individual roles and teams, providing different interesting types of analysis of the SMA dynamics.

Scerri et al. [10] have focused on studies about the properties and the performance of teams with hundreds of members. They developed a model of teamwork to address the limitations of others models applied to very large teams. The model organizes team members into subteams that evolve dynamically, and into overlapping subteams that work on subgoals of the overall team goal. They experimented these very large teams evolving in distinct domains, such as control of fire trucks responding to an urban disaster and simulated unmanned aerial vehicles in a battlespace.

Grossi et al. [4] [11] argue that "organizational structures should be seen along at least three dimensions: power, coordination, and control". They provided a technical terminology for describing the notion of structural organization and its properties. The concepts are defined rigorously by means of concepts from graph theory. In addition to be useful for describing the organizational structure, they can be employed to provide a formal analysis of the effect of such structures on the activities in the SMA. Their formal tool can be useful to provide numerical analyses of the organizational structures, and for evaluating to what extent an organizational structure exhibits some characteristic properties such as robustness, flexibility and efficiency.

Machado et al. [12] developed a detailed discussion related to multi-agent patrolling and an empirical evaluation of possible patrolling solutions in domains such as computer networks and computer war games. The authors proposed different architectures of multi-agent systems, various evaluation criteria, applied in two experimental settings. They implemented a patrol simulator. The results show that some kinds of architecture can patrol an area, in certain circumstances, more adequately than others.

Furtado and Filho [13] described a simulator of crimes in an urban area. The user configures and allocates police forces in certain geographical regions and then interact with the simulation, watching the crime behavior in the presence of preventive police. They described how studies involving simulations can help to determine whether a reorganization is necessary and how it should be performed. The simulation results illustrate how to exploit opportunities for the system to be reorganized.

Although some of these detailed studies about structures had tried to answer the question of which structures are best suited to solve problems in environments that can change (or not) frequently, they do not address other properties that are intrinsic in hard task environments, as the case of a partially observable and unknown environment, neither any kind of analysis relating to patterns detected in this environment.

3 The Suitable Team Structure

The design of a rational agent team can be a very complex problem. The hardest case corresponds to an environment that is partially observable, nondeterministic, sequential, continuous, unknown and dynamic [1]. One of the main challenges associated with the design problem, involves the search for solutions for the synthesis problem, namely, "which structures are best suited to solve optimally certain types of problems?" [4]. In this context, the relation between the set of possible patterns of the environment and the set of possible organizational structures for a team is a necessary information for the designer of a rational team, i.e., that must be able to use this knowledge to revise the team formation, changing for a suitable structure when the team discovers the environment patterns or when these patterns change [14].

3.1 Environmental Patterns

We focus in this work on agent teams designed to confront opponent teams with technical and numerical equivalence, and aiming to conquer areas where there are scarce resources of high economic value, that are distributed in locations within a territory whose topology is unknown, but that can be discovered by the teams as the agents interact with their environment. The agent teams' territory can be a physical environment, as a battlefield in an unconventional war, where military groups are formed in order to realize campaigns to conquer certain areas in big cities, which were contained by traffickers and militias [15]. It can be a virtual battlefield in a war electronic game, multiplayer, online first-person shooter [16]. The battle can happen in the cyberspace as well, where occurs anonymous cybernetic attacks, that are directed at political leaders, military systems, and any ordinary citizen, anywhere in the world [17].

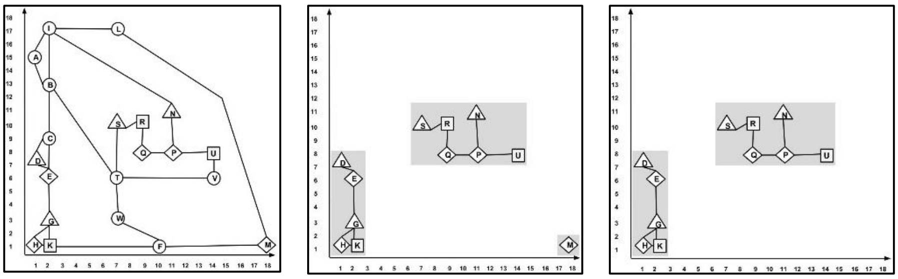
These battlefields can be represented by a vertex-edge-labeled graph (weighted graph with two vertex functions), where the vertices denote resources and possible locations for the agents, and each edge indicates the possibility of crossing from one vertex to another with a cost for the agent. Each vertex in the graph has a resource value and, optionally, an identifier. Formally, the vertex-edge-labeled graph $G = (V, E)$ consists of a set of vertices V and a set of edges E, which contains unordered pairs of distinct elements of V, a vertex value function vvf: $G(V) \rightarrow N$, a vertex labeling function vlf: $G(V) \rightarrow Idv$, and an edge value function evf: $G(E) \rightarrow R$, where N is the set of natural numbers employed to value the resources in the vertices in G, Idv is a subset of the natural numbers (e.g.: {1, ..., $|V|$}) employed to identify the $|V|$ vertices in G, and R is the set of real numbers employed to represent the cost of edge crossing.

A path (of length n) from a vertex u to a vertex v in a graph G is a collection of edges of the form (u, v_1) $(v_1, v_2), \dots, (v_n, v)$. A graph is connected if and only if: (a) it contains just one vertex, or (b) it contains a path between every pair of its vertices. A graph $G1 = (V1, E1)$ is a sub-graph of a graph $G2 = (V2, E2)$ whenever $V1 \subseteq V2$ and $E1 \subseteq E2$. The notion of environmental patterns in this work tries to capture the diversity of environments, i.e., how many "valuable areas", sub-graphs representing the higher valued resources, appear in a graph G. In our work, these “valuable areas” are called clusters. In order to formally define the notion of cluster, let infVc denotes an inferior limit value of resource capacity by vertex in a cluster, and infCc an inferior limit value of resources capacity by cluster. We define the notion of cluster in the graph $G = (V, E)$ as any sub-graph $C = (V', E')$ that satisfies three conditions:

(1) $\forall v \in C(V'), \text{vvf}(v) \geq \text{infVc}$
(2) C is a connected sub-graph
(3) infCc $\leq \sum_{v \in C(V')} \text{vvf}(v) \leq \sum_{v \in G(V)} \text{vvf}(v)$

The first condition ensures that the resource value associated with each vertex in a cluster C is greater than the inferior limit value of resource capacity by vertex. The second condition ensures that the clusters in the graph G can be a sub-graph C with only one (isolated) vertex with high value, or a sub-graph in which all vertices in C are not isolated. Finally, the third condition ensures that the sum of the resource values associated with all vertices in a cluster C is greater than the inferior limit value of resources capacity by cluster, that must be lesser than or equal the graph capacity, i.e., the sum of all values of resource capacity in the graph G.

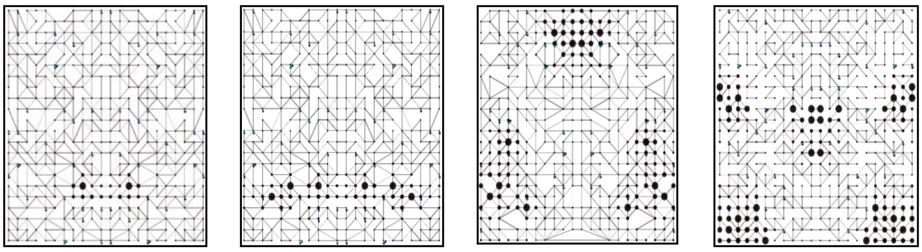
Figure 1 illustrates the notion of a cluster $C = (V', E')$ in a graph $G = (V, E)$, when $|V| = 21$ vertices and $|E| = 52$ edges. The vertices that are not graphically described by a circle in Figure 1(a) represent the vertices v in V that satisfy the first condition to be a member of V' when $\text{InfVc} = 1$, i. e., $\text{vvf}(v) \geq 1$. The three selected sub-graphs in Figure 1(b) represent those sub-graphs G' in G that satisfies the first and the second condition to be a cluster C in G. The two selected sub-graphs in Figure 1(c), C1 and C2, represent the sub-graphs G' in G that are clusters, i.e, those that satisfy the three conditions when $\text{InfCc} = 10$, i.e., $\sum_{v \in C1(V1)} \text{vvf}(v) \geq 10$ and $\sum_{v \in C2(V2)} \text{vvf}(v) \geq 10$. When considering the distribution of values of vertices of a cluster C in G, two extremal cases are possible: highly diverse clusters with the maximum possible number of distinct values, and homogeneous clusters having vertices of the same value.



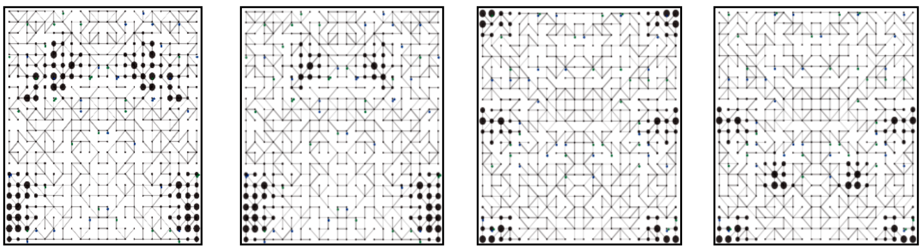
(a) First condition (b) Second condition (c) Third condition

Fig. 1. Clusters

The notion of environmental pattern associated with a vertex-edge-labeled graph G is defined as a tuple $EP(G) = <S, N, H, D>$, considering four attributes associated with the distribution of all clusters in G, such that: S represents the mean size of the clusters in G; N represents the number of clusters in G; H (homogeneity) represents how much the clusters in G have, approximately, the same value; and D (dispersion) represents the approximate mean distance between the clusters in G. So we have at least eight extremal case associated with these four attributes. Figure 1 shows graphs with different cluster patterns, setting the parameters in the cluster conditions as $infVc = 1$ and $infCc = 10$.



(a) SC1 (*S* = small) (b) SC2 (*S* = large) (c) SC3 (*N* = 3) (d) SC4 (*N* = 5)



(e) SC5 (*H* = hom.) (f) SC6 (*H* = non-hom.) (g) SC7 (D = disp.) (h) SC8 (D = non-disp.)

Fig. 2. Environments with different patterns

The larger vertices in black color identify the vertices and edges that satisfy the three cluster's conditions. Figure 2(a)-(b) illustrate different sizes for the clusters. Figure 2(c)-(d) illustrates different number of clusters, three clusters in SC3 and five clusters in SC4. The graphs in Figure 2(e)-(f) have different homogeneity values, i.e., the vertices in the clusters in SC5 have approximately the same value and the vertices in SC6 have not. Finally, different dispersion values are illustrated in Figure 2(g)-(h), i.e., where the six clusters in SC7 are more dispersed on the graph and the six clusters in SC8 are more close to each other.

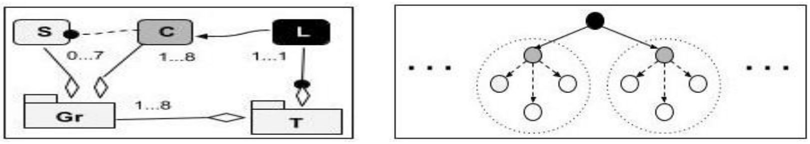
3.2 Teams' Structural Organization

A team is a human organizational pattern that can be characterized by a set of cooperative agents which have agreed to work together attempting to maximize their utility [18]. The designer must solve two related problems to design an agent team: the design of individual cooperative heterogeneous agents and the design of a rational team organization for these agents. The design of each kind of agent requires the formalization of the function that maps each sequence of perceptions into an action, that is necessary to realize tasks like to find and to conquer the best clusters, to defend these clusters, to attack the agents in the adversary team and to help the agents in the same team.

In relation to the agent organization, the designer must formally describe the team's organizational specification OS. In this work, this notion is based on the Moise organizational modeling language [19, 20]. It decomposes the organizational specification OS $= <\mathrm{SS}, \mathrm{FS}, \mathrm{DS}>$ in three dimensions. The Structural Specification (SS) defines the roles, relationships between roles, and the team's groups. The Functional Specification (FS) defines the global goals and how these are decomposed into subgoals and missions. The deontic specification (DS) relates these two dimensions, identifying subsets of missions and goals in FS that are permitted and/or required for each role in SS.

The MOISE Structural Specification SS is built in three levels. Let Agents represents the set of N agents in the team, and Roles the set of M roles that can be played by any agent $\mathrm{Ai} \in \mathrm{Agents}$. Let $\mathrm{AgRj} = \{\mathrm{Ai} \mid \mathrm{Ai} \in \mathrm{Agents\ plays\ a\ role\ Rj} \in \mathrm{Roles}\}$. At the individual level, the SS defines the set of behaviours that an agent $\mathrm{Ai} \in \mathrm{AgRj}$ is responsible for when he adopts the role Rj. At the social level, the SS defines three kinds of links between roles: authority, communication and acquaintance. An authority link $\mathrm{link}(\mathrm{Ai}, \mathrm{Ai}', \mathrm{aut})$ implies the existence of a communication link $\mathrm{link}(\mathrm{Ai}, \mathrm{Ai}', \mathrm{com})$, the latter implies the existence of an acquaintance link $\mathrm{link}(\mathrm{Ai}, \mathrm{Ai}', \mathrm{acq})$. For example, in the case where the link type is acquaintance, an agent source $\mathrm{Ai} \in \mathrm{AgRj}$, playing a source role Rj, is allowed to have a representation of the agent destination $\mathrm{Ai}' \in \mathrm{AgRj}'$, playing a destination role Rj'. At the collective level, the SS defines the aggregations of roles in groups.

A group is created from a group specification indicating the subset of roles that should be played in the group and their respectively min-max cardinality (how many agents can play those roles), its set of subgroups and their respectively min-max cardinality and the sets of intra-group and inter-group links [20]. Figure 3(a) illustrates the MOISE structural specification of a team composed of groups of agents Gr $(\min = 1; \max = 8)$, that can play two roles C $(\min = 1; \max = 8)$ and S $(\min = 0; \max = 7)$, and one group of one agent that can play one role L $(\min = 1; \max = 1)$.



(a) MOISE SS (b) SS graph representation

Fig. 3. An agent team's organizational structure

Hence, a MOISE structural specification SS can be viewed as a vertex-labeled graph $G = (V, E)$, whose vertexes V are playing role agents and whose edges E represent authority, communication and acquaintance links. Figure 3(b) illustrates the previous SS graph representation. The graph in Figure 3(b) is one of the valid representations of the team organizational structures that can be generated from the MOISE SS in Figure 3(a), where the structure of each sub-group Gr is represented by a tree, generated by the authority inter-group link between role L (agent in the black vertex) and role C (agents in the grey vertices), and the communication intra-group link between roles C and role S (agents in white vertices).

3.3 The Team Composition Problem

The team composition problem considers that the designer has worked previously on the team's organizational specification $OS = <SS, FS, DS>$, specifying which roles the agents can play in the groups and their links (SS), the team's current goals in its missions, the max-min number of agents that can commit with them (FS), and the agents that are allowed to or should commit to the goals and missions (DS). Given these previous information, the team composition task is to find a graph $Go = (Vo, Eo)$ that represents a suitable organizational structure in an environment represented by the graph $Ge = (Ve, Ee)$ with a known pattern $EP(Ge) = <S, N, H, D>$. More specifically, to find Go the designer must solve two related choice problems: (a) to choose the number of groups and (b) to choose the cardinality of each group, such as the team structural specification SS is not violated and aimed to maximize the team's performance evaluation measure in the environment Ge.

We believe that the knowledge about the relation between the set of possible patterns for the environment and the set of possible organizational structures for a team can be learned by some agents in the organization in order to eventually revise the team formation; this may be done when patterns are discovered in a partially observable and unknown environment, or when they change in a dynamic environment. We describe in the sequence how we experiment different structural specifications with distinct environmental patterns possibilities.

4 Experimental Evaluation

Our experimental approach consists of proposing different organizational specifications that are suitable for different environmental patterns (EPs), and can be adapted to generate knowledge about the agent teams' performance.

4.1 Evaluation Scenario

The application domain used in this work is the Agents on Mars scenario, developed in the Multi-Agent Programming Contest [6]. The task environment consists of a battlefield in an unconventional war, in which artificial agents have special sensors and actuators to explore and to conquer rich areas in natural resources, which are located in unknown places. Although it seems a specific type of war application, where the ground squads realize campaigns for the installation of forces in the rich resource areas, the chosen specific scenario has the major components that occur in campaigns in a diversity of application domains.

The scenario environment is represented by a graph where the vertices denote water wells and possible locations for the agents, and the edges indicate the possibility of crossing from one vertex to another with an energy cost. A zone is a sub-graph covered by a team according to a coloring algorithm based on the notion of domain [6]. Agents from different teams can be located in a single vertex. The team with the highest number of agents dominates the vertex, which receives the dominant team color. An uncolored vertex inherits the color from its neighborhoods dominant team. If the graph contains a sub-graph with a colored border, all the nodes that are within this boundary receive the same color.

At the beginning of the simulation, the map is unknown to the agents. Each team consists of 28 players that can be of five different types: explorers, sentinels, saboteurs, inspectors and repairers. These types define the characteristics of each agent such as life level, maximum energy, strength, and visibility. The roles also limit the possible actions that the agent can perform in the environment. For instance, explorers can discover water wells and help to explore the map, while sentinels have long-distance sensors and thus can observe larger areas, saboteurs can attack and disable enemies, inspectors can spy opponents, and repairers can repair damaged agents.

A team receives a cash reward whenever it reaches a major milestone. This reward can be used to empower the agents, increasing their maximum energy or strength. Different milestones can be reached during a competition, such as dominating areas with fixed values, having performed a successful number of attacks or well-succeeded defenses. If not used, the reward is added to the team's total score. The goal of each team is to maximize its score, defined as the sum of the values obtained by the occupied zones with the earned (and not yet spent) rewards in each step of the simulation.

4.2 Agent Teams

In our work, we decided to design BDI agents and to give emphasis on the functioning of the teams' organizations in graphs with different EPs. Each agent is composed of plans, a belief base and its own world model, that consists of a graph. It captures every detail received from the environment, such as explored vertices and edges, opponents' position, disabled teammates, etc.

In addition to the agents' types defined by the Scenario, we defined additional different roles in our system. Each of these roles has a mission associated with it, and can be played by one or more agents. The coordinator is a kind of agent internal to our system, which does not communicate with the MAPC server. Whenever the world model is updated, he computes which are the best zones in the graph and send this information to the other agents. Although he determines the best areas of the map, each agent decides for himself which empty vertex he will occupy to form a zone or increase it. The coordinator is also responsible for creating the organizational specification, in the beginning of a competition, and for distributing the groups, roles and missions among the other agents. Table 1 describes the mission related to each role.

Table 1. Types and missions of agents

Type	Mission
Explorer	Explores the whole graph by probing every vertex and surveying all edges on its path
Soldier	Tries to occupy one of the best zones indicated by the coordinator agent
Guardian	Defends the squad by attacking any opponent that is close to the team's zone
Medic	Repairs the agents in the squad
Soldier exp.	Explores the team's zone by probing the vertices whose values are unknown
Saboteur	Attacks any close opponent
Sentinel	Sabotages the opponent by moving inside its zone
Repairer	Repairs the saboteur and the sentinel
Coordinator	Builds its local view of the world through the percepts broadcasted by the other agents

Figure 4 shows a diagram to illustrate the team's *SS*. Our team was developed using a platform for MAS programming called JaCaMo [21], which supports all levels of abstractions – agent, environment, and organization – that are required for developing sophisticated MAS, by combining three separate technologies: Jason for programming autonomous agents [22], CArtAgO for programming environment artifacts [23, 24], and MOISE for programming multi-agent organizations [25].

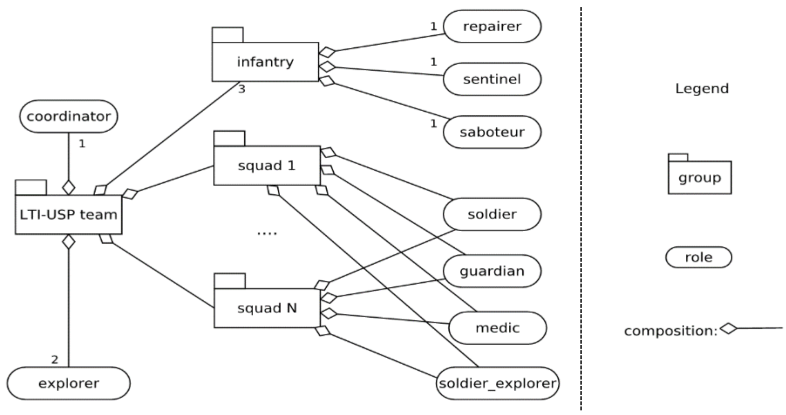


Fig. 4. Team structural specification

Jason is a Java-based interpreter for an extended version of the AgentSpeak programming language, for programming BDI agents. CArtAgO is a framework for environment programming based on the A&A meta-model [24]. In CArtAgO, the environment can be designed as a dynamic set of computational entities called artifacts, organized into workspaces, possibly distributed among various nodes of a network [21].

Each artifact represents a resource or a tool that agents can instantiate, share, use, and perceive at runtime. We made use of the organizational artifacts provided in Moise.

4.3 Description of the Experiments

The goal of our experiments was to evaluate the impact of the environmental patterns over teams' performance, by modifying the environmental pattern $EP(G) = <S, N, H, D>$ in the Agent on Mars map, and measuring the performance of two adversary teams composed by the same BDI agents, but with two different organizational specifications $OS = (SS, FS, DS)$. For each team, the performance measure adds the values of its conquered clusters with the earned rewards, as the team reaches a major milestone. The experiments tried to perceive the impact of the EP over teams' performance.

The experiments consisted of seven teams: TG1, TG2, TG3, TG4, TG6, TG8 and TG10. These seven teams competed in 14 environments with different EPs, including the eight scenarios illustrated in Figure 1. In relation to teams' OS, we fixed the attribute values associated with the organizational dimensions FS and DS, and modified the number of squads and the cardinality of each squad in the structural dimension SS. Each team TGN is composed of one infantry group, two explorers and the other 23 agents divided into n squads. Each squad had nearly the same number of agents, and respected the number of agents of each type when assigning the roles.

Each team played against each other three times, and the team that wins most matches wins the overall tournament. Each match had 750 steps and the map was randomly generated. From one match to another we changed the number of vertices, edges and high-valued areas. For each environment, we performed 10 simulations for each of the possible matches. The data collected in all simulation was the winner and the final score of each team. These metrics were used to indicate the performance of each team. We used the Wilcoxon T test as a hypothesis test to define for each match if the 10 simulations were sufficient or not to conclude that a team was better than other in a determined environment. The results of the Wilcoxon T test for the environment SC3 (Figure 1) is shown in Table 2.

Table 2. Results of the Wilcoxon T test for SC3

	TG1	**TG2**	**TG3**	**TG4**	**TG6**	**TG8**	**TG10**
TG1		0,0059	0,0195	0,0020	0,0039	0,0371	0,0137
TG2	0,0059		0,6953	0,0488	0,0273	0,7695	0,6250
TG3	0,0195	0,6953		0,1602	0,0371	0,3750	0,0371
TG4	0,0020	0,0488	0,1602		0,1602	0,0840	0,0273
TG6	0,0039	0,0273	0,0371	0,1602		0,1602	0,2324
TG8	0,0371	0,7695	0,3750	0,0840	0,1602		0,6250
TG10	0,0137	0,6250	0,0371	0,0273	0,2324	0,6250	

The Wilcoxon T test is a non-parametric test for dependent samples that can indicate with some stated confidence level if a particular population tends to have larger values than other. Based on the results of the Wilcoxon T test, it's possible to represent the partial order of a certain number of teams given their performance in a specific environment by a partial order graph. Figure 5 illustrates the partial order graph obtained by drawing an edge from team A to team B whenever a significant difference (p – value < 0.05) exists between the performances of two teams.

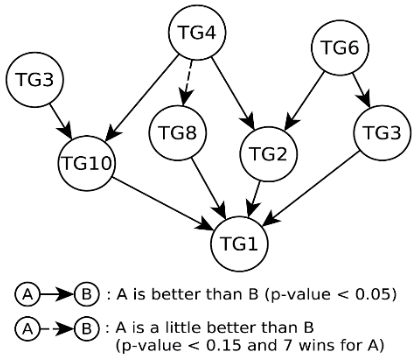
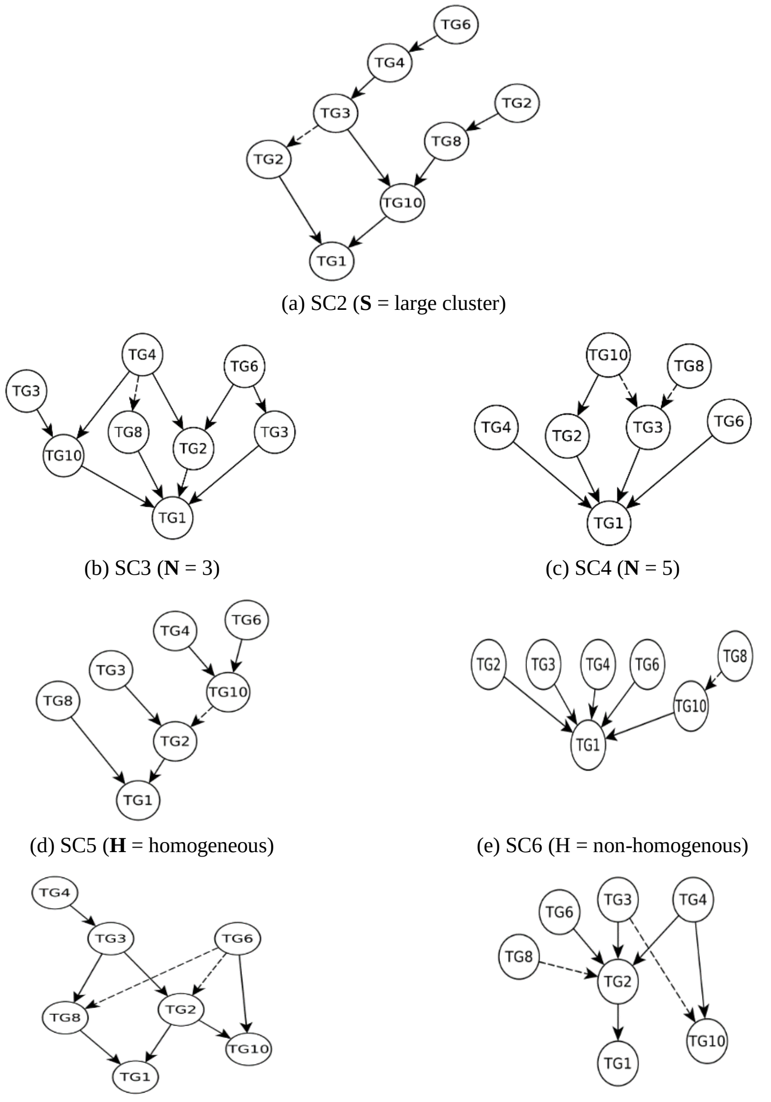


Fig. 5. Example of a partial order graph

The direction of the edge will be from the team with better performance to the team with lower performance, and we omit the edges that can be extrapolated through a transitive closure, that is, those edges for which there is already a path connecting two teams. The dashed edges represent the matches for which the Wilcoxon T test came close to detect a significant difference between two teams (p – value < 0.15 and at least 7 wins for one team). Finally, it's possible that the same team appears twice or more times in graph as shown in Figure 5, in which TG3 appears twice. This is necessary so that we can represent that TG3 achieved a better performance than TG10, but the same was not observed for TG6 and TG10. This analysis was performed on all environments and the results obtained are presented in the following subsections.

4.4 Obtained Results

Here, a cluster is any subgraph of the maps formed by vertices with value greater than 1 (infVc), in which the sum of all vertices is greater than 10 (infCc). In relation to clusters' size (S), in general, a team with a number of squads equal to the number of zones (a zone is a sub-graph covered by a team according to a coloring algorithm based on the notion of domain [6]) in the cluster does better. In the environment SC2 it's possible to divide the cluster in 6 zones to be occupied by the squads and, as shown in Figure 6(a), TG6 is better than almost all the other teams, because: (i) a smaller number of squads is not effective in securing and defending bigger clusters, and (ii) a larger number ends up by causing some squads to disperse out of the cluster, making it difficult to conquer it.



(a) SC2 (**S** = large cluster)

(b) SC3 (**N** = 3)

(c) SC4 (**N** = 5)

(d) SC5 (**H** = homogeneous)

(e) SC6 (H = non-homogenous)

(f) SC7 (**D** = dispersed)

(g) SC8 (**D** = non-dispersed)

Fig. 6. Partial order graphs for different environments

Regarding the number of clusters (N), the team must have a number of squads equal or closer to the number of clusters on the map. If the number of squads is smaller than the number of clusters, the team will not cover all good areas, which can then be easily occupied by the opponent. If the number of squads is greater than the number of clusters, some agents will be placed in areas of small values, weakening the squads' attack and defense. For example, in Figure 6(b) it is possible to see that for the environment SC3 with 3 clusters, TG3 performs better than TG10, while TG10 outperforms TG3 for SC4, which has 5 clusters.

Regarding the homogeneity (H), the experiments showed that finding and occupying the clusters with the highest values is critical in non-homogeneous environments. In these cases, it is better to balance among the teams (Figure 6(c)) since the winner ends up being defined by the team that occupies the bigger clusters. This is good for teams with small number of squads, because there is a chance that they occupy the clusters with highest values, while the opponent with a larger number of squads ends up by spreading its agents in smaller and not valuable clusters.

Regarding the clusters' dispersion (D), the results showed that less dispersed clusters help teams with a larger number of squads to form larger areas than they could if the clusters (and the agents) were dispersed (Figure 6(d)). Moreover, with the proximity of the clusters, the squads tend to be closer of each other, making it easier to attack the opponent. For example, TG3 and TG4 can dominate large areas in SC7, which often comprise two neighbor clusters, while TG8 occupies smaller areas and does not attack efficiently the opponents. However, in SC8, with the proximity of the clusters, TG8's squads tend to be closer, making it easier to attack the opponent, and thus helping TG8 to occupy sometimes areas that cover more than one cluster at the same time.

5 Conclusions and Further Work

Our evaluation tried to assess the impact of environmental patterns in the performance of a team of agents, designed to confront an "equivalent" opponent team, aiming to conquer areas with scarce resources of high economic value, in unknown locations in the environment. In our work, these areas were denoted as clusters and were employed to generate the notion of environmental pattern, which was essential to operationalize the evaluation.

We performed the evaluation considering the impact of the environmental patterns in a valued graph map over the performance of teams with the same functional and deontic specifications, but a with different structural specification, due the different numbers of squads and the cardinality of each squad in the team. Although our results are preliminary ones, we believe that they provide at least two contributions that can be exploited in the design of agents' teams when the task environment is hard, but that can be described in terms of environmental patterns.

The first contribution is related to the knowledge the designer can learn about these task environments, in order to assess whether a team will be able to selectively search for solutions in the map. The results of our evaluation provided some knowledge about the different organizational specifications that are suitable to different environmental patterns in the maps. In this sense, we hope to intensify this initial evaluation in two directions: (1) from the environment' side, to evolve scenarios that aggregate diversified environmental patterns, obtained from the combination of the different attribute values employed to represent patterns; and (2) from the team's side to extend the approach to other attribute values associated with the three dimensions employed to define organizational specifications.

The second contribution is related to the proper concretization of the notion of environmental pattern realized in the work. Considering this notion as a complementary representation of the state of the environment, and the consistent knowledge that can be provided by a more intensive evaluation of the impact of environmental patterns over the performance of teams, we believe that two possibilities are generated for the designer (or any artificial agent properly designed): (1) to predict the behavior of a team if he knows its goals, its organizational specification, and the current environmental patterns; (2) to define a suitable organizational specification for a team and the properties of its behavior if the agent knows the current environment pattern and team's goal. These are hypotheses that we hope to prove in the near future.

6 Acknowledgements

Jaime Simão Sichman is partially supported by CNPq, Brazil, grant agreement no. 303950/2013-7.

7 References

1. Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson Education, 3nd. Edition (2010).
2. Boissier, O., Sichman, J.S.: Organization oriented programming. Tutorial Notes, 3rd. International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), New York (2004).
3. Coutinho, L., Sichman, J., Boissier, O.: Modeling dimensions for agent organizations. In V. Dignum, editor, Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models. IGI Global (2009).
4. Grossi, D., Dignum, F., Dignum, V., Dastani, M., Royakke, L.: Structural Aspects of the Evaluation of Agent Organizations. Coordination, Organizations, Institutions, and Norms in Agent Systems II, Lecture Notes in Computer Science, Volume 4386, pp 3-18 (2006).
5. Harary, F.: Status and contrastatus. Sociometry, vol. 22, pp. 23– 43 (1959).
6. Tristan, B., Michael, K., Federico, S., Jürgen, D., Jomi, F. H. 2012: The Multi-agent Programming Contest 2011: A Résumé. In Louise Dennis, Olivier Boissier, and Rafael Bordini, editors, Programming Multi-Agent Systems, vol. 7217 of Lecture Notes in Computer Science, pp. 155–172 Springer, Heidelberg (2012).
7. Horling, B., Lesser, V.: A Survey of Multi-Agent Organizational Paradigms. Computer Science Technical Report 04-45. University of Massachusetts (2004).
8. van der Broek, E. L., Jonker, C. M., Sharpanskykh, A., Treur, J., Yolum, P. Formal modeling and analysis of organizations. In: Boissier, O., Dignum, V., Matson, E., and Sichman, J. (eds) Proceedings of the International Workshop on Organizations in Multi-Agent Systems (OOOP), Utrecht (2005).
9. Hodgson, A., Roennquist, R., Busetta, P., Howden, N.: Team Oriented Programming with Simple Team, in Proc. of SimTecT 2000, pp. 115-122. Sydney, Australia (2000).
10. Scerri, P., Xu, Y., Liao, E., Lai, J., Sycara, K.: Scaling teamwork to very large teams. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04). ACM Press. New York (2004).
11. Grossi, D., Dignum, F., Dastani, M., Royakkers, L. Foundations of organizational structures in multiagent systems. Proceedings of AAMAS' 05, Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 690–697. ACM Press. Utrecht (2005).

12. Machado, A., Ramalho, G., Zucker, J-D., Drogoul, A.: Multi-agent Patrolling: An Empirical Analysis of Alternative Architectures. Multi-Agent-Based Simulation II. Vol. 2581 of the series Lecture Notes in Computer Science, pp 155-170 (2003).
13. Furtado, V., Filho, E. J.: A Multi-Agent Simulator for Teaching Police Allocation. In: Proceedings IAAI'05 Proceedings of the 17th conference on Innovative applications of artificial intelligence - Volume 3 Pages 1521-1528 (2005).
14. Lu, H: Team formation in agent-based computer games. In: Pisan, Y. (Ed.), 2nd Australasian conference on Interactive Entertainment, IE 2005. ACM Press, pp. 121–124. Sydney, Australia (2005).
15. Melo, A., Belchior, M., Furtado, V.: Analyzing Police Patrol Routes with the Simulation of the Physical Reorganization of Agents. In Proc. of the 6th Int. Workshop on Multi-Agent Based Simulation (MABS'05), Springer-Verlag (2005).
16. The Gamasutra Quantum Leap Awards: First-Person Shooters, GamaSutra, http://www.gamasutra.com/view/feature/1832/the_gamasutra_quantum_leap_awards_.php, (accessed February 8, 2016).
17. Czosseck, C. and Geers, K.: The Virtual Battlefield: Perspectives on Cyber Warfare. In: Cryptology and Information Security Series, vol. 3, IOS Press (2009).
18. Carley, K., Gasser, L.: Computational organization theory. In: Weiss, G. (ed) Multiagent Systems, pp 299–330. MIT Press, Cambridge, MA (1999).
19. Hübner, J., Sichman, J., Boissier, O.: A model for the structural, functional, and deontic specification of organizations in multi-agent systems. Advances in Artificial Intelligence Volume 2507 of the series Lecture Notes in Computer Science, pp 118-128 (2002).
20. Hübner, J., Sichman, J., Boissier, O.: Developing organised multiagent systems using the MOISE+ model: programming issues at the system and agent levels. International Journal of Agent-Oriented Software Engineering, pp. 1–27 (2007).
21. Boissier, O., Bordini, R., Hübner, J., Santi, A.: Multi-agent oriented programming with JaCaMo. Science of Computer Programming, 78(6): 747-761 (2011).
22. Bordini, R., Hübner, J., Wooldridge, M.: Programming multi-agent systems in AgentSpeak using Jason. Wiley-Blackwell (2007).
23. Ricci, A., Piunti, M., Viroli, M.: Environment programming in multi-agent systems: an artifact-based perspective. Autonomous Agents and Multi-Agent Systems, 23(2): pp. 158–192 (2010).
24. Andrea, O., Alessandro, R., Viroli, M. Artifacts in the A&A meta-model for multi-agent systems. Autonomous Agents and Multi-Agent Systems, 17(3): pp. 432–456 (2008).
25. Hübner, J., Boissier, O., Kitio, R., Ricci, A. Instrumenting multi-agent organisations with organisational artifacts and agents. Autonomous Agents and Multi-Agent Systems, 20(3): pp. 369–400 (2009).

On the Minimal Recognition of Rights in Holonic Institutions

Jeremy Pitt[1], Jie Jiang[1] and Ada Diaconescu[2]

[1] Imperial College London, Exhibition Road, SW7 2BT, UK
[2] Télécom ParisTech CNRS LTCI, 46 rue Barrault F-75634 Paris Cedex 13, France

Abstract. In one aspect of her study of collective action, Ostrom proposed eight design principles for the supply of institutions for sustainable common-pool resource management. Computational logic has been used to formalise an executable specification of six of these principles for resource allocation in open multi-agent systems and networks. However, the eighth principle, *nested enterprises*, is structural rather than procedural, and the seventh principle, *minimal recognition of rights*, concerns a critical relationship between the components of that structure – not just the *right to self-organise*, but essentially *enough* (i.e. minimal) rights to self-organise. The idea of holonic institutions has been proposed to satisfy the requirement of polycentric self-governance in complex systems of nested enterprises. This paper investigates the axiomatic specification of Ostrom's seventh principle as a constraint on the holonic structure, as a prelude to a more systematic investigation into values, conflict resolution and the trade-off between rights and powers in holonic institutions.

Keywords: Electronic Institutions, Holonic Architectures, Self-Organisation, Rights, Powers, Conflict Resolution.

1 Introduction

Based on extensive fieldwork, Ostrom showed [12] that it is possible for a community (of *appropriators*) to develop its own solution to a "tragedy of the commons" situation, i.e. a collective action situation featuring open access to a communal resource, in which appropriators are incentivised to act (rationally) to maximise their short-term utility, even if that results in (irrationally) the depletion of the resource in the long term. These solutions, which successfully sustained the resource over extended periods of time, were based on sets on conventional rules, with which the appropriators voluntarily agreed to comply and so constrain their own otherwise unrestricted actions. There was also a mutual agreement not to repudiate these rules: as conventional rather than physical rules they could be broken. Therefore the appropriators could not refuse to accept that the rules should be monitored for compliance, that they should be enforced, or that proportional graduated sanctions should be imposed for transgression.

Ostrom called these sets of rules *self-governing institutions*, based on the idea that those affected by the rules should participate in their formation, selection and modification, hence self-governance. However, just positing an arbitrary set of rules was not in itself enough: sometimes there were such rules and the common-pool resource was *not* sustained. She observed that the self-governing institutions that successfully sustained the resource had eight features in common, otherwise one or more of the features were weak or missing altogether.

She then went one step further, which was to turn to the issue of *supply*. She argued that, faced with a collective action situation, it was not necessary to trust in 'evolution' (in the sense that the appropriators would form, select and modify an institution which exhibited the necessary characteristics). Instead, since it is now known what the necessary characteristics are, it should be possible to *design* institutions with those characteristics from the beginning. To address this issue, Ostrom proposed eight *institutional design principles* for the supply of institutions for sustainable common-pool resource management.

Computational logic, specifically the Event Calculus [8], has been used to formalise an executable specification of the first six of these principles, related to boundary conditions, congruence, self-determination, monitoring, sanctions and conflict resolution (see Section 2.1). Simulations of resource allocation in open multi-agent systems showed that as more of these principles were axiomatised, the more likely it was that the agents could self-organise an 'institution' that maintained participation and sustained the system [19]. Additionally, it was shown that by axiomatising a theory of distributive justice [20] in the context of an economy of scarcity, over time a fair allocation (as measured by the Gini index) could be achieved, despite the allocation at any one timepoint being unfair (given that at each timepoint, since some agents received zero allocation, there was no metric that would return a indicator that the allocation was fair) [13].

The six principles used in these experiments could all be given a procedural interpretation and hence an executable (declarative) specification. Furthermore, all of these simulations used either a single institution, or when there were multiple institutions, they were all independent and their internal operations did not interfere with each other. However, the eighth principle, *nested enterprises*, is structural rather than procedural, and is specifically concerned with the interaction and inter-dependence between multiple institutions and their relational arrangements. The seventh principle, *minimal recognition of rights*, concerns a critical relationship between the components of that structure – not just the *right to self-organise*, but essentially *enough* (i.e. minimal) rights to self-organise.

These last two principles are critical for issues of *scale*. This was a particular concern for Ostrom in her later work: e.g. whether or not global scale collective action problems (like climate change) demanded top-down solutions [11]. For Ostrom, the answer was at best ambiguous, but one factor was clear: that as systems scaled up and became more complicated, it was the weakness of the seventh principle and the lack of *polycentric* self-governance (multiple centres of decision-making) that was the root cause of systemic failure. By contrast, polycentric self-governance enabled meaningful and sustainable behaviours were observed to emerge from a seemingly chaotic complex system [10].

The idea of *holonic institutions* [3, 18] has been proposed to satisfy the requirement of polycentric self-governance in complex systems of nested enterprises. This enables an investigation of the axiomatic specification of Ostrom's seventh principle, as a constraint on the holonic structure, pursued in this paper.

Accordingly, the paper is structured as follows. Section 2 reviews the background to this work, covering Ostrom's institutional design principles, their axiomatic specification in a dialect of the Event Calculus, and the relation to dynamic norm-governed multi-agent systems [1]. Section 3 reviews the idea of holonic institutions, which converges the benefits of institutions and conventional rules for regulating behaviour with the management of complexity, scale, diversity, stability and robustness afforded by holonics. Section 4 contains the critical contribution of the paper, with an attempt to characterise the "minimal recognition of rights" (to self-organise) in terms of an 'empowerment' component and an 'entitlement' component (cf. voting in [14]), and axiomatise these two components in terms of the Event Calculus. The outcome is that while the eighth principle is structural, its characterisation in terms of holonic institutions does enable an analysis and formalisation of the seventh principle within the same framework. Further and related work is considered in Section 5, and some concluding remarks are made in Section 6, specifically as the trade-off between rights and powers impacts the *values* manifested in, or by, holonic institutions.

2 Background

This section reviews some of the background to this work. We begin with a summary of Ostrom's institutional design principles in Section 2.1, continue with the axiomatisation of the principles in the Event Calculus in self-organisng electronic institutions in Section 2.2, and consider these as a special sub-class of dynamic norm-governed multi-agent systems in Section 2.3. This lays the foundations for the formal analysis of the "nested enterprise" and "minimal recognition of rights" design principles in respectively Section 3 and Section 4.

2.1 Ostrom's Institutional Design Principles

Ostrom's institutional design principles are, of course fully specified in [12] and have been well-documented in previous works (e.g. [19]). They are reproduced here for completeness (to make this paper self-contained) and for a point of clarification (see below). Therefore the design principles for a self-governing institution to *sustain* a common-pool resource, as originally specified in [12, p. 90], are listed in Table 1. Subsequent extensive fieldwork has corroborated these principles with only minor modifications [2], and that institutions for sustainable common-pool resource management do exhibit these eight features. Correspondingly, research has also revealed numerous examples where absence of one or more of the principles led to depletion of the resource.

Design principle P1 is concerned with ensuring clear distinctions between who is and is not a member of the institution, and which common-pool resources are, or are not, managed by the institution. The second principle P2 is about

Table 1: Ostrom's institutional design principles

P1	Clearly defined boundaries	**P2**	Congruence between rules and local conditions
P3	Collective-choice arrangements	**P4**	Monitoring
P5	Graduated sanctions	**P6**	Conflict-resolution mechanisms
P7	Minimal recognition of rights to organise	**P8**	Nested enterprises

ensuring that appropriation rules are congruent with local environmental conditions (e.g. ration in times of shortage and free-hand in times of excess, and not the other way round) and to provision rules which themselves may incur costs (time, labour, money, etc.). The third principle P3 concerns self-determination, and ensuring that "most" appropriators who are affected by the rules participate in the decision-making processes controlling the selection and modification of the rules. Principle P4 requires that monitors, who audit appropriator behaviour, are themselves appropriators or are appointed by and accountable to them. Principle P5 states that non-compliant appropriators should be sanctioned according to a principle of proportionality (although see [17]), and Principle P6 specifies that conflict resolution mechanisms should be "rapid" and "low-cost".

As indicated above, Principle P8 is concerned with structure, specifically that all the activities covered by the first six principles, i.e. provision, appropriation, self-determination, monitoring, enforcement, conflict resolution and governance, are organised in multiple layers of nested enterprises. Principle P7 concerns a particular constraint on the relationships between layers.

Particularly interesting successes and failures in complex systems, where the eighth principle is required, have often been highlighted by the corresponding presence or absence of principle P7. One notable success story is the rice plantations in Bali, where crop planting is a careful trade-off between water scarcity and pest dispersion dynamics, and was carefully managed by a process of prayer and signalling which connected base-level farmers to entire regions through the a meso-level of water temples [9]. A notable failure documented by Ostrom [12, p. 175-177] is the example of fisheries in Canada, where a federal insistence of a "one size fits all" policy over-rode local arrangements that could take specific contextual, seasonal, historical and other environmental factors into account. This led to collapse of the local fishing stocks and industries.

In general, we are concerned with axiomatisation of all the principles for self-organising multi-agent systems, although it is the formalisation of Principle P7, in the context of a representation of structure in Principle P8 in terms of *holonic* institutions (see Section 3), that is the primary subject of investigation. For this, though, we first outline our overall approach to the axiomatisation.

2.2 Axiomatisation of the Principles

To address the problem of resource allocation in open multi-agent systems, where there is no centralised controller, institutional design principles P1-P6 were for-

Table 2: Main Predicates of the Event Calculus (EC).

Predicate	**Meaning**
Act happensAt T	Action Act occurs at time T
initially $F = V$	The value of fluent F is V at time 0
$F = V$ holdsAt T	The value of fluent F is V at time T
Act initiates $F = V$ at T	The occurrence of action Act at time T initiates a period of time for which the value of fluent F is V
Act terminates $F = V$ at T	The occurrence of action Act at time T terminates a period of time for which the value of fluent F is V

mally specified in computational logic, specifically using a dialect of the Event Calculus (EC) [8], which could then be operationalised as a logic program. As such, the logic program constitutes both a specification and executable code for algorithmic self-governance [19].

The EC is a logic formalism for representing and reasoning about actions or events and their effects. The EC is based on a many-sorted first-order predicate calculus. For the dialect used here (referred to as "the" Event Calculus), the underlying model of time is linear, so we use non-negative integer time-points (although this is not an EC restriction). We do not assume that time is discrete (the numbers need not correspond to a uniform duration) but we do impose a relative/partial ordering for events: for non-negative integers, $<$ is sufficient.

An *action description* in EC includes axioms that define: the action occurrences, with the use of happensAt predicates; the effects of actions, with the use of initiates and terminates predicates; and the values of the fluents, with the use of initially and holdsAt predicates. Table 2 summarises the main EC predicates. Variables, that start with an upper-case letter, are assumed to be universally quantified unless otherwise indicated. Predicates, function symbols and constants start with a lower-case letter.

Where F is a *fluent*, which is a property that is allowed to have different values at different points in time, the term $F = V$ denotes that fluent F has value V. Boolean fluents are a special case in which the possible values are *true* and *false*. Informally, $F = V$ holds at a particular time-point if $F = V$ has been *initiated* by an action at some earlier time-point, and not *terminated* by another action in the meantime. In our case, we are particularly interested in those fluents which specify the (institutionalised) powers (**pow**), permissions (**per**) and obligations (**obl**) of an agent, i.e. we want to know when $\mathbf{pow}(Agent, Action) = true$, $\mathbf{per}(Agent, Action) = true$, and $\mathbf{obl}(Agent, Action) = true$.

Events initiate and terminate a period of time during which a fluent holds a value continuously. Events occur at specific times (when they *happen*). A set of events, each with a given time, is called a *narrative*.

The utility of the Event Calculus comes from being able to reason with narratives. Therefore the final part of an EC specification is the domain-independent 'engine' which computes what fluents hold, i.e. have the value *true* in the case of boolean fluents, or what value a fluent takes, for each multi-valued fluent. This can be used to compute a 'state' of the specification in terms of the fluents representing institutional facts. This state changes over time as events happen, and includes the roles, powers, permissions and obligations of agents, and the values assigned to each of the fluents (in particular, which method is used for access control, which method is used for winner determination, and so on).

For example, considering design principle P1, suppose an agent G is assigned to the role of *gatekeeper*, and so is empowered to admit an agent A as a *member* to the institution I, by an *assign* action, depending on the access control method.

$$
\begin{array}{l}
\mathit{assign}(G, A, \mathit{member}, I) \quad \textsf{initiates} \quad \mathit{role\_of}(A, \mathit{member}, I) = \mathit{true} \quad \textsf{at} \quad T \quad \leftarrow \\
\qquad\qquad \mathbf{pow}(G, \mathit{assign}(G, A, \mathit{member}, I)) = \mathit{true} \quad \textsf{holdsAt} \quad T \\
\mathbf{pow}(G, \mathit{assign}(G, A, \mathit{member}, I)) = \mathit{true} \quad \textsf{holdsAt} \quad T \quad \leftarrow \\
\qquad\qquad \mathit{applied}(A, I) = \mathit{true} \quad \textsf{holdsAt} \quad T \quad \wedge \\
\qquad\qquad \mathit{acMethod}(I) = \mathit{attribute} \quad \textsf{holdsAt} \quad T \quad \wedge \\
\qquad\qquad \mathit{role\_of}(G, \mathit{gatekeeper}, I) = \mathit{true} \quad \textsf{holds} \quad T \quad \wedge \\
\qquad\qquad \mathit{role\_conditions}(\mathit{member}, A, I) = \mathit{true} \quad \textsf{holdsAt} \quad T \\
\mathbf{pow}(G, \mathit{assign}(G, A, \mathit{member}, I)) = \mathit{true} \quad \textsf{holdsAt} \quad T \quad \leftarrow \\
\qquad\qquad \mathit{applied}(A, I) = \mathit{true} \quad \textsf{holdsAt} \quad T \quad \wedge \\
\qquad\qquad \mathit{acMethod}(I) = \mathit{discretionary} \quad \textsf{holdsAt} \quad T \quad \wedge \\
\qquad\qquad \mathit{role\_of}(G, \mathit{gatekeeper}, I) = \mathit{true} \quad \textsf{holdsAt} \quad T
\end{array}
$$

If the agent A has applied to join the institution (i.e. the gatekeeper cannot arbitrarily assign membership) and *acMethod* is *attribute*, then the agent G occupying the role of gatekeeper is empowered to assign the role *member* provided the applicant satisfies certain (external) role conditions. The conditions could include, for example, not exceeding a fixed number of non-compliant actions, a duration since the last non-compliant action, and so on. Similarly, if the *acMethod* is *discretionary*, then the *gatekeeper* is empowered to assign the role without conditions, according to its (internal) decision-making, which could yet make reference to external conditions.

Given a sequence of events, i.e. a narrative, it is then possible to animate (query) the specification to determine what powers, permissions and obligation hold at the start of the narrative, at the end, and at each time point in-between.

2.3 Dynamic Norm-Governed Multi-Agent Systems

In our experience with formalising the principles, the first six are, or can be given, an operational, functional or procedural reading which is amenable to declarative specification in this form. However the eighth principle is structural and the seventh principle is concerned with specific relationships between layers in that structure.

For principle P7, the sub-text states that "the rights of appropriators to devise their own institutions are not challenged by external governmental authorities" [12, p. 90]. In some earlier works, which were dealing with only a single institution, the significance of this principle was underestimated (and when paraphrased in terms of autonomy, "whatever rules the members agree to govern their affairs, no external authority can overrule them", even somewhat inaccurately). The key issue is that the *right* to self-organisation should not be challenged by external authorities, but specific outcomes of the self-organisation can be.

For example, suppose an 'inner' institution specified its system of graduated sanctions as "First offence: fine €5. Second offence: fine €10. Third offence: execution". An external authority, as the name implies, one with *power* over the inner institution, could well challenge and deny this on the grounds that executing people is not something considered acceptable; in fact, it is illegal and subject to sanction. The external authority should not be able to challenge or deny the *right* of the inner institution to self-organise its system of graduated sanctions, although it can deny specific configurations. This is the nuance that we need to capture in the formalisation of the seventh principle, for which the framework of dynamic norm-governed (multi-agent) systems [1] is required.

This framework allows agents to modify the rules or protocols of a norm-governed system at runtime. The framework defines three components: a specification of a norm-governed system, a protocol stack for defining how to change the specification, and a topological space for expressing the 'distance' between one specification instance and another.

Firstly, the norm-governed specification expresses five aspects of social constraint: the physical capabilities; the institutionalised powers; the permissions, prohibitions and obligations of the agents; the sanctions and enforcement policies that deal with the performance of prohibited actions and non-compliance with obligations; and the designated roles of empowered agents.

Secondly, the protocol *stack* is used by the agents to modify the rules or protocols of a norm-governed system at runtime. This stack defines a set of object level protocols, and assumes that during the execution of an object protocol the participants could start a meta-protocol to (try to) modify the object-level protocol. The meta-protocol could initiate a meta-meta protocol to modify the meta-protocol, and so on. Finally 'transition' protocols define the conditions in which an agent may initiate a meta-protocol, who occupies which role in the meta-protocol, and what elements (the *degrees of freedom*: DoF) of an object protocol can be modified as a result of the meta-protocol execution. Given a set of DoF, by assigning a value to each DoF, we get a *specification instance*.

Thirdly, a set of rules R implicitly defines a specification space $\mathcal{L}$, where each instance of the specification space is characterised by a different assignment of values to each parameter in each rule. The size of this space is given by

$$
\begin{aligned}
\mid \mathcal{L} \mid = {} & (V_{1,1} \times V_{1,2} \times \ldots \times V_{1,P_1}) \times (V_{2,1} \times V_{2,2} \times \ldots \times V_{2,P_2}) \times \ldots \times \\
& (V_{R,1} \times V_{R,2} \times \ldots \times V_{R,P_R})
\end{aligned}
$$

where $V_{i,j}$ is the number of values that the jth parameter of rule i can take, P_i is the number of parameters of rule i, and R is the number of rules in the set.

This is the basis for defining a *specification space* as a 2-tuple, where one component is the set of all possible specification instances and the other component is a function d which defines a 'distance' between any pair of elements in the set. It then possible to define rules which prohibit certain instances, or which limit the 'distance' that movement between specification instances is 'allowed', as illustrated in Figure 1.

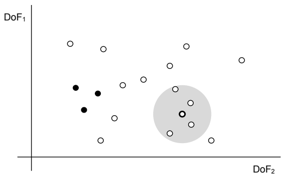


Fig. 1: An illustrative specification space with two DoF

In this figure, there are two DoF. Unfilled circles represent 'allowable' specification instances, and filled circles are prohibited instances. The circle with the bold perimeter in the middle of the shaded area is the current specification instance, the rule specifying the 'distance' that can be moved when changing instances limits the changes to the four instances included within the gray area. (Note that this distance is also a DoF, and therefore also changeable.)

3 Holonic Institutions

3.1 Holonic Systems

A *holonic system* (or *holarchy*) is composed of interrelated subsystems, each of which are in turn composed of sub-subsystems and so on, recursively, until reaching a lowest level of 'elementary' subsystems. As emphasised by Koestler [7], each such intermediary sub-system must play a dual role and be both an autonomous whole controlling its parts; and a dependent part of a supra-system. This helps construct large systems with macro-goals from intermediary components able to achieve partial goals.

3.2 Holonic Institutions

Figure 2-a depicts a generic conceptual model (abstract architecture) of holonic institutions to help address the questions above. In short, each holonic institution features two complimentary regulatory components implementing their dual roles, for 'inward' and 'outward' regulation. *Inward regulation* includes the internal rules, governance and adaptation functions for achieving a goal. *Outward*

regulation merges, via conflict resolution and negotiation, the institution's own common goal with the (supra-)institutions' common goals. This results in the compromise goal that the institution agrees to pursue. Each holonic institution is encapsulated within a *membrane* providing *membership-control* functions. At a high level of abstraction, this approach addresess the issue of the *composition* of institutions (Figure 2-b). Institution adaptation relies on feedback from members and from the institution's evaluation of its goal achievement; it is propagated progressively from lower to upper holonic levels. Furthermore, this component-oriented design helps formalise, understand and analyse composite institutions, providing a key basis for addressing the challenge of institutional complexity.

A somewhat superficial use of holonic systems would be just to capture the hierarchical decomposition implicit in the simple expression of Ostrom's eighth principle as nested enterprises, concerning just the provision and appropriation systems. However, Ostrom's nested enterprises considers all forms of provision, appropriation, monitoring, enforcement, conflict resolution and other governance activities as different enterprises operating in multiple layers. Therefore, invoking the full 'power' of holonic systems offers the possibility of considering the distinct enterprises as social (i.e. interacting) constructs in their own right, even if it is atomic level actors who are performing these actions, "as if" the enterprise did it for itself (cf. the notion of "counts as" in the Jones and Sergot account of institutionalised power [6]). This also enables the characterisation of the holon's dual role, both inward ('selfish', or dealing with an 'inner realm'), and outward ('transcendental', or dealing with an 'outer realm'), and lines of demarcation between the two, for example in terms of boundaries, jurisdictions and sovereignty (i.e. the authority to self-govern).

Therefore the characterisation of Ostrom's principle P8 in terms of holonic institutions is concerned with much more than hierarchical decomposition. It is more significantly concerned with capturing the requirements of polycentric governance [3, 18] and the management of multi-scale, multi-criteria (sub-)optimization. Not that "sub" is used here in the sense that there will be conflicting goals, not all of which can be satisfied, and the checks, balances, compromises and effective conflict resolution mechanisms that enable meaningful and sustainable behaviours to emerge from complex systems need to be carefully considered. Principle P7 is a crucial element of this.

The above considerations provide a generic architectural overview on the manner in which *holonic institutions* can be constructed and maintained to address the aforementioned questions and achieve the advantages enabled by holonic principles. However, it also provides a handle on how to formalise the minimal recognition of rights, by characterising the requirements of the rulesets of the 'inner' and 'outer' (supra) institutions in Figure 2-b.

4 The Minimal Right to Self-Organise

In this section, we try to be precise about what the "minimal recognition of rights" entails for the structural relationship between an institutional holon and

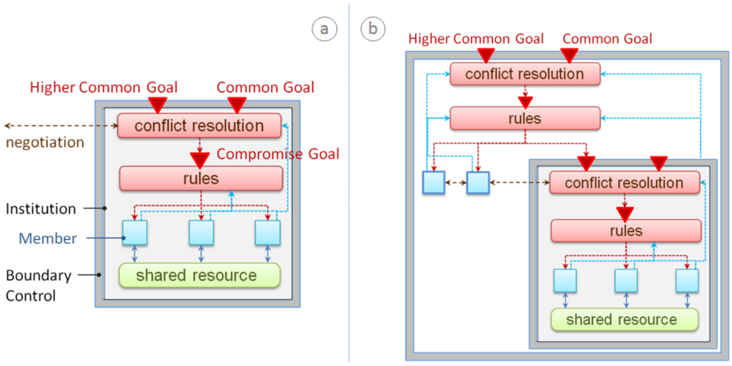


Fig. 2: a) institution *holon* with dual role: inward/selfish & outward/transcendental; b) supra-institution with several institutions / members

its supra-institutional holon(s); and similarly with its sub-institutional holons. We start by giving a characterisation of the right (to self-organise), before considering how this could be recognised as conditions on the rulesets of the inner and outer holonic institutions. Finally, we give some thought to the issue of 'minimality'.

4.1 Right = Empowerment + Entitlement

In previous work on voting [14], the notion of *enfranchisement* (the right to vote) was characterised by two components, an *empowerment* component and an *entitlement* component. The empowerment component required having the institutionalised power to establish conventional facts (i.e. a vote for or against), that no one could object to 'appropriate' exercise of the power, and that removing the power would result in sanction. The entitlement aspect required unhindered access to the voting 'machinery', an obligation that the vote would be counted correctly, and an obligation for the result to be declared correctly.

Analogously, the right to self-organise – or the recognition by the supra-level institution of the right of any of its sub-institutions to self-organise – can be characterised by an *empowerment* component and an *entitlement* component. For the empowerment component we have that:

– an institution should be empowered (have the power) to self-organise, i.e. its own institutionalised power should give control over, and responsibility for, representation, participation, and rule-selection etc. to its member entities;
– there should be no entity in the supra-level institution that is empowered to object to 'appropriate' exercise of these institutionalised powers; and
– inappropriate exercise of such a power in the supra-level institution should be subject to sanction.

For the entitlement component we have that:

– the sub-institution should be represented in the deliberative assemblies of the supra-level institution;
– the supra-level institution should provide an appeals procedure for conflicts that cannot be resolved by the dispute resolution processes specified by the sub-institution; and
– the sub-institution should be entitled to access and operate the 'machinery' of self-organisation.

We next develop a logical axiomatisation of each aspect of the two components.

4.2 The Empowerment Component

For an institutional holon to be empowered to self-organise, it simply requires operational-, collective- and constitutional-choice rules that implement the first six of Ostrom's principles. That is, there should be some protocol and associated institutionalised power(s) that determine: boundaries over representation and participation; selection and modification of collective choice arrangements; appointment and performance of monitors and monitoring; and the system of sanctions and appeals. These are, of course, precisely the rules specified in [19].

Preventing objection to 'appropriate' exercise of these powers requires the following. First, we define a specification space $\mathcal{L}$ as described in Section 2.3. Then, let $\mathcal{R}$ be a set of rules. For each rule $R \in \mathcal{R}$ identify the set of *changeable components*, or degrees of freedom (DoF), and for each DoF, identify the set of values it can take. (For example, for the collective choice rule deciding which operational choice rule to decide a resource allocation, the DoF is the resource allocation method; and its values are ration, queue, priority, etc.) Then the specification space $\mathcal{L}$ is every permutation of values that can be assigned to each DoF in each rule, and a *specification instance* is one such assignment of values.

Given a set of entities (holons) $\mathcal{H}$, the state of an institution at time ($I_t \in \mathcal{H}$) is defined by:

$$I_t = \langle \mathcal{M}, L, \epsilon \rangle_t$$

where:

– $\mathcal{M}$ is the set of member holons, such that $\mathcal{M} \subseteq \mathcal{H}$
– L is a specification instance of $\mathcal{L}$
– ϵ is the local environment consisting of brute facts and institutional facts.

However, some of the instances might be prohibited by the supra-institution $I' \in \mathcal{H}$. Therefore, to represent the fact that no supra-institution can object to an 'appropriate' exercise of the power to self-organise, in the rules of I' something of the following form is required:

$$\begin{aligned}
&\mathbf{pow}(H, \mathit{object}(H, L, I, I')) = \mathit{true} \quad \mathsf{holdsAt} \quad T \quad \leftarrow \\
&\quad \mathit{role}(H, I') = \mathit{monitor} \quad \mathsf{holdsAt} \quad T \quad \wedge \\
&\quad \mathit{specification_instance}(I) = L \quad \mathsf{holdsAt} \quad T \quad \wedge \\
&\quad \mathit{prohibited}(L) = \mathit{true} \quad \mathsf{holdsAt} \quad T
\end{aligned}$$

$$
\begin{aligned}
& \mathit{object}(H, L, I, I')) \quad \textsf{initiates} \quad \mathit{objected}(L, I) = \mathit{true} \quad \textsf{at} \quad T \quad \leftarrow \\
& \qquad \mathbf{pow}(H, \mathit{object}(H, L, I, I')) = \mathit{true} \quad \textsf{holdsAt} \quad T
\end{aligned}
$$

This axiom states that an holon H in supra-institution I' has the power to object to a specification instance used by component institution I only if it occupies the necessary role in I' (*monitor*, say), and the specification instance L selected by I is not prohibited in I'. This means that the right of I to self-organise into whatever specification instance it chooses is not challenged by any external authority, like I', although specific applications of that right can be challenged.

Finally, let us suppose, firstly, that we take the approach of Robert's Rules of Order to self-organisation, i.e. that anything is allowed unless someone objects; and secondly, that there is a holon in I' that, by occupying the role of *franchiser* is empowered to veto specification instances selected by I. However, while this holon may be empowered to veto specification instances, it may not be *permitted* to exercise that power, unless a holon that is empowered to object (by the above axiom) has done so. Using the veto without permission initiates a sanction.

$$
\begin{aligned}
& \mathbf{pow}(H, \mathit{veto}(H, L, I, I')) = \mathit{true} \quad \textsf{holdsAt} \quad T \quad \leftarrow \\
& \qquad \mathit{role}(H, I') = \mathit{franchiser} \quad \textsf{holdsAt} \quad T \quad \wedge \\
& \mathbf{per}(H, \mathit{veto}(H, L, I, I')) = \mathit{true} \quad \textsf{holdsAt} \quad T \quad \leftarrow \\
& \qquad \mathit{role}(H, I') = \mathit{franchiser} \quad \textsf{holdsAt} \quad T \quad \wedge \\
& \qquad \mathit{objected}(L, I) = \mathit{true} \quad \textsf{holdsAt} \quad T \quad \wedge \\
& \mathit{veto}(H, L, I, I') \quad \textsf{initiates} \quad \mathit{sanction}(H, I') = 404 \quad \textsf{at} \quad T \quad \leftarrow \\
& \qquad \mathit{role}(H, I') = \mathit{franchiser} \quad \textsf{holdsAt} \quad T \quad \wedge \\
& \qquad \mathbf{per}(H, \mathit{veto}(H, L, I, I')) = \mathit{false} \quad \textsf{holdsAt} \quad T
\end{aligned}
$$

Note here that '404' is an 'error code' used in I' to denote particular misuses of institutionalised power, in this case vetoing a specification instance without permission. The consequence of this sanction may be a penalty: for example, it may be fined, removed from the role of *franchiser*, or both.

4.3 The Entitlement Component

It may be supposed from the previous section that the empowerment component is a constraint on an institution's freedom to manoeuvre, because some arrangements of the rules (i.e. certain specification instances) are prohibited, and can be vetoed. In return for this loss, the right confers some entitlements, which are satisfied by certain obligations in the supra-level institution.

The first of these is that there is an entitlement to representation and participation in the deliberative assemblies of I'. This can be formalised using the same form of constitutional choice rules in I' that were present in I, based on axiomatising Ostrom's principles P1–P6. The extent to which this entitlement is satisfied can be evaluated using a framework for procedural justice [16].

Similarly, a process should be defined in I' which deals with the resolution of conflicts in I which cannot be handled by I itself (for example, the Court of Arbitration for Sport resolves legal disputes which the governing authorities of the sports themselves cannot resolve themselves, or considers appeals against the decisions of this authorities). Complex dispute resolution procedures can be defined in the Event Calculus [15].

Finally, the entitlement to access the 'machinery' of self-organisation is an abstract concept that we will not develop further here. Suffice to say it (probably) involves a number of basic 'freedoms', such as the freedom of assembly and freedom of speech; and material rather than illusional choice (as discussed above, and also avoiding any inevitability about self-organisation).

4.4 Minimal Recognition

Finally, we consider the implications of the *minimal* recognition of the right to self-organise. However, several case studies where the principle is not met have been studied in some detail [12, ch. 5], and although 'minimal' could be taken to imply such kind of quantifiable threshhold or an identifiable set of baseline conditions, it is difficult to identify a precise specification.

Therefore, if we were to try to define this for ourselves, one way might be to use the specification space and the distance function defined between 'allowable' specification instances. This suggests three possible minimality conditions.

Firstly, we could compute the outer institution tolerance as the ratio of allowable specification instances to the total number of specification instances. A higher number indicates that the outer institution is more tolerant of inner institution self-organisation than a lower number.

A second metric could be the inner institution total freedom to manoeuvre, as measured by the total distance that the specification instances are allowed to change. If there were n specification instances $l_1, \ldots, l_n$, this is computed by:

$$\Sigma_{i=1}^{n} \Sigma_{j=1}^{n} d(l_i, l_j)$$

Thirdly, a metric could be defined accruing to the average number of specification instances that can be accessed from any one specification instance. Suppose that τ is the upper limit on the distance that can be moved. Then the number of specification instances that can be reached from any given specification instance l_i is:

$$\mathbf{card}(\{l_j | d(l_i, l_j) < \tau\})$$

If we computed the average number of specification instances that could be reached from an arbitrary specification instance, then this would be an indicator of the extent of self-organisation available from any instance.

However, with all of these metrics, what constitutes a *minimal* tolerance is not clear, and would most likely be application specific. Even then, while a quantitative evaluation might be indicative, it seems that minimal is most likely to be a relative, qualitative, assessment. We return to this issue in Section 6.

5 Further and Related Work

5.1 Further Work

One direction of further research is, of course, to complete the formal specification in the Event Calculus, define some metrics for minimal recognition, and then build a simulation environment comparable to those used in previous experiments [19, 13]. The experimental hypothesis is that those systems with layered institutions would sustain themselves (and their resources) for longer if the rule-sets of inner and outer institutions contain provisions of the kind specified here.

A second direction of work is, in the context, to deepen the analysis of the sources of, the resolution of, conflicts between inner and outer institutions and between two or more inner institutions. A preliminary investigation of this issue can be found in [5], but it seems that there is a need to distinguish different types of conflict (within a holon, between peer holons, and between inner/outer holons) as different conflict resolution mechanisms may well be required.

However, while this research would confirm the principle as a necessary condition of sustainable common-pool resource management, it would be a somewhat blunt instrument. What would be really interesting to know is where the *balance* is in the trade-off between the rights of the inner institution to self-organise on the one hand, and the power of the outer institution to constrain that self-organisation. This is an investigation that is also being actively pursued, as we believe this will shed some light on this concept of *minimal* recognition, which remains far from clear at the moment. Note that this will probably also be an adaptive, context-sensitive balance.

5.2 Related Work

The formal characterisation of rights has, of course, been the subject of study in legal, social and organisational theory, ethics and moral philosophy for some considerable time. However, we are not aware of any similar attempt to characterise the *right to self-organise* in computational logic, nor to situate it in the context of structured interacting entities like holonic institutions.

There are though related studies that can be of significant relevance. For example, one such is the concept of *duty*. Right and duty have been characterised as 'correlatives' [21], in the sense that when one agent has a right against another, then that other owes a duty to the first. An enrichment of the analysis of "right = empowerment + entitlement" could entail an associated duty component which could become useful in characterising the minimality conditions.

6 Summary and Conclusions

There is an ongoing research programme into the formalisation of Ostrom's institutional design principles in computational logic, and specifically the Event

Calculus. The aim of this research is to design self-organising electronic institutions to address the problem of sustainable resource allocation in open multi-agent systems. This has a potentially wide range of applications, for example in ad hoc, vehicular and sensor networks; in cloud and grid computing; in virtual organisations; and in infrastructure management using socio-technical systems.

This paper has particularly focused on the formalisation of the seventh principle, the minimal recognition of rights. Although this is clearly still work in progress, the primary contributions of this paper are:

- to characterise the *right to self-organise* in terms of empowerment and entitlement components, and to begin a formal specification in terms of the Event Calculus;
- to identify the *recognition* of that right (to self-organise) as conditions on the rulesets of 'inner' and 'outer' holonic institutions; and
- to consider the *minimal* recognition of that right, as conditions or metrics defined on a specification space.

However, perhaps the concept of "minimality" presents the greatest challenge, as it may be resistant to mere quantification with respect to the specification space. Our intuition though, is that the minimality is actually a function of *qualitative values* held by the members themselves: that it is not the total number of available specification instances, or the freedom to manoeuvre as specified by the total movable distance; or indeed any other objective metric. Instead, it is the value that the members themselves put on having certain specification instances available to them, and the freedom to self-organise between these specific instances, that is the critical element of minimality. But this is, of course, rather harder to identify, let alone measure; although this a promising line of inquiry is offered by the study of *interactional justice*, whereby subjective individual opinions of fairness (or value) are aggregated into objective collective opinions.

Furthermore, it may not even be certain specification instances that are directly of value, but that these are the specification instances that indirectly create other social, moral or fungible values, with which the institution's members, either individually or collectively, are really concerned. In such circumstances, the link between the value and the minimal set of specification instances that support that value may not even be recognised by the agents themselves. This may have severe implications in trying to build socio-technical systems for self-organising digital communities, say, and indicates that aspects of value-sensitive design [4] and other initiatives that emphasise values derived from a study of social practices are critical in the future development of open multi-agent systems.

Acknowledgements

The authors would particularly like to thank Pompeu Casanovas for conversations which have significantly helped to clarify numerous issues in rights and powers, but any persistent misunderstandings are our own. We are also very grateful for the many helpful comments of the anonymous reviewers.

The first author has been partially supported by the UK EPSRC Grand Challenge project No. EP/I031650/1 *The Autonomic Power System*.

References

1. Artikis, A.: Dynamic specification of open agent systems. Journal of Logic and Computation 22(6), 1301–1334 (2012)
2. Cox, M., Arnold, G., Villamayor Tomás, S.: A review of design principles for community-based natural resource management. Ecology and Society 15(4), 38 (2010)
3. Diaconescu, A., Pitt, J.: Holonic institutions for multi-scale polycentric self-governance. In: Proceedings COIN, LNCS, vol. 9372, pp. 19–35. Springer (2015)
4. Friedman, B., Kahn, P., Borning, A.: Value sensitive design and information systems. In: Himma, K., Tavani, H. (eds.) The Handbook of Information and Computer Ethics, pp. 69–101. Wiley (2008)
5. Jiang, J., Pitt, J., Diaconescu, A.: Rule conflicts in holonic institutions. In: IEEE SASO Workshops (FoCAS). pp. 49–54 (2015)
6. Jones, A., Sergot, M.: A formal characterisation of institutionalised power. Journal of the IGPL 4(3), 427–443 (1996)
7. Koestler, A.: The Ghost in the Machine. Hutchinson Publisher (1967)
8. Kowalski, R., Sergot, M.: A logic-based calculus of events. New Generation Computing 4, 67–95 (1986)
9. Lansing, J., Kremer, J.: Emergent properties of Balinese water temple network: coadaptation on a rugged fitness landscape. American Anthropologist 95, 97–114 (1993)
10. Ostrom, E.: Beyond markets and states: Polycentric governance of complex economic systems. In: Grandin, K. (ed.) Les Prix Nobel. The Nobel Prizes 2009, pp. 408–444. Stockholm: Nobel Foundation (2010)
11. Ostrom, E.: Thinking about climate change as a commons. In: 15th Annual Philip Gamble Memorial Lecture, pp. 1–34. UMass Amherst (2011)
12. Ostrom, E.: Governing the commons: The evolution of institutions for collective action. Cambridge, UK: Cambridge University Press (1990)
13. Pitt, J., Busquets, D., Macbeth, S.: Distributive justice for self-organised common-pool resource management. ACM Trans. Auton. Adapt. Syst. 9(3), 14:1–14:39 (2014)
14. Pitt, J., Kamara, L., Sergot, M., Artikis, A.: Voting in multi-agent systems. Computer Journal 49(2), 156–170 (2006)
15. Pitt, J., Ramirez-Cano, D., Kamara, L., Neville, B.: Alternative dispute resolution in virtual organizations. In: Artikis, A., O'Hare, G., Stathis, K., Vouros, G. (eds.) Proc. ESAW'07. LNCS, vol. 4995, pp. 72–89. Springer (2007)
16. Pitt, J., Busquets, D., Riveret, R.: Procedural justice and 'fitness for purpose' of self-organising electronic institutions. In: PRIMA 2013: Principles and Practice of Multi-Agent Systems, LNCS, vol. 8291, pp. 260–275. Springer (2013)
17. Pitt, J., Busquets, D., Riveret, R.: The pursuit of computational justice in open systems. AI & Society 30(3), 359–378 (2015)
18. Pitt, J., Diaconescu, A.: Structure and governance of communities for the digital society. In: Workshop on Self-Improving System Integration at IEEE International Conference on Autonomic Computing (ICAC). pp. 279–284 (2015)
19. Pitt, J., Schaumeier, J., Artikis, A.: Axiomatisation of socio-economic principles for self-organising institutions: Concepts, experiments and challenges. ACM Trans. Auton. Adapt. Syst. 7(4), 39:1–39:39 (Dec 2012)
20. Rescher, N.: Distributive Justice. Bobbs-Merrill (1966)
21. Sergot, M.: A computational theory of normative positions. ACM Transactions on Computational Logic 2(4), 581–622 (2001)

COIN Invited Talk

Security, Privacy, and Accountability: Foundation in Sociotechnical Systems

Munindar P. Singh

North Carolina State University, Raleigh, NC, USA
singh@ncsu.edu

(joint work with Nirav Ajmeri, Amit Chopra, Ozgur Kafali)

Abstract.
Despite sustained research effort in security, current security practice conveys a decidedly ad hoc flavor—find a bug; patch it; find the next bug; and so on. This methodology is sometimes termed "engineering", using the term in the narrow sense of developing solutions to specific problems. The past few years have seen a growing push to develop a Science of Security (SoS), viewed as a systematic body of knowledge with strong theoretical and empirical underpinnings that inform the engineering of secure information systems.

I introduce SoS, briefly describing its key elements. I then motivate some of the foundational challenges of SoS from the standpoint of systems of autonomous participants, such as cross-organizational service systems. I describe how security is an element of the governance of such systems. I introduce an approach for governance based on a new formulation of norms and accountability, showing how it addresses the challenges of security pertaining to secure collaboration and would facilitate the development of flexible and secure cross-organizational service engagements.

I present our recent approaches for software engineering for sociotechnical systems that support the above considerations and demonstrate them via examples from the healthcare domain.

Joint COIN/EMAS Invited Talk

Implementing Norms
Why is it so difficult?

Frank Dignum

Utrecht University, The Netherlands F.P.M.Dignum@uu.nl

Abstract. Many people have made implementations of norms or normative systems over the years. However, the implementations differ widely and no uniform methodology to implement normative systems seemed to have been developed. Why is it so difficult to implement norms? Can't we just have a Norms module that can be added to a system? I will discuss these issues and also point to some possible ways forward.

1 Introduction

In [3] we already described some of the issues that come up when trying to implement norms in agent systems. Because the norms influence the behaviour of the agents, they somehow have to be taken into account during the planning and execution of their actions. It is this *influence* relation between the norms and the plans and actions that is difficult to capture. Let me give a simple example. There is a norm that bikes should stop for a red traffic light. In a typical Dutch city like Amsterdam it would be hard to deduce this norm from just looking at the actions of the bikes. So, there is certainly not a one-on-one relation or rule that makes bikes stop whenever the traffic light is red. From own experience it seems that the norm functions as a heuristic that states that when you are on a bike and getting to a red light you have to check whether you can cross and have to stop when there is traffic coming from the sides streets (that you cannot avoid in any way). However, bikes will also stop when there are children already waiting at the traffic light (because you have to give a good example) or when they spot a policeman that could give you a fine when passing a red light. These examples show that the way a norm influences the actions is at least partly situational. One could argue that thus one has to take both the norm and the current situation as input for the influence relation. Theoretically this is correct, but it can be quite difficult to determine how the norm and situation should be combined. Are there just a lot of special situations that give exceptions to a general rule or are there types of situations that each combine in a different way with the norm? This is not very clear and also has not been investigated in a systematic way (although this has, of course, strong links to modeling norms as a kind of default logic).

However, the situation gets even more complex as not only the present situation influences the way norms influence behaviour, but also the social structure of which this situation is a part. E.g. a biker might stop for a red light if everyone else also stops, but ignores the light if most people do so. The biker also might disregard red lights habitually (at certain crossings or times) if he knows that in the past this never led to any danger. Finally, if the biker is a school teacher riding ahead of a class of children on the bike he will never pass a red light. So, we have to include social status, roles, practices and history in the list of aspects that should be regarded when modeling the influence of norms on behaviour. The list gets longer and longer and also involves more and more complex concepts. Thus it is clear that modeling the influence of norms on behavior can be very complex and will differ depending on which of these aspects are taken into account.

Of course, norms do not exist in isolation. So, besides the above norm there might be a norm that you have to come in time for a meeting with your boss. When you go on the bike to work and are a bit late it might be that you get a conflict between the norm to stop for a red light and being in time for the meeting. In a good logic tradition one could solve this problem by giving a preference order over the norms such that one will be fulfilled and the other violated when they are in conflict. This could be conveniently handled within the norms module. However, life is not as simple as that. Often the preferences of the norms are situation dependent. If I have a meeting with my boss in which I am going to ask a favor (or promotion) I certainly do not want to be late and possibly annoy him before even starting the meeting. Thus I might risk going through red in order to be in time for the meeting. But if I see a policeman near the traffic light I might stop. The reason being, not that I give priority to the traffic norm, but rather that if the policeman is going to fine me, it will take even more time than just waiting for the light! So, at this moment the planning actually influences the norm preference and which norm might be violated. To further complicate matters one might even consider that passing through the red light and being fined one certainly will be late, when you pass through the red light and not get a fine you certainly will be in time and if you stop for the red light you still have a chance of making it in time to the meeting. It is clear that in these more complex cases it does not suffice to just check which norms are applicable at a certain moment and use those as a kind of filter on the potential plans. There is a interdependence between the planning and the norms that does not (always) allow for a one way influence relation (as would be prefered for any deliberation architecture)!

2 Acting with Norms

Given the above arguments one would think it is better not to incorporate norms in agents at all. Things are not as grim as they seem though. Norms have many aspects and not all aspects are equally important in every application. The first step to take when implementing norms should thus be to determine what the function should be of the norms in the system that is developed. The aspects that relate to that function have to be modeled and implemented. In many cases the norms are seen as constraints that (in principle) can still be violated (either in specific circumstances, randomly in some cases, or based on another clear criteria). In this case one can have a norms module that is used to check all potential plans and orders them based on how many norms they violate and how efficient the plans are.

However, if the norms should lead to emerging behavior as perceived in reality this will not be enough. In that case more elaborate mechanisms should be designed. This will often also necessitate a more complex deliberation cycle of the agents incorporating more social concepts. This is not very easy, because it is not clear on forehand which concepts are needed exactly and there are no architectures that incorporate these concepts in a systematic way. I.e. there are all kinds of extensions of BDI architectures, but often with only one or two new concepts and usually for a very specific purpose or application area.

What is needed is a richer social model in which norms can play a role and agents have all social concepts available. Based on such a rich conceptual model designers can make choices of which parts are needed for their application and what are the consequences of choices they make (both for possible (emerging) behaviour and also for efficiency). Until such a more elaborate social model exists people will have to start from scratch every time they want to implement norms with a slightly different perspective or function in mind. Although this is useful in its own right as all these implementations might support some particular rich social model and give ideas on how to build this, the danger is that people get tired of using norms because they are difficult and try to circumvent the problems in current day systems with very primitive means. This will lead to poorly designed systems that are not well understood and might lead to unforeseen or unwanted emerging behavior. The research that I have started in [1] and [2] should be seen in this light. Although norms do not feature prominently in these papers, they are the underlying motivation to take this broader perspective and start working on a more encompassing social framework. As stated in the vision paper, we hope that other researchers get inspired by this and will join the research.

3 Biography

Dr. Frank Dignum received his PhD in the previous century and has since been working in Swaziland, Portugal and The Netherlands. He is working on social aspects of software agents with applications in serious gaming and social simulations. He is well known for his work on norms and agent communication and lately for the combination of agents and games. His latest research focuses on creating new agent architectures to build agents that operate in real-time environments and have to cooperate with humans and other agents. He has organized many workshops and conferences on the topics and given tutorials at most major conferences and summer schools on them.

References

1. F. Dignum, R. Prada, and G.J. Hofstede. From autistic to social agents. In *AAMAS 2014*, May 2014.
2. V. Dignum, C.M. Jonker, F. Dignum, and R. Prada. Situational deliberation; getting to social intelligence. In *SocialPath workshop*, 2014.
3. Loïs Vanhee, Huib Aldewereld, and Frank Dignum. Implementing norms? In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 03*, WI-IAT '11, pages 13–16, Washington, DC, USA, 2011. IEEE Computer Society.

CARE

Practical results from Samsung Digital Education Research Platform

Andrew Koster

Samsung Research Institute, Campinas, Brazil
andrew.k@samsung.com

Abstract.
At the SRBR we have developed a research platform for technology-enhanced learning focusing on using tablets in the classroom and collecting fine-grained usage data from the education material. Throughout field trials, we have curated a database of such usage signals, that we are using to better understand how students learn, are engaged and how to measure student performance in a more organic manner. We can use this data to calibrate an agent-based simulation of a classroom in an attempt to understand how changes in the classroom or content would affect the real-world students. In the introduction to the 2016 CARE@AAMAS workshop, I will present this platform and some early results.

Multi-Agent Simulation for Group Recommendation in E-Learning Environments

Mhd Irvan[1] and Takao Terano[2]

[1,2]Department of Computational Intelligence and Systems Science
Tokyo Institute of Technology, Japan
[1]irvan@trn.dis.titech.ac.jp
[2]terano@dis.titech.ac.jp

Abstract. Recommender Systems have become an essential part for web services. Many people now rely on recommended contents to decide what movies to watch, which music to listen, or what products to buy. Typically this kind of recommendation is tailored to individual users. However, in e-learning environments, many activities may not be intended for personal usage, but rather for group consumption. Group recommendation is a tough problem to solve because different members of a group may have different preference. This short paper investigates ways to recommend courses depending on which members happen to be online during the same time frame. We propose a multi-agent model to generate recommendations. Members' activities are simulated by multiple intelligent agents, evolved by genetic algorithms, autonomously interacting within a virtual world simulating an e-learning environment populated with hundreds of randomly generated courses that mimic actual online courses. Recommending ideal group courses to multiple members at the same time may lead to a fun and highly interactive session and we believe the result from this early research can give positive values to the world of online learning.

Keywords: Multi-Agent Simulation, Recommender System, e-Learning

1. Introduction

These days, many web services rely heavily on recommender systems [1]. Social networks recommend you people to follow, streaming websites recommend you videos to watch, or e-commerce websites recommend you products to buy. People deal with these services daily. Once the number of items being offered to users is greatly increased, users could easily become overwhelmed by a situation commonly referred as information overload [2]. People gets confused as they just don't know what to do after the visit the website. Recommender systems try to address this problem by recommending things to do, such as what items to buy or what music to listen, to users based on their preferences.

Recommender systems are typically tailored toward individual user [3]. However, there are situations where recommendations are necessary to be made for several users as a group. E-learning can be a situation where group recommendation can play as a vital role [4]. In an e-learning environment, to create a virtual classroom atmosphere, courses can be designed to be taken by a group of students. These virtual courses may not be intended for personal usage, but rather, for group consumption. Tasks within classes may be offered as group works and questions may lead to an interactive discussion amongst students.

However, group recommendation is a tough problem to address [5]. A group can consist of students with various learning preferences. Students being online may differ from the group of the previous session. This short paper introduce a method to recommend courses depending on the types of students who happen to be online during the same time frame. We propose a model, implementing multi-agent systems, to calculate predictions and to generate recommendations.

We generate an e-learning environment, filled with hundreds of randomly generated courses that resemble actual online courses. The environment is populated by multiple intelligent agents that simulate students' activities. These agents are evolved by genetic algorithms [6] and autonomously interact with each other and giving feedback in forms of reinforcement learning [7].

2. Proposed Method

Our proposed method uses a Learning Classifier System (LCS) that combines the features of genetic algorithms and the feedback measures of reinforcement learning [8]. LCS has been found to provide good recommendation [9] and has also been proven to work effectively in a multi-agent environment [10]. This research aims to improve the multi-agent LCS model by adding a capability to also support recommendations for group of multiple users, in addition to recommendation for individual users.

LCS is essentially a collection of input-action classifiers illustrating known knowledge it has learned from previous situations. It collects input from the e-learning environment, such as the students currently online, their profiles, and favorite topic. Based on these inputs, it proposes actions (Fig. 1) in form of which course recommendation to take.

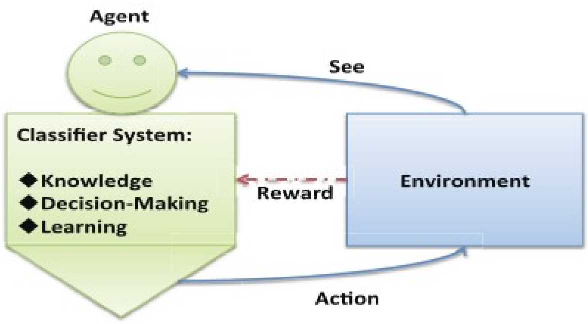


Fig 1. Illustration of a Learning Classifier System

For each group of students, the system extracts their profile (gender, age, and location) and their favorite topic (science, social, or language).

Genetic algorithms are used to search and test the possible permutations for LCS classifiers by applying permutation and crossover upon them. Genetic algorithm is triggered (*GA_TRIGGER*) when the average time period for classifiers within the set *[A]* since the last occurrence of genetic algorithm is greater than the frequency parameter *θ:*

$$GA\_TRIGGER([A]) \; if: \frac{\sum_{C \in [A]} t - C.g}{[A].n} > \theta$$

If the current environment situation matches the classifiers within the population set *[A]*, the system allocates those matched classifiers into a match set *[M]*. The algorithm continues to calculate the prediction array *P(A)* of the set *[M]* using the value of prediction (*C.p*) and fitness (*C.f*) of the classifiers within [M]:

$$P(A) = \frac{\sum_{C.a=a \wedge C \in [M]} C.p \times C.f}{\sum_{C.a=a \wedge C \in [M]} C.f}$$

Recommendation proposed by the classifiers with the highest *P(A)* is selected. Reinforcement learning is then applied after the LCS makes a recommendation to give feedback hinting the quality of the proposed recommendation. The feedback is in the form of reward R to be allocated to the classifiers in the set *[A]*, derived from the maximum prediction within the prediction array of the current iteration *P(A)*, discounted by *γ*:

$$Q = \max_A P(A)$$
$$R = r + \gamma Q$$

3. Early Experiment

To test our method, we generated 200 courses (categorized into science, social, or language studies) to be recommended to 1000 agents randomly generated through Gaussian distribution with standard deviation from 0.01 to 0.05 to fill their profiles in form of gender (male or female), age (kids, teens, or adults), location (at home or outside), and favorite subject (science, social, or language studies). Each course is designed to be taken by a group of three agents (experiment 1) and four agents (experiment 2). At each time frame during the simulation three or four agents are randomly selected to be "online". The system then try to recommend a course that suits the combination of these agents profiles. We calculate the precision value as the occurrence of true positive (*tp*) divided by the sum of the true positive (*tp*) and false positive (*fp*):

$$\text{Precision} = \frac{tp}{tp + fp}$$

We ran the simulation up until 10,000 simulation steps to see how the precision values progress over time. Fig. 2 illustrates the Precision percentage of experiment 1 and experiment 2 after every 1000th simulation steps.

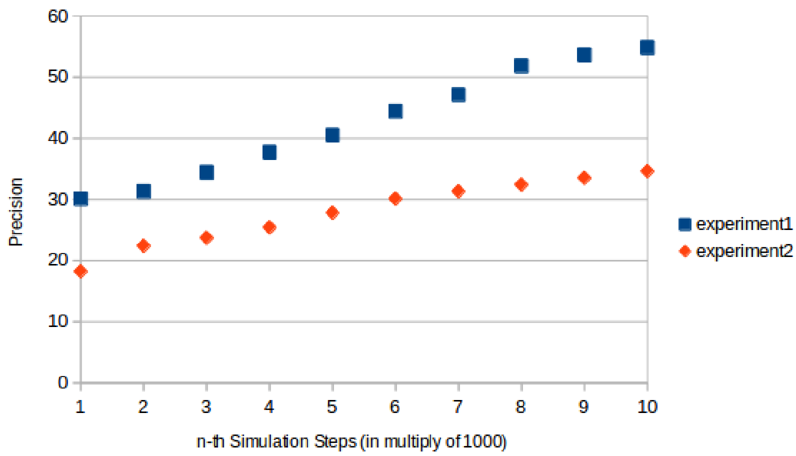


Fig. 2 Precision value of Experiment 1 and Experiment 2

Fig. 2 shows that recommending courses to a group of four students need much longer time to produce the same range of precision than recommending to a group of three agents. This is expected as more agents means more permutation to test and much deeper search space for the genetic algorithms to explore.

At some points during the experiment 1, we saw a sudden big increase in precision. This happened because after a long run, more and more agents with similar profile showed up and the LCS has built a big enough collection of classifiers that were already proven to be accurate. For similar reason, we saw precision gap was widening between both experiments during latter-half of the simulation because by the time experiment 1 had built a collection of accurate classifiers, experiment 2 still had more permutation to test and search.

4. Future Works

Recommending ideal group courses to student groups may lead to a fun and highly interactive session and we believe the result from this early research can give positive values to the world of online learning. Future research include algorithm optimization, and testing with real data. A longer run of simulation may also provide more insights to what is happening during the learning process.

References

1. Bobadilla, J., et al.: Recommender Systems Survey. In: Knowledge-Based Systems, Vol. 46, pp. 109-132 (2013)
2. Bollen, D., et al.: Understanding choice overload in recommender systems. In: Proceedings of the fourth ACM conference on Recommender systems, pp. 63-70 (2010)
3. Adomavicius, G.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. In : IEEE Transactions on Knowledge and Data Engineering, Vol. 17, Issue 6, pp. 734-749 (2005)
4. Zhang, D., et al.: Can e-learning replace classroom learning?. In: Communications of the ACM, Vol. 47, Issue 5, pp. 75-79 (2004)
5. Jameson, A.: More than the sum of its members: challenges for group recommender systems. In: Proceedings of the working conference on Advanced visual interfaces, pp. 48-54 (2004)
6. Goldberg, D.E., and Holland, J.H.: Genetic Algorithms and Machine Learning. In: Machine Learning, Vol. 3, Issue 2-3, pp. 95-99 (1988)
7. Kaelbling, L.P., et al.: Reinforcement Learning: A Survey". In: Journal of Artificial Intelligence Research, Vol. 4, pp.237-285 (1996)
8. Holland, J.H., et al.: What is a Learning Classifier System?. In: Learning Classifier Systems: From Foundation to Applications, Lecture Notes in Computer Science, Vol. 1813, pp. 3-32. Springer Berlin Heidelberg (2000)
9. Irvan, M., Chang, N., and Terano, T.: Designing a Situation-aware Movie Recommender System for Smart Devices. In: Proceeding of The Sixth International Conference on Information, Process, and Knowledge Management, pp. 24-27 (2014)
10. Irvan, M., Yamada, T., and Terano, T.: Influence of Organizational Learning for Multi-Agent Simulation based on an Adaptive Classifier System. In: IEEJ Transactions on Electronics, Information and Systems, Vol. 133, No. 9, pp. 1752-1761 (2013)

Gamification in Teaching Multi-Agent Systems

J. Baldeón, M. Lopez-Sanchez, I. Rodríguez, and A. Puig

University of Barcelona, Faculty of Mathematics,
Applied Mathematics and Analysis Department,
Gran Via, 585, 08007 Barcelona, Spain
{johan.baldeon,maite_lopez,inmarodriguez,annapuig}@ub.edu
http://www.maia.ub.es

Abstract. Traditional theoretical classroom dynamics suffer from student feedback and interaction. Unfortunately, attendance rates also represent a common problem. In order to mitigate these issues, we propose the inclusion of novel teaching resources. On the one hand, Multi-Agent Systems (MAS) core concepts of distribution, autonomy and interaction can be mapped into collaborative classes, where students can experience theoretical concepts in hands-on activities. On the other hand, class gamification can help to enhance students motivation and engagement. Nevertheless, applying gamification requires the usage of a suitable framework. This paper proposes an extension of the Gamification Model Canvas. This extension includes MAS principles as well as those of Lego Serious Play and Audience Response Systems. Additionally, we illustrate its applicability by means of a case study that designs and gamifies a multi-agent systems class, which has been positively evaluated by students.

Keywords: Gamification, Teaching MAS, LSP, ARS

1 Introduction

Nowadays, learning of multi-agent systems (MAS) concepts is often an individual experience with a unidirectional communication between teachers and students, and where both methodology and environment are those of a traditional classroom. In this context, theoretical classes are boring, include many abstract concepts [3], and, as a result, students simply do not engage, and there are low attendance rates in theoretical classes.

But, what about if teachers design MAS classes as if they were multi-agent systems? In this case, teachers and students would work collaboratively, acting as agents that (distributively) interact, and promoting emergent behavior [1]. In this scenario, both existing knowledge and skills could be put into practice, keeping in mind that this experiential approach would ensure that teaching ideas, principles and concepts have a lasting value [5].

In addition, if session activities would include emulating MAS applications in this collaborative setting, it would consolidate learned concepts and would also enhance students experience [20]. Moreover, these activities would exploit transversal skills such as teamwork, social applications, collective thinking, argumentation and social intelligence [4].

Furthermore, if we gamify these activities, students would become more engaged and motivated. Gamification is the use of game design elements and game mechanics in non-game contexts [6]. In fact, the application of gamification techniques in different contexts has increased in the last years[1]. Specifically, it is a current trend in education since it is used to increase students' engagement and to promote certain learning behaviors on them [7]. Nevertheless, gamification is not a straightforward process and, thus, it should be driven by a formal gamification design framework to prevent failures in the implementation. Gamification Model Canvas[2] (GMC) is a framework which has proven to be an agile, flexible, and a systematic tool that helps to design and to evaluate a gamified system [17]. However, to the best of our knowledge, GMC has been applied only in business contexts rather than in academic ones.

This paper proposes an extension of GMC framework and applies it to gamify a university class. The extension is done at two different levels. On the one hand, at an inner level, we propose the inclusion of a new relevant element within the framework. On the other hand, at an outer level, we consider the addition of external supporting systems, Lego Serious Play (LSP) [8] and Audience Response System (ARS) [13], that help to put into practice the design framework. LSP to generate new ideas (brainstorm) about how to improve theoretical classes through building models with metaphorical meaning using Lego bricks, and Game-Based ARS to involve all students in a classroom, enhance learning feedback and interactivity within the class. Additionally, we illustrate its applicability through a case study of an undergraduate MAS subject, where students revisit MAS theoretical concepts and design a specific MAS. Moreover, when working collaboratively, MAS principles were somehow mapped into an experiential and immersive design, where students had autonomy to design and play agents themselves and their interaction was key to orchestrate the overall designed system.

2 Related Work

The popularity of gamification in the education field has increased in the last years. Many experiences have introduced gamification into elementary, high school, and even higher education settings, obtaining uneven results in educational attainment and motivation [19]. Usually, unfruitful results can be due to the fact that gamification is being applied by merely adding game rewards to an existing set of learning activities, instead of using gamification design frameworks to make activities more attractive and engaging.

Game designers state that the core of a game for learning should be aligned with those competencies students are aimed to acquire [12]. This should be taken

[1] http://www.gartner.com/smarterwithgartner/five-key-trends-in-gartners-2015-digital-marketing-hype-cycle/
[2] http://www.gameonlab.com/canvas

into account in order to guarantee the success of a gamified class. Actually, the effective gamification of courses is still a challenge. This is specially the case in the higher education context [11], where students must assimilate high-level concepts in a short period of time.

To the best of our knowledge, few gamified experiences for teaching and learning in computing areas have been proposed in high education. Some of them are encouraging experiences related to teaching MAS concepts. For instance, [1] included role playing games in teaching the content of an MAS, where players understood the basis, were motivated and had fun.

In other experiences, we found the effectiveness of brainstorming driven by Lego Serious Play, [14] gamified the process of formulating and refining use cases in software development. In general, students who used LSP attained a higher level of skills in the areas of comprehension, application, and analysis. Moreover, the quality of software projects submitted at the end of the course improved, and student engagement increased significantly.

Related to the use of gamification models, other approaches applied design frameworks to introduce Computer Science [15] and C-programming language [10] in introductory courses. Briefly, [15] followed the MDA framework [9] to encourage students to increase participation in social and learning activities by using PeerSpace, an online social network for collaborative learning. As for [10], students achieved higher understanding and engagement in programming in the context of a C-programming language gamified course. Its design was based on Nicholson's theoretical framework [18]. In this engaging experience, most students continued working even after earning the maximum amount of grade points. They also continued mastering unexplored C-programming topics.

Pointing out the encouraging results attained in these experiences, we propose to use both LSP and GMC framework to gamify classes of multi-agent systems, a subject that implies the introduction of theoretical abstract concepts, which may become tiresome and dull to follow.

3 Gamification Model Canvas - GMC

Gamifying effectively an activity is not as straightforward as to use points, medals, and badges to engage users. Therefore, gamification frameworks become a useful guidance for designers [17]. Specifically, GMC constitutes an agile, flexible and systematic instrument that helps to find play-based solutions to develop behaviors in non-game environments. It is based on the MDA game design Framework [9]– which lets collect interests of users about aesthetics and dynamics– and on Business Model Canvas, a tool to design a business model [3].

GMC considers nine ordered elements: [4]

- *Revenues* element describes either the economic or social return of the gamified solution.

[3] http://www.businessmodelgeneration.com/canvas
[4] Note that references to these elements along the paper are highlighted in italics.

- *Players* element focuses on who are the users, how are they, and what are their expectations. They can be considered *newbies*, *masters* or *designers*[5], and each profile involves different Bartle and Marczewski's gamification user types [2][16]. The *newbie* profile is an early user of the system and could be *killer*, *self-seeker*, *consumer* or *exploiter*. The *master* is a regular user who needs more meaningful incentives to become an expert or designer user and could be *explorer*, *achiever* or *socializer*. The *designer* is a very committed user who helps the system and needs self-realization opportunities to develop himself and could be *philanthropist*, *free spirit* or *disruptor*.
- *Behaviours* element describes the behaviours or actions we want to develop in the players in order to get revenues from the project. For example, go to a website, read content, answer a survey, buy something, etc.
- *Aesthetics* element describes the desirable emotional responses elicited in players during the playing experience. For example, challenge (game as an obstacle course), fellowship (game as a social framework), etc.
- *Dynamics* element defines how to create the aesthetics. For example, fellowship aesthetics can be facilitated by means of dynamics that allow sharing information within a group or by tasks (or winning conditions) that cannot be achieved by single players. Example of dynamics are altruism, scarcity, identity, status, etc.
- *Components* element defines those elements involved in the creation of dynamics and feedback that will create game mechanics. For example, points, badges, levels, missions, avatars, etc.
- *Mechanics* are the various actions, behaviours and control mechanisms afforded to the player within a game context. It describes the rules of the game with components for creating game dynamics.
- *Platforms* element defines the environment physical or virtual, on which to implement game mechanics.
- *Costs* element describes the investment needed to develop the game.

In the following section we present the proposed extensions to GMC framework.

4 Proposal

4.1 GMC new element - End Game

Each iteration in GMC helps to further define and validate the hypothesis associated to those elements of the model defined in previous steps. Specifically, the *mechanics* element, which is core in the gamification process, is the section that requires more iterations and refinements. This is the case even if all the components of the rest of the elements are well-designed, since the complexity of designing game mechanics is intrinsically high. In fact, some initiatives plan to provide tools to aid gamification designers with the transformation of selected components in the GMC workflow into specific mechanics that turn out to be suitable for the gamification process at hand.

[5] Gamification World Congress 2015, https://gamification.world/congress/gwc-2015

Moreover, Mora et al. [17] point out that specific hypothesis refereed to the logics of the game, such as game on-boarding or end game rules, do not appear explicitly in any element of the GMC framework. Certainly, although one may argue that some of these rules could be defined during the subsequent refinement steps of the *mechanics* element, its design becomes unavoidably complex when combining all actions, behaviours and components to define all the rules of the gamified activity. Therefore, to isolate the specific rules of the end of the game,would help in the modularity of the overall design process.

Thus, we propose to extend the GMC framework with a new element named *end game* (see Fig. 1). This element integrates different actions and behaviours to create specific game dynamics that will reach the end of the game. These dynamics will lead to a final state of victory or goal accomplishment which, most often, will stretch players to the limits of their abilities in their pursuing of this final (desirable) state. In the case of a learning serious activity, our claim is that students will make further progress in their skills and knowledge acquisition. To become top user –with highest score in a competitive context– or to become most popular player –with highest peer recognition in a collaborative context– constitute some examples of end-game rules.

4.2 GMC Extensions

As we consider *revenues*, *players*, *aesthetics* and *mechanics* rough elements in the GMC framework, we propose to extend it with two external systems that may help designers in these elements. Fig. 1 depicts this extended GMC framework.

On the one hand, we suggest the use of tools that involve users to: corroborate *revenues* (we also refer to them as *benefits*); refine user profiles (*players*); and let them to provide suitable and creative solutions (*aesthetics* and *mechanics*). The Lego Serious Play is one representative example of these tools.

On the other hand, *mechanics* can be deployed by using Audience Response Systems. For example, ARS quizzes create a competitive atmosphere, including timed responses, real-time feedback and points. Additionally, ARS Surveys, which lack of any competitive aspect, can be used for getting feedback from learners, being more suitable for group discussions.

5 Case Study

In order to explore the application of our extended GMC framework, we performed a case study of teaching Multi-Agent Systems, an optional subject in the fourth-year of a Computer Engineering degree. It has 6 ECTS (European Credit Transfer System), students attend two 2-hour-long in-class activities per week.

As for class attendance, a total number of 26 students were enrolled in the course. Nevertheless, just a few of them (less than 20%) attended all theoretical classes. In fact, none of the theoretical classes did surpass fifty percent attendance. Additionally, when attending, the majority of students did not actively participate in class discussions. Therefore, our gamification goals (i.e., *benefits*)

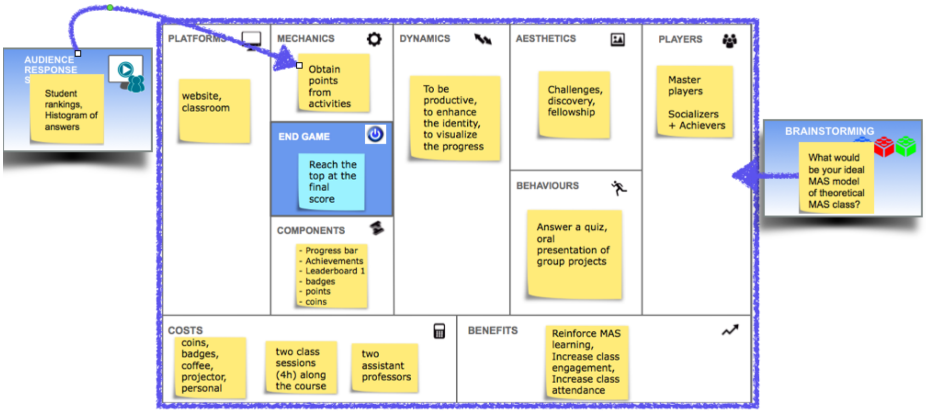


Fig. 1. Design of the gamification experience applying the Extended GMC.

are to increase both class engagement and attendance, as well as to reinforce MAS learning, considering that concepts introduced in MAS theoretical classes are often abstract and difficult to grasp.

5.1 Design

The design of the gamification was conducted by a team conformed by a game designer coach, the teacher of the MAS subject, and two collaborator teachers who also actuated as assistants during sessions.

In the starting point of the design process we involved a group of students in a brainstorming session to corroborate our diagnosis of the problem and our definition of the solution (i.e. gamification goals). We also aimed at obtaining students' feedback as well as collecting their opinions and ideas. To do so, we scheduled a Lego Serious Play activity (see blue box in the right most part of Fig. 1) with eight voluntary students distributed in two groups of four students each.

Upon coach's requests, students built a model about a MAS application in the real world, built a model of a novel MAS model that could improve the world, and built a model of their ideal class to learn MAS topics Finally, students answered a survey that gathered their opinions about the LSP activity and current dynamics of (non gamified) MAS theoretical classes.

All students agreed that the activity of modelling a MAS application with Lego bricks had helped them conceptualise and reinforce concepts of the subject. They also claimed that they had learned from the MAS application proposals made by their fellows. They imagined their ideal class as an engaging and collaborative activity with the teacher giving support to groups of students, who interact among themselves and apply theoretical concepts to real applications. It is noteworthy that during the LSP activity they had fun and felt committed and motivated.

These results helped us to confirm the suitability of the gamification project. Moreover, some students suggestions about their ideal class inspired our class gamification design, which is depicted in Fig. 1 and detailed in the following.

The *revenues* (benefits) to achieve with the gamified class are: reinforce MAS learning, increase class engagement and increase class attendance.

Related to *players*, students could be considered *master players* in GMC framework, because they are familiar with the basic concepts of MAS. Furthermore, the design team agreed that user types were socializers and achievers.

As students usually have difficulties with the first contact with MAS concepts, and also to be continuously engaged during the class, our proposal of *behaviors* to develop in the players were how to ice-break and how to overcome mental exertion. Then, we propose to include several quizzes with real time feedback along the class, and oral presentations of group projects.

The *aesthetics* –desirable emotional responses evoked in students– we considered are: to overcome a challenge, explore MAS concepts, and fellowship.

As for *dynamics*, we chose to: be productive or contribute to reinforcing concepts for all class students, enhance the group identity and progression within the classroom, and visualize the progress of each student inside the gamified MAS session.

Components we used are: progress bar, achievements, leaderboards, badges, coins, and points.

The *mechanics* we defined revolve around point rankings. First, students answer an interactive quiz –performed by means of an Audience Response System– and obtain individual points. The student at the top of the ranking earns a badge, and each student adds his points to a paper scoreboard. Afterwards, students design a MAS project consisting of several activities that include the build-share-reflect sequence from Lego Serious Play. Along this sequence, students share proposals and get rewards: physical coins that correspond to a number of points in the scoreboard. Badges are also awarded for most valued (voted) projects. Additionally, a shared coffee break was included to promote group identity.

The *end game* consists in, firstly, choosing (and rewarding) the three students with the higher scores (that is, the number of points accumulated along the performed activities) and, secondly, declaring the absolute winner.

Regarding *platforms* where to apply gamification, we required to use both the physical classroom (and resources) environment as well as an Audience Response System website.

Finally, some *costs* are identified to be necessary to get some of the resources required to develop the gamification project. These are coins, badges, class projector and personnel.

Next section explains the actual execution of these *mechanics* and *end game* as well as the usage of *platforms* and incurred *costs*.

5.2 Gamified class

We conducted a two-hour-long theoretical class with 24 students which were familiar with the subject, and thus, they could be considered as *master players*.

The session was first introduced by recalling our gamification goal (i.e., expected *benefit*) and describing its outline: reviewing theoretical concepts and designing a Multi-Agent System.

Thus, the class started by reviewing theoretical concepts for half an hour. This activity was designed to consolidate MAS concepts by means of an Audience Response System that induced the 'answer a quiz' *behaviour* with 'individual challenge' *aesthetics*.

Specifically, students participated in an interactive quiz we created at Kahoot! [6] web *platform*. It consisted of 5 consecutive theoretical questions posted on the class projection screen. Top left image in Fig. 2 shows the first question. Students answered (and received feedback) by means of their mobile phones. Fig. 2 also depicts the colour-and-shape-code user-friendly interface that students are provided to select their individual answer (See top right image in Fig. 2) where left-most image shows possible answers to choose from whereas the rest provides student feedback. Cross-marked red screen stands for time-out or incorrect answers (no points awarded in any case), for both cases the correct answer is provided. Tick-marked green screen indicates the answer is correct as well as the number of awarded points. They were awarded up to 1000 *points*[7] if they answered each question correctly and within a 30 second timeout period, so that the quicker they answered, the most points they got. For each question/answer cycle, the class projection screen showed an histogram of aggregated answers indicating how many students had chosen each option. When figures showed that some concepts were not clear enough, the teacher opened a discussion about most common errors (see bottom left image in Fig. 2).

The *mechanics* of this ARS incorporated different ways of providing immediate feedback to students. On the one hand, as top right image in Fig. 2 shows, each mobile phone indicated individual achievements: if the answer was correct or wrong, the awarded points, and current position in the class ranking. On the other hand, the class projection screen showed student rankings after each question (see bottom right image in Fig. 2). Afterwards, at the end of the quiz, the winner got a badge and students were asked to write their total awarded points in a paper *scoreboard*. We had provided beforehand this scoreboard for including subsequent activity scores, and thus, it also acted as a *progress bar*.

Afterwards, we devoted most of the session to experience with the design of a MAS. Here the teacher first presented subsequent parts a Multi-Agent Based Simulation research project, and then we encouraged the students to collaboratively design related examples of MAS —so we induced a MAS design *behaviour*.

Simweb is a former European research project that aimed at defining and studying alternative market models for the distribution of electronic contents –such as music or pieces of news. It involved the definition of a market as a MAS including both market stakeholders as well as those products being traded. Thus, it required characterising: i) buyer and provider agent roles and associated be-

[6] https://getkahoot.com

[7] For the sake of readability, we also use italics to highlight previously designed components from Fig. 1.

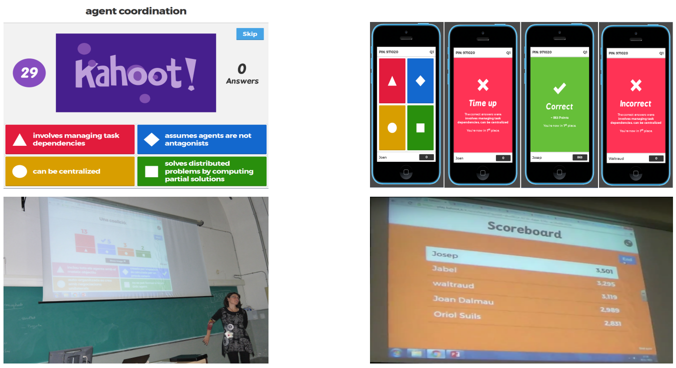


Fig. 2. Top-left: Question projected. Top-right: Mobile student interface. Bottom-left: Histogram of answers. Bottom-right: quiz scoreboard.

haviours; and ii) product ontologies. Different actions and strategies were defined for each role: "imitation", "innovation", or "reputation leadership" constituted alternative provider strategies; whereas "buy cheapest offers", "satisfy requests exactly", or "be loyal to provider" were some buyer strategies. Additionally, products were defined in terms of its attributes and associated value domains. These definitions were key to define provider product offers as well as buyer preferences.

Students were asked to design a MAS market collaboratively. Initially, student pairs had assigned 3 different products (a party organizing service, a dron, and an electronic book), and the various provider and buyer strategies mentioned before. Each pair had a specific product, buyer and provider strategies assigned such that there were 12 different pair assignments but 4 pairs traded the same product —so that they could form 3 bigger groups afterwards. When working collaboratively, MAS principles were somehow mapped into an experiential and immersive design, where each student had autonomy to design and play one agent (i.e., its behaviour) but student interaction was key to agree on the overall system orchestration (i.e., common product definition and interaction).

MAS market design was partitioned in two phases, each following a (i) design -(ii) share -(iii) reward sequence similar to the one from Lego Serious Play. First phase required students to define the product, provider offers, and buyer preferences. Whereas in the second, students specified the agent interaction protocol and executed the system so that agent decision making –based on computing buyer requirement satisfaction from considering providers' offers– become apparent. For each phase: (i) design was done in pairs; then (ii) each pair shared their definition with those 3 other pairs trading the same product and finally, (iii) students awarded each others' work by awarding physical *coins* that were initially equally distributed. Specifically, each pair agreed on the design they liked the most and awarded this other pair by giving 2 of their coins (one coin from and for each student). Moreover, it is worth mentioning that these tree stages had a 3-to-5-minute time limitation that kept students focused on their tasks.

This activity was quite demanding. Therefore, a shared coffee break was introduced to relax students a bit. Students had an opportunity to socialize in a more relaxed atmosphere and to *enhance the group identity.*

Considering the overall activity from a gamification point of view, our final goal was to increase the mastery (i.e., knowledge) of 'master' players, but students where also considered as both 'socializer' and 'achiever' players, since they worked collaboratively to achieve their design task. Furthermore, the *aesthetics* were that of 'discovery' and 'fellowship' and main *dynamics* pursued students to be productive while learning to design a MAS market.

Students obtained points from different activities so that the *End game* was defined by accounting for the students that reached the top at the final *score.* Thus the *mechanics* consisted in:

1. Sharing: The pair having most coins within each group explained their market design to the rest of the class. As a result, 3 pairs explained market examples that traded different products.
2. Reward: Classmates voted for the preferred one. Each member of the winner pair won a badge.
3. Teacher's feedback: The teacher provided final remarks on MAS design such as model visualization or MAS execution indicators.
4. Ranking: Students updated their paper *scoreboards* by computing their total number of points and by considering that both *coins* and *badges* accounted for 1000 *points* each.
5. Winner and price assignment: top-three-scored students raised their subject grades proportionally. The overall winner was also acclaimed by classmates.

6 Results

Based on data gathered from surveys filled by students after the gamified class, we present the following results:

- 96% of students considered that their knowledge or skills in MAS concepts had increased after hands-on activities during the gamified MAS class.
- 96% of students indicated that tools used in class (interactive game, participatory activities, group work, awards, etc.) had helped them to be engaged, motivated, and also to reinforce and further learn MAS concepts.
- 92% of students had fun during the gamified theoretical session.
- 88% of students believed that classes could include gamified dynamics.
- 83% of students would attend more classes, if they were gamified.
- 96% of students would recommend their fellows to attend gamified sessions.

Analyzing these results, we can conclude that the majority of students perceived knowledge acquisition during the session. We can also observe that students were a bit more engaged and motivated than took pleasure. It could be due to the fact

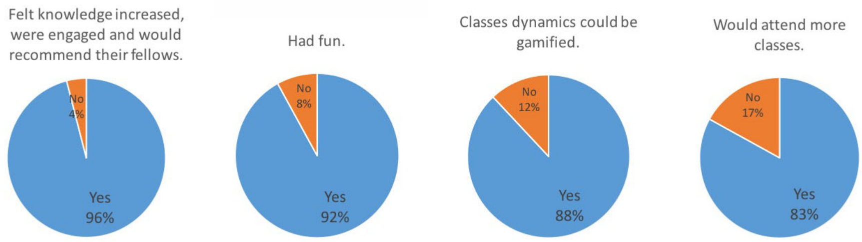


Fig. 3. Results of gamified MAS class about students.

that they probably did not follow well the activities due to an overloaded agenda, as they themselves said. Then, as a lesson learned, we could say that gamified activities must be carefully designed to facilitate participation and engagement of students in class.

Related to students' opinions about the use of gamification in theoretical classes and if they would attend classes more frequently, the reason of the decrease of values until 88 and 83 percent respectively may result from (i) the preferences of some students of "learning by example", which demands from them a less cognitive effort (they play a passive rol) in contrast to their active rol during gamified classes, (ii) and the coincidence of studies and work timetables. Recall that this is a subject of last year of studies, so that many students work and study at the same time. Additionally, it should be noted that the majority students had fun with ARS quiz and found it interesting. Nevertheless, some others did not liked the competitive aspect of gamification.

7 Conclusions and Future Work

This paper presents a methodology to improve traditional classroom dynamics and knowledge acquisition. Specifically it aims at promoting student's performance, in-class participation and attendance. This methodology is based in the Gamification Model Canvas framework, but extends it to include MAS principles, as well as those of Lego Serious Play and Audience Response Systems, so that its application in current academic contexts become most suitable.

We have developed a use case consisting of the design and implementation of a gamified class in the subject of MAS in a university level course. 96% of students believed that activities and tools used in the gamified class had helped them to consolidate and deepen their knowledge, besides got them engaged and motivated during class. As future work, we expect to be able to further evaluate the proposal extending this experience to next year MAS course as well as to other subjects.

References

[1] Barreteau, O., Bousquet, F., Attonaty, J.M.: Role-playing games for opening the black box of multi-agent systems: method and lessons of its application to senegal river valley irrigated systems. J. of Artif. Societ. Social Simul. 4(2), 5 (2001)

[2] Bartle, R.: Hearts, clubs, diamonds, spades: Players who suit muds. Journal of MUD research 1(1), 19 (1996)

[3] Beer, M.: Multi-Agent Systems for Education and Interactive Entertainment: Design, Use and Experience: Design, Use and Experience. IGI Global (2010)

[4] Bellanca, J.A.: 21st century skills: Rethinking how students learn. Solution Tree Press (2011)

[5] Casasola, E., De, V., Cliffe, O., Padget, J.: Teaching mas in the uk and in latinamerica. Innovations in Teaching And Learning in Information and CS (2005)

[6] Deterding, S., Dixon, D., Khaled, R., Nacke, L.: From game design elements to gamefulness: defining gamification. In: International Academic MindTrek Conference: Envisioning Future Media Environments. pp. 9–15. ACM (2011)

[7] Domínguez, A., Saenz-de Navarrete, J., De-Marcos, L., Fernández-Sanz, L., Pagés, C., Martínez-Herráiz, J.J.: Gamifying learning experiences: Practical implications and outcomes. Computers & Education 63, 380–392 (2013)

[8] Gauntlett, D.: Creative explorations: New approaches to identities and audiences. Routledge (2007)

[9] Hunicke, R., LeBlanc, M., Zubek, R.: Mda: A formal approach to game design and game research. In: AAAI Workshop on Challenges in Game AI. vol. 4 (2004)

[10] Ibanez, M.B., Di-Serio, A., Delgado-Kloos, C.: Gamification for engaging computer science students in learning activities: A case study. Learning Technologies, IEEE Transactions on 7(3), 291–301 (2014)

[11] Iosup, A., Epema, D.: An experience report on using gamification in technical higher education. In: Proc. 45th ACM technical symposium on Computer science education. pp. 27–32. ACM (2014)

[12] Kapp, K.M.: The gamification of learning and instruction: game-based methods and strategies for training and education. John Wiley & Sons (2012)

[13] Kay, R.H., LeSage, A.: Examining the benefits and challenges of using ARS: A review of the literature. Computers & Education 53(3), 819–827 (2009)

[14] Kurkovsky, S.: Teaching software engineering with lego serious play: conference workshop. Journal of Computing Sciences in Colleges 30(6), 13–15 (2015)

[15] Li, C., Dong, Z., Untch, R.H., Chasteen, M.: Engaging computer science students through gamification in an online social network based collaborative learning environment. Journal of Information and Education Technology 3(1), 72–77 (2013)

[16] Marczewski, A.C.: Even Ninja Monkeys Like to Play. CreateSpace Independent Publishing Platform (2015)

[17] Mora, A., Riera, D., González, C., Arnedo-Moreno, J.: A literature review of gamification design frameworks. In: Games and Virtual Worlds for Serious Applications (VS-Games), 2015 7th International Conference on. pp. 1–8. IEEE (2015)

[18] Nicholson, S.: A user-centered theoretical framework for meaningful gamification. In: Proc. of GLS 8.0 Games+Learning+Society Conference. pp. 223 – 229 (2012)

[19] Richter, G., Raban, D.R., Rafaeli, S.: Studying gamification: The effect of rewards and incentives on motivation. In: Gamification in education and business, pp. 21–46. Springer (2015)

[20] Sakellariou, I., Kefalas, P., Stamatopoulou, I.: Teaching intelligent agents using netlogo. In: ACM-IFIP IEEIII (2008)

Argumentation Support Tool with Advisory Function

Yuki Katsura, Kei Nishina, Shogo Okada, and Katsumi Nitta

Department of Computational Intelligence and Systems Science
Tokyo Institute of Technology
4259 Nagatsuta-cho, Midori-ku, Yokohama, Kanagawa, Japan
{katsura_y,nishina,okada,nitta}@ntt.dis.titech.ac.jp

Abstract. In a school, it is a matter of importance to educate on discussion skills. On-the-job training (OJT) is an effective method for discussion education. To reduce the burden of the supervisor taking care of several discussion groups, we have developed a discussion support tool for use on a tablet PC. Each student is assumed to participate in the discussion and exchange messages through a computer network. This tool has an easy input interface by which a student inputs his message in the form of an argumentation diagram. Then, this message is converted into logical formulas, and furthermore, converted into an argumentation framework, which represents a logical structure of the discussion. Then, the semantics of the discussion is calculated and they are displayed on the screen to the student. This information helps students to understand the current status of the discussion and to select the next utterance.

Key words: Argumentation theory, argumentation framework, argumentation support tool, logical analysis of real complicated discussions, reliability of argument, modular structure

1 Introduction

In order to educate on discussion skills in a school, on-the-job training (OJT) is often used. To proceed OJT in a class, a supervisor explains the subject of discussion to students, divides students into several groups, and each group discusses the subject. During the discussion, the supervisor circulates around these groups, observes their discussion, and gives advice. A discussion contains several kinds of utterances such as claim, concede, denial, argument, counter argument, and so on. The supervisor recognizes these speech acts, detects attack and support relations between utterances, draws possible conclusions and gives advice based on these observations. Though OJT is a useful method for educating discussion skills, as the supervisor has to take care of several groups at once, it is difficult to give sufficient advice to everyone involved.

On the other hand, several on-line discussion support tools have been developed. By using these tools, users can exchange messages by way of a computer network. Some tools have a user interface which accepts users' input in the form of a diagram. As this diagram shows the logical structure between messages, it helps users to proceed in a discussion effectively. However, these tools don't have the function to give advice during discussion.

Therefore, we aim at developing an on-line discussion support tool which solves problems of OJT. This system recognizes the logical structure of discussion and gives advice to students by calculating semantics of argumentation.

In Section Two, we introduce theory of computational argumentation as background theory. In Sections Three and Four, we show an overview of an argumentation support tool which has the function of giving advice instead of a supervisor In Section Five, we will show an example of discussion by using this tool.

2 Theoretical Background

This argumentation support tool is based on Toulmin Diagram and the theory of Argumentation Framework (AF), which represents structure of discussion in the form of a graph, and defines its semantics.

2.1 Toulmin Diagram

Stephen Toulmin, a British philosopher, insisted that for a good argument, it needs to provide good justification for a claim. He proposed a model of argument which contains six interrelated components such as claim (conclusion), data, warrant, backing, rebuttal, and mode (qualifier).

Fig. 1 is an example of Toulmin Diagram which shows six components as the justification of a claim "Tweety flies." Toulmin Diagram is useful to make clear the logical structure of argument.

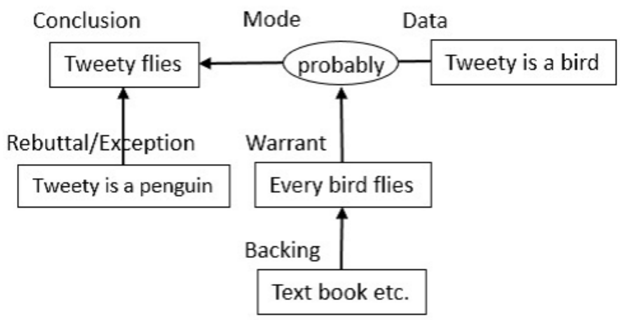


Fig. 1. Toulmin Diagram

2.2 Argumentation Framework

An argumentation framework [3] is a tuple $AF = (Ar, attacks)$, where Ar is a set of arguments, and $attacks \subseteq Ar \times Ar$ is a binary attack relation on Ar. An attack from an argument $a \in Ar$ to an argument $b \in Ar$ is expressed as $(a,b) \in attacks$. The semantics of AF is defined as follows:

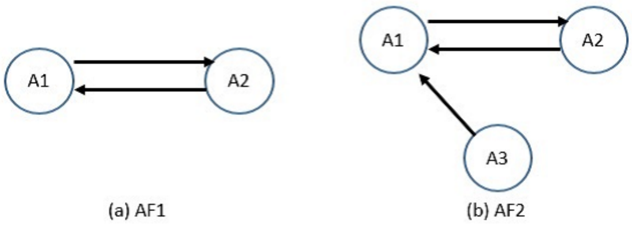


Fig. 2. Example of Argumentation Framework

- The set $S \subseteq Ar$ is *conflict-free* iff $^{\forall}x, y \in S, (x, y) \notin attacks$.
- For any $x \in Ar$, x is $acceptable$ with respect to some $S \subseteq Ar$ iff $^{\forall}y \in Ar$ s.t. $(y, x) \in attacks$ implies $^{\exists}z \in S$ s.t. $(z, y) \in attacks$.
- $F_{AF} : 2^{Ar} \rightarrow 2^{Ar}$, and $F_{AF}(S) = \{a \in Ar \mid a$ is acceptable w.r.t. $S\}$.
- $S \subseteq Ar$ is a *complete extension* iff S is *conflict-free* and $F_{AF}(S) = S$.
- The smallest element (with respect to set inclusion) among the complete extension is called a *grounded extension*.
- The maximal element (with respect to set inclusion) among the complete extension is called a *preferred extension*.

An argument which belongs to a ground extension is an argument which is not defeated by any counter argument, and an argument which belongs to any preferred extension is an argument which is not defeated completely and which has chance to hold. An argument which doesn't belong to any complete extension is an argument which is defeated by other counter argument.

For example, let's consider the following arguments.

A1: A nuclear power plant is necessary.

A2: A nuclear power plant is not necessary.

These arguments are represented by $AF1 = (\{A1, A2\}, \{(A1, A2), (A2, A1)\})$ (Fig.2(a)). The complete extension CE, the gounded extension GE and the preferred extension PE of AF1 are $CE = \{\{\}, \{A1\}, \{A2\}\}, GE = \{\}, PE = \{\{A1\}, \{A2\}\}$, respectively.

Here, let's consider following argument A3 be added to AF1.

A3: A solar power plant may supply sufficient energy.

Then, the new structure is represented by $AF2 = (\{A1, A2, A3\}, \{(A1, A2), (A2, A1), (A3, A1)\})$ (Fig.1(b)). Semantics of AF2 become $CE = \{\{A2, A3\}\}, GE = \{\{A2, A3\}\}, PE = \{\{A2, A3\}\}$, respectively.

2.3 Reliability-Based Argumentation Framework and Modularization of AF

AF theory defines its semantics using attack relation between arguments which appeared in discussion. However, AF theory doesn't consider the quality (the reliability) of arguments. Furthermore, when the theme of discussion contains difficult issue points, AF becomes large and complex. In such case, students cannot grasp the overall logical structure easily and to calculate semantics takes much time. To solve this problem,

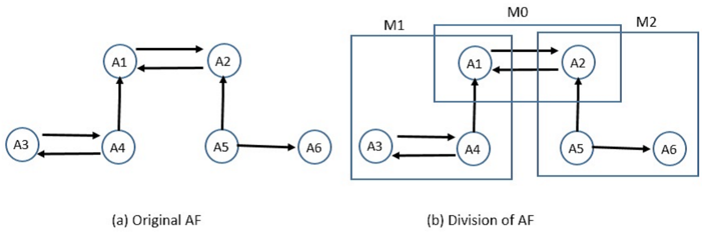


Fig. 3. Example of Modular Argumentation Framework

dividing an AF into several modules is a promising approach for understanding logical structure and for calculating semantics.

To support this modularization method, a new argumentation theory which integrates semantics of modules is necessary. Therefore, we proposed Reliability-Based Argumentation Framework Theory (RAF theory). RAF is a tuple $(Ar, attacks, ST)$, where Ar is a set of argument, $attacks \subseteq Ar \times Ar$, and ST is a function mapping from each argument in Ar to any one element in the set of reliabilities $\{sk, cr, def, unc\}$. Reliability sk, cr, def and unc are abbreviation of $skeptical$, $credulous$, $defeated$ and $uncertain$, respectively. If the reliability of an argument is $defeated$, the argument will be ignored because such argument is unreliable. If the reliability is $credulous$, the argument is on insufficient grounds. Therefore, pessimistic people will not allow a credulous argument to become a member of the grounded extension. Furthermore, when an argument is attacked by an argument whose reliability is cr, an optimistic person may ignore this attack. Therefore, the semantics of RAF is more complex than that of AF. For example, in the case of Fig. 2(b), if $st(A1) = sk, st(A2) = sk$, and $st(A3) = cr$, then the complete extension, the grounded extension and the preferred extension of AF2 becomes $CE = \{\{\}, \{A1\}, \{A2, A3\}\}, GE = \{\}, PE = \{\{A1\}, \{A2, A3\}\}$, respectively from pessimistic view.

When a large AF is divided into several modules, semantics of an AF is calculated by integrating semantics of each module by RAF theory. For example, let consider AF (= $(\{A1, A2, A3, A4, A5, A6\}, \{(A1, A2), (A2, A1), (A3, A4), (A4, A3), (A4, A1), (A5, A2), (A5, A6)\})$) is divided into three modules, M0, M1 and M2 (Fig.3). As a module M0 is higher than modules M1 and M2, semantics of AF in M0 is affected by semantics of M1 and M2. In M1, an argumentation framework AF1 (=({A1, A3, A4}, {(A3,A4), (A4,A3), (A4,A1) })) exists and an argument A1 belongs to a preferred extension $\{A1, A3\}$ of AF1. Therefore, in M0, the reliability of A1 becomes cr. On the other hand, in M2, an argumentation framework AF2 (=({A2, A5, A6 }, {(A5,A2), (A5,A6) })) exists and an argument A2 doesn't belong to the complete extension of AF2. Therefore, in M0, the reliability of A1 becomes def. Then, in M0, as reliability of A2 is def, A2 is ignored, and a grounded extension GE becomes $\{A1\}$ on the optimistic view, and becomes $\{\}$ on the pessimistic view.

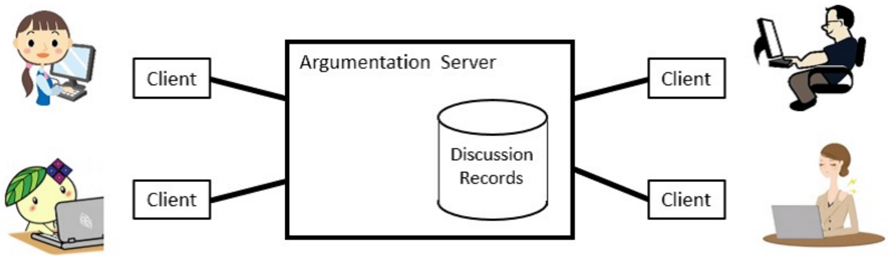


Fig. 4. On-line Argumentation Support System

3 Argumentation Support Tool on Tablet PC

3.1 Overview of Argumentation Support Tool

Our argumentation support tool [18] is designed to be used in the course of discussion training (argumentation training) in the school. This tool is installed on a tablet PC, and each student is assumed to participate in the discussion using his (her) tablet PC. These PCs are connected to the argumentation server, and students can exchange messages (utterances) via this server (Fig. 4). The argumentation server stores old discussion records in the discussion database, and has a function to search similar discussion scenes (parts of scenario) by comparing the sequence of utterances in the discussion database.

During discussion, this tool shows not only the logical structure of discussion in two types of diagrams such as an argument diagram and an argumentation framework but semantics of discussion. These informations correspond to advices by an instructor.

In a client system, a user interface appears on the screen (Fig.5) by which students communicate with others.

Fig. 6 shows the architecture of our argumentation support tool. At first, users input utterances in the form of Argument Diagram from the input interface, then the information of the diagram is sent to the argumentation server. In the server, the argumentation diagram is converted into logical expressions. Logical expressions are stored in the argumentation database. In this database, old argumentation records are stored. During argumentation, the argumentation server recognizes the latest three utterances and searches for similar utterance pattern (similar scene) in the database. If the argumentation server finds similar scene, it is sent the client.

The logical expressions are converted into an argumentation framework (AF) and its graph structure is shown on the client's screen by the output interface. Then the semantics of AF is calculated and its semantic information is shown on the client's screen, too.

3.2 Function of Argumentation Support Tool

3.2.1 Input Utterance in The Form of The Argument Diagram

The Argument Diagram represents the logical structure of utterances consisting of Data nodes, Claim nodes and Support links, which indicate the reasoning which is from the

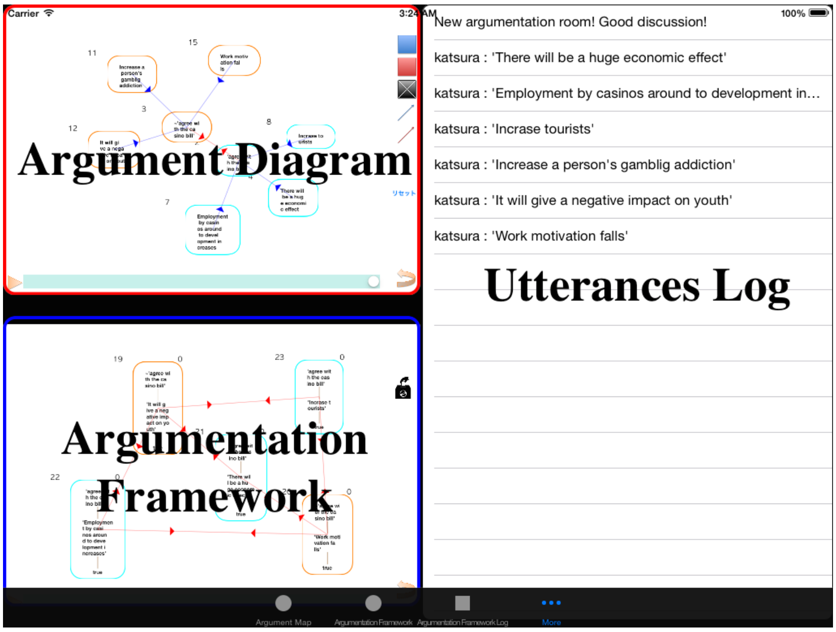


Fig. 5. Snapshot of the screen of the argumentation support tool

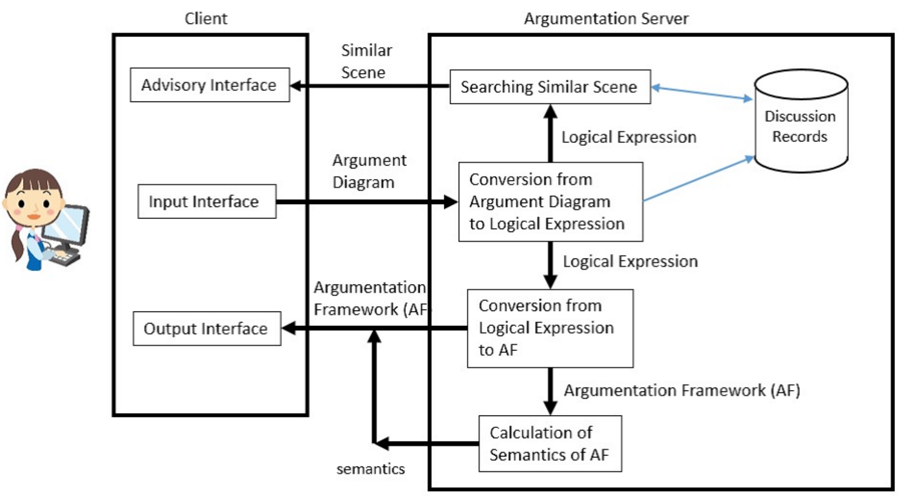


Fig. 6. Architecture of Argumentation Support System

contents of Data nodes to the Claim node, and Attack relation between the components of them. The Argument Diagram is based on Toulmin Diagram but is very simple because we want to make it easy for users to input during the discussion. This diagram saves time-series information of the argument, while accepting utterances input of the user.

Using the input interface, the user creates a node on the screen, makes a link which combines two node, and inputs his text message in the node.

Definition 1. *The argument diagram is a tuple,* $(A, attacks, supports)$ *, where, A indicates a set of utterances, while* $attacks$ *and* $supports$ *indicates binary relations between utterances in A. A corresponds to nodes on the screen, while* $attacks$ *and* $supports$ *indicates two kinds of links between nodes.*

3.2.2 Conversion of Argument Diagram into Logical Expressions

The argument diagram is converted into the formula. The formula is classified to R_s(A set of strict inference rules), R_d(A set of defeasible inference rules), K_p(A set of the ordinary premises) and K_n(A set of the axioms).

Let an argument diagram be $(A, attacks, supports)$. For any member of attacks and supports, R_d and K_p are obtained as follows.

- $\forall a, b \in A,\ (b, a) \in attacks$
  $\Longrightarrow \{\neg a \Leftarrow b\} \in R_d,\ \{b\} \in K_p$
- $\forall a, b \in A,\ (b, a) \in supports$
  $\Longrightarrow \{a \Leftarrow b\} \in R_d,\ \{b\} \in K_p$

3.2.3 Conversion of Logical Expressions into Argumentation Framework

Conversion is carried out according to the conversion process of Aspic+ [19]. *Example 1* shows a conversion example.

Example 1.

$K_p = \{s, u, x\}$	$K_n = \{p\}$	$R_s = \{s_1, s_2\}$	$R_d = \{d_1, d_2, d_3, d_4, d_5\}$
$d_1 : p \Rightarrow q$	$d_2 : s \Rightarrow t$	$d_3 : t \Rightarrow \neg d_1$	$d_4 : u \Rightarrow v$
$d_5 : v, x \Rightarrow \neg t$	$s_1 : p, q \rightarrow r$	$s_2 : v \rightarrow \neg s$	

Above formulas are converted into following argumentation framework.

$AF = (\{A_1, A_2, A_3, A_4\}, \{(A_3, A_1), (A_2, A_3), (A_3, A_4), (A_4, A_3)\})$

Here, $A_1 : \{d_1, s_1\},\ A_2 : \{d_4, s_4\},\ A_3 : \{d_2, d_3\},\ A_4 : \{d_5, d_4\}$

By showing this AF on the screen in the form of a graph structure, students can understand the overall logical structure of argumentation. However, if the AF contains a lot of arguments, the graph becomes too complex to understand at a glance. In such case, students can divide the AF into several modules by pointing pivot argument on the screen. Then, the AF graph is divided into two subgraphs at the designated node and each subgraph is treated in the different modules.

3.2.4 Calculation of Semantics of Argumentation Framework

During discussion, when an argumentation framework is updated, its semantics is calculated. After the complete extension, the grounded extension and the preferred extension are calculated, each argument is checked to see if it is included in the grounded extension or if it is included in the complete extension. If an argument is contained in the grounded extension, the color of its node becomes blue on the screen. If an argument is contained in the complete extension, its color becomes yellow, and if it isn't the complete extension, its color becomes red.

In the case of *Example 1*, the complete extension CE, the preferred extension PE and Grounded Extension GE are followings.

$$CE = PE = GE = \{A_1, A_2, A_4\}$$

Therefore, the color of A_1, A_2 and A_4 is blue, and the color of A_3 is red. Students can recognize the status of argument by its color, which is useful information needed to select the next utterance.

When the AF is divided into several modules, the semantics of the higher module is calculated by semantics of lower modules using RAF theory introduced in 2.3.

3.2.5 Searching Similar Scene

The argumentation server stores old discussion records (argumentation records) in the Discussion Database. Each discussion record is a sequence of logical formulas as following example. In this example, P1, Q1, R1, S1 and T1 are text data.

Sequence of formula in the discussion database:

...	
$id201 : P1 \Leftarrow Q1$	Q1 supports P1
$id202 : \neg Q1 \Leftarrow R1$	R1 attacks Q1
$id203 : \neg R1 \Leftarrow S1$	S1 attacks R1
$id204 : R1 \Leftarrow T1$	T1 supports R1
...	

During discussion, the server recognizes the sequence of arguments and searches similar sequence in the database. For example, if the latest three arguments of the current discussion are $id105$, $id106$ and $id107$ as follows, and if $P1$ and $P2$, $Q1$ and $Q2$, $R1$ and $R2$, and $S1$ and $S2$ are similar texts, then the argumentation server judges that the sequence from $id201$ and $id203$ is similar to the sequence from $id105$ and 107. Similarity between two texts are measured by the number of common key words.

Sequence of formula of current discussion:

...
$id105 : P2 \Leftarrow Q2$
$id106 : \neg Q2 \Leftarrow R2$
$id107 : \neg R2 \Leftarrow S2$

When a student considers the next utterance as $id108$, by referring to $id204$, he (she) may make a proper argument.

3.3 Example

To show the effectiveness of this argumentation support tool, we compared the usual face-to-face discussion without this tool and the discussion using this tool. The subject of discussion is "The expenses of the university should be free or not." Before the experiment, students are passed several documents concerning expenses of the university and they are given time to read and consider the discussion strategy.

Fig. 7 and Fig. 8 are Argumentation Diagram (AD) and Argumentation Framework (AF) of face-to-face discussion without the support tool. Fig. 9 and Fig. 10 are AD and AF of discussion using this support tool. By comparing Fig. 8 and Fig. 10, the discussion using this tool tends to go to deeper level. Also Fig.11 which is the comparison of depth of each argument shows the same result.

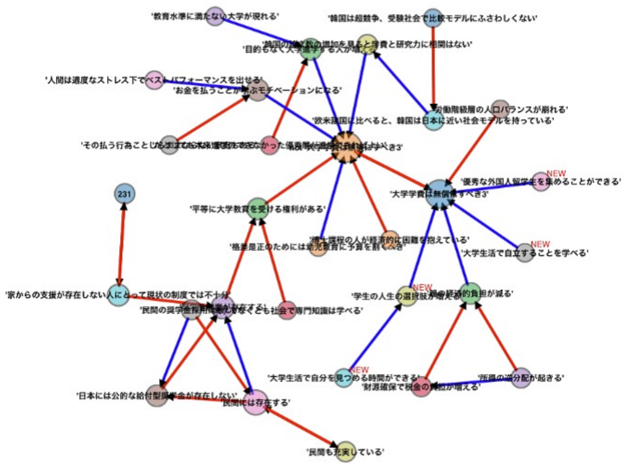


Fig. 7. Argument Diagram (without Tool)

4 Conclusion

We introduced an argumentation support tool based on the theory of AF and RAF. This tool helps users by supplying the environment for on-line discussion, by visualizing logical structure of arguments, by calculating semantics and by searching similar scene in old records. The information of semantics and similar scene in the discussion database helps students to grasp the situation of ongoing discussion and to select the next argument. These informations correspond to a part of advice made by a supervisor.

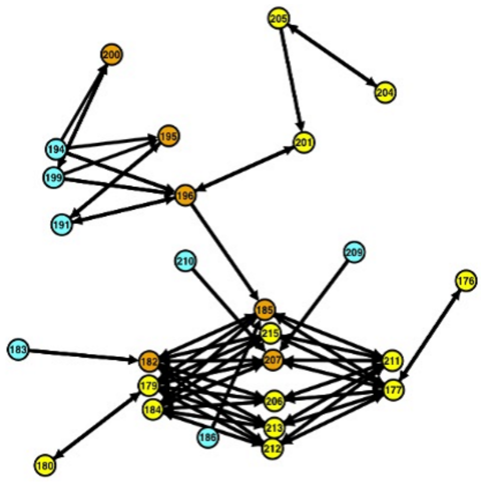


Fig. 8. Argumentation Framework (without Tool)l

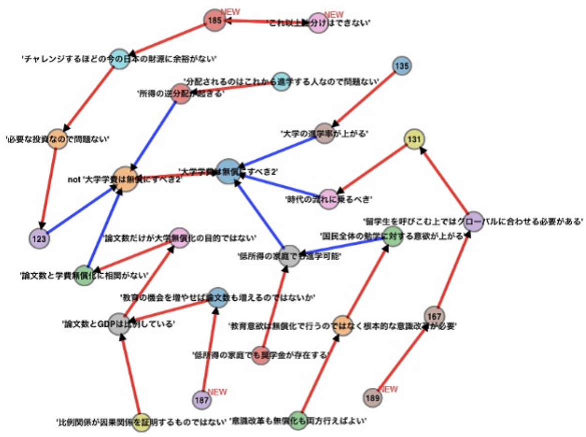


Fig. 9. Argument Diagram(with Tool)

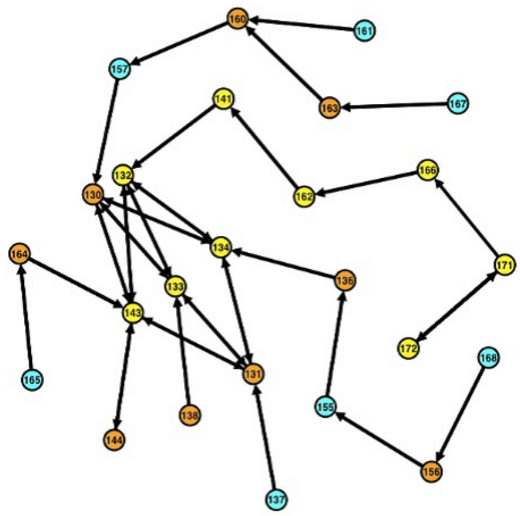


Fig. 10. Argumentation Framework (with Tool

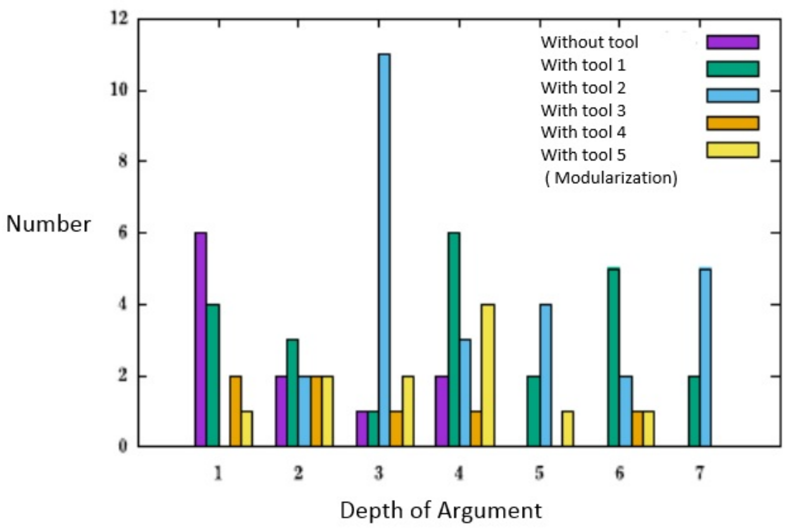


Fig. 11. Comparison of Depth of Argument

References

1. Chris Reed and Glenn Rowe. Araucaria: Software for argument analysis, diagramming and representation. *International Journal on Artificial Intelligence Tools*, 13(04):961–979, 2004.
2. Toulmin Stephen. The uses of argument. *Cambridge University Press, Cam bridge*, 1958.
3. Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.
4. Katie Atkinson, Trevor Bench-Capon, and Paul E DUNNE. Uniform argumentation frameworks. *Computational Models of Argument: Proceedings of COMMA 2012*, 245:165, 2012.
5. Paul E Dunne, Anthony Hunter, Peter McBurney, Simon Parsons, and Michael Wooldridge. Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artificial Intelligence*, 175(2):457–486, 2011.
6. Leendert VANDER TORRE and Serena Villata. An aspic-based legal argumentation framework for deontic reasoning. *Computational Models of Argument: Proceedings of COMMA 2014*, 266:421, 2014.
7. Niels Pinkwart, Collin Lynch, Kevin Ashley, and Vincent Aleven. Re-evaluating largo in the classroom: Are diagrams better than text for teaching argumentation skills? *Intelligent Tutoring Systems*, pages 90–100, 2008.
8. Phan Minh Dung, Phan Minh Thang, and Nguyen Duy Hung. Modular argumentation for modelling legal doctrines of performance relief. *Argument and Computation*, 1(1):47–69, 2010.
9. Gerhard Brewka and Thomas Eiter. Argumentation context systems: A framework for abstract group argumentation. In *Logic Programming and Nonmonotonic Reasoning*, pages 44–57. Springer, 2009.
10. Sanjay Modgil. Reasoning about preferences in argumentation frameworks. *Artificial Intelligence*, 173(9):901–934, 2009.
11. Leila Amgoud and Claudette Cayrol. On the acceptability of arguments in preference-based argumentation. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 1–7. Morgan Kaufmann Publishers Inc., 1998.
12. Trevor JM Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.
13. Yuqing Tang, Kai Cai, Peter McBurney, Elizabeth Sklar, and Simon Parsons. Using argumentation to reason about trust and belief. *Journal of Logic and Computation*, page exr038, 2011.
14. Hengfei Li, Nir Oren, and Timothy J Norman. Probabilistic argumentation frameworks. In *Theorie and Applications of Formal Argumentation*, pages 1–16. Springer, 2012.
15. Andrea Pazienza, Floriana Esposito, and Stefano Ferilli. An authority degree-based evaluation strategy for abstract argumentation frameworks.
16. Phan Minh Dung, Robert A Kowalski, and Francesca Toni. Assumption-based argumentation. In *Argumentation in Artificial Intelligence*, pages 199–218. Springer, 2009.
17. Martin Caminada. On the issue of reinstatement in argumentation. *Logics in artificial intelligence*, pages 111–123, 2006.
18. Yuki Katsura, Shogo Okada, and Katsumi Nitta. Dynamic argumentaion support tool using argument diagram. *Annual Conference of the Japanese Society for Artificial Intelligence*, 29:1–4, 2015. in Japanese.
19. H. Prakken. An overview of formal models of argumentation and their application in philosophy. *Studies in logic*, 4(1):65–86, 2011.