Pre-Proceedings of the 3rd International Workshop on Collaborative Agents – REsearch and development (CARE) 2011

Christian Guttmann
Michael Luck
Milind Tambe
(CARE Organisers)

Program Committee

Sherief Abdallah	British University in Dubai.
Abdulla Al Zaabi	Etisalat BT Innovation Center (EBTIC)
Magnus Boman	SICS
Rafael Bordini	Federal University of Rio Grande do Sul
Birgit Burmeister	Daimler AG
Cristiano Castelfranchi	Institute of Cognitive Sciences and Technologies
Wei Chen	Intelligent Automation Inc.
Leonardo Garrido	Centro de Computación Inteligente y Robótica. Tecnológico de Monterrey, Campus Monterrey.
Christian Guttmann	Etisalat British Telecom Innovation Centre (EBTIC)
Fredrik Heintz	Linköping University
Gal Kaminka	Bar Ilan University
Samin Karim	Accenture
Matthias Klusch	DFKI
Victor Lesser	University of Massachusetts Amherst
Sonenberg Liz	Melbourne University, Department of Information Systems
Michael Luck	King's College London
Zakaria Maamar	Zayed University
Gord Mccalla	University of Saskatchewan
Pasquier Philippe	The Melbourne University, Department of Information Science
Goss Simon	DSTO Air Operations Division
Milind Tambe	University of Southern California
Michael Thielscher	The University of New South Wales
Rainer Unland	University of Duisburg-Essen, ICB
Birna Van Riemsdijk	TU Delft
Toby Walsh	NICTA and UNSW
Kumari Wickramasinghe	Monash University
Cees Witteveen	Delft University of Technology
Wayne Wobcke	University of New South Wales
Neil Yorke-Smith	American University of Beirut and SRI International
Inon Zuckerman	UMIACS

Towards Optimal Planning for Distributed Coordination Under Uncertainty in Energy Domains

Jun-young Kwak[1], Pradeep Varakantham[2], Milind Tambe[1], Laura Klein[1], Farrokh Jazizadeh[1], Geoffrey Kavulya[1], Burcin B. Gerber[1], David J. Gerber[1]

[1] University of Southern California, Los Angeles, CA, 90089
{junyounk,tambe,lauraakl,jazizade,kavulya,becerik,dgerber}@usc.edu
[2] Singapore Management University, Singapore, 178902
pradeepv@smu.edu.sg

Abstract. Recent years have seen a rise of interest in the deployment of multiagent systems in energy domains that inherently have uncertain and dynamic environments with limited resources. In such domains, the key challenge is to minimize the energy consumption while satisfying the comfort level of occupants in the buildings under uncertainty (regarding agent negotiation actions). As human agents begin to interact with complex building systems as a collaborative team, it becomes crucial that the resulting multiagent teams reason about coordination under such uncertainty to optimize multiple metrics, which have not been systematically considered in previous literature. This paper presents a novel multiagent system based on distributed coordination reasoning under uncertainty for sustainability called SAVES. There are three key ideas in SAVES: (i) it explicitly considers uncertainty while reasoning about coordination in a distributed manner relying on MDPs; (ii) human behaviors and their occupancy preferences are incorporated into planning and modeled as part of the system; and (iii) various control strategies for multiagent teams are evaluated on an existing university building as the practical research testbed with actual energy consumption data. We empirically show the preliminary results that our intelligent control strategies substantially reduce the overall energy consumption in the actual simulation testbed compared to the existing control means while achieving comparable average satisfaction level of occupants.

1 Introduction

Over the decades, energy issues have been getting more important. In the U.S., 48% of energy consumption is from buildings, of which 25% is associated with heating and cooling [12] at an annual cost of $40 billion [12]. Furthermore, on an annual basis, buildings in the United States consume 73% of its electricity. Recent developments in multiagent systems are opening up the possibility of deploying multiagent teams to achieve complex goals in such energy domains that inherently have uncertain and dynamic environments with limited resources.

This paper focuses on a novel planning method for distributed coordination under uncertainty (regarding agent negotiation actions) to optimize multiple competitive objectives: i) amount of energy used in the buildings; ii) occupant's comfort level; and iii) practical usage considerations. There have been some trials to balance energy consumption and enhancement of building services and comfort levels [7, 11] and to monitor and collect energy consumption data [7, 8] in energy domains. Other works have explicitly focused on design optimization and use of multiagent systems [5, 9] in different domains. In addition, some multiagent systems [3, 4, 13] have been employed to model home automation systems. Unfortunately, past work in the energy domain has three key weaknesses. First, they do not consider uncertainty while reasoning about coordination and mostly rely on deterministic plans. Second, they limitedly incorporate intelligence of occupancy or occupancy preferences into the system and thus occupants are not explicitly modeled as agents in the system. Third, they are mostly evaluated in their own simulation environments, which are not constructed on the actual energy data and occupants' responses in the buildings. Thus, their assumptions may not be realized in real-world problems.

This paper presents a novel multiagent system based on distributed coordination reasoning under uncertainty for sustainability called SAVES (Sustainable multi-Agent systems for optimizing Variable objectives including Energy and Satisfaction). SAVES provides three key contributions to overcome limitations in past work. First, we explicitly consider uncertainty while reasoning about coordination in a distributed manner. In particular, we rely on MDPs (Markov Decision Problems) to model agent interactions, specifically focusing on rescheduling meetings, which will be extended to decentralized MDPs. Second, human behaviors and their occupancy preferences are incorporated into planning and modeled as part of the system. As a result, SAVES is capable of generating an optimal plan not only for building usage but also for occupants. Third, the influence of various control strategies for multiagent teams is evaluated on an existing university building as the practical research testbed with actual energy consumption data in the simulation. Since the simulation environment is based on actual data, this result can be easily deployed into the real-world. Preliminary results show that our intelligent control strategies substantially reduce the overall energy consumption in the actual simulation testbed compared to the existing control means while achieving comparable average satisfaction level of occupants.

2 Related Work

With rising energy costs, the need to design and integrate scalable energy consumption reduction strategies in buildings calls for novel approaches. There are numerous challenges associated with energy resources such as supply and depletion of energy resources and heavy environmental impacts [11] (ozone layer depletion, global warming, climate change, etc.). The rise in energy consumption in buildings can be attributed to several factors such as enhancement of building services and comfort levels [7, 11], through heating, cooling and lighting needs and increased time spent indoors [11].

To model and optimize buildings' energy consumption, building owners and facility managers are demanding robust, intelligent and adaptable performance monitoring techniques. These techniques are important in energy consumption data collection [7, 8] and ambient environmental conditions control [7]. Existing heating, cooling, ventilation, and lighting systems generally operate with no intelligence of occupancy or occupancy preferences and therefore are unable to optimize operations. Even more, no feedback is available to occupants about how their actions and schedules impact building energy consumption. To realize both tangible benefits such as energy and operation savings, value property, reduction in occupant complaints as well as the intangible benefits such as occupant comfort and satisfaction, designers must develop energy adaptive capabilities within the building environmental control systems.

Abras *et al.* [3], Conte *et al.* [4] and Roy *et al.* [13] have employed multiagent systems to model home automation systems (or smart homes) and simulating control algorithms to evaluate performance. While there is relevance in terms of the problem domain and employing multiagent systems, our representation and approaches are different in having to account for human preferences and decisions directly.

Research by Fahrioglu *et al.* [6] and Mohsenian-Rad *et al.* [10] provide incentive compatible mechanisms for distribution of energy among interested parties. This thread of research is complementary, especially in designing incentives for humans to reveal their true energy preferences. However, these approaches assume a centralized controller with whom all the members interact, which is not present in our domain. Instead, there are peer-to-peer negotiations between humans regarding their energy consumption and comfort level.

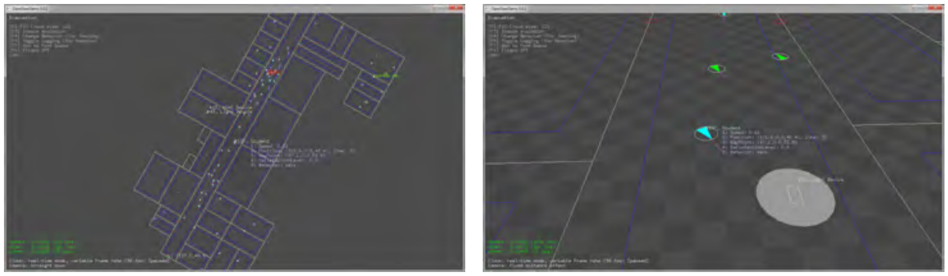
3 Design Decisions

The SAVES system consists of a simulation module, an input/output module to communicate with agents, and an underlying reasoning and planning module. Figure 1 shows a generic loop of the system. In particular, the input/output module first collects data and constructs the world model. Given the world model, the reasoning and planning module generates policies to achieve the given objectives in the context of coordination. With these world model and generated policies, the simulation module models agents' physical and behavioral interactions in the system and realize the coordination in the actual world via the input/output module. We now describe the modules as well as the particular instantiations of these modules in the energy domain.

Fig. 1. Overall System Design

The simulation module provides a 2D, OpenGL environment based on the open-source project OpenSteer [3] as shown in Figure 2. The simulation module consists of two different types of agents as described below, modeling their physical and behavioral interactions. It can be used for efficient statistical analysis of different control strategies

[3] http://opensteer.sourceforge.net/



(a) (b)

Fig. 2. Screen capture

in buildings before deploying the system to an actual physical world. The input/output module makes a connection among different modules in the system by collecting actual data in the domain, transferring data to the reasoning module, sending output results to the simulation or deployed module in the world to represent outputs, and providing means to communicate with agents via proxy and handheld devices. The coordination and planning module generates optimal policies to achieve the given team missions considering multiagent interactions in the context of coordination in the mutiagent setup.

Here we describe the design issues regarding agents, first introducing building component agents and human agents, then detailing the method to calculate the properties of agents, and finally discussing different control strategies considering agent interactions.

3.1 Building Component Agents

We consider three building component agent categories: a HVAC (Heating, Ventilating, and Air Conditioning) agent, a lighting agent, and an appliance agent. The HVAC agent is modeled based on the principles of thermodynamics, fluid mechanics, and heat transfer. We assume that this agent mainly controls the temperature of the assigned zone. The lighting agent controls the lighting level of the room. For the appliance agent, we only include the computer device including the desktop and laptop computers in this work. These agents have two possible actions: “on” and “standby”. When the lighting or appliance agents are “on”, they consume some fixed amount of energy. We measure the average amount of energy used by these agents, which will be detailed in Section 5.

Since the energy consumption of HVAC agents relies on a set of parameters including the temperature change in the space, air flow, and number of people, etc., the average value cannot be simply measured. Instead, we describe how to compute the energy use by HVAC agents below.

Calculating Total Energy Consumption: Since the building is composed of a large number of HVACs and they are the main consumers of the energy, it is important to choose the right set of parameters and reasonable values for them. In particular, the energy consumption of HVAC agents is calculated as following [2] mainly based on changes in air temperature and airflow speeds:

$$Q = \frac{1.1 \times CFM \times \Delta T}{3412.3},\ \Delta T = log(\frac{CFM}{C}),$$

where Q is the amount of energy used (kWh), CFM is an air volume flow rate (ft^3/min), which is typically ranged between 500–1500 (ft^3/min), ΔT is the temperature change in a zone (°F), and C is a scale factor.

3.2 Human Agents

There are four different types of human agents such as a faculty, staff, graduate student, and undergraduate student. Each agent has access to a subset of the six available behaviors according to their types — wander, attend the class, go to the meeting, teach, study, and perform research, any one of which may be active at a given time, where the behavior is selected via the given class and meeting schedules. During execution of these behaviors, individual travelers may move at integer speeds from 0 to 3. Each agent also has specific levels of emotions and information about the environment. Specifically, every agent has a property about the satisfaction level based on the current environmental condition and knows his or her current location without any noise. A more extended discussion of the satisfaction property will take place below.

Calculating Satisfaction Level: The satisfaction level (SL) of an individual human agent is modeled as a percentage value between 0 and 100 (0 is fully unsatisfied, 100 is fully satisfied). SL of the individual occupant is calculated as following:

$$SL = 100.0 - PPD,$$

$$PPD = 100.0 - 95.0 \cdot \exp^{-(0.03353 \cdot PMV^4 + 0.2179 \cdot PMV^2)},$$

where SL is the satisfaction level (%), PPD is the Predicted Percent Dissatisfied (%) [1], PMV is the Predicted Mean Vote. The PMV index is calculated from an equation of thermal balance for the human body in ASHRAE Standard [1].

4 Control Strategies

In a given scenario, all agents within the simulation will use the same strategy. Possible strategies include: i) manual control strategy, ii) reactive control strategy, iii) proactive control strategy, and iv) proactive control strategy based on multiagent coordination.

4.1 Manual Control

The manual control strategy simulates the current building control method maintained by USC facility managers. Specifically, we assume that HVAC agents are not controlled by human agents and that appropriate temperature points are centrally set by facility managers. For HVAC agents, the CFM values are fixed throughout the simulation. In this control setting, HVAC agents always attempt to reach the pre-set temperature using the fixed CFM value regardless of the presence of human agents and their preferences in terms of temperature. Lighting agents are controlled by only human agents. Control actions (i.e., turning on/off the light) of human agents are either deterministic or stochastic according to the type of action. In particular, when human agents enter the space, they always turn on the light. When they leave the space, they stochastically turn off the light. For appliance agents, we simply assume that they are always on.

4.2 Reactive Control

Since the manual control strategy simply follows the pre-defined policy provided by the facility managers, it is fairly easy to come up with action plans of building component agents. However, it does not adapt the given policies based on actual schedules or preferences of occupants in the building, and thus the building component agents are limited to adapt their control policies appropriately according to the dynamic changes. Particularly, HVAC agents keep operating to reach the desired point, even though the space is empty, which ends-up wasting energy. At the same time, since they do not consider occupants' preferences in the space and instead prioritize the pre-determined points, the average satisfaction level of occupants can decrease.

Here we discuss about another control strategy that building component agents reactively respond to the behaviors of human agents. In this setting, we assume that HVAC agents are not controlled by human agents and that appropriate temperature points are measured based on the average preference of human agents in the specific space. HVAC agents automatically turn on and off according to the presence of people and temperature set points, and the CFM values are adjusted accordingly. In the reactive control strategy, the lighting and appliance agents are now automatically controlled. In particular, they are turned on and off according to the presence of people. For instance, when people enter the specific room, the lighting and appliance agents are automatically turned on, and when people leave the room, they are turned off. While human agents follow their given schedules, with the reactive setting, the building component agents can act more intelligently than the manual policy as they operate based on human agents' actual needs. As a result, we can reduce the cases where the energy is wasted for unnecessary spaces, which will contribute to the reduction of the overall energy consumption.

4.3 Proactive Control

Although the reactive control strategy can adapt their policies based on actual needs of occupants in the building, this approach is still limited in a sense of optimality. In practice, there is a delayed effect in changing temperature. In other words, HVAC agents can only change a certain amount of degree in temperature per hour. This property exposes the weakness of the reactive control strategy. Although HVAC agents know the desired temperature of human agents at a specific time point, it takes a certain amount of time to reach the desired temperature point from the current air temperature, and the satisfaction level of occupants in the space will decrease during that time.

To overcome limitations of the reactive setting, we suggest a third strategy controlled in a proactive manner. Given the meeting and class schedules of human agents, the building component agents can predict: i) what their preferences are in terms of temperature, ii) how long it will take to reach the preferred temperature point from the current condition, iii) what CFM value is required, etc. In this setting, the building component agents can access the meeting/class schedules of human agents. Based on that prior knowledge, they now generate more optimal policies to reduce the overall energy consumption while maximizing the average satisfaction level of occupants in the building. For instance, since the HVAC agent knows the desired temperature points in advance based on the given schedules of human agents, it can generate plans to control temperature with less power. At the same time, when human agents get to the space, the current air temperature is already met with their preferences, and thus their satisfaction level increases.

4.4 Modeling Multiagent Coordination: MDP representation

With the existence of human agents, agent interactions are a fundamental aspect of our energy simulation. In SAVES , all agents share a common architecture based on MDP (Markov Decision Problem) frameworks, possessing varying degrees of knowledge about the world and other building agents (i.e., local knowledge).

This section describes our MDP representation in the energy domain for illustration. The MDP model represents a class of MDPs covering all types of meetings for which the agent may take rescheduling actions. In our work, we construct a MDP for each meeting as shown in Figure 3. Alternatively, we can model all meetings in the building as a single MDP. However, if we consider a gigantic MDP model for rescheduling all meetings together, the number of states and actions exponentially explodes as the number of agents increases. In addition, the complexity to handle all possible coordinations among agents significantly increases, which is burdensome to handle within a reasonable amount of time.

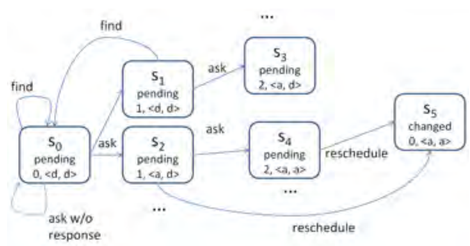


Fig. 3. Simplified MDP — d: disagree, a: agree

As preliminary work, we construct a simplified MDP model for rescheduling meetings. For each meeting, a meeting agent can perform any of three actions — reschedule, find another slot, and ask. For the "ask" action, an agent can autonomously reduce its own autonomy and ask a human agent whether he or she agrees with rescheduling the meeting. The human agent can respond to the meeting agent with "agree" or "disagree".

The agent may choose to perform any of these actions in various states of the world. State is composed of three features: $\langle f_1, f_2, f_3 \rangle$, where f_1 is the status whether meeting location and time is changed (i.e., pending or changed), f_2 is the number of "ask" actions invoked so far, and f_3 is a set of responses from all meeting attendees: $\langle rp_{i,1}, rp_{i,2}, ..., rp_{i,n} \rangle$, where n is the number of attendees of meeting i and $rp_{i,k}$ is a response of agent k to rescheduling meeting i (i.e., agree or disagree).

The MDP's reward function has its maximum value when the meeting agent invokes the "reschedule" action in the state where all meeting attendees agreed to reschedule. We denote the component of the reward function that focuses on the expected energy gain by rescheduling the meeting as r_{energy}. However, there is clearly a high team cost incurred by forcing all of the attendees to rearrange their schedules. This team cost is incorporated into the MDP's reward function by adding a negative reward, $r_{rearrange}$. The magnitude is also an increasing function in the number of attendees (e.g., rescheduling a meeting of a large group is more costly than rescheduling a

Table 1. Parameter Values for the Satisfaction Level Calculation

Parameter	Value Range from [1]	Value
Clothing	0.5–1.0 (light to heavy clothing)	1.0
Metabolic Rate	1.0–2.0 (low to high activity)	1.2
External Work	0	0
Air Temperature	20–28 (°C)	Zone temperature
Radiant Temperature	20–28 (°C)	70°F
Air Velocity	0–0.2 m/s	0.1 m/s
Relative Humidity	30–60 %	40 %

one-on-one meeting). The overall reward function for taking the "reschedule" action, $a_{reschedule}$, in a state s is a weighted sum of these components:

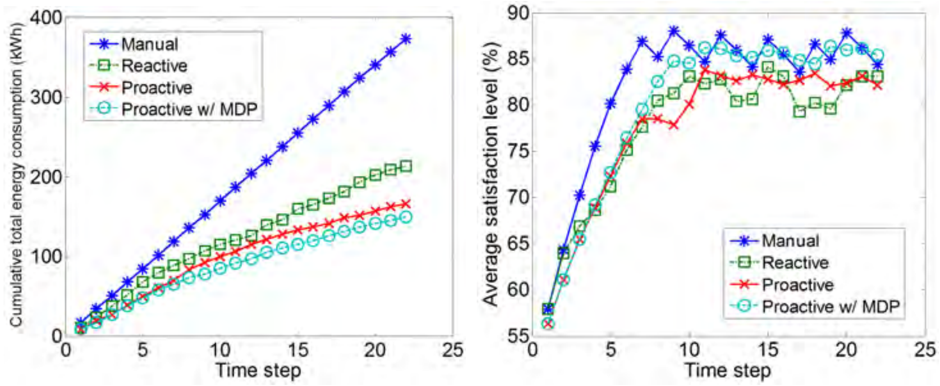
$$R(a_{reschedule}, s) = \alpha \cdot r_{energy} + (1 - \alpha) \cdot r_{rearrange}$$

, where $0 \leq \alpha \leq 1$. In addition, a small amount of cost is incurred to invoke actions of "ask" and "find another slot".

The MDP's transition probabilities represent the likelihood over possible action outcomes. Specifically, the transition function is defined considering four factors: i) meeting constraints of attendees; ii) level of energy consciousness, which determines how much they care about energy; iii) degree of intimacy among occupants; and iv) the current status of responses, which can be related to emotional contagion within the group. Since we store the current set of responses from individual agents and number of "ask" actions called so far, the repeated "ask" action may result in different transitions. In particular, the "ask" action, by which the agent queries the human agent, has $2^{n_i} + 1$ possible outcomes, where n is the number of attendees of the meeting i. First, the human agent may not respond at all, in which case, the agent is performing the equivalent of a "wait" action for a given timeout. Other set of possible outcomes are decided depending on responses of meeting attendees as illustrated in Figure 3. We assume that the "find" action reset values of features in the state.

5 Empirical Validation

We evaluate the performance of SAVES in our energy domain and compare four different control techniques: 1) manual control, 2) reactive control, 3) proactive control, and 4) proactive control with MDP. We focus on measuring two different criteria — total energy consumption (kWh) and average satisfaction level of occupants (%). For the HVAC agent, in the manual setting, 65–70°F was set to the desired temperature by facility managers, and in other control strategies, the desired temperature was decided based on the average preference values of building occupants. In the manual setting, the likelihood value for human agents to "turn off" lights was 50% and CFM was set to 1500.0. To calculate the energy consumption by the HVAC agent, we set the scale factor to 100.0. For the satisfaction calculation, we used the same parameter values in Table 1 (column 3) while performing the experiments across four different control strategies. The experiments were run on Intel Core2 Quadcore 2.4GHz CPU with 3GB main memory. All techniques were evaluated for 100 independent trials throughout this section. We report the average values.



(a) Total Energy Consumption (b) Average Satisfaction Level

Fig. 4. Comparison

5.1 Experimental Domain Description

We have identified an educational building in conjunction with USC Facilities Management Services, as our practical testbed. This campus building is composed of classrooms, offices for faculty and staff, and conference rooms for meetings. Specifically, we use one floor of the actual university building in the experiments, which has 18 zones and 33 rooms. There is one HVAC agent for each zone, and one lighting agent for each room. We also assume that each person in the office has either one desktop or laptop computer, and conference room and class room has two computers, respectively. There are four human agent categories: faculty, staff, graduate student and undergraduate student. Throughout the entire simulation, we consider a typical winter season in southern California (i.e., starting indoor temperature is 55°F in the simulation). During the simulation, indoor temperature goes down by -1°F per 30 minutes. Possible temperature range in the building is between 50 and 90°F. Students follow 2010 Fall class schedule, and we generated the arbitrary meeting schedules for faculty, staff, and student agents. The measurement is performed during the working hour (i.e., 8:00am–7:00pm), and the preference value of each occupant in temperature is randomly drawn between 60–70°F. To calculate the energy consumption of the lighting and appliance agents, we collected actual energy consumption data in the testbed building and obtained the average values. In particular, when the lighting agents are on, we use 0.128 *kW/h* for the office, 0.192 *kW/h* for the conference room, and 0.768 *kW/h* for the classroom. When they are off, they do not spend the energy. For the appliance agents, the desktop computer spends 0.150 *kW/h* and 0.010 *kW/h* when it is on and standby, respectively. The laptop computer spends 0.050 *kW/h* when it is on and 0.005 *kW/h* when it is on standby.

5.2 Comparison: Total Energy Consumption

We compared the cumulative total energy consumptions measured during work hours for all control strategies in the energy domain. Figure 4(a) shows the cumulative total energy consumption on the y-axis in *kWh* and the time step on the x-axis. Time step 1 indicates 8:00am and each time step increases by 30 minutes. As shown in the figure, the manual control strategy showed the worst result since it does not take into account behaviors or schedules of human agents and building component agents simply follow the predefined policies. The reactive and proactive control strategies showed lower energy consumptions than the manual setting by 43.0% and 55.6%, respectively. The proactive control strategy with the MDP model showed the best results among all different control strategies and statistically significant improvements (via t-tests) in terms of energy used in the testbed building, relative to other control strategies. Specifically, the proactive control with MDP reduced the energy consumption by 59.9% than the manual control strategy. Although we did not tune the parameter values and only applied the simplified MDP model, with considering multiagent coordination in SAVES, we could achieve significant improvements. These outcomes are still preliminary results and yet only tested in the simulation environment, all experimental results were measured based on the actual data and testbed. Later, we will be able to show even more improvement with the optimally tuned parameters and extend our work to deploy it into the actual building with proxy agents. Furthermore, as we revise the equations shown in Section 3.1, we will be able to get more exact results for analysis.

Now, we analyze how various control strategies can cause different results. Figure 5 shows the energy consumption distribution over zones for all control strategies. In the figures, the x-axis shows the group number of data obtained by each control strategy and the y-axis displays the total energy consumption for each zone in *kWh*. The floor plan we used in the simulation has four different types of zones, which decides the total energy consumptions. Specifically, zones 1–4 (blue), 9 (green), and 12 (yellow) have two offices per zone, zones 5–7 (light blue or cyan) are class rooms, zones 13–15 (orange or red) are conference rooms, and zones 11 (yellow), 16 (light red), and 17 (red) have three offices per zone. As shown in the first group of Figure 5, the manual control strategy results in the similar level of energy consumptions according to the different types of zones. This result clearly indicates that the manual setting is only impacted by the physical constraints of the building space itself, which never considers the interactions among agents. The normalized standard deviation was 0.134. In the reactive (the second group in Figure 5) and proactive setting (the third group in Figure 5), it now started showing the difference in terms of the amount of energy used even within the same type of zones since those methods consider the actual behavioral patterns and schedules of human agents, and building component agents respond and adapt their policies based on them. Their normalized standard deviations are 0.205 and 0.312, respectively, which are higher than the value of the manual setting. Lastly, the

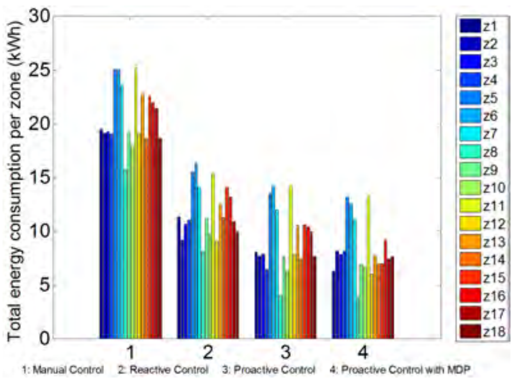


Fig. 5. Energy Consumption Distribution

proactive control strategy with the MDP model considers rescheduling of meetings. The target meetings to reschedule are ones with less than 4 people in the conference rooms in zones 13–15. We only considered the location reallocation and did not assume the meeting time can be also changed. New candidate locations are small faculty offices in zones 1–4. As shown in the fourth group of Figure 5, it showed increased energy consumptions in zones 1–4 due to the reallocated meetings, but simultaneously showed much more reduction in zones 13–15, as a result the overall energy consumption decreased. The normalized standard deviation was 0.313, which was the highest among different control strategies. These results give us a lesson that multiagent coordination/negotiation can benefit our model in SAVES, and by considering higher degree of coordination among agents, we will be able to achieve the significant energy reduction in this domain.

5.3 Comparison: Average Satisfaction Level

Here, we compare the average satisfaction level of human agents under different control strategies in the simulation. We used the equations discussed in Section 3.2.

Figure 4(b) shows the average satisfaction level in percentage on the y-axis and time step on the x-axis, which are the same as mentioned in the previous section.[4] As shown in the results, all methods were able to achieve at least 80% or higher results on average, and the manual and proactive with MDP settings showed the best results among them. Note that the equations to calculate the individual satisfaction level are based on the average model about the responses according to different environmental conditions, which is mostly related to air temperature, and they do not consider individual preferences. Thus, although the reactive and proactive control strategies act more intelligently by additionally considering the preferences of human occupants, we could not obtain explicit benefits to improve the satisfaction level and even in some cases, the solution quality may be harmed. On the other hand, the manual setting just make HVAC agents attempt to reach the desired temperature set point over time. Once HVAC agents get to the desired point, they are turned off, which will decrease the satisfaction level. If the temperature is again away from the scope of desired temperature point, HVAC agents are turned on and the satisfaction level increases. As a result, the manual setting shows a race condition in the graph, which means it eventually cannot go over a certain point in terms of the satisfaction level. With revised equations considering more factors from the coordination perspective such as preferences, energy awareness, emotional contagion effect, etc., we expect significant improvements in terms of the satisfaction level.

In our work, we still only separately consider two different optimization criteria — the energy consumption and the satisfaction level since this is still preliminary work. However, as we will eventually optimize multiple objectives in SAVES, we will be able to achieve effective multiagent team coordinations to minimize the total energy consumption while maximizing occupant's comfort level.

---

[4] Note that the starting indoor temperature of the building is 55°F in the simulation, which causes the low average satisfaction level for a while.

6 Conclusion

This paper aims to open a new area of research for multiagent systems: in many real-world problems, specifically in energy domains, we see many different levels of agent interactions and coordinations involved, and hence multiagent systems must address such complex situations to achieve the given objectives under uncertainty. In this work, we presented a new framework called SAVES based on distributed coordination reasoning for sustainability. There are three major new ideas in SAVES. SAVES: (i) explicitly considers uncertainty while reasoning about coordination in a distributed manner relying on MDPs; (ii) incorporates human behaviors and their occupancy preferences into planning and models them as part of the system; and (iii) evaluates various control strategies for multiagent teams on an existing university building as the practical research testbed with actual energy consumption data. We justified our design decisions in SAVES through a preliminary empirical evaluation and showed that SAVES can provide solutions to significantly reduce the energy consumption while achieving the comparable satisfaction level of building occupants. For future work, we will consider opportunities for direct occupant participation and incentivization via handheld devices and deploy our system to the real-world.

References

1. Thermal environmental conditions for human occupancy. *ANSI/ASHRAE Standard 55*.
2. Ventilation for acceptable indoor air quality. *ANSI/ASHRAE Standard 62*.
3. S. Abras, S. Ploix, S. Pesty, and M. Jacomino. A multi–agent home automation system for power management. In *ICINCO*, 2006.
4. G. Conte and D. Scaradozzi. Viewing home automation systems as multiple agents systems. In *Multi-agent system for industrial and service robotics applications, RoboCUP*, 2003.
5. J. Dijkstra and H. Timmermans. Towards a multi-agent model for visualizing simulated user behavior to support the assessment of design performance. *Automation in Construction*, 11(2):135 – 145, 2002.
6. M. Fahrioglu and F. L. Alvardo. Designing incentive compatible contracts for effective demand managements. *IEEE Trans. Power Systems*, 15:1255–1260, 2000.
7. Y. Gao, E. Tumwesigye, B. Cahill, and K. Menzel. Using data mining in optimization of building energy consumption and thermal comfort management. In *SEDM*, 2010.
8. Q. Khalid and R. Langhe. Evaluation and monitoring of energy consumption patterns using statistical modeling and simulation: Case study of commercial and residential buildings. In *International Conference on Emerging Technologies (ICET)*, 2010.
9. K. Kim and K. J. Kim. Multi-agent-based simulation system for construction operations with congested flows. *Automation in Construction*, 19(7):867 – 874, 2010.
10. A.-H. Mohsenian-Rad, V. Wong, J. Jatskevich, and R. Schober. Optimal and autonomous incentivebased energy consumption scheduling algorithm for smart grid. In *ISGT*, 2010.
11. L. Perez-Lombard, J. Ortiz, and C. Pout. A review on buildings energy consumption information. *Energy and Buildings*, 40:394–398, 2008.
12. A. H. Rosenfeld, J. J. Romm, H. Akbari, and A. C. Lloyd. Painting the town white – and green. *Technol. Rev.*, 100:52–59, 1997.
13. N. Roy, A. Roy, and S. K. Das. Context-aware resource management in multi-inhabitant smart homes: A nash h-learning based approach. In *PerCom*, 2006.

A formalism to recognise and monitor patient behaviour for intervention in intelligent collaborative care management.

First author, second author
{name.lastname, othername.otherlastname}@insititut.country

Institute, Country

Abstract. By 2020, chronic disease will be the cause for approximately 75% of all deaths worldwide [?]. chronic diseases, once diagnosed, persist through life and require adequate management. Members of the iCare group at EBTIC and at Monash University are investigating Intelligent Collaborative Care Management (ICCM) [?,?] for chronic diseasesto coordinate the are for patients with chronic disease.
ICCM has been compared to Customer Life Cycle Management (CLCM) [?], where it is of critical importance to monitor the patient's behaviour for the occurrence or absence of key events. This article argues for the adaption of an existing approach (used to enable psychological evaluation of game playing behaviour) into the ICCM framework. It draws on models from behavioural psychology, relies on standard formalisms and proposes an automated process to monitor patient's behaviour with the aim to facilitate a timely and appropriate intervention.

1 Introduction

The World Health Organisation (WHO) has released statistics, indicating that by the year 2020, chronic disease will be the cause for approximately 75% of all deaths worldwide [?]. Chronic diseases persist through life and require adequate management. Members of the iCare group at EBTIC in Abu Dhabi, United Arab Emirates and Monash University in Melbourne, Australia have previously been working on an Intelligent Collaborative Care Management (ICCM) project [?,?], focusing on heath care plans for patients with chronic diseases in Australia.

In this context ICCM has been used as Customer Life Cycle Management (CLCM) [?], and it has been pointed out to be of critical importance to monitor the patient's behaviour to recognise the occurrence or absence of key events. 4 Issues have been identified: Monitoring, Recognition, Intervention and the associated Cost [?]. Within the domain of health care, the aspects of recognition and intervention are left to the medical experts, and the matter of cost is not considered at this stage. It should be noted that the people involved in the process of recognition and intervention (i.e. observing and enforcing behaviour change in a health plan) are ranging from doctors, dentists, nurses, nutritionists, dieticians and physiotherapists to health visitors, health promotion practitioners, psychologists and psychiatrists, counselors, health educators and fitness instructors [?].

The formalism we propose is aiming to provide the means to monitor patients and is supposed to be useful to all of the above listed practitioners. Due to this it has to be of broad applicability. The presented formalism for the definition of behaviour is based on accepted standards from the field of behavioural psychology. Implementing it will enable a practitioner to monitor the behaviour of individual patients as well as all patients in a database on the basis of precisely defined behaviour patterns. It will also require the practitioner to adhere to a structure which will allow the cross evaluation and comparison of the defined behaviour by other practitioners who may or may not share the same background.

2 Self management of health and disease

The models presented in this chapter have either been frequently referred to in the literature (specifically: HBM, ToRA, ToPB, [?,?,?,?,?]) or have been used in the ICCM and iCARE project as well as in the health care literature before (mainly: BDI [?,?,?,?]). In what follows we introduce these models and relate them to adherence in self management of health and disease.

2.1 Adherence and non-adherence to health regimens

A study on the adherance rates in (all) published empirical studies from 1948 to 1998 reported on adherence to medical treatment (prescribed by a non-psychiatrist). The study found that the average non-adherence rate was 24.8 % [?]. Roughly a third of scheduled appointments are missed and the deviation from instructions regarding the taking of prescribed medicine is significant [?].

In recent years the investigation of patient's compliance has been widened to encompass an increasing number of activities and actions preformed by the patient. This is because research suggests that the number of factors that play into the decision to follow some self managed health plan is much larger than previously thought and that the scope of the relevant activities does not just restrict itself to the domain of the actual health care or the disease [?].

What drives patients to adhere to some regimen is a complex combination of factors, references to behavioural psychology (which is traditional concerned with the behaviour of ill people) are drawn and this article briefly introduces a number of increasingly complex behavioural models.

One of the important considerations for the suggested approach is to include the influence of peers, a community and the tailored interaction with heath care professionals into the model. A recent study [?] has been dedicated to investigate the inter-group behaviour with respect to health care. While the main focus on the study was on surveying the existing literature and conducted studies with respect to gaps between social groups, its findings include that groups and group behaviour is an important (and well studied) aspect when considering human behaviour with respect to their health care.

The reasons why humans engage in health related activities have been long since classified to belong to one of three categories [?]:

1. Prevention: before there is any symptoms, to avoid health problems at all.
2. Diagnosis: when there are (real or perceived) indications or symptoms for an illness but the actual cause is yet unknown.
3. Treatment: for existing and diagnosed issues.

The first two of the above categories are referring to (potential) patients who are not yet certain about their health problems and the resulting implications for them. In this article, we are only considering the behaviour of patients who have already been diagnosed with a (chronic) disease and who have been provided with a health care plan to assist them in the struggle, i.e. patients who are aware of having an illness which is going to impact their life increasingly; and increasing inversely proportionally to their adherence to a health plan at that.

2.2 The Health Belief Model (HBM) [?]

The Health Belief Model (HBM) shown in Figures 1 and 2 was developed by the US Public health Service in the 1950s [?]. Its initial aim was to understand the non rational, but very common failure of people to partake in early screening tests and undergo preventive treatments for diseases.

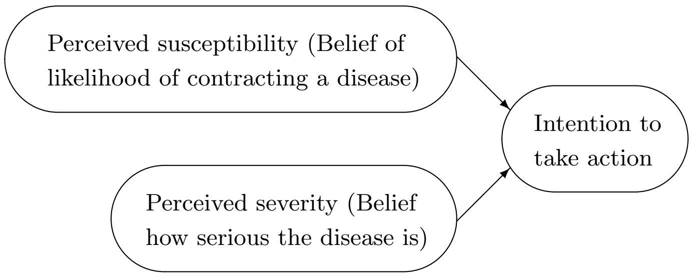


Fig. 1. The Health Belief Model [?] (simplified illustration)

Subsequently it was used to understand the patients' responses to symptoms and to following (prescribed) health plans. In this model, only two considerations (or *beliefs*) are assumed to be forming the basis for human decision making: the subjective value of the outcome of some action A and the belief whether A will bring about that outcome (compare to Figure 3, showing another model). The HBM model does not consider the environment of the patient and does not the actions of others (as mentioned above) other than to the extent of others having an impact by changing these two beliefs. The HBM has been shown to be able to predict patient behaviour reasonable well for a number of behaviours and actions, yet comes short and is inconsistent in the prediction of others [?].

It should be noted that in the domain of health services not all actions can be traced back to attitudes and beliefs, as habitual (e.g. drug abuse) and instinctive (e.g. competing for sexual partners) behaviours can not be modeled in these terms, and are thus not represented in the HBM. These behaviours are generally omitted in the models presented here and the justification for this is the fact that those behaviours are excluded as their recognition is part of the practitioners fields of expertise, which the approach is not aiming to model.

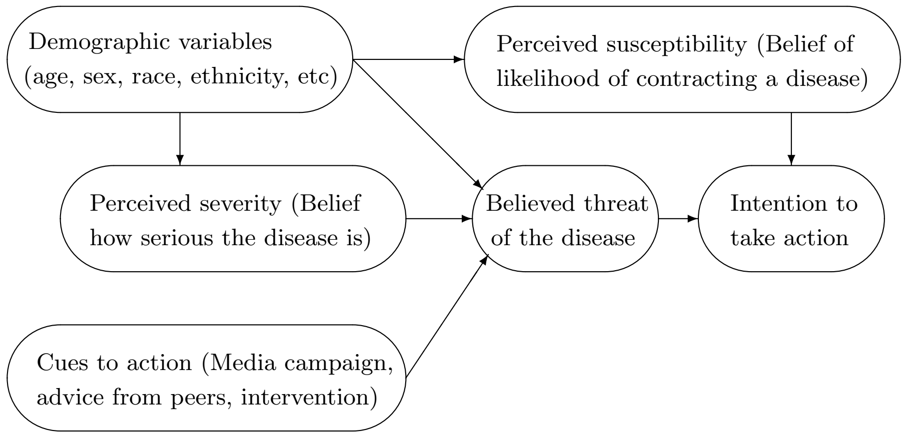


Fig. 2. The Health Belief Model [**?**] (adopted from [**?**])

2.3 Belief, Desire and Intentions (BDI) [?]

The above mentioned HBM was developed by social psychologists and draws on a well established body of psychological and behavioural theory. It uses two concepts which are difficult to define: *Intention* and *Belief*. These terms do appear in this article with a number of interpretations. The reader is advised that the use of these terms is to be understood in the context of the respective section.

In this section we briefly outline another paradigm, namely the BDI paradigm [**?**], deriving its acronym from the triplet of concepts *Belief*, *Desire* and *Intention*. BDI is based on folk psychology, in the sense that it is based on higher level concepts which we as humans consciously recognise. It is not based on unambiguous definitions nor on rigorously gathered empirical evidence; it considers a simplified view on the problem of decision making and behaviour.

This not withstanding, the BDI approach has found merit in computer science (where it is used very successfully as a model for computer agent's future planning) as well as child psychology (especially for work related to autistic children [**?**] lacking recognition of even these simplified concepts in others).

BDI is mentioned here for two reasons: Firstly, because it has been used in work directly related to the Health Care Plan project currently being undertaken by the iCARE group at EBTIC [**?**,**?**], in the context of which the reported research is being undertaken. Reviewing it here and putting it in context with some more developed theories from the fields of social and behavioural psychology seems recommendable to avoid later confusion.

Secondly, and equally important, it does offer a very useful first level of conceptual distinction of aspects related to the process of decision making and behaviour. The Theory of Reasoned Action (ToRA) [?,?,?] and later the Theory of Planned Behaviour (ToPB) [?] (which will provide the core of the formalism) both consider beliefs and intentions, and, to some extent, desires. These theories are far more elaborate, but, previous research [?] (taking the same approach as the one presented here) has found that the level of detail often exceeds the level required. Therefore, and because the proposed approach is to be applied by experts in a practical manner, specific implementations might find that reducing the level of complexity or even reverting to the simplified model of BDI will yield acceptable results and be preferable over the complex models.

As evident from the name, BDI distinguishes 3 concepts:

1. Beliefs: The informational state of the agent i.e. the model of the world as maintained by the software agent as well as how the world evolves and functions (i.e. inference rules).
2. Desires: In case of software agents desires are the states of the world the agent is designed to bring about. Traditionally these include a subset of the desires which at a given moment of time is being considered the priority of the desires (and might contradict other desires). These are called goals.
3. Intentions: As the agent is considered to operate in an environment which is not totally under the agent's control the difference between a desire and an intention needs to be made. An agent which has committed to pursue a specific desire. This requires a plan which the agent has already initiated.

In addition to the above the model recognises are so-called *events*, i.e. changes in the environment or the state of the agent itself which force the modification of a goal or cause a plan to be executed.

2.4 The Theory of Planned Behaviour (ToPB) [?]

In the previous section we have mentioned Bratman's [?] folk psychology, which is sufficient to model software agent's desired (by the programmer) behaviour. We will now discuss monitoring the human patient on the basis of their behaviour and as such we need a more detailed model for human decision making.

In the section below we will briefly outline theories and well-established models for human decision making from the field of behavioural psychology that are considered for our approach.

The theory of Planned Behaviour in psychology, i.e. a theory regarding the link between attitudes and behaviour (proposed by Ajzen in 1985 [?,?]) provides a model for human behaviour that treats the attitude a person has towards some action as a relevant factor in the decision making process whether to execute this action. This theory implies that the attitude someone has towards a certain behaviour influences that persons likeliness to subsequently exhibit that behaviour.

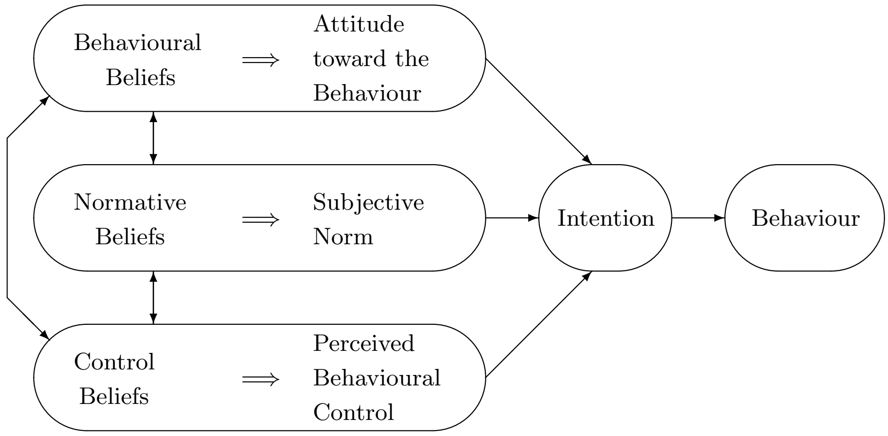


Fig. 3. The Theory of Planned Behaviour (simplified illustration) [?,?]

According to *ToPB* and with respect to actions and behaviour, human decision making is guided by three different considerations and beliefs:

1. Behavioural beliefs: Someone's expectations about the likely outcome of actions, paired with the subjective view on these outcomes.
2. Normative beliefs: Commonly known as *peer pressure*, normative beliefs are the opinion of others regarding the outcomes of actions, the personal intention to adhere to these peer standards as well as the desire of the individual to live up to these expectations of ones peers.
3. Control beliefs: The confidence towards having control over all relevant factors required to bring about an outcome.

Figure 3 (page 6) illustrates this model and how the mentioned beliefs and considerations influence ones intentions and subsequently ones behaviour. This theory has formed the basis for the previous work on the evaluation and the assessment of human game playing behaviour [?,?,?,?,?] referenced in this article.

3 Intelligent Collaborative Care Management (ICCM)

Intelligent collaborative care management aims to provide a unified model for the composition, and management of patient care services. This encompasses 4 phases, namely the design of the health care composition system, the composition of a health plan for an individual patient, the distribution of the elements of that plan to a number of individual service providers and, ultimately, the implementation and management of this plan.

The composition of the health plan for an individual can be done by a human and it can be supported by an expert system / decision support system. Such a system can be checking for consistencies and or used to verify the plan against a legal database (e.g. checking for medication the patient is not be allowed to posses or verifying whether the patient is entitled to certain procedures).

The distribution of the services to the relevant practitioners can be done by software agents [?] with minimal human supervision. When it comes to the management of the health plan, specifically the monitoring of the adherence to the health plan, this process should be automated, i.e. done by a machine and only supervised by a human. The machine will be in charge of the rigorous and complete monitoring, the human expert tasked with intervention planning will be alerted if flags are raised. This however requires a well designed monitoring process, which will depend on the unambiguous definition of critical behaviour, as defined by the human expert for the specific case of an individual patient.

3.1 Backgound: Intelligent collaborative care management

The Intelligent Collaborative Care Management (ICCM) project [?,?] investigates Customer Life Cycle Management (CLCM) settings , focusing on BDI (see Section 2.3) agents that are not always following their designed behaviour. The Intelligent Collaborative Care Management architecture consists of three stages to address the functionalities of customer life cycle management:

1. Care plan design: Developing a care plan by selecting services intended to realise the objectives of service providers and customers;
2. Care plan distribution: Managing contractual relationships between practitioners (service providers) and patients (service customers);
3. Care plan management: Managing the execution of contractual obligations by practitioners and monitoring the care plan adherence of the patients.

The three stages are implemented using two specification layers: a coordination and a task layer. These specification layers define the properties of a CLCM domain. The coordination layer specifies the domain information shared among all the three stages (post-condition of delivering a specific service). The task layer specifies the domain information associated to each stage.

The architectural design for these stages is a non trivial undertaking. We start by providing the specification for a basic customer care model. This model makes some simplifying assumptions and aims to provide a view on the complete set of interactions occurring in the scenario.

This basic model is then extended to remove the mentioned simplifications, thereby increasing in complexity and allowing for advanced concepts:

- **Basic Model of Customer Care.** This model specifies a complete world and system with no uncertainty, incompleteness, and unbounded computational resources. Building this model is complex as it captures interaction and negotiation protocols among diverse types of service providers and customers, as well as constraints during the contractual commitments and service delivery.
- **Extended Model of Customer Care.** This model addresses even greater complexities, where a world is associated with uncertainty, incompleteness,

and bounded resources. It offers recognition and intervention support specifications that are not captured by the basic model. A special case of this model is a world where humans are involved with their own mental attitudes.

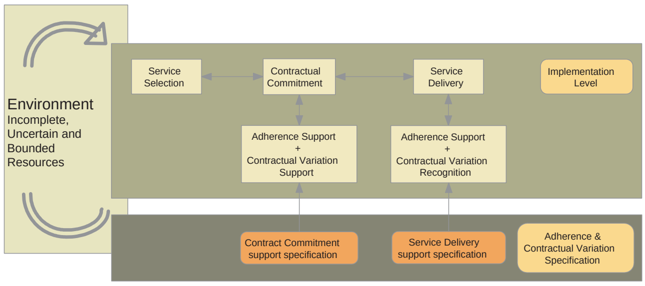


Fig. 4. ICCM architecture: The extended model of customer care with adherence and variation support to cope with uncertainty, incompleteness and bounded resources

3.2 Monitoring, adherence and intervention

The level of complexity varies with the problem and the care plan. While some can be modeled easily, the example given in [**?**] refers to cardiovascular disease and lists disease related factors, the patient, the practitioner, the health service, the policies and the social context as relevant factors, all contributing to the success or the failure of the health plan. It further states that success on one level can be canceled out by failure on another.

Examples of aspects which can be monitored include the consistency which which appointments are made and followed up, which medications are purchased, and of course measurements recorded by the practitioners. Generally, a monitoring system should be able:

1. to monitor providers (automatic).
2. to monitor the patient's adherence to the health plan (automated)
3. to monitor and adapt / amend the plan (partly automated, at least the flagging when thresholds are passed (up or down), also requires a human with access to the health plan and expert knowledge)
4. to monitor for problems unknown to the experts. This can mean:
   - problems of which the individual expert has no knowledge (dentist might not know about stomach problems, a well designed monitoring process would watch out for that),
   - problems resulting from two experts not having access to their respective databases (legal constraints, however a software agent could check without breaching this legal constraint) or
   - problems with a whole population of patients, of which the data would simply be too much to be monitored by a single person (outbreak of some disease, some medication provoking alergic reactions in combination with some medication which has never been observed before, ect)

As pointed out above in Section 1 there is a number of people from a variety of backgrounds which can be involved in an intervention. A viable approach must provide sufficient freedom to all of them if it is expected to be accepted by the practitioners in the field. As far as interventions to behaviour deviating from the prescribed health care plan are concerned, there are more than one theory on how to successfully implement an intervention (cf. e.g. [**?**]) and it is not our aim to suggest one over the other. In the following section we will outline a formalism to state behaviour in the context of a health care plan. Such a formalism will enable a practitioners from a variety of backgrounds to define behaviour which they consider warning signs for insufficient adherence to a health plan. The system we propose is purely aiming at the automated monitoring of patient behaviour coupled with a mechanism to alert the right experts, who will then be tasked with the implementation of an intervention.

4 Defining, evaluating and monitoring behaviour

The caretaker's ability to monitor the patient's behaviour is an important element in the diagnosis of the patient and supervision of any rehabilitation process. One of the drawbacks of using questionnaires when investigating human behaviour is the subjective nature of introspective statements [**?**].

4.1 Defining behaviour

With respect to observable aspects of behaviour we continue [**?**,**?**,**?**] to use the works of Ajzen as reference point, specifically the TACT (Target, Action, Context and Time) paradigm that was suggested for the design and the evaluation of questionnaires (within the context of ToPB related research). We motivate our choice by the fact that the ToPB has been *"the explicit theoretical basis for 222 studies published in the Medline database, and 610 studies published in the PsycINFO database, from 1985 to January 2004."* [**?**].

In his work [**?**] Ajzen argues that in order to define behaviour sufficiently the above mentioned four aspects have to be distinguished and identified in (introspective) statements regarding behaviour. His running example is *"walking on a treadmill in a physical fitness centre for at least 30 minutes each day in the forthcoming month"*. It is not always clear how to distinguish between the four aspects, a matter to which we will return further below. The labelling of behaviour as well as the distinction of which of the four aspects to assign to a part of a statement describing behaviour is subjective and therefore has to be decided upon by a medical expert during the composition phase.On a higher level abstract behavioural monitoring plans can be designed in advance and offered to the expert during the composition level.

The subjectivity of defining which behaviour should be observed, and how (and to which extend) opens the door for ambiguity. However, we are not making any claims towards providing an objective way to assess behaviour and, as a matter of fact, do not intend this to be understood as a subjective means to monitor *all* patients. The medical expert composing the health care plan should be familiar with the patient. Consider the case where the patient is known to be someone who has in the past strictly followed a health regimen, but happens to live in the outback or has a job that will make regular attendance of monitored events at fixed times impossible or unlikely. Such a person would not be checked for attending rehabilitation regularly. However, a patient with a known strong tendency to drop out of health care plans should be monitored very closely on regular attendance. In this example the requirement for the second subject to be on time might actually be seen as another, additional, treatment. Generally, heath care is always to a certain extend a subjective matter, as it is based on the subjective impression of the provider and the subject under consideration.

However, in the context of this paper this is not of much relevance. We merely provide well-defined and consistent means to the assess behaviour. The classification, just like the eventual interpretation of the collected data, will remain the task of the person investigating the behaviour. Ajzen himself points out that there is this ambiguity and that there are many possible additions to the basic TACT paradigm as proposed by him originally (e.g. *"within next month"* can include *"next Tuesday"*). Within the scope of this paper the 4 TACT aspects suffice, however, we point out that we neither expect these 4 nor the TACT approach itself to be the optimal solution to all monitoring requirements possible. As mentioned above, there are a number of theories in the field of health care related psychology (some of which we mentioned) and the adopted theory and paradigm will depend on the specific focus of the application, the Theory of Planned Behaviour is our personally preferred choice, but it can be replaced by the one the experts prefer. We pointed out earlier in Section 2.3 that the model chosen should only be as complex as required by the application. If the experts can agree on a simpler model than this is the one which should be selected. It is important to have a solid basis for the proposed description of behaviour, which specific theory is chosen is open for discussion.

The specifics of the project for which the formalism is eventually used will determine the extent to which a finer grained distinction if TACT (or indeed a different paradigm) is required; one of the first tasks in the design stage of a project is for the designed, the expert composing the health plan and the practitioner in the field to liaise on that matter and to reach a consensus. Furthermore, complicated extensions will complicate the formalism presented in this paper without adding value to the conceptual approach and are therefore omitted here. For demonstration purposes the presented level will suffice and for more demanding requirements the extensions might become very specific and complex, but not conceptually different. Therefore we argue that the extend of the presented material suffices for this article.

4.2 Formally stating and evaluating behaviour

Propositional logic (PL) is the logic of propositions, meaning it is only concerned with statements and facts which can be decided to be either true or false. In our context these are entries in the database. Using the usual [**?**] connectives *not*, *and*, *or*, *if ... then* and *if, and only if* we define a language that allows us the expression of complex statements over these entries in a database.

On a relational model we have a succession of *states*, each representing a set of propositional statements and thus model temporal statements . In the context of a health care database we can then make statement relating to both a temporal ordering of events (on the basis of the time stamp of an entry) as well as to frequency and spread of events (on the basis of the date of the entry).

As far as the usability of such a formal basis is concerned it should be noted that there is an algorithm which automatically translates the formal statement into natural language (and vice versa). There is of course a limit on how complex these sentences can be before they become very hard to read [**?**] but this limitation is not imposed by the algorithm.

With respect to the monitoring of behaviour through a database, i.e. the verification process of a formula under a specific valuation the following can be said: The process of investigating whether a statement about a patient is true, given a database, can be automated. The time this will take increases linearly with the length of the statement [**?**], however the time it takes to check whether a specific statement does make sense at all increases exponentially (worst case) with the length of the statement. The latter will not be of importance to us and used, if at all, only during the generation of a monitoring event. This indicates that the complexity of the approach is well within the feasible limits.

4.3 Automating the monitoring of patient behaviour

The justification to formalise behaviour with respect to health care is on the basis of two considerations: Firstly, there is a subjective bias, which the individual patient will inevitably exhibit (being an individual with his / her own view on the world). It does not matter whether the patient is actively trying to deceive the health care official or whether it is happening subconsciously, if the patient is asked to report on his or her behaviour a certain degree of subjective bias has to be assumed. The second reason for investigating a formalism to describe and subsequently assess human behaviour is the fact that a health plan will include a number of practitioners which means that the patient's behaviour, even if it was observed, would be observed by a number of practitioners who can not be burdened to report and collaborate on their personal experiences with each individual patient. The traces a patient leaves when executing a health care plan can be used as basis for the evaluation of his / her behaviour.

Observing patient behaviour The Intelligent Collaborative Care Management (ICCM) project [**?**,**?**] calls for a central database to which all practitioners involved in an individual patient's care plan submit certain updates. We assume therefore that the relevant aspects of the patient's behaviour are already recorded in a database and can be subjected to queries.

With respect to these queries two tasks can be distinguished:

1. To define which behaviour we are interested in, and

2. to enable the translation of such a behavioural statement into a query.

In addition, we add another requirement to the two tasks above, namely that the described behaviour is stated in a manner that will allow a precise and unambiguous evaluation. The person in charge of describing the behaviour might not have received formal psychological training or might not be aware of the underlying theory used in our approach. As a matter of fact it will have to be expected that a number of people tasked with the design of monitoring queries will have different views on the matter of *how* to describe behaviour.

Therefore, we will reference to one of the aforementioned theories from behavioural psychology, namely the theory of planned behaviour [**?**] (see Section 2.4), and use it as the basis for our formalism. The claim made here is not that this is indeed the only right theory, but only that this is *a* theory which has been investigated extensively in the field (cf. [**?**,**?**,**?**,**?**,**?**,**?**]) and as such can be used as an example here. it will eventually be up to the practitioners tasked with the design of this system to make the final decision on which theory they want to base their monitoring processes on.

Using the work of Ajzen, we then can make use of the same author's work on questionnaire design [**?**] and adopt his TACT paradigm mentioned in Section 4.1. As detailed in [**?**] (for the evaluation of cooperative versus competitive behaviour, see below), complex behaviour statements can be formulated such that they can be broken down to individual boolean queries, i.e. atomic statements which can be verified to be either true or false. The process can be implemented efficiently enough to be used by applications running on mobile phones [**?**,**?**] and thus lends itself for the use as a server side based monitoring system.

3 exemplary scenarios We briefly outline 3 possible scenarios where the approach could be used. These are contrived but realistic, for more specific examples, including the seperation of high level / low level behaviours and the infinite nesting of statements the reader is referred to [**?**,**?**].

1. A patient X in a rehabilitation center. In this setting the patient is in a clinic for a duration of time during which he / she is expected to partake in a number of exercises. These are staged and supervised by a number of practitioners which may not have the access rights to few each others files on the patient. The files are considered to be stored in some central database which can be accessed by the monitoring system controlled by a different practitioner, tasked with the organising of interventions. Exemplary behaviours which could be expressed in the formalism are (e.g.):
   - **A**: X has reported to the exercises (`action`) for cardiac monitoring (`context`) for at least 2 sessions per day (`target`) in the last week (`time`).
   - **B**: X has completed at least 4 (`target`) swimming exercises (`action`) under supervision (`context`) in the last week (`time`).
   - **C**: X has exhibited behaviour (`action`) **A** or **B** (`target`) in clinic (`context`) this each week (`time`).

2. A patient Y is monitored for general aversion to adhere to health care appointments, maybe because the insurance suspects Y to be skiving or maybe because his / her GP is worried about some general aversion to health care exercises.
   – **D**: Y has realised (`action`) his dentist appointment (`target`) on wednesday (`time`) to have plaque removed (`context`).
   – **E**: Y was present (`action`) during doctor's visit (`target`) for checkup (`context`) on monday (`time`).
   – **F**: Y has had an ECG (`context`) done (`action`) in GP's clinic (`target`) on a day last week (`time`).
   – **G**: Y has exhibited (`action`) at least 2 (`target`) of the behaviours **D** or **E** or **F** (`context`) in the last week (`time`).
3. An elderly patient Z is partaking in a trial for which an application is installed in his mobile phone (e.g. iPhone). There is some central care unit which is enabled to monitor the overall day to day behaviour of Z:
   – **H**: Z has been active (moving) `action`) for at least 30 minutes (`target`) in the morning (0500h-1200h) (`context`) and in the afternoon (1201h-2030h) (`context`).
   – **I**: Z has been active outside (`action`) the apartment (`target`) for a total of at least 120 minutes (`context`) today (`time`).
   – **J**: Z has done either **H** or **J** (`action`) on at least 6 days (`target`) per week (`context`) during the last month (`time`).

Proof of concept: investigating human behaviour in computer games The proposed formalism has previously been investigated in the context of serious games [**?**] where a prototype computer game has been used to show the computational feasibility of the approach. The prototype implementation (for mobile devices) produced data which was meeting the requirements set out in [**?**] and could be evaluated completely automatically. As a side result it was shown that the approach did not just enable the evaluation of individual behaviour but that overlapping behaviours patterns could be identified for large groups of participants. The drawback of the proof of concept application is that it was implemented to show the feasibility of the approach from a computational point of view, for a proper psychological investigation the participation of experts from that field would be required. Analogously, this articles suggestion to use the formalism in the monitoring of patient behaviour can only be shown to be computationally feasible and according to standards in the field of health care. The actual use and thus verification is a matter that relies on the inclusion of the ongoing ICCM project.

5 Conclusion

This article argues for the adaption of an existing approach (used to enable psychological evaluation of game playing behaviour) into the ICCM framework. We have provided an argument for the need for intervention and supported the claim that intervention can only be on the basis of monitoring the patient's behaviour. We have provided the reader with a brief overview of models used in the health care and behavioural psychology literature. We have referenced to the formal definition of behaviour given in the context of our preferred model and subsequently used it for a formalism which can be applied to a health care database. The claim that this can be implemented has been supported by previous research into a similar application using the same formalism. Finally we have provided the reader with examples of scenarios where such a monitoring would be of use and have given exemplary behaviour within these scenarios.

A Delegation-Based Collaborative Robotic Framework ⋆

Patrick Doherty, Fredrik Heintz, and David Landén
Dept. of Computer and Information Science, Linköping University,
581 83 Linköping, Sweden
{patdo, frehe, davla}@ida.liu.se

Abstract. Collaborative robotic systems, such as unmanned aircraft systems, are becoming technologically mature enough to be integrated into civil society. To gain practical use and acceptance, a verifiable, principled and well-defined foundation for interactions between human operators and autonomous systems is needed. In this paper, we propose and specify such a formally grounded collaboration framework. Collaboration is formalized in terms of the concept of delegation and delegation is instantiated as a speech act. Task Specification Trees are introduced as both a formal and pragmatic characterization of tasks and tasks are recursively delegated through a delegation process. The delegation speech act is formally grounded in the implementation using Task Specification Trees, task allocation via auctions and distributed constraint solving. The system is implemented as a prototype on Unmanned Aerial Vehicle systems and a case study targeting emergency service applications is presented.

1 Introduction

Collaborative robotic systems, such as unmanned aircraft systems, are becoming technologically mature enough to be integrated into civil society. To gain practical use and acceptance, a verifiable, principled and well-defined foundation for interactions between human operators and autonomous systems is needed. This interaction is going to be mixed-initiative in nature. Humans will request help from robotic systems and robotic systems will request help from humans when collaborating to achieve complex missions in unstructured and challenging environments. In developing a principled framework for such sophisticated interactions, many interdependent conceptual and pragmatic issues arise and need clarification both theoretically and pragmatically.

The complexity of developing deployed architectures for realistic collaborative activities among robots that operate in the real world under time and space constraints is very high. We tackle this complexity by working both abstractly at a formal logical level and concretely at a systems building level. More importantly, the two approaches are related to each other by grounding the formal abstractions into actual software implementations. This guarantees the fidelity of the actual system to the formal specification.

This paper presents a principled formal framework for collaborative robotic systems based on delegation. The basis for the principled framework for interaction between

⋆ This work is partially supported by grants from the Swedish Research Council (VR) Linnaeus Center CADICS, VR grant 90385701, the ELLIIT Excellence Center at Linkoping-Lund for Information Technology, NFFP5-The Swedish National Aviation Engineering Research Program, and the Center for Industrial Information Technology CENIIT.

human operators and robotic systems is a triad of fundamental, interdependent conceptual issues: delegation, mixed-initiative interaction and adjustable autonomy. These concepts are used to clarify, validate and verify different types of interaction between robotic platforms and human operators. The concept of delegation is particularly important and provides in a sense a bridge between mixed-initiative interaction and adjustable autonomy.

Delegation – In any mixed-initiative interaction, humans may request help from robotic systems and robotic systems may request help from humans. One can model such requests as a form of delegation, $Delegate(A, B, task, constraints)$, where A is the delegating agent, B is the contractor, $task$ is the task being delegated which consists of a goal and possibly a plan to achieve the goal, and $constraints$ represents a context in which the request is made and the task should be carried out.

Adjustable Autonomy – In solving tasks in a mixed-initiative setting, the robotic systems involved will have a potentially wide spectrum of autonomy, yet should only use as much autonomy as is required for a task and should not violate the degree of autonomy mandated by a human operator. One can begin to develop a principled means of adjusting autonomy through the use of the $task$ and $constraint$ parameters in $Delegate$. A task delegated with only a goal and no plan, with few constraints, allows the robot to use much of its autonomy in solving the task, whereas a task specified as a sequence of actions and many constraints allows only limited autonomy.

Mixed-Initiative Interaction – By mixed-initiative, we mean that interaction and negotiation between a robotic system, such as a UAV, and a human will take advantage of each of their skills, capacities and knowledge in developing a mission plan, executing the plan and adapting to contingencies during the execution of the plan. Mixed-initiative interaction involves a very broad set of issues, both theoretical and pragmatic. One central part of such interaction is the ability of a ground operator (GOP) to be able to delegate tasks to a UAV, $Delegate(GOP, UAV, task, constraints)$ and in a symmetric manner, the ability of a UAV to be able to delegate tasks to a GOP, $Delegate(UAV, GOP, task, constraints)$. Issues pertaining to safety, security, trust, etc., have to be dealt with in the interaction process and can be formalized as particular types of constraints associated with a delegated task.

2 Delegation as a Speech Act

Delegation is central to the conceptual and architectural framework we propose. Consequently, formulating an abstraction of the concept with a formal specification amenable to pragmatic grounding and implementation in a software system is paramount. As a starting point, Falcone & Castelfranchi provide an illuminating, but informal discussion about delegation as a concept from a social perspective [4, 10]. Their approach to delegation builds on a BDI model of agents, that is, agents having beliefs, goals, intentions, and plans [5]. However, their specification lacks a formal semantics for the operators used. Based on intuitions from their work, we have previously provided a formal characterization of their concept of strong delegation using a communicative speech act with pre- and post-conditions which update the belief states associated with the delegator and contractor, respectively [9]. Strong delegation means that the delegation is explicit.

The formal characterization of the speech act is expressed in KARO [12]. The KARO formalism is an amalgam of dynamic logic and epistemic/doxastic logic, augmented with additional modal operators to deal with the motivational aspects of agents.

The target for delegation is a *task*. The Merriam-Webster dictionary definition of a task is "a usually assigned piece of work often to be finished within a certain time". Assigning a piece of work to someone by someone is in fact what delegation is about. In computer science, a *piece of work* in this context is generally represented as a composite action. There is also often a purpose to assigning a piece of work to be done. This purpose is generally represented as a *goal*, where the intended meaning is that a task is a means of achieving a goal. We will require both a formal specification of a task at a high-level of abstraction in addition to a more data-structural specification flexible enough to be used pragmatically in an implementation. For the formal specification, the definition provided by Falcone & Castelfranchi will be used. For the data-structure specification used in the implementation, Task Specification Trees will be defined in a Section 3.

Falcone & Castelfranchi define a task as a pair $\tau = (\alpha, \phi)$ consisting of a goal ϕ, and a plan α for that goal, or rather, a plan and the goal associated with that plan. Conceptually, a plan is a composite action. At this abstraction level, the definition of a task is purposely left general. For instance, timing and resource issues are abstracted away although they will be dealt with explicitly in the implementation.

From the perspective of adjustable autonomy, the task definition is quite flexible. If α is a single elementary action with the goal ϕ implicit and correlated with the post-condition of the action, the contractor has little flexibility as to how the task will be achieved. On the other hand, if the goal ϕ is specified and the plan α is not provided, then the contractor has a great deal of flexibility in achieving the goal. There are many variations between these two extremes and these variations capture the different levels of autonomy and trust exchanged between two agents.

Paraphrasing Falcone & Castelfranchi into KARO terms, we consider a notion of strong delegation represented by a speech act $S-Delegate(A, B, \tau)$ of A delegating a task $\tau = (\alpha, \phi)$ to B, where α is a possible plan and ϕ is a goal.

Preconditions:
(1) $Goal_A(\phi)$
(2) $Bel_A Can_B(\tau)$ (Note that this implies $Bel_A Bel_B(Can_B(\tau)))$
(3) $Bel_A(Dependent(A, B, \tau))$
(4) $Bel_B Can_B(\tau)$

Postconditions:
(1) $Goal_B(\phi)$ and $Bel_B Goal_B(\phi)$
(2) $Committed_B(\alpha)$ (also written $Committed_B(\tau)$)
(3) $Bel_B Goal_A(\phi)$
(4) $Can_B(\tau)$ (and hence $Bel_B Can_B(\tau)$, and by (1) also $Intend_B(\tau)$)
(5) $Intend_A(do_B(\alpha))$
(6) $MutualBel_{AB}$("the statements above" $\wedge$ $SociallyCommitted(B, A, \tau))$[1]

This particular characterization of delegation follows Falcone & Castelfranchi closely. One can easily foresee other constraints one might add or relax in respect to the basic specification resulting in other variants of delegation [6, 7].

[1] A discussion pertaining to the semantics of non-KARO modal operators may be found in [9].

3 Task Specification Trees

Both the declarative and procedural representation and semantics of tasks are central to the delegation process. The relation between the two representations is also essential if one has the goal of formally grounding the delegation process in the system implementation. A task was previously defined abstractly as a pair (α, ϕ) consisting of a composite action α and a goal ϕ. In this section, we introduce a formal task specification language which allows us to represent tasks as *Task Specification Trees* (TST's).

For our purposes, the task representation must be highly flexible, sharable, dynamically extendible, and distributed in nature. Tasks need to be delegated at varying levels of abstraction and also expanded and modified because parts of complex tasks can be recursively delegated to different robotic agents which are in turn expanded or modified. Consequently, the structure must also be distributable. Additionally, a task structure is a form of compromise between an explicit plan in a plan library at one end of the spectrum and a plan generated through an automated planner [15, 14] at the other end of the spectrum. The task representation and semantics must seamlessly accommodate plan representations and their compilation into the task structure. Finally, the task representation should support the adjustment of autonomy through the addition of constraints or parameters by agents and human resources.

The task specification formalism should allow for the specification of various types of task compositions, including sequential and concurrent, in addition to more general constructs such as loops and conditionals. The task specification should also provide a clear separation between tasks and platform specific details for handling the tasks. The specification should focus on what should be done and hide the details about how it could be done by different platforms.

In the general case, A TST is a declarative representation of a complex multi-agent task. In the architecture realizing the delegation framework a TST is also a distributed data structure. Each node in a TST corresponds to a task that should be performed. There are six types of nodes: sequence, concurrent, loop, select, goal, and elementary action. All nodes are directly executable except goal nodes which requires some form of expansion or planning to generate a plan for achieving the goal.

Each node has a *node interface* containing a set of parameters, called *node parameters*, that can be specified for the node. The node interface always contains a platform assignment parameter and parameters for the start and end times of the task, usually called P, T_S and T_E. These parameters can be part of the constraints associated with the node called *node constraints*. A TST also has *tree constraints*, expressing precedence and organizational relations between the nodes in the TST. Together the constraints form a constraint network covering the TST. In fact, the node parameters function as constraint variables in a constraint network, and setting the value of a node parameter constrains not only the network, but implicitly, the degree of autonomy of an agent.

Consider a small scenario where the mission is to first scan $\mathsf{Area_A}$ and $\mathsf{Area_B}$, and then fly to $\mathsf{Dest_4}$. A TST describing this mission is shown in Figure 1. Nodes N_0 and N_1 are composite action nodes, sequential and concurrent, respectively. Nodes N_2, N_3 and N_4 are elementary action nodes. Each node specifies a task and has a node interface containing node parameters and a platform assignment parameter. In this case only temporal parameters are shown representing the intervals tasks should be completed in.

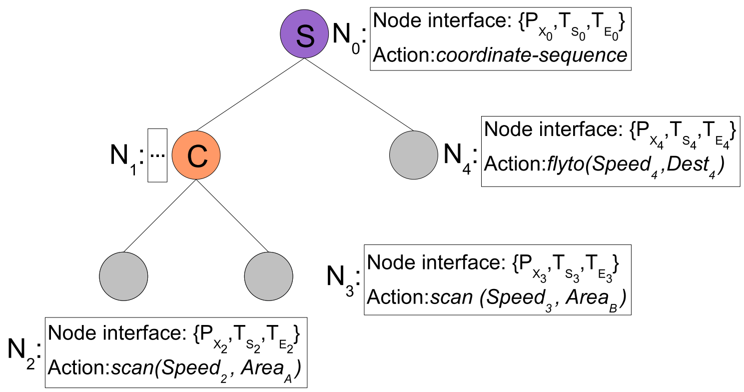


Fig. 1. An example TST.

3.1 TST Syntax

The syntax of a TST specification has the following BNF:

TST ::= NAME ('(' VARS ')')? '=' (**with** VARS)? TASK (**where** CONS)?
TSTS ::= TST | TST ';' TSTS
TASK ::= <elementary action> | <goal> | **sequence** TSTS | **concurrent** TSTS
while <cond> TST | **if** <cond> **then** TST **else** TST
VAR ::= <var name> | <var name> '.' <var name>
VARS ::= VAR | VAR ',' VARS
CONSTRAINT ::= <constraint>
CONS ::= CONSTRAINT | CONSTRAINT **and** CONS
ARG ::= VAR | <value>
ARGS ::= ARG | ARG ',' ARGS
NAME ::= <node name>

Where <elementary action> is an elementary action $name(p_0, ..., p_N)$, <goal> is a goal $name(p_0, ..., p_N)$, $p_0, ..., p_N$ are parameters, and <cond> is a FIPA ACL query message requesting the value of a boolean expression..

The TST clause introduces the main recursive pattern. The right hand side of the equality provides the general pattern of providing a variable context for a task (using **with**) and a set of constraints (using **where**) over the variables previously introduced.

Example: Consider the TST depicted in Figure 1. The nodes N_0 to N_4 have the task names τ_0 to τ_4 associated with them. This TST contains two composite actions, *sequence* (τ_0) and *concurrent* (τ_1), and two elementary actions, *scan* (τ_2, τ_3) and *flyto* (τ_4).

τ_0(T_{S_0},T_{E_0}) =
with $T_{S_1}, T_{E_1}, T_{S_4}, T_{E_4}$ **sequence**
τ_1(T_{S_1},T_{E_1}) =
with $T_{S_2}, T_{E_2}, T_{S_3}, T_{E_3}$ **concurrent**
τ_2(T_{S_2},T_{E_2}) = scan(T_{S_2},T_{E_2},$Speed_2$,$Area_A$);

$\tau_3(T_{S_3},T_{E_3})$ = scan(T_{S_3},T_{E_3},$Speed_3$,$Area_B$)
where $cons_{\tau_1}$;
$\tau_4(T_{S_4},T_{E_4})$ = flyto(T_{S_4},T_{E_4},$Speed_4$,$Dest_4$)
where $cons_{\tau_0}$

$cons_{\tau_0} = T_{S_0} \leq T_{S_1} \wedge T_{S_1} < T_{E_1} \wedge T_{E_1} \leq T_{S_4} \wedge T_{S_4} < T_{E_4} \wedge T_{E_4} \leq T_{E_0}$
$cons_{\tau_1} = T_{S_1} \leq T_{S_2} \wedge T_{S_2} < T_{E_2} \wedge T_{E_2} \leq T_{E_1} \wedge T_{S_1} \leq T_{S_3} \wedge T_{S_3} < T_{E_3} \wedge T_{E_3} \leq T_{E_1}$

3.2 TST Semantics

A TST specifies a complex task (composite action) under a set of tree-specific and node-specific constraints which together are intended to represent the context in which a task should be executed in order to meet the task's intrinsic requirements, in addition to contingent requirements demanded by a particular mission. The leaf nodes of a TST represent elementary actions used in the definition of the composite action the TST represents and the non-leaf nodes essentially represent control structures for the ordering and execution of the elementary actions. The semantic meaning of non-leaf nodes is essentially application independent, whereas the semantic meaning of the leaf nodes are highly domain dependent. They represent the specific actions or processes that an agent will in fact execute. The procedural correlate of a TST is a program.

During the delegation process, a TST is either provided or generated to achieve a specific set of goals, and if the delegation process is successful, each node is associated with an agent responsible for the execution of that node.

Informally, the semantics of a TST node will be characterized in terms of whether an agent believes it *can* successfully execute the task associated with the node in a given context represented by constraints, given its capabilities and resources. This can only be a belief because the task will be executed in the future and even under the best of conditions, real-world contingencies may arise which prevent the agent from successfully completing the task. The formal semantics for TST nodes will be given in terms of the logical predicate $Can()$ which we have used previously in the formal definition of the S-Delegate speech act, although in this case, we will add additional arguments. This is not a coincidence since our goal is to ground the formal specification of the S-Delegate speech act into the implementation in a very direct manner.

Recall that in the formal semantics for the speech act S-Delegate (described in Section 2), the logical predicate $Can_X(\tau)$ is used to state that an agent X has the capabilities and resources to achieve task τ. An important precondition for the successful application of the speech act is that the delegator (A)believes in the contractor's (B) ability to achieve the task τ, (2): $Bel_A Can_B(\tau)$. Additionally, an important result of the successful application of the speech act is that the contractor actually has the capabilities and resources to achieve the task τ, (4): $Can_B(\tau)$. In order to directly couple the semantic characterization of the S-Delegate speech act to the semantic characterization of TST's, we will assume that a task $\tau = (\alpha, \phi)$ in the speech act characterization corresponds to a TST. Additionally, the TST semantics will be characterized in terms of a Can predicate with additional parameters to incorporate constraints.

In this case, the Can predicate is extended to include as arguments a list $[p_1, \ldots, p_k]$ denoting all node parameters in the node interface together with other parameters provided in the (**with** VARS) construct[2] and an argument for an additional constraint set $cons$ provided in the (**where** CONS) construct.[3] Observe that $cons$ can be formed incrementally and may in fact contain constraints inherited or passed to it through a recursive delegation process. The formula $Can(B, \tau, [t_s, t_e, \ldots], cons)$ then asserts that an agent B has the capabilities and resources for achieving task τ if $cons$, which also contains node constraints for τ, is consistent. The temporal variables t_s and t_e associated with the task τ are part of the node interface which may also contain other variables which are often related to the constraints in $cons$.

Determining whether a fully instantiated TST satisfies its specification, will now be equivalent to the successful solution of a constraint problem in the formal logical sense. The constraint problem in fact provides the formal semantics for a TST. Constraints associated with a TST are derived from a reduction process associated with the $Can()$ predicate for each node in the TST. The generation and solution of constraints will occur on-line during the delegation process. Let us provide some more specific details. In particular, we will show the very tight coupling between the TST's and their logical semantics.

The basic structure of a Task Specification Tree is:

TST ::= NAME ('(' VARS$_1$ ')')? '=' (**with** VARS$_2$)? TASK (**where** CONS)?

where VARS$_1$ denotes node parameters, VARS$_2$ denotes additional variables used in the constraint context for a TST node, and CONS denotes the constraints associated with a TST node. Additionally, TASK denotes the specific type of TST node. In specifying a logical semantics for a TST node, we would like to map these arguments directly over to arguments of the predicate $Can()$. Informally, an abstraction of the mapping is

$$Can(agent_1, TASK, VARS_1 \cup VARS_2, CONS) \tag{1}$$

The idea is that for any fully allocated TST, the meaning of each allocated TST node in the tree is the meaning of the associated $Can()$ predicate instantiated with the TST specific parameters and constraints. The meaning of the instantiated $Can()$ predicate can then be associated with an equivalent Constraint Satisfaction Problem (CSP) which turns out to be true or false dependent upon whether that CSP can be satisfied or not. The meaning of the fully allocated TST is then the aggregation of the meanings of each individual TST node associated with the TST, in other words, a conjunction of CSP's.

One would also like to capture the meaning of partial TST's. The idea is that as the delegation process unfolds, a TST is incrementally expanded with additional TST nodes. At each step, a partial TST may contain a number of fully expanded and allocated nodes in addition to other nodes which remain to be delegated. In order to capture this process semantically, one extends the semantics by providing meaning for an unallocated TST node in terms of both a $Can()$ predicate and a $Delegate()$ predicate:

$$\exists agent_2\, Delegate(agent_1, agent_2, TASK, VARS_1 \cup VARS_2, CONS) \tag{2}$$

[2] For reasons of clarity, we only list the node parameters for the start and end times for a task, $[t_s, t_e, \ldots]$, in this article.

[3] For pedagogical expediency, we can assume that there is a constraint language which is reified in the logic and is used in the CONS constructs.

Either $agent_1$ can achieve a task, or (exclusively) it can find an agent, $agent_2$, to which the task can be delegated. In fact, it may need to find one or more agents if the task to be delegated is a composite action.

Given the $S\text{-}Delegate(agent_1, agent_2, TASK)$ speech act semantics, we know that if delegation is successful then as one of the postconditions of the speech act, $agent_2$ can in fact achieve $TASK$ (assuming no additional contingencies):

$$\begin{aligned} &Delegate(agent_1, agent_2, TASK, VARS_1 \cup VARS_2, CONS) \\ &\quad \to Can(agent_2, TASK, VARS_1 \cup VARS_2, CONS) \end{aligned} \tag{3}$$

Consequently, during the computational process associated with delegation, as the TST expands through delegation where previously unallocated nodes become allocated, each instance of the $Delegate()$ predicate associated with an unallocated node is replaced with an instance of the $Can()$ predicate. This recursive process preserves the meaning of a TST as a conjunction of instances of the $Can()$ predicate which in turn are compiled into a (interdependent) set of CSPs and which are checked for satisfaction using distributed constraint solving algorithms.

Sequence Node For a *sequence node*, the child nodes should be executed in sequence, from left to right, during the execution time of the sequence node.

$Can(B, S(\alpha_1, ..., \alpha_n), [t_s, t_e, \ldots], cons) \leftrightarrow$
$\exists t_1, \ldots, t_{2n}, \ldots \bigwedge_{k=1}^{n}(Can(B, \alpha_k, [t_{2k-1}, t_{2k}, \ldots], cons_k)$
$\quad \vee \exists a_k Delegate(B, a_k, \alpha_k, [t_{2k-1}, t_{2k}, \ldots], cons_k))$
$\wedge\, consistent(cons)$[4]
where $cons = \{t_s \leq t_1 \wedge (\bigwedge_{i=1}^{n} t_{2i-1} < t_{2i}) \wedge (\bigwedge_{i=1}^{n-1} t_{2i} \leq t_{2i+1}) \wedge t_{2n} \leq t_e\} \cup cons'$

Concurrent Node For a *concurrent node*, the child nodes should be executed during the time interval of the concurrent node.

$Can(B, C(\alpha_1, ..., \alpha_n), [t_s, t_e, \ldots], cons) \leftrightarrow$
$\exists t_1, \ldots, t_{2n}, \ldots \bigwedge_{k=1}^{n}(Can(B, \alpha_k, [t_{2k-1}, t_{2k}, \ldots], cons_k)$
$\quad \vee \exists a_k Delegate(B, a_k, \alpha_k, [t_{2k-1}, t_{2k}, \ldots], cons_k))$
$\wedge\, consistent(cons)$
where $cons = \{\bigwedge_{i=1}^{n} t_s \leq t_{2i-1} < t_{2i} \leq t_e\} \cup cons'$.

Observe that the constraint sets $cons_k$ in the semantics for the concurrent and sequential nodes are simply the constraint sets defined in the (**where** CONS) constructs for the child nodes included with the sequential or concurrent nodes, respectively. Additionally, the definition of the constraint set $cons$ in the semantics for the concurrent and sequential nodes contains the structural temporal constraints which define sequence and concurrency, respectively, together with possibly additional constraints, denoted by $cons'$ that one may want to include in the constraint set. Note also, that we are assuming that scoping and overloading issues for variables in embedded TST structures are dealt with appropriately in the recursive expansion of the $Can()$ predicates in the definitions.

Selector Node Compared to a sequence or concurrent node, only one of the *selector node*'s children will be executed, which one is determined by a test condition in the selector node. The child node should be executed during the time interval of the selector

[4] The predicate $consistent()$ has the standard logical meaning and checking for consistency would be done through a call to a constraint solver which is part of the architecture.

node. A selector node is used to postpone a choice which can not be known when the TST is specified. When expanded at runtime, the net result can be any of the node types.

Loop Node A *loop node* will add a child node for each iteration the loop condition allows. In this way the loop node works as a sequence node but with an increasing number of child nodes which are dynamically added. Loop nodes are similar to selector nodes, they describe additions to the TST that can not be known when the TST is specified. When expanded at runtime, the net result is a sequence node.

Goal A *goal node* is a leaf node which can not be directly executed. Instead it has to be expanded by using an automated planner or related planning functionality. After expansion, a TST branch representing the generated plan is added to the original TST.

$Can(B, Goal(\phi), [t_s, t_e, \ldots], cons) \leftrightarrow$
$\exists\alpha\,(GeneratePlan(B, \alpha, \phi, [t_s, t_e, \ldots], cons) \wedge Can(B, \alpha, [t_s, t_e, \ldots], cons))$
$\wedge\, consistent(cons)$

Observe that the agent B can generate a partial or complete plan α and then further delegate execution or completion of the plan recursively via the $Can()$ statement in the second conjunct.

Elementary Action An *elementary action node* is a leaf node that specifies a domain-dependent action. The semantics of Can for an elementary action is platform dependent.

$Can(B, \tau, [t_s, t_e, \ldots], cons, \ldots) \leftrightarrow$
$Capabilities(B, \tau, [t_s, t_e, \ldots], cons) \wedge Resources(B, \tau, [t_s, t_e, \ldots], cons)$
$\wedge\, consistent(cons)$

There are two parts to the definition of Can for an elementary action node. These are defined in terms of a *platform specification* which is assumed to exist for each agent potentially involved in a collaborative mission. The platform specification has two components.

The first, specified by the predicate $Capabilities(B, \tau, [t_s, t_e, \ldots], cons)$ is intended to characterize all static capabilities associated with platform B that are required as capabilities for the successful execution of τ. If platform B has the necessary static capabilities for executing task τ in the interval $[t_s, t_e]$ with constraints $cons$, then this predicate will be true.

The second, specified by the predicate $Resources(B, \tau, [t_s, t_e, \ldots], cons)$ is intended to characterize dynamic resources such as fuel and battery power, which are consumable, or cameras and other sensors which are borrowable. Since resources generally vary through time, the semantic meaning of the predicate is temporally dependent.

Resources for an agent are represented as a set of parameterized resource constraint predicates, one per task. The parameters to the predicate are the task's parameters, in addition to the start time and the end time for the task. For example, assume there is a task $flyto(dest, speed)$. The resource constraint predicate for this task would be $flyto(t_s, t_e, dest, speed)$. The resource constraint predicate is defined as a conjunction of constraints, in the logical sense. As an example, consider the task $flyto(dest, speed)$ with the corresponding resource constraint predicate $flyto(t_s, t_e, dest, speed)$. The constraint model associated with the task for a particular platform P_1 might be:

$t_e = t_s + \frac{distance(pos(t_s, P_1), dest)}{speed} \wedge (Speed_{Min} \leq speed \leq Speed_{Max})$

4 Allocating Tasks to Platforms

The Delegate speech act requires that the delegating agent believes that the contractor has the ability to achieve the task. One central problem is therefore to find an agent which can achieve a particular, potentially very complex, task. When a task becomes complex it is highly likely that no single agent can achieve it alone. Since an agent can achieve a task by delegating parts of it to another agent, recursive delegation can solve the problem. The problem is therefore to find a set of agents who together can achieve a complex task with time, space and resource constraints through recursive delegation. This can be seen as a task allocation problem. The problem is to allocate tasks to platforms and assign values to parameters such that each task can be carried out by its assigned platform and all the constraints are satisfied.

When a platform is assigned an elementary action node in a TST, the constraints associated with that action are instantiated and added to the constraint store of the platform. The resource constraint is connected to the constraint problem defined by the TST through the node parameters. A platform can be allocated more than one node in a TST. This may introduce implicit dependencies between the actions since each allocation adds resource constraints to the constraint problem of the platform. There can for example be a shared resource that both actions use.

A *complete allocation* is an allocation which allocates every node in a TST to a platform. A completely allocated TST defines a constraint problem that represents all the constraints for this particular allocation of the TST. As the constraints are distributed among the platforms it is a distributed constraint problem. If the constraint problem is consistent then a *valid allocation* has been found and each solution can be seen as a potential execution schedule of the TST. The consistency of an allocation can be checked by a distributed constraint satisfaction problem (DCSP) solver such as the Asynchronous Weak Commitment Search (AWCS) algorithm [21] or ADOPT [18].

However, solving the task allocation problem as a single DCSP problem is currently not possible since the problem is too large even for modest TSTs. Instead we have developed a heuristic search approach for allocating tasks to platforms which uses marginal cost auctions to guide the search [16]. This allows reasonably large TSTs to be allocated.

Example The constraint problem for a TST is derived by recursively reducing the Can predicate statements associated with each task node with formally equivalent expressions, beginning with the top-node τ_0 until the logical statements reduce to a constraint network. Below, we show the reduction of the complex task α_0 represented by the TST in Figure 1 when there are three platforms, P_0, P_1 and P_2, with the appropriate capabilities, P_0 has been delegated the composite action α_0 and has recursively delegated α_2 and α_4 to P_1 and α_3 to P_2 while keeping α_1. α_i is the composite action described by the TST rooted in node τ_i.

$$\begin{aligned}
&Can(P_0, \alpha_0, [t_{s_0}, t_{e_0}], cons) = Can(P_0, S(\alpha_1, \alpha_4), [t_{s_0}, t_{e_0}], cons) \leftrightarrow \\
&\exists t_{s_1}, t_{e_1}, t_{s_4}, t_{e_4} (Can(P_0, \alpha_1, [t_{s_1}, t_{e_1}], cons_{P_0}) \vee \exists a_1 Delegate(P_0, a_1, \alpha_1, [t_{s_1}, t_{e_1}], cons_{P_0})) \\
&\qquad \wedge (Can(P_0, \alpha_4, [t_{s_4}, t_{e_4}], cons_{P_0}) \vee \exists a_2 Delegate(P_0, a_2, \alpha_4, [t_{s_4}, t_{e_4}], cons_{P_0}))
\end{aligned}$$

Let us focus on the reduction of first element in the sequence, α_1. Since P_0 has not delegated α_1 we expand the Can predicate one more step:

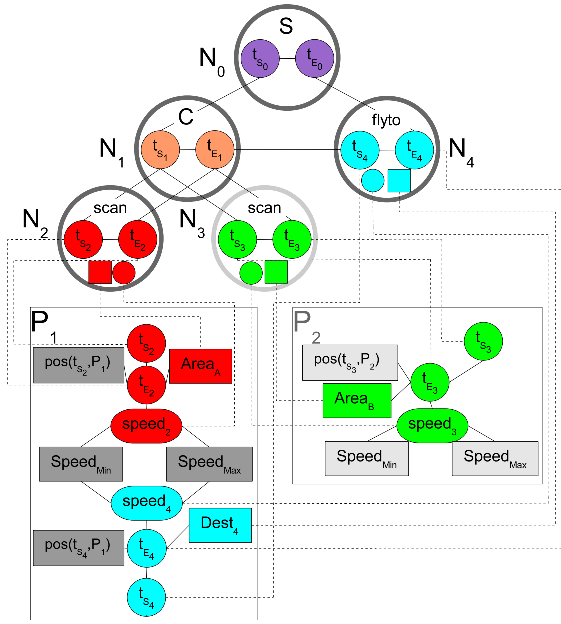


Fig. 2. The completely allocated and reduced TST showing the interaction between the TST constraints and the platform dependent constraints.

$$
\begin{aligned}
&Can(P_0, \alpha_1, [t_{s_1}, t_{e_1}], cons_{P_0}) = Can(P_0, C(\alpha_2, \alpha_3), [t_{s_1}, t_{e_1}], cons_{P_0}) \leftrightarrow \\
&\exists t_{s_2}, t_{e_2}, t_{s_3}, t_{e_3} (Can(P_0, \alpha_2, [t_{s_2}, t_{e_2}], cons_{P_0}) \vee \exists a_1 Delegate(P_0, a_1, \alpha_2, [t_{s_2}, t_{e_2}], cons_{P_0})) \\
&\qquad \wedge (Can(P_0, \alpha_3, [t_{s_3}, t_{e_3}], cons_{P_0}) \vee \exists a_2 Delegate(P_0, a_2, \alpha_3, [t_{s_3}, t_{e_3}], cons_{P_0}))
\end{aligned}
$$

Since P_0 has recursively delegated α_2 to P_1 and α_3 to P_2 the $Delegate$ predicates can be reduced to Can predicates:

$$
\begin{aligned}
&Can(P_0, \alpha_1, [t_{s_1}, t_{e_1}], cons_{P_0}) = Can(P_0, C(\alpha_2, \alpha_3), [t_{s_1}, t_{e_1}], cons_{P_0}) \leftrightarrow \\
&\exists t_{s_2}, t_{e_2}, t_{s_3}, t_{e_3} Can(P_1, \alpha_2, [t_{s_2}, t_{e_2}], cons_{P_1}) \wedge Can(P_2, \alpha_3, [t_{s_3}, t_{e_3}], cons_{P_2})
\end{aligned}
$$

Since P_0 has recursively delegated α_4 to P_1 we can complete the reduction and end up with the following:

$$
\begin{aligned}
&Can(P_0, \alpha_0, [t_{s_0}, t_{e_0}], cons) = Can(P_0, S(C(\alpha_2, \alpha_3), \alpha_4), [t_{s_0}, t_{e_0}], cons) \leftrightarrow \\
&\exists t_{s_1}, t_{e_1}, t_{s_4}, t_{e_4} \\
&\quad \exists t_{s_2}, t_{e_2}, t_{s_3}, t_{e_3} Can(P_1, \alpha_2, [t_{s_2}, t_{e_2}], cons_{P_1}) \wedge Can(P_2, \alpha_3, [t_{s_3}, t_{e_3}], cons_{P_2}) \\
&\ \wedge Can(P_1, \alpha_4, [t_{s_4}, t_{e_4}], cons_{P_1})
\end{aligned}
$$

The remaining tasks are elementary actions and consequently the definition of Can for these are platform dependent. When a platform is assigned an elementary action node the resource constraints for that action is added to the local constraint store. The local constraints are connected to the distributed constraint problem through the node parameters of the assigned node. All remaining Can predicates in the recursion are replaced with constraint sub-networks associated with specific platforms as shown in

Figure 2. To check that distributed constraint problem is consistent we use local CSP solvers together with a DCSP solver.

5 A Collaborative UAS Case Study

One important application area for unmanned aircraft systems is to assist emergency services. Here we consider an emergency services assistance scenario where an unmanned aircraft system (UAS) should scan a disaster area for injured people and deliver relief packages to them. In the first part of the scenario, the UAS scans the disaster area and creates a map over the locations of the identified survivors [20]. In the second part, the UAS delivers boxes with supplies to the survivors. To transport a box it can either be carried directly by an unmanned aircraft or it can be loaded onto a carrier which is transported to a key position from where the boxes can be distributed to their final locations.

In this particular scenario, there is a UAS consisting of two platforms (P_1 and P_2) and an operator (OP_1) which has found five survivors (S_1–S_5). The UAS has access to a carrier. Both platforms have the capability to transport a single box while only platform P_1 has the capability to transport a carrier. Both platforms also have the capabilities to coordinate sequence and concurrent tasks. At the same time another operator, OP_2, is performing a mission with the platforms P_3 and P_4 north of the survivors. P_3 is currently idle and OP_1 is therefore allowed to borrow it if necessary.

From the map, a TST is created that will achieve the goal of distributing relief packages to all survivors (Figure 3). The TST contains a sub-TST (N_1–N_{12}) for loading a carrier with four boxes (N_2–N_6), delivering the carrier (N_7), and unloading the packages from the carrier and delivering them to the survivors (N_8–N_{12}). A package must also be delivered to a survivor (S_5) far away from where most of the survivors were found (N_{13}). The delivery of packages can be done concurrently to save time, while the loading, moving, and unloading of the carrier is a sequential operation.

To achieve the mission, OP_1 delegates the TST to P_1. P_1 is now responsible for N_0 and for recursively delegating the nodes in the TST that it is not able to do itself. The allocation algorithm traverses the TST in depth-first order. P_1 will first find a platform for node N_1. When the entire sub-TST rooted in N_1 is allocated then it will find an allocation for node N_{13}. Nodes N_1 and N_2 are composite action nodes which have the same marginal cost for all platforms. P_1 therefore allocates N_1 and N_2 to itself. The constraints from nodes N_0–N_2 are added to the constraint network of P_1. The network is consistent since the composite action nodes describe an unrestricted schedule.

Below node N_2 are four elementary action nodes. Since P_1 is responsible for N_2, it tries to allocate them one at the time. For elementary action nodes, the choice of platform is the key to a successful allocation. The candidates for node N_3 are platforms P_1 and P_2. P_1 is closest to the package depot and therefore gives the best bid for the node. P_1 is allocated to N_3. For node N_4, platform P_1 is still the best choice, and it is allocated to N_4. Given the new position of P_1 after being allocated N_3 and N_4, P_2 is now closest to the depot resulting in the lowest bid and is allocated to N_5 and N_6. The schedule initially defined by nodes N_0–N_2 is now also constrained by how long it takes for P_1 and P_2 to carry out action nodes N_3–N_6. The constraint network is distributed among platforms P_1 and P_2.

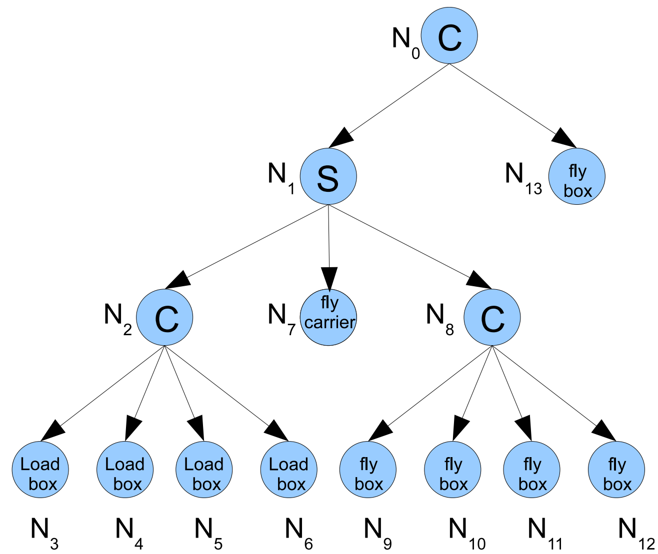


Fig. 3. The TST for the supply delivery case study.

The next node for P_1 to allocate is N_7, the carrier delivery node. P_1 is the only platform that has the fly carrier capabilities and is allocated the node. Continuing with nodes N_8–N_{12}, the platform with the lowest bid for each node is platform P_1, since it is in the area after delivering the carrier. P_1, is therefore allocated all the nodes N_8–N_{12}. The last node, N_{13}, is allocated to platform P_2 and the allocation is complete.

The only non-local information used by P_1 was the capabilities of the available platforms which was gathered through a broadcast. Everything else is local. The bids are made by each platform based on local information and the consistency of the constraint network is checked through distributed constraint satisfaction techniques.

The total mission time is 58 minutes, which is much longer than the operator expected. Since the constraint problem defined by the allocation of the TST is distributed between the platforms, it is possible for the operator to modify the constraint problem by adding more constraints, and thereby potentially changing the task allocation. The operator puts a time constraint on the mission, restricting the total time to 30 minutes.

The added time constraint makes the current allocation inconsistent. The last allocated node must therefore be re-allocated. However, no platform for N_{13} can make the allocation consistent, not even the unused platform P_3. Backtracking starts. Platform P_1 is in charge, since it is responsible for allocating node N_{13}. The N_1 sub-network is disconnected. Trying different platforms for node N_{13}, P_1 discovers that N_{13} can be allocated to P_2. Since removing all constraints due to the allocation of node N_1 and its children made the problem consistent, the backjump point is in the sub-TST rooted in N_1. Removing the allocations for sub-tree N_8 does not make the problem consistent so further backjumping is necessary. Notice that with a single consistency check the algorithm could deduce that no possible allocation of N_8 and its children can lead to a consistent allocation of N_{13}. Removing the allocation for node N_7 does not make a difference either. However, removing the allocations for the sub-TST N_2 makes the problem consistent. When finding an allocation of N_{13} after removing the constraints from N_6 the allocation process continues from N_6. When a consistent allocation is found, P_1 informs the operator. The operator inspects the allocation and approves it, thereby confirming the delegation and starting the execution of the mission.

6 Related Work

Due to the multi-disciplinary nature of the work considered here, there is a vast amount of related work too numerous to mention. In addition to the work referenced in the article, we instead consider a number of representative references from the areas of cooperative multi-robot systems and task allocation from a robotic perspective.

Cooperative multi-robot systems have a long history in robotics, multi-agent systems and AI in general. One early study presented a generic scheme based on a distributed plan merging process [2]. Another early work is ALLIANCE [19], which is a behavior based framework for instantaneous task assignment of loosely coupled subtasks with ordering dependencies. M+ [3] integrates mission planning, task refinement and cooperative task allocation. It uses a task allocation protocol based on the Contract Net protocol with explicit pre-defined capabilities and task costs. The M+CTA framework [1] is an extension of M+, where each robot has an individual plan and tasks are initially decomposed and then allocated. After the planning step, robots negotiate with each other to adapt their plans in the multi-robot context. The MURDOCH system [11] uses a publish/subscribe protocol for communication and an auction mechanism for task allocation. The result is very similar to the Contract Net protocol. Other Contract-Net and auction-based systems similar to those described above are COMETS [17], Hoplites [13] and TAEMS [8].

7 Conclusions

We have proposed and specified a formally grounded collaboration framework for robotic systems and human-operated ground control systems. Collaboration is formalized in terms of the concept of delegation and delegation is instantiated as a speech act. The formal characterization of the Speech act has a BDI flavor and KARO, which is an amalgam of dynamic logic and epistemic/doxastic logic, is used in the formal characterization. Tasks are central to the delegation process. Consequently, a flexible, specification language for tasks is introduced in the form of Task Specification Trees. Task Specification Trees provide a formal bridge between the abstract characterization of delegation as a speech act and its implementation in the collaborative system shell. Using this idea, the semantics of both delegation and tasks is grounded in the implementation in the form of a distributed constraint problem which when solved results in the allocation of tasks and resources to agents. We show the potential of this approach by targeting a real-life scenario consisting of UAV's and human resources in an emergency services application. The results described here should be considered a mature iteration of many ideas both formal and pragmatic which will continue to be pursued in additional iterations as future work. We will for example explore the expansion of select and loop nodes in much more detail.

References

1. Alami, R., Botelho, S.C.: Plan-based multi-robot cooperation. In: Advances in Plan-Based Control of Robotic Agents (2001)
2. Alami, R., Ingrand, F., Qutub, S.: A scheme for coordinating multirobot planning activities and plans execution. In: Proc. ECAI (1998)
3. Botelho, S., Alami, R.: M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement. In: Proc. ICRA (1999)
4. Castelfranchi, C., Falcone, R.: Toward a theory of delegation for agent-based systems. In: Robotics and Autonomous Systems. vol. 24, pp. 141–157 (1998)
5. Cohen, P., Levesque, H.: Intention is choice with commitment. AI 42(3), 213–261 (1990)
6. Cohen, P., Levesque, H.: Teamwork. Nous 25(4), 487–512 (1991)
7. Davis, E., Morgenstern, L.: A first-order theory of communication and multi-agent plans. Journal Logic and Computation 15(5), 701–749 (2005)
8. Decker, K.: TAEMS: A framework for environment centered analysis and design of coordination mechanisms. In: Foundations of Distributed AI. Wiley Inter-Science (1996)
9. Doherty, P., Meyer, J.J.C.: Towards a delegation framework for aerial robotic mission scenarios. In: Proc. International Workshop on Cooperative Information Agents (2007)
10. Falcone, R., Castelfranchi, C.: The human in the loop of a delegated agent: The theory of adjustable social autonomy. IEEE Transactions on Systems, Man and Cybernetics–Part A: Systems and Humans 31(5), 406–418 (2001)
11. Gerkey, B., Mataric, M.: Sold!: Auction methods for multi-robot coordination. IEEE Transactions on Robotics and Automation (2001)
12. W. van der Hoek, B.v.L., Meyer, J.J.C.: An integrated modal approach to rational agents. In: Wooldridge, M., Rao, A. (eds.) Foundations of Foundations of Rational Agency (1998)
13. Kaldra, N., Ferguson, D., Stentz, A.: Hoplites: A market-based framework for planned tight coordination in multirobot teams. In: Proc. ICRA (2005)
14. Kvarnström, J.: Planning for loosely coupled agents using patrial order forward-chaining. In: Proc. ICAPS (2011)
15. Kvarnström, J., Doherty, P.: Automated planning for collaborative UAV systems. In: Proc. International Conference on Control, Automation, Robotics and Vision (2010)
16. Landén, D., Heintz, F., Doherty, P.: Complex task allocation in mixed-initiative delegation: A UAV case study (early innovation). In: Proc. PRIMA (2010)
17. Lemaire, T., Alami, R., Lacroix, S.: A distributed tasks allocation scheme in multi-UAV context. In: Proc. ICRA (2004)
18. Modi, P., Shen, W.M., Tambe, M., Yokoo, M.: Adopt: Asynchronous distributed constraint optimization with quality guarantees. AI 161 (2006)
19. Parker, L.E.: Alliance: An architecture for fault tolerant multi-robot cooperation. IEEE Trans. Robot. Automat 14(2), 220–240 (1998)
20. Rudol, P., Doherty, P.: Human body detection and geolocalization for UAV search and rescue missions using color and thermal imagery. In: Proc. IEEE Aerospace Conference (2008)
21. Yokoo, M.: Asynchronous weak-commitment search for solving distributed constraint satisfaction problems. In: Proc. CP (1995)

Bounded Rationality in Maintaining Agreements

Christian Guttmann[1] and Abdulla Al Zaabi[1]

Etisalat BT Innovation Centre (EBTIC)
Khalifa University of Science, Technology and Research
P.O.Box 127788, Abu Dhabi, UAE
{christian.guttmann,abdulla.alzaabi}@kustar.ac.ae

Abstract. Agreements define the roles (e.g., responsibilities, objectives and tasks) of agents in executing a collaborative plan. Plans often need to be modified due to changing conditions of the social and natural environment (e.g., governmental regulations, collaborators' time, resources and objectives). With these plan changes, agreements also need to be revised which is a complex process. This process includes evaluating the participation in a plan, renegotiating the terms and conditions, reassessing agreements made with other agents, and predicting the exact transaction cost of these changes. In many domains, individuals are often not able to efficiently manage agreements and make sure they are compliant. This is because individuals are bounded rational – they are not able to completely capture and process the full complexity of issues that would guaranty a decision to be optimal (e.g., taking actions towards achieving compliant agreements that are reached efficiently). An individual's decision making process is limited by its cognitive abilities, the time available to capture and process information, and the incompleteness and uncertainty associated with the information itself. This position paper is a first step towards raising key questions and issues associated with building a comprehensive approach to assist bounded rational actors in managing agreements. The collaborative management of a patient with a chronic disease is used as an example.

1 Introduction

Planned activities often need to be modified to achieve agreed goals. We consider plans where several parties perform activities to achieve joint goals (and call those collaborative plans). If a collaborative plan is changed (which happens frequently in many domains), then the participants are often required to change the underlying agreements, too. If not managed well, significant costs can be incurred by rearranging agreements as well as the potential to leave a plan invalid (not compliant). In some cases, if only few activities in a plan are changed, a very large number of agreements may have to be changed (it may depend on how many individuals are affected by the plan change). The challenge is to cope with the bounded rationality of individual actors as this is a primary reason of optimal decisions not being made. Bounded rational actors have limited cognitive abilities, insufficient time, and incomplete/uncertain information available to make a decision, and hence, will influence the extend to which a compliant and efficient agreement change can occur. This requires a system that minimises transaction cost and ensures compliance when agreement changes are required. A transaction is represented by an exchange of information or activity made in relation to changing an agreement structure. The cost of a transaction can vary depending on how significant the changes are. For example, agreements of one individual or agreements of one activity in a plan, or changing agreements of the entire team or all activities. A more advanced approach is to consider heuristics how much other providers are influenced by changes made to the care plan.

One of the many practical domains in which the above is an issue is the collaborative care management of chronic disease [6]. A team of health care professionals performs a plan to care for a patient with a chronic disease aiming to maintain a healthy life and reduce cost to the health care system. The plan itself may need to be updated frequently due to changes in best practice guidelines, government regulations and a patient's health status to which a care plan has to comply with. When a plan is updated, each caregiver needs to reassess and renegotiate his/her role in the plan to make sure the patient can still be treated effectively and efficiently. For example, requirement of a collaborator's involvement (is a podiatrist services still required in a new plan?), review a collaborators's ability to perform a requested task ("loosing weight" activity is removed from a plan for a still obese patient, a podiatrist will not be able to guarantee the maintenance of healthy limbs), renegotiate the terms and conditions of financial outcomes (podiatrist receives rebate four times a year for a "foot review", but new plan consists of only two "foot reviews", hence less income for the podiatrist), and the validity of the plan ("taking a blood sample" is a legal requirement before giving a "nutrition education"). The complexity comes in as health care providers need to frequently review and possibly modify their role in executing a plan. If not managed well, plan changes can lead to a poor execution or non-compliance.

One main research question of interest is: How to effectively assist the management of agreement changes in collaborative care settings if the agreement parties are bounded rational (and potentially make suboptimal decisions regarding the efficiency and effectiveness of changing an agreement)? Developing "intelligent assistance" is well studied in the literature [10, 5]. Our interest is in building systems different in the following way.

- **Assist in managing agreed activities, not their performance.** The agent is not assisting in *performing* an individual's task (e.g., to help a GP to diagnose a patient's disease). Instead, an agent assists an actor in managing tasks. An actor has limited cognitive abilities to achieve and manage its tasks (due to complex and dynamic world), and an agent can extend this rationality by knowing how to achieve and manage the actor's task. Both, the agent and actor know the commitments of the actor, but only the actor acts, and the agent advises.
- **Enable exchange with other agents, not only one actor.** An agent aims for activities to be all in tune and agreed across all actors. An agent connects with other agents to exchange relevant information that makes the maintainance of agreements possible. This is different to connecting an agent to only one actor.
- **Design towards group oriented outcomes.** The behaviour of an agent is designed to assist the management of tasks associated to a collaborative plan and joint goals, and not the management of tasks of an individual.

Once such a system is realised, bounded rational caregivers are better able to manage the complex and dynamic conditions with an effect on the following three criteria.

- how to efficiently reduce the transaction cost during agreement changes possible involving transactions performed the entire group.
- how to observe whether plans and agreements are valid.
- how to identify whether a service is still needed in a collaborative plan.

This paper argues that current software architectures for collaborative management platforms are limited in supporting bounded rational actors in managing changes in joint plans. Intelligent agents can be used to extend such architectures by capturing mental attitudes of individuals, and more important to identify limitations which then require interventions to continue the care. We discuss the issues surrounding a support element in a software architecture architecture, including issues occurring at the conceptual, formal, and implementation level.

1.1 Intuitive Problem Statement

Intuitively, the problem is to reduce the transaction cost associated with modifying underlying agreement structures of a plan, where individual decision makers are bounded rational. We assume that there is a constant cost in changing the plan itself as an algorithm can recalculate the logical order of the loosely coupled activities. But the agreements that need to change are assigned with a cost.

Given a set of agents A, an agreement structure AS, a plan P (a loosely coupled set of activities), a revised plan P', a mapping of activities in P to agents A, how can we arrive at a new agreement structure AS' that satisfies two criteria.

- Overall plan validity. Satisfies plan constraints (e.g., X can only be executed before Y). Can the plan outcomes be achieved? Is a plan valid?
- Overall agreement validity. Satisfies agreement constraints (e.g., agent 1 must find an agreement with agent 2). Are the agreements valid?

The efficiency of an algorithm may be measured by a value representing the incurred transaction costs, that is, the cost of all transactions that have been performed in order to achieve an agreement change. This could be a combination of the number of transactions and the weighing of each type of transaction.

1.2 Why is solving this problem important?

Currently, agreements are organized such that individuals are committed to a rigid framework of rules and plans. This would all be fine and useful and predictable, if there were no changes in the social and natural environment during the lifecycle of a plan. Common sense dictates that if a change in the environment occurs that we just adapt the plan. What is often forgotten in this simple process is the significant transaction cost that is incurred when plans are reevaluated, reassessed and renegotiated. Some processes are so complex that an individual is not able to foresee the entire consequence to an action (e.g., is the overall care plan still achievable? What implications does an action have to other members of the team?). There is currently no comprehensive solution to this significant issue, and the research community often investigates idealized models of rationality and assume that transaction costs are non-existent or minimal in making changes to plans.

1.3 Bounded Rational Actor

According to Herbert Simon [9], bounded rationality means that we are limited in making perfect decisions. This is due to a lack of information available to a decision maker (incomplete and uncertain information), a limit of cognitive abilities to process information, and a limited time to make decisions. We simplify a decision problem so much that we can solve it, but may remove critical information during the simplification process, and hence make a decision on "'too abstract"' information – missing potentially the true optimal solution of a problem. According to Simon [9], we merely satisfice an optimisation problem, rather than optimally solving it. Rubinstein extends this work on bounded rationality and provides a model of a bounded rational decision maker [8]. Following from this work, we introduce the term "'bounded rational actor (BRA)"' – a rational actor with certain limits. And we consider such an actor that makes decisions towards achieving the goals of a collaborative plan.

Given the problem statement in Section 1.1, our interest is in investigating this problem under bounded rationality and how it effects the cost of transactions. This might be due to not fully realising the possibility to reduce agreement structures effectively.

A simple example of a basic transaction cost function is where one cost unit corresponds to an individual performing a request and response interaction towards changing an agreement structure. Consider a team of health care provider, where each provider agreed to perform an individually assigned tasks (specified in a careplan), and has agreed to the overall workflow and outcomes of the plan. During the lifecycle of this plan, the patient realises that he/she can not walk the distance as specified in the care plan. This requires to modify this activity currently in the care plan. As the GP is responsible for the overall execution of the plan, he/she has to agree with changes. Indeed, each team member may need to agree with the changed care plan, and at least each of them is actioned to re-asses and re-agree to the revised plan. So, we have at least two times the number of caregivers, for each caregiver there is a request and respond action. Since this is a change that can occur frequently and primarily effects the GP and patient, one basic approach may be to only involve the GP and patient in this transaction, and to have a simple cutoff point on the influence of plan changes to individuals.

2 Questions and Issues on Building Intellgent Systems to Support Bounded Rational Actors

In the context of organisations, [4] suggests that the world is too complex, dynamic and diverse to be fully understood. Therefore, to overcome this limitation there are two paths forward: 1) to reduce the scope of analyses for a satisfactory outcome (that is, to simplify a problem to make it computable) and 2) to develop tools, techniques and arrangements to extend the cognitive limitations of actors [4]. We focus on the latter point and discuss the issues that arise in building a comprehensive system that assists bounded rational agents to act in complex domains. We confine our discussion to the conceptual, formal, and implementational issues that need to be addressed to build such a framework.

2.1 Conceptual Layer

- What should be cared for? Which entity is the subject of care?
  - Compliance, goal achievements?
  - Individual, group, care plans?
- How can we care for it?
  - Broadcast wide interventions?
  - Replanning?
  - Monitoring and observing?

In the context of chronic disease management, there are various ways of defining the subject of care – though generally, we care about maintaining the health of a patient. We can consider the lifecycle of the actual disease, or the collaborative status of the patient, the status of the collaboration. One way to maintain the patient's health is by achieving the health care goals defined by the team of caregivers. Goals are achieved by performing a health care plans, and this plan must be flexible to cope with the changes that happen in the social and natural environment. The modification and execution of a care plan is at the core of an intelligent collaborative care system. The question of what we care about influences the structures that need to be maintained. It determines which features are important to , and how we represent those features.

How do we need to view a patient in this system? What are the critical stages through which a care plan goes? At what time does the patient exit the lifecycle? What is the lifetime value of care plan? Figure 1 is a first attempt to outline the stages through which a patient goes during the lifecycle of a plan. These steps are defined below.

- **Patient enters system:** Patient feels sick, shows symptoms of a chronic disease, seeks help from medical professional.
- **Diagnose patient:** Patient is diagnosed with a disease using the pathological tests, physical examinations, and historical medical data of a patient. At the end of this process, the general practitioner knows the patient conditions, and can hence define the treatment and the specific needs of a given patient.
- **Create customised plan:** Given the patient's conditions and the treatment goals, the GP can decide on the goals and outcomes of the care plan. This will define activities according to best practice that will assist the treatment of a patient.
- **Negotiate tasks:** Caregivers are approached to potentially be assigned to the tasks. The GP and health care authorities know which activities can be performed by whom, and the specific caregivers understand the responsibilities to take in the care plan. However, there is usually the option of refining a task, possibly modifying activities specific to the caregiver in question that need be executed.
- **Assign tasks:** At this stage, all activities and goals have been identified, negotiated, refined and agreed. A formal agreement by individual entities is now seeked.

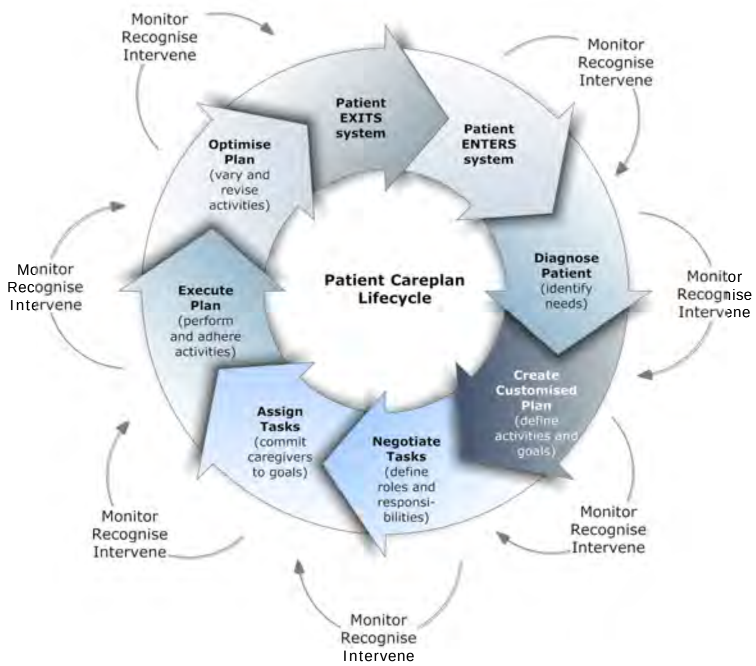


Fig. 1: Patient care plan lifecycle

– **Execute plan:** Individual caregivers (including patients) now need to follow on their agreed committments by performing the activities as specified.
– **Optimise plan:** Changing environmental and patient conditions as well as changing patient objectives may require changes to the care plan. This may go as far as to change the entire plan. The most “costly” part in this stage is to manage the existing agreements that have been made in the negotiation and assignment stage.
– **Patient exists system:** The care plan has ceased to be relevant. There could various reasons why this is so, e.g., , death of a patient, or the patient removal from a care plan due to a patient not making any progress.

During each of these cycles, it is critical to make sure that the activities, and commitments are aligned with reaching the desired outcomes of this care plan. This is a challenge as actors are bounded rational, and require assistance to make sure the overall goals can be achieved.

Further questions of conceptual importance are. To what extend are we responsible to care for individuals, and when do they need to care for themselves? What implication on the system architecture and design? When to intervene and when not to? Costs to health care providers? What is a care plan lifecycle exactly? What exactly constitutes a basic relationship between two entities? How can we form, maintain and extend (contractual) relationships? And why is it important? A relationship between two entities defines which agreements can be established (which is discussed in the next section).

Hence, it is important to characterize the features of each entity, and the nature of this relationship. In this section, we consider relationships as partnerships, where groups share responsibility to achieve a goal. For example, "effective language learning is a partnership between school, teacher and student", "the action teams worked in partnership with the government". And in this paper we consider how a care team aims to move a patient through a case plan lifecycle. Other important questions are: What is an agreement? What are the mechanism that each agent performs to reevaluate the agreements? What exactly needs to change in the agreement structure? How to manage agreement restructuring in incomplete and uncertain domains, particularly domains consisting of bounded rational humans? How is it done now (with rational humans)? How are agreements formed and maintained? How can we ensure that things stay on track? What are collaborative plan structures? What are collaborative care agreement structures?

2.2 Formal Layer

Since this section builds heavily on the section of concepts, we only outline first questions relevant to the formalisation of a framework.

- What needs to be represented, and what can be represented? What is an appropriate level of abstraction for the agreements and plans?
- What logic can represent what we need to represent? What logic to use and/or extend? Will it be a PCTL or PCTL + BDI type logic that is required?
- (How) can we formally represent the interaction and agreement processes in the collaborative care management system?
- What is the algorithm, and how can we formalise it?
- How can we formally define the input and output of the algorithm?
- What and how can we measure the outcomes of the algorithm? What benchmark? Generally, what we need to measure is the overall outcome of the algorithmic solution. The transaction costs would be the measure. In chronic disease,

2.3 Implementation/Evaluation Layer

Similar to Section 2.2, we only outline first questions relevant to the implementation and evaluation of a framework.

- BDI agent platform/representation?
- What are the BDI Agents?
- Why BDI Agents? - justify why it best suits our framework.
- What questions can we answer?

The Belief-Desire-Intention (BDI) is one of various multi-agent programming models that aim to model human behavior [1, 7]. Computer agents have three mental states about the outside world.

- Beliefs: represent information the agent currently has about the world, what the agent beliefs to be true or valid.

– Desire: the options an agent has to enable it to achieve a goal. Not all of them has to be executed, which option to choose is based on different elements.
– Intention: represent what the agent committed to achieve. They may represent goals, in the basic scenario; intention is what the agent decided to work towards.

The BDI model for computer agents represents intelligent computer agents based on the above mental attitudes. The BDI agents use these mental states to reason or decide how to act. This is a practical reasoning model because one can reason with a set of mental states to justify a course of action with relative ease. When an agent has different approaches to achieve a goal, the agent chooses what it "thinks" it is best based on its' set of mental states. From an implementation point of view, we consider using different approaches for the implementation of the agreement structure variation framework.

3 Related Research

3.1 Service Oriented Architectures and Web Services

Web services are self-contained and self-describing, and provide functionality and interoperation for business processes. Web services rely on fully standardised interfaces and well described information flows to enable interoperability between processes according to the Service Oriented Architecture (SOA) paradigm [2]. The interfaces are well defined, and the data and control path are well known. Description languages, such as Web services Description Language (WSDL), are used to define a service. Web services provide a common standard mechanism (e.g., communication protocols) for interoperable integration among disparate systems, and the key to their utility is their standardization.[1]

One of the biggest issues of web services and SOA is that the assumed rigid standardisation (considered to be one of the greatest benefit of SOA), is possibly one of the web services greatest disadvantages. There is no adequate answer to what will happen if services change their interface and their quality of service (QoS) during the time of invocation (the interface/QoS are simply assumed to remain entirely static once invoked – an assumption that can not hold in many domains, particularly health care). Indeed, what if the web service itself does not "realise" that interfaces and QoS have changed or are not valid according to new certain governmental or partnership regulations. A health care provider is often not able to realise or adapt to these changes due to its bounded rationality and the complexity of the system changes. This can be a critical issue in the management of a chronic disease, where service providers are assigned to patients, have developed a close relationship with that patient, should stay with that patient, but where frequent changes in regulations make this engagement invalid (as per the SOA paradigm). In other words, government regulation can influence a health care provider's interface and QoS, and thus the provider's involvement in a care plan. However, we can't just remove care providers, as patients have built a relationship with him/her (and the care provider has established a relationship with other care providers).

[1] http://www.ibm.com/developerworks/webservices/library/ws-soaintro.html

And it is also in the care provider's interest to stay within a valid framework to receive the rebate by the government for the delivered services. There is a high demand in flexibly reacting to the changing environmental conditions.

3.2 Contract Agreement Frameworks

Related research has been conducted on norms and contractual agreements [3]. In the CONTRACT project, each contract is associated with Critical States (CS) and Danger of Violation (DOV) states. CS are compulsory for the successful execution of a contract. That is, at the service delivery stage, if a CS state does not occur, it is identified as a violation of the contract. The DOV states indicate a possible violation of the contract, but are not explicitly stated in a contract. This work considers a framework in which global states are identified that show norm violations or close norm violations. Global states and norms assume that there is some global and high-level regulative system. However, many agreements are done among partners. The detection of states may need to be done across the support agents.

4 Concluding Remarks

This paper argues that we need to assist bounded rational agents to follow and organise agreements. Bounded rational here means that individuals are cognitively overwhelming with the ever changing environmental conditions of complex collaborative arrangements. A lifecycle of a care entity could be a two year long care plan for a patient with a chronic disease (which is used as an example in this paper). Such a plan is not likely to stay static for two years due to changing health conditions of the patient, and changing governmental policies that influence how care plans are conducted. The importance of a solution to this problem is that it can significantly reduce transaction costs and increase plan validity in dynamic and uncertain domains. The novelty of this approach is to enable assistance to execute a complex collaborative care plan. This is different to related work where agents assist an individual to achieve its own plans (without necessarily involving the constraints and goals of a collaborative plan), and different to detecting global states that indicate the status of a regulative system execution. The realisation of this proposed research will manage agreements and plans significantly better than existing approaches. The framework enables bounded rational actors to assist in complex and dynamic environments, and achieve joint outcomes for the cared entity.

References

1. M.E. Bratman, D.J. Israel, and M.E. Pollack. Plans and resource-bounded practical reasoning. *Computational intelligence*, 4(3):349–355, 1988.
2. T. Erl. *Service-oriented architecture: a field guide to integrating XML and web services*. Prentice Hall PTR Upper Saddle River, NJ, USA, 2004.
3. N. Faci, S. Modgil, N. Oren, F. Meneguzzi, S. Miles, and M. Luck. Towards a monitoring framework for agent-based contract systems. *Cooperative Information Agents XII*, pages 292–305, 2008.

4. J. Kooiman. *Governing as governance*. Sage Publications Ltd, 2003.
5. K. Myers, P. Berry, J. Blythe, K. Conley, M. Gervasio, D. McGuinness, D. Morley, A. Pfeffer, M. Pollack, and M. Tambe. An Intelligent Personal Assistant for Task and Time Management. *AI MAGAZINE*, 2007.
6. V.L. Patel, K.N. Cytryn, E.H. Shortliffe, and C. Safran. The collaborative health care team: the role of individual and group expertise. *Teaching and Learning in Medicine*, 12(3):117–132, 2000.
7. A.S. Rao and M.P. Georgeff. Modeling rational agents within a BDI-architecture. *Readings in agents*, pages 317–328, 1997.
8. A. Rubinstein. *Modeling bounded rationality*. The MIT Press, 1998.
9. H.A. Simon. Rationality as Process and as Product of Thought. *The American Economic Review*, 68(2):1–16, 1978.
10. J. Tweedale, N. Ichalkaranje, C. Sioutis, B. Jarvis, A. Consoli, and G. Phillips-Wren. Innovations in multi-agent systems. *Journal of Network and Computer Applications*, 30(3):1089–1115, 2007.

Shared Mental Models for Decision Support Systems and Their Users

Iris van de Kieft, Catholijn M. Jonker, and M. Birna van Riemsdijk

Man Machine Interaction Group, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands

Abstract. Decision support systems (DSSs) aim to assist people in their decision making process. We argue that a shared mental model between the human user and the DSS enables and enhances their collaboration. This paper presents an approach based on shared mental models, for analysing user-DSS cooperation, which results a set of concepts that a shared mental model of the user and the NSS should contain. This analysis also indicates the various reasons that discrepancies between the individual mental models may arise. The results of this analysis provide a basis for improving the sharedness of the mental models of the user and the DSS, and thereby, improving their cooperation. We illustrate our approach with an example: negotiation support systems.

1 Introduction

Decision makers often seek various forms of external information support to aid their decision making process [6], in order to aid their cognitive deficiencies in judgement and decision making. This had led to the development of interactive computer-based systems that aid users in judgement and choice activities: decision support systems (DSS) [5].

A DSS and its user can be regarded as a team, which has the task of making a decision. They form a specific kind of team, in which each has their own, complementary capabilities. It is well-known from the social psychology literature that performance of human teams is positively influenced by the team members having a shared understanding or *shared mental model* (SMM) of the task and the team work involved ([11, 14]). We maintain that having an SMM is not only important in human teams, but also in human-agent teams. Discrepancies between the mental models of the DSS and the user may at best result in innocent misunderstandings, but at its worst may result in a dysfunctional cooperation. We thus argue that the SMM concept is important for DSS development.

In this paper, we present a sketch of an SMM-based analysis of the user-DSS task and interaction. We believe this analysis provides a basis for improving the shared mental model of the user and the DSS, which in turn should improve user-DSS cooperation.

We illustrate our analysis with an example: a negotiation support system (NSS). We analyze the negotiation task and the interaction between user and

NSS, to determine the essential components of their SMM. Furthermore, the analysis serves to determine the possible causes of discrepancies between mental models, for example, the constructive nature of preferences and the bounded rationality of humans. Future work will address how this analysis can be used to determine how the SMM, and thus cooperation, can be improved.

This paper is organized as follows: Sect. 2 presents an introduction to shared mental models and an outline of our SMM-based analysis method. Sect. 3 presents the NSS example, in which we illustrate how our SMM-based analysis can be applied to negotiation. Finally, we discuss future work in Sect. 4.

2 Shared Mental Models in User-DSS Cooperation

2.1 Shared Mental Models

Mental models have received a lot of attention in literature regarding team performance. Several studies have shown a positive relation between team performance and similarity between mental models of team members (e.g., [2, 11, 14]). That is, it is important for team performance that team members have a shared understanding of the team and the task that is to be performed, i.e., that team members have a SMM. The concept of SMM is defined in [3] as:

> 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 behavior to demands of the task and other team members.

SMMs thus help *describe, explain and predict the behavior of the team*, which allows team members to coordinate and adapt to changes.

We maintain that SMM theory, as developed in social psychology, can be used as inspiration for the development of techniques for improving teamwork in human-agent teams.

We emphasize that not all knowledge in the mental models needs to be shared. This is especially true for user-DSS teams, in which each team member has complementary skills, and correspondingly, a distinct role. Therefore, the first step in our approach is to determine what knowledge is complementary and what should be shared. Based on this, a *desired SMM* can be defined. The actual SMM consists of the knowledge actually shared (similar) between the two mental models. The desired SMM consists of the knowledge that ideally should be present in the SMM. In this desired SMM, we want the knowledge not only to be similar but accurate.

We consider a *discrepancy* between mental models to exist when one model contains information regarding an element, and the other model contains either conflicting information regarding this element, or no information regarding this element. Once a discrepancy is detected, it can be resolved by adapting (one of) the mental models. Note that discrepancies may also exist for elements that need not be part of the desired SMM. In this paper, we are interested in those discrepancies that impair the desired SMM, and hence, possibly the cooperation. We will sometimes use the term 'problematic discrepancy' to emphasize that we are referring to a discrepancy regarding an element of the desired SMM.

2.2 SMM-Based Analysis of User-DSS cooperation

We argue that when a user does not understand or agree with the DSS advice, there is a discrepancy between their mental models. However, it is not always immediately clear how the discrepancy should be resolved, as there are different kinds of discrepancies. In order to reduce and resolve these possible discrepancies, we first must determine what they can be, when they arise and why. We therefore propose an SMM-based analysis of user-DSS task and teamwork. This analysis aims to gain insight into the user-DSS cooperation and how it may be improved. The analysis consists of two steps:

1. Analysis of the user-DSS task, interaction and their different roles. This allows us to determine what should knowledge is complementary and what should be part of the desired SMM.
2. Analysis of what humans find difficult about the task, and other possible reasons that sharedness may be difficult to achieve. This allows us to determine what areas might need extra focus when trying to achieve and maintain sharedness.

The results of this analysis can form the basis for improving the SMM of user and DSS.

3 Example: Negotiation Support Systems

In this section, we illustrate our approach of analyzing user-DSS cooperation, based on SMM, with an example: negotiation (support systems). Negotiation is an interactive decision-making process between two or more parties. The following four major stages can be discerned in integrative negotiation: private preparation, joint exploration, bidding, and closing. *Private preparation* is about information gathering and reflection before meeting the other party. In *joint exploration* the negotiating parties talk to each other, but do not place bids on the table. During *bidding*, both negotiators exchange bids according to the agreed protocol, typically a turn-taking protocol. For each incoming bid, the negotiator has to decide whether to accept, to make a counteroffer, or to stop. During the *closing* stage the outcome of the bidding stage is formalized and confirmed by both parties.

3.1 Related Work

Negotiation is a complex process that involves emotions as well as computational complexity. As a result, even experienced human negotiators can fail to achieve efficient outcomes [15]. This has motivated the development of negotiation support systems (NSSs), which assist a human negotiator (user) in negotiation by, for example, aiding communication, enhancing negotiation skills, and reducing cognitive task load.

A number of NSSs have been or are being developed [7]. Inspire[1] is a web-based NSS with a facility for specification of preferences and assessment of offers, an internal messaging system, graphical displays of the negotiation's progress, and other capabilities. It has been used to support humans in negotiation, as well as to collect data about such negotiations for research purposes. Another example of an NSS is Athena[2]. This system has primarily been used in education. In both Inspire and Athena, users have to build content models themselves. That is, users have to provide the domain structure. The provided support does not include predefined repositories of content models, interaction support, or assistance in selecting a bidding strategy. Smartsettle[3] is a commercial NSS, which also provides bidding support.

The Pocket Negotiator project [7] strives for synergy between NSS and the human negotiator that it assists by exploiting their complementary skills. The aim of the Pocket Negotiator is to provide focus and structured support, which will increase the user's capacity for structuring and exploring the negotiation space, and to reduce the cognitive task load while doing so. The aim is not to supplant the human in negotiation, but to create an intelligent artificial partner.

In general, an NSS does not engage directly in the negotiation. Its purpose is to provide assistance during the negotiation process by structuring the process and possibly offering analysis support [7]. This can be contrasted with automated negotiating agents, which do engage directly in a significant part of negotiation, acting on behalf of their human or artificial principal [9].

Several technical challenges must be faced when developing an NSS [7]. However, in this article, we assume that an NSS has been created successfully, and has the technical means to assist with preference elicitation, domain and opponent modelling, and strategic bidding. NSSs can differ in the number of parties they support, and also the type of negotiation they support: bilateral or multilateral. For example, an NSS may be built to support both parties during a bilateral negotiation, or one party during a multilateral negotiation. We are interested in the cooperation between the NSS and one of the parties it supports, i.e., the user. In this illustrative example, we assume the user is involved in bilateral negotiation.

3.2 The Interaction Between Human Negotiator and NSS: What to Share?

In this section we analyze the interaction between user and NSS to gain insight into the task division between user and NSS. This helps determine the contents of the desired SMM that needs to be cultivated between user and NSS.

[1] http://invite.concordia.ca/inspire/
[2] http://www.athenasoft.org
[3] http://www.smartsettle.com

In this type of human-machine collaboration, the human weaknesses are covered by the strengths of the machine, and the weaknesses of the machine are covered by the strengths of the human. This implies that tasks should be divided between user and NSS in a way that respects their complementary capabilities:

- The user has a wealth of knowledge about the world and about interacting with other humans, but need not be a specialist in negotiation. The NSS specializes in negotiation. It makes generic negotiation knowledge available to the human.
- The NSS remains rational at all times. The user has emotions that might hinder the negotiation. However, it has been argued that emotions are sometimes needed for decision making [4]. With negotiation, it seems that both the rational NSS and the more emotional human can each provide a useful perspective on the situation, and together achieve a good outcome.
- The user can recognize emotions from voice, face, and body language, but might be at a loss how to deal with them. The NSS has generic negotiation knowledge about dealing with emotions.
- The user has limited working memory and limited computational power, i.e., bounded rationality. The NSS typically has better memory and can search much more quickly through much larger outcome spaces. Nevertheless, the NSS also can have bounded rationality, i.e., in some cases it may lack sufficient information and/or reasoning capabilities. However, for our purposes, we assume that the NSS's computational power suffices for the domains in which it is used.

Because the user and the NSS have complementary skills and tasks, they need not share all their knowledge. However, some shared information is necessary to cooperate and understand each other, hence the need for an SMM. The information and knowledge exchange between these two team members is as follows: during the preparation and exploration stage the user needs to inform the NSS about the current negotiation, e.g., the Opponent, the set of issues I, and outcome space V, and the utility functions of himself (Self) and Opponent. We assume that the user is also responsible for informing the NSS about the exchanged bids. The NSS needs this user input in order to provide assistance during the bidding stage, when strategic, tactical and bidding decisions have to be made.

For this information exchange to be successful, the user must fully understand the process of negotiation and what is expected of him/her by the NSS, and what can be expected in return. This implies that during the negotiation stages, the NSS needs to provide the user (upon request) with generic negotiation information, but also current negotiation information regarding the Opponent, I, V, and utility functions, in as far as such information is available to the NSS.

The user and the NSS thus need shared information about the current negotiation and about their capabilities and knowledge. More formally, the desired SMM of a human negotiator and an NSS contains submodels on:

- domain knowledge D

- I: set of issues
- $\forall i \in I$: V_i the value range of issue i

– knowledge about Self
  - u_S: the utility function of Self
  - the emotional status of Self as far as Self is aware of that state
  - the coping style of Self
  - the negotiation model of Self
  - the capabilities and types of knowledge of Self and of NSS

– knowledge about the Opponent
  - u_O: the utility function of the Opponent in as far as known to Self or NSS
  - the emotional status of the Opponent as far as perceived by Self
  - the coping style of the Opponent as far as known to Self or NSS
  - the negotiation model of the Opponent, in as far as this is known to Self or NSS

– bidding knowledge
  - bidding history: the sequence of bids that have been exchanged so far
  - the current bidding strategy for Self
  - the bidding protocol, including information about available time

3.3 The Weaknesses of the Human Negotiator

In this second part of our analysis, we discuss the problems humans have with negotiation, assuming there is no NSS support. There are two ways to categorize the problems humans have with negotiation: related to *outcome*, or related to the negotiation *process*. The outcome related pitfalls in negotiation are: leaving money on the table, settling for too little, rejecting a better offer than any other available option, and settling for terms worse than alternative options [1, 15].

The outcome related pitfalls are caused by the problems people have during the negotiation process, which are related to the following:

– *Lack of training* Humans have difficulty in structuring negotiation problems and thinking creatively about such problems. Moreover, just negotiating in practice does not alleviate these problems due to faulty feedback and self-reinforcing incompetence. Faulty feedback refers to the problem of not getting accurate, immediate, and specific feedback, which can only be solved through regular training. Self-reinforcing incompetence means not being aware of one's limitations, thus not seeing the need to improve one's skills.
– *Lack of preparation* Preparation is insufficient when it leaves the negotiator unaware of an important part of the issues, underlying interests, the preferences and/or circumstances of the parties involved, see e.g., [1, 15].
– *Structural barriers to agreement* This refers to such problems as die-hard bargainers, a bad atmosphere [12], power imbalance [13], cultural and gender differences [8], disruptive or incommunicative people, and a lack of information. The last point can be caused by insufficient preparation, but also by communication problems. See [1] for more information.

– *Mental errors* Parties commit mental errors such as the escalation error, biased perception, irrational expectations, overconfidence, and unchecked emotions. The escalation error is the continuation of a previously selected course of action beyond the point where it makes sense. Biased perception is the problem of perceiving the world with a bias in your own favour [1, 15].
– *Satisficing* Due to uncertainty of the future, the costs of acquiring information, and the limitations of their computational capacities, people have only bounded rationality, forcing them to make decisions by satisficing, not by maximization [15].

NSSs aim to relieve some of these problems. At the same time, these problems are also precisely what may make it difficult to achieve the desired SMM, and thus, for the user to understand the reasoning and advice of the NSS.

3.4 Mental Model Discrepancies

We have now identified what should ideally be in the SMM of the user and the DSS. We have also identified what may make achieving such a desired SMM difficult. Together, this provides insight into where problematic discrepancies may arise.

Section 3.2 showed that the NSS relies upon the user for most of its knowledge about the current negotiation. If the user does not provide enough input, the mental model of the NSS may be incomplete.

Section 3.3 discussed the problems humans have with negotiation. These can cause the mental model of the user to lack (accurate) information. For example, lack of training, lack of preparation and/or bounded rationality can cause the user to have incomplete knowledge of the current situation. At the same time, the NSS has generic negotiation knowledge that the user may lack, as well as superior computational abilities. The NSS's mental model can thus contain more accurate information than the user's. In such situations, the user's mental model should be adapted to that of the NSS.

One particular aspect that may lead to discrepancies is the *constructiveness* of domain and preference information. Even with proper preparation, information on the domain and preferences of Self and Opponent is often difficult to determine fully at the start of the negotiation. Humans have been found to discover this information along the way. Due to this constructiveness, the user may discover new knowledge during the negotiation that the NSS does not yet have, thus resulting in a discrepancy.

Table 1 provides an overview of some possible causes of discrepancies between mental models. For each submodel defined above, and for each team member, the table lists what may cause their mental model to lack (accurate) information. This knowledge about problematic discrepancies helps identify for what elements of the desired SMM it may be particularly difficult to achieve similarity.

Table 1. Causes for lack of (correct) information in mental models

submodel	User mental model	NSS mental model
domain $D = \langle I, V \rangle$	lack of preparation, bounded rationality, constructive domain	lack of user input, constructive domain
knowledge about Self, e.g., u_S	lack of training, lack of preparation, bounded rationality, constructive preferences	lack of user input, constructive user preferences
knowledge about Opponent, e.g,. u_O	lack of training, lack of preparation, constructive Opponent preferences, bounded rationality	lack of user input, constructive Opponent preferences
bidding knowledge	lack of training, lack of preparation, bounded rationality	lack of user input

4 Summary and Future Work

We presented an approach for improving user-DSS cooperation. This approach involves a SMM-based analysis of user-DSS task and teamwork, which provides insight into what knowledge should be shared between user and DSS, and what might make achieving such a SMM difficult. This analysis can then form the basis for improving cooperation. We illustrated our approach with an example: NSSs.

Future work first of all calls for a more precise specification of our analysis method. This requires formalizing the concepts of mental model and shared mental model for user-DSS teams. Moreover, the analysis steps should be made more concrete. This then allows us to apply our method in a more thorough manner to different DSS domains, thereby generating a description of the contents of the desired SMM in those domains.

Once a formal analysis method has been developed, we will investigate ways to achieve and maintain the desired SMM. One technology that we believe is suitable for this is *explanation.* Explanation can serve various purposes, such as improving effectiveness (helping users make good decisions), increasing the user's trust in the system and improving *transparency* of the system [16]. The latter is particularly relevant, as this facilitates detecting and resolving discrepancies between mental models of NSS and user. Transparency means explaining how the system works, thus giving the user a better understanding of the NSS's reasoning process. This allows the user to detect any discrepancies between the mental models, and subsequently to resolve these by updating the mental models where necessary. The results of our analysis can form a basis for these explanations, determining the requirements. In [10], we have made some initial steps herein, where explanation is used to resolve mental model discrepancies regarding preferences.

There are also other ways in which our analysis might be used to improve user-DSS cooperation. The results of our analysis could assist DSS design. For example, the analysis results could provide guidelines for the knowledge that needs to be stored in the DSS database(s). They could also provide guidelines for the user interface design, by indicating what type of interaction is necessary between user and DSS.

This work should be implemented and user tests should be performed to determine if our approach indeed succeeds in improving user-DSS cooperation.

Acknowledgements This research is supported by the Dutch Technology Foundation STW, applied science division of NWO and the Technology Program of the Ministry of Economic Affairs. It is part of the Pocket Negotiator project with grant number VICI-project 08075.

References

1. Harvard Business Essentials: Negotiation. Harvard Business School Publishing Corporation, Boston (2003)
2. Bolstad, C., Endsley, M.: Shared mental models and shared displays: An empirical evaluation of team performance. In: Human Factors and Ergonomics Society Annual Meeting Proceedings. vol. 43, pp. 213–217 (1999)
3. Cannon-Bowers, J., Salas, E., Converse, S.: Shared mental models in expert decision making teams. Current issues in individual and group decision making pp. 221–246 (1993)
4. Damasio, A.: Descartes' error: Emotion, reason, and the human brain. Putnam Publishing (1994)
5. Druzdzel, M., Flynn, R.: Encyclopedia of Library and Information Science, chap. Decision Support Systems. New York: Marcel Dekker, 2nd edn. (2003)
6. Gönül, M., Önkal, D., Lawrence, M.: The effects of structural characteristics of explanations on use of a DSS. Decision Support Systems 42(3), 1481–1493 (2006)
7. Hindriks, K., Jonker, C.: Creating human-machine synergy in negotiation support systems: towards the pocket negotiator. In: Proceedings of the 1st International Working Conference on Human Factors and Computational Models in Negotiation. pp. 47–54. ACM (2008)
8. Hofstede, G., Hofstede, G., Minkov, M.: Cultures and organizations: Software of the mind: Intercultural cooperation and its importance for survival. McGraw-Hill New York, NY (2005)
9. Kersten, G., Lai, H.: Negotiation support and e-negotiation systems: an overview. Group Decision and Negotiation 16(6), 553–586 (2007)
10. van de Kieft, I., Jonker, C., van Riemsdijk, M.: Explaining negotiation: Obtaining a shared mental model of preferences. In: Modern Approaches in Applied Intelligence, Lecture Notes in Artificial Intelligence. Springer Verlag (to appear)
11. Lim, B., Klein, K.: Team mental models and team performance: A field study of the effects of team mental model similarity and accuracy. Journal of Organizational Behavior 27(4), 403 (2006)
12. Mastenbroek, W.: Negotiate. Basil Blackwell (1989)
13. Mastenbroek, W.: Negotiating as emotion management. Theory, Culture & Society 16(4) (1999)

14. Mathieu, E., Heffner, T., Goodwin, G., Salas, E., Cannon-Bowers, J.: The influence of shared mental models on team process and performance. The Journal of Applied Psychology 85(2), 273–283 (2000)
15. Thompson, L.: The mind and heart of the negotiator. Pearson Prentice Hall, NJ (2005)
16. Tintarev, N., Masthoff, J.: A survey of explanations in recommender systems. In: ICDE'07 Workshop on Recommender Systems and Intelligent User Interfaces. Citeseer (2007)

Reducing Incentives for Agent Collaboration
(Position Paper)

Neil Yorke-Smith[1,2]

[1] Olayan School of Business, American University of Beirut, Lebanon.
[2] SRI International, Menlo Park, CA, USA.
nysmith@aub.edu.lb

Abstract. The benefits of collaborative agents include achievement of complex goals that are difficult or impossible to attain for an individual agent. Incentive schemes, market mechanisms, and organizational structures can be designed to foster collaboration. On the other hand, unwanted collaboration between small numbers of agents, at the expense of the benefit to the greater society, demands measures to disincentivize such negative, collusive collaboration.

1 Why Collaboration is Not Always Desirable

In an ideal world, cooperative agents work together to achieve greater goals for the good of the larger society. This ideal is not the world we live in. Self-interested agents act for their own selfish benefit at the expenses of others; groups of self-interested agents collude for their benefit at the expense of the larger society. Collaborators break their promises, fail to comply with norms or regulations, and even straight-out deceive. Agents can be at best careless and at worst treacherous.

Just as incentive schemes, market mechanisms, and organizational structures can be designed to foster collaboration, we argue that some situations require careful design of anti-collaboration provisions.

Now famous is the 1995 US FCC spectrum auction [1]. The rules of the auction prohibited companies from openly colluding to divide the spectrum at low cost to themselves and hence low value for the public. The major players in the auction neatly circumvented the rules, however, by using the least significant digits of their public messages to coordinate their bidding strategies. "In other words, these parties used the auction protocol itself to cheat" [15].

Electoral voting machines have been another headline topic for agents collaborating in anti-social manners [14]. How can the public have confidence that their vote will be properly recorded and accounted, and that fallacious votes will not be accounted? Supposing there is an inspection, what if certain agents (such as the machine manufacturers) collude with the inspection agents?

This position paper draws attention to the need to disincentivize such negative, collusive collaboration, and to otherwise mitigate their effects.

2 Example: Maritime Customs Collaboration

Ocean-based freight, according to the International Maritime Organization, account for 90 percent of world trade by weight [12]. The inspection of container contents and application of regulations and tariffs is a significant part of the import-export process at ports worldwide. The progress of containers through customs, however, is more often an exercise in negotiation rather than a structured queuing process. As soon as such a regulatory process involves negotiation, corruption in its various manifestations becomes a possibility.

Fig. 1 shows some possible deviations from an archetypal customs import process. These include inaccurate, incomplete, or fictitious documentation; under- or over-inspection; inaccurate value estimation; waiving true fines or imposing additional fines; and delaying or expediting certain containers. In some situations, a whole grey 'parallel customs' system evolves.

Extra-process negotiation is the most common entry point for non-standard behaviour within customs processes [11]. Agents willingly or unwillingly make private, collusive agreements. Such non-standard practices fall into three categories [16, 7]. First, deviations based on the relationship between agents, where these is no obvious monetary or physical bribe. Relationship levers in negotiation can arise from family connection (nepotism), political tie (patronage), or favour owed. Second, deviations be based on monetary considerations, where there is a tangible bribe, whether cash or gift, or a debt forgiven. Third, negotiation levers based on threats or extortion, whether physical, financial, or reputation-based.

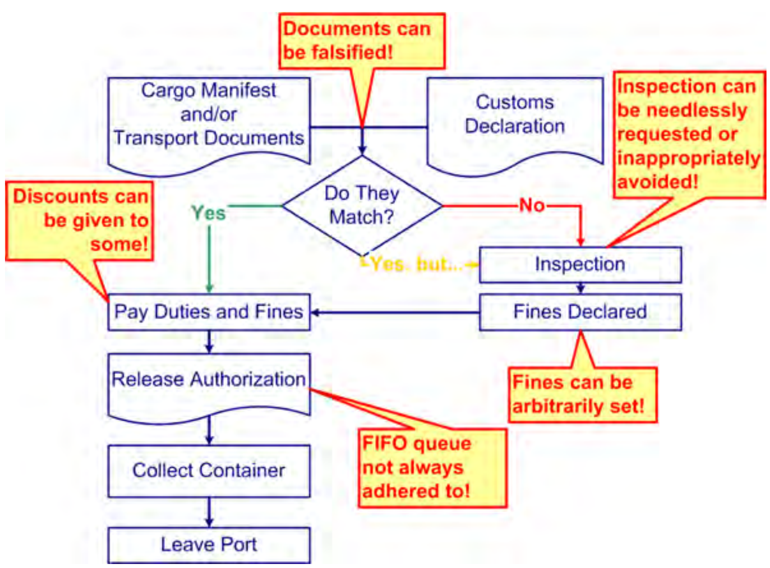


Fig. 1. An archetypal import process. Some opportunities for deviations from the published process are highlighted.

The Organization for Economic Co-operation and Development (OECD) notes, particularly for developing countries, that customs revenue is a significant component of public finances, but that customs efficiency is often hampered by widespread corruption, creating "a major disincentive and obstacle to trade expansion" and leading to "disastrous consequences in terms of national security and public finance" [8].

The effect of corruption burden communities and nations, weighing especially on the disenfranchised. It hinders development, being "one of the most serious barriers to overcoming poverty", with a strong correlation manifest between perceived corruption and national per capita income [18].

3 Disincentives, Incentives, and Process Design

Given that corruption can enter a market or process whenever there is opportunity for agents to negotiate, what can be done to reduce the opportunities and the incentives for and the impact of collusive behaviour? In certain situations, guaranteed collusion-free protocols can be deployed (see Shelat [15] for a brief overview). When it is not possible to eliminate negative collaboration entirely,[3] how can the system be made more robust to its presence?

A perfusion of legal or normative rules provide no guarantee of disincentivizing collusion. As Tacitus observed, "The more corrupt the state, the more laws." Studies by the OECD and other organizations report that customs corruption is not easily combated by policy changes, for example [8]. Further, extended processes can provide more opportunities for negotiation and hence foster rather than disincentivize deviations from the ideal.[4]

Hence a call can be issued to re-examine the research on agent collaboration and market design [6]. The call is to disincentivize the socially bad behaviours, not just incentivize socially good behaviours. It is not enough to be able to check correctness of published contracts [2]; nor can we assume agents are cooperative.

Catalysts and culture. The problem of collusion is, then, challenging for those who would wish to tackle it. Collusion can arise from within a group of agents, needing no external catalyst agent. It can be fostered, however, by the existence of external mediating agents, such as 'fixers' in a customs process. Dignum et al. [4] point to the interaction of many elements—economic, social, personal, structural, environmental—as determining the existence and role of mediator agents. Further, the broad socio-cultural environment shapes agent behaviour in negotiation [3]. Hence relevant is study of organizational behaviour, norm emergence, and societal culture (for a computational study of culture's influence on human–agent negotiation, see Gal et al. [5]).

[3] In the case of customs, such an effort would be prohibitively expensive and unrealistic. Further, draconian efforts can have adverse implications for personal freedoms and fundamental human rights.

[4] "Systems and procedures [evolve] to maximise the number of steps and approvals—to create as many opportunities as possible for negotiation" [8].

Protocols and decision aids. If we take 'agents' to refer to autonomous entities, encompassing human actors and businesses as well as automated agents, then the failed Covisint business-to-business market in the automotive industry [13] is another demonstration of the problems of cooperation—besides that of collusion—that can emerge in an auction setting.

Karlsson et al. [10] show how positive cooperation (among humans) can be incentivized through market-based protocols that allow complex bids.[5] While such protocols have theoretical and computational advantages, behavioural economics assures us that humans are not rational decision-makers. Decision aids may be needed in order for complex market-based protocols to be effective.

Mechanism design and simulation. Spectrum auction markets is one domain where collusion-resistant market mechanisms have been developed [19], motivated by the FCC experience; another is online reputation mechanisms [9]. How do these mechanisms transfer to more ill-defined processes and agent systems such as in the domain of customs? How can elusive notions such as 'benefit to society' be quantified, and taken into algorithmic account?

The connection to the agents community arises naturally through game theory and mechanism design. In addition, we suggest that simulation has a role in the study of complex multiagent processes and systems, aiding modelling, analysis, and evaluation [17, 7]. A broader question is whether agent technology can be used to build automated, semi-automated, or decision-aided systems that are more reliable than processes carried out solely by human actors.

4 Research Outlook

We conclude by enumerating relevant research questions, adapted from the Call for Papers for the CARE workshop:

- How do we design markets that hinder collusion?
- What interventions and incentives can disincentivize negative collaboration?
- How do we enforce prohibitions on illicit joint agreements and contracts?
- How do we build agent systems that work efficiently in partially-regulated environments where negative collaboration is not necessarily prohibited?
- How do we build systems or mechanisms robust to unreliable or non-conformant collaborators, and to colluding groups of agents?
- How do organizational structures influence the negotiation of agents and collusive behaviour?
- How can lessons learned in game theoretic computation inform mitigation of collusion?

[5] To 'disincentivize' is standard procedure in market programming, in the sense that protocols should be robust to speculation, i.e., speculation being unwarranted in every practical case by any rational agent in the market.

Acknowledgements. Thanks are due to Magnus Boman, Tony Feghali, and F. Jordan Srour, and to the anonymous referees. This work was partially supported by University Research Board grant A88813 from the American University of Beirut.

References

1. Cramton, P., Schwartz, J.A.: Collusive bidding: Lessons from the FCC spectrum auctions. J. Regulatory Economics 17, 229–252 (2000)
2. Desai, N., Narendra, N.C., Singh, M.P.: Checking correctness of business contracts via commitments. In: Proc. AAMAS. pp. 787–794. Estoril, Portugal (2008)
3. Dignum, V., Dignum, F., Osinga, S.A., Hofstede, G.J.: Normative, cultural and cognitive aspects of modelling policies. In: Proc. Winter Simulation Conference. pp. 720–732. Baltimore, MD (2010)
4. Dignum, V., Tranier, J., Dignum, F.: Simulation of intermediation using rich cognitive agents. Simulation Modelling Practice and Theory 18(10), 1526–1536 (2010)
5. Gal, Y., Kraus, S., Gelfand, M., Khashan, H., Salmon, E.: An adaptive agent for negotiating with people in different cultures. ACM Transactions on Intelligent Systems and Technology (to appear)
6. Guttmann, C., Dignum, F., Georgeff, M. (eds.): Collaborative Agents – REsearch and development (CARE 2009/2010), vol. LNAI 6066. Springer (2011), to appear
7. Harb, H., Srour, F.J., Yorke-Smith, N.: A case study in model selection for policy engineering: Simulating maritime customs. In: Proc. AAMAS'11 Workshop on Agent-based Modeling for Policy Engineering. Taipei, Taiwan (2011)
8. Hors, I.: Fighting corruption in customs administration: What can we learn from recent experiences? OECD Development Centre Working Paper 175 (2001)
9. Jurca, R., Faltings, B.: Mechanisms for making crowds truthful. J. Artificial Intelligence Research 34, 209–253 (2009)
10. Karlsson, M., Ygge, F., Andersson, A.: Market-based approaches to optimization. Computational Intelligence 23(1), 92–109 (2007)
11. Klitgaard, R., MacLean-Abaroa, R., Parris, H.L.: Corrupt Cities: A Practical Guide to Cure and Prevention. ICS Press, Oakland, CA (2000)
12. Maritime Knowledge Centre, International Maritime Organization: International shipping and world trade: Facts and figures. `www.imo.org` (2009)
13. McGee, M.K.: Covisint and AT&T enter health care exchange market. InformationWeek (25 Feb 2008), `www.informationweek.com/news/software/soa_webservices/showArticle.jhtml?articleID=206801143`
14. Mercuri, R.T., Camp, L.J.: The code of elections. Communications of ACM 47, 52–57 (2004)
15. Shelat, A.: Collusion-free protocols. In: Proc. Behavioral and Quantitative Game Theory: Conf. on Future Directions. pp. 91:1–91:1. Newport Beach, CA (2010)
16. Srour, F.J., Harb, H., Yorke-Smith, N.: Maritime customs negotiation with corrupt agents. In: INFORMS Annual Meeting. Austin, TX (2010)
17. Tesfatsion, L.: Agent-based computational economics: Growing economies from the bottom up. Artificial Life 8, 55–82 (2002)
18. Transparency International: The Global Corruption Report 2009. `www.transparency.org/publications/gcr` (2009)
19. Zhou, X., Zheng, H.: Breaking bidder collusion in large-scale spectrum auctions. In: Proc. ACM Intl. Symposium on Mobile Ad Hoc Networking and Computing. pp. 121–130. Chicago, IL (2010)