Decommitting in Multi-agent Execution in Non-deterministic Environment: Experimental Approach

Jiří Vokřínek
Agent Technology Center
Czech Technical University in Prague
vokrinek@agents.felk.cvut.cz

Antonín Komenda
Agent Technology Center
Czech Technical University in Prague
komenda@agents.felk.cvut.cz

Michal Pěchouček
Agent Technology Center
Czech Technical University in Prague
pechoucek@agents.felk.cvut.cz

ABSTRACT

The process of planning in complex, multi-actor environment depends strongly on the ability of the individual actors to perform intelligent decommitment upon specific changes in the environment. Reasoning about decommitment alternatives during the planning process contributes to flexibility and robustness of the resulting plan. In this article we formally introduce and discuss three specific decommitment rules: (i) relaxation, (ii) delegation and (iii) full decommitment. We argue that appropriate selection, setting and preference ordering of the decommitment rules contributes to robustness (measured as a number of failures) of the overall plans. The presented claims are supported by empirical experiments.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Artificial Intelligence—*Intelligent agents*; I.2.8 [**Computing Methodologies**]: Artificial Intelligence—*Plan execution, formation, and generation*

General Terms

Measurement, Performance, Reliability, Experimentation, Verification

Keywords

social commitment, decommitment rule, commitments based planning, non-deterministic environment

1. INTRODUCTION

The process of planning in complex, multi-actor environment depends strongly on the ability of the individual actors to perform intelligent decommitment upon specific changes in the environment. Reasoning about decommitment alternatives during the planning process contributes to flexibility and robustness of the resulting plan.

Multi-agent research community has provided a viable formalism for representing agents commitment towards their individual as well as joint intentions. Wooldridge and Jennings have formalized such mental attitude of the agents by means of special knowledge structures, referred to as *social commitments* [13].

Social commitment is a knowledge structure describing agent's obligation to achieve or maintain a specific goal, under specific conditions. The commitment does not capture description how the committed goal can be achieved. An important knowledge component in the commitment – *decommitment rule*, sometimes referred to as a *convention* [12] – provides information about how and under which circumstances the commitment can be dropped. Reasoning activities of a rational agent such as individual planning for a goal achievement, plan execution and monitoring as well as replanning, plan reparation or plan merging and plan coordination are supported by the information encoded in the social commitments.

In the context of classical planning, agents deliberate about primitive (or compound) actions, components of the plan, in order to form appropriate ordering (or decomposition) representing a result of a specific planning problem. While a typical action in a plan contains only a set of preconditions and resulting postconditions, we suggest extending the action representation with the commitment-related information. A planning agent will not only reason about preconditions and effects of an action but also about how much and in which way it can rely on someone implementing the given action. This capability is critical for agents to be able to coordinate their actions and to perform *multi-agent planning* in the sense of forming plans (*i*) by interaction among multiple autonomous agents and (*ii*) to be executed by multiple autonomous agents.

The product of planning with commitments is a set of partially ordered terminal actions, allocated to individual actors who agreed to implement the actions under certain circumstances, clustered into two categories :

- *commitment condition* that may be (*i*) a specific situation in the environment (such as completion of a particular precondition) or (*ii*) a time interval in which the action has to be implemented no matter what the status of the environment is or (*iii*) a combination of both.
- *decommitment conditions* specifying under which condition the actor is allowed to decommit from the commitment once the task is finished (e.g. notification) or once the task cannot be completed (e.g. a failure)

Planning with commitments results in a plan where agents commit themselves to carry out actions leading towards achievement of e.g. a joint persistent goal [5]. Planning strate-

Cite as: Decommitting in Multi-agent Execution in Non-deterministic Environment: Experimental Approach, Jiří Vokřínek, Antonín Komenda, Michal Pěchouček, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 977–984

gies used for establishing such a plan are usually based on the delayed-commitment principle [1]. Other approach uses eager-commitment strategy as showed in [11].

In this paper we formally introduce and discuss three specific decommitment rules: (i) relaxation, (ii) delegation and (iii) full decommitment. We argue that appropriate selection, setting and preference ordering of the decommitment rules contributes to robustness (measured as a number of failures) of the overall plans. The particular focus of this contribution is the preference ordering of the decommitment rules in various non-deterministic environments. Besides formal specification of the decommitment rules, the key research contribution lies in experimental analysis of the use of the various decommitment strategies and their mutual dependence.

The article is structured as follows. Section 2 gives a brief overview of works most relevant to our approach. In Section 3 we adapt the formal description of commitment by Wooldridge and introduce a particular commitment convention to improve the flexibility of the commitments. The verification and experimental evaluation is presented in Section 4. Finally, the last section concludes the paper.

2. RELATED WORK

Formalization of commitments has been extensively studied in the past using various formalisms, most of all building on and extending the BDI framework [9] when describing obligations the agents adopt. Fasli [4] uses a model based on branching temporal components from Computational Tree Logics (CTL) [3], which is much more expressive than our model based on temporal intervals, however it is too complex for experimental deployment. Fasli also defines strategies regarding conditions for a successful decommitment from the agent's obligations, which in several aspects correlate with Wooldridge's convention [12] and thereby with our decommitment rule set.

Another formal representation of commitments considering temporal account has been introduced in [7]. CTL has been extended to capture features usually not considered in common approaches (but relevant for realistic environments), namely time intervals considered in commitments satisfaction, *maintenance* type of commitments next to *achieve* type of commitments and vague specification of time. However, most of these aspects can be also captured using BDI and Wooldridge's social commitment framework showed in [12] in combination with an appropriate temporal model.

The uncertainty in agent's commitments has been studied in [14]. The authors have extend the commitment with *"... uncertainty by explicitly describing the possibility of future modification/revocation of the commitment ..."*. The paper concentrated on the uncertainty in the quality of the commitment fulfilment (quality of service) rather than on decommitting conditions.

The last but not least is an extensive related research field of planning in dynamic and/or uncertain environment. There is a wide range of approaches, for example probabilistic planning – MAXPLAN [6], contingency planning – system CIRCA [8], planning under uncertainty [2]. These approaches focus on creating plan alternatives to avoid uncertainty, in contrary to the commitment based planning where the emphasis is put on individual commitments and its decommitment strategies.

3. DECOMMITMENT RULES

As stated in the Introduction, the targeted topic of the research proposed in this paper is the impact of the decommitment rules on the plan execution. In this section, we provide formalization for the three most commonly used decommitment rules.

We require the agents that perform intelligent planning and replanning by means of social commitments to be able to perform at least basic reasoning about the decommitment rules attached to the particular commitments. This is needed at the time of replanning, when an agent needs to decide which decommitment rule (i.e. a new commitment) to adopt, provided that conditions for more than one of them are satisfied. Similarly, when negotiating about who will accept which commitment, the agents shall be able to analyze not only properties of the goal and costs associated with the goal completion process but also the various decommitment rules when considering likelihood of the particular failure to happen. Ideally, the agent shall be able to estimate costs of each decommitment rule. In the scope of this paper we are not addressing the agents' decision making, but we focuss on the performance and usability of several decommitment strategies settings during execution of the commitments in dynamic environment.

Michael Wooldridge in [12] defines the commitments formally as follows:

$$\begin{aligned}&(\mathsf{Commit}\ A\ \psi\ \varphi\ \lambda),\\&\quad\lambda = \{(\rho_1,\gamma_1),(\rho_2,\gamma_2),\ldots,(\rho_k,\gamma_k)\},\end{aligned} \tag{1}$$

where A denotes a committing actor, ψ is an activation condition, φ is a commitment goal, and λ is a convention. The convention is a set of decommitment rule tuples (ρ,γ) where ρ is a decommitment condition and γ is an inevitable outcome. The convention describes all possible ways how the commitment can be dropped. Generally speaking, the actor A has to transform the world state in such a way that the φ goal becomes true if ψ holds and any γ has not been made true yet. The actor is allowed to drop the commitment if and only if $\exists i : \rho_i$ which is true. A decommitment is fulfilled provided that γ_i is made true. A formal definition in modal logic (working with the models of mental attitudes like Believes, Desires, Intentions, [9], and temporal logic where the operator AG denotes inevitability and operator $\frown$ denotes the temporal until) follows as defined in [12]:

$$\begin{aligned}&(\mathsf{Commit}\ A\ \psi\ \varphi\ \lambda) \equiv\\&\quad((\mathsf{Bel}\ A\ \psi) \Rightarrow \mathsf{AG}((\mathsf{Int}\ A\ \varphi)\\&\qquad\wedge(((\mathsf{Bel}\ A\ \rho_1) \Rightarrow \mathsf{AG}((\mathsf{Int}\ A\ \gamma_1))) \frown \gamma_1)\\&\qquad\ldots\\&\qquad\wedge(((\mathsf{Bel}\ A\ \rho_k) \Rightarrow \mathsf{AG}((\mathsf{Int}\ A\ \gamma_k))) \frown \gamma_k)\\&\quad) \frown \bigvee_i \gamma_i).\end{aligned} \tag{2}$$

This definition is used in a declarative way. Provided that whatever the agent does during a specific behavior run complies with the above defined commitment, the expression 2 is valid throughout the whole duration of the run.

The structure of our commitments is based on this definition and the decommitment rule set is in detail discussed in the next section.

We have recognized three main types of decommitment usually used in commitments: (i) *Full Decommitment* for dropping the commitment, (ii) *Delegation* of the commitment to another agent, and (iii) *Relaxation* of the time frame of the commitment. The decommitment condition for each decommitment strategy is defined to enable flexibility of the commitment under various circumstances. During the planning process, the preference relation over the decommitments is defined as a part of the decommitment rule set. The decommitment rules are unordered according to the definition (1) and thus we must slightly change the definition of the commitment:

$$\begin{array}{l}(\mathsf{Commit}\ A\ \psi\ \varphi\ \lambda),\\ \quad\lambda = \big((\rho_1,\gamma_1),(\rho_2,\gamma_2),\ldots,(\rho_k,\gamma_k)\big),\end{array} \tag{3}$$

where the ordering is fixed and the rules are processed in a specific order. The processing of the rule means that dropping a part of the commitment definition in (2)

$$\bigvee_i \gamma_i \tag{4}$$

simplifies to

$$\gamma^*, \tag{5}$$

where γ^* is the inevitable outcome of an active decommitment rule. There is only one active rule for each commitment and the rules are switching from the first rule to the last one. The switch is performed only if γ^* is not realizable and ρ of the next active rule holds. The switching process uses defined fixed ordering of the rules to determine the correct succeeding rule.

According to our understanding, each decommitment rule set (corresponding to Wooldridge's commitment convention) must contain two basic rules, which ensure the racionality of the agent's decision making process. These rules are based on the definition of the *open-minded commitment* defined in [12]:

$$\begin{array}{l}(\mathsf{Commit}\ A\ \varphi) \equiv\\ \quad \mathsf{AG}((\mathsf{Int}\ A\ \varphi) \curvearrowleft ((\mathsf{Bel}\ A\ \varphi) \vee \neg(\mathsf{Bel}\ A\ \mathsf{EF}\varphi))),\end{array} \tag{6}$$

where the operator EF denotes future possibility. Thus in each and every commitment the initial rule should be the success rule

$$(false, (\mathsf{Bel}\ A\ \varphi)) \tag{7}$$

and the decommitment rule set should be always closed with a fail-safe rule turning a violated commitment eventually off

$$(false, \neg(\mathsf{Bel}\ A\ \mathsf{EF}\varphi)), \tag{8}$$

which is used provided that no other rule can be used and the commitment became unrealizable.

The decommitment condition ρ in rules (7) and (8) cannot become true (as $\rho = false$), which means the agent will never intend to the decommitment rule outcome γ according to the definition (2)

$$\begin{array}{lll}((\mathsf{Bel}\ A\ \rho) & \Rightarrow & \mathsf{AG}((\mathsf{Int}\ A\ \gamma))) \curvearrowleft \gamma\\ (false & \Rightarrow & \mathsf{AG}((\mathsf{Int}\ A\ \gamma))) \curvearrowleft \gamma,\end{array} \tag{9}$$

nevertheless the rule can drop-out the commitment using the drop-out part

$$\bigvee_i \gamma_i \text{ or } \gamma^* \tag{10}$$

respectively. This principle can be used for any drop-out rule with no need for explicit intention.

Definition 3.1. *Each commitment can be decommitted if the commitment goal φ is achieved (the commitment succeeded) or if the commitment goal φ can not be achieved any more (the commitment is violated).*

The formal definition of the decommitment rule set in each commitment follows

$$\begin{array}{l}(\mathsf{Commit}\ A\ \psi\ \varphi\ \lambda),\\ \quad\lambda = \big((false, (\mathsf{Bel}\ A\ \varphi)),\\ \qquad\quad \ldots \text{investigated decommitment rules} \ldots,\\ \qquad\quad (false, \neg(\mathsf{Bel}\ A\ \mathsf{EF}\varphi))\big),\end{array} \tag{11}$$

where the three investigated rules are injected between two basic rules and thereby the last violation rule can be avoided. The rate of avoidance is one of the experimental metrics and is discussed in Section 4.

Three proposed decommitment rules can be defined using the adopted formalism as follows:

Definition 3.2. *Full decommitment decommits the original commitment if and only if the commitment goal φ is unrealizable.*

Definition 3.3. *Delegation decommits the original commitment if and only if the commitment goal φ is unrealizable and the new commitment on the other agent's side is formed.*

Definition 3.4. *Relaxation decommits the original commitment if and only if the commitment goal φ is unrealizable, the negotiated relaxation conditions hold and the relaxed commitment is formed.*

In the three following subsections, we describe the rules in more details and we formalize them using a temporal model, based on the duration time interval of the commitment being the only constraint of the commitment goal φ. This model is suitable for commitment-based planning, since the plan is a (partially) ordered list of temporally successive commitments.

3.1 Full Decommitment

The basic decommitment strategy is dropping the commitment. Under defined circumstances the agent is completely released from the commitment.

Let the commitment time interval $T_\varphi = \langle t_s, t_e \rangle$, where t_s is the starting time and t_e is the ending time of the commitment time interval. The commitment duration is defined as $t_d = t_e - t_s$ Let the commitment goal condition φ contain only defined temporal properties, then the decommitment rules can be described as:

$$\begin{array}{l}(\mathsf{Commit}\ A\ \psi\ \varphi\ \lambda),\\ (t_s^{est} > t_s \Rightarrow update(t_s, t_s^{est}), false) \in \lambda,\\ (t_e^{est} > t_e, t_e^{est} > t_e) \in \lambda,\\ t_e \in \varphi,\end{array} \tag{12}$$

where t_s^{est} and t_e^{est} are estimations of the real start and end of the activity. The first part of the rule describes continuous adjustment of the commitment's start time in the case the agent is forced to postpone its execution (which may not affect the end time condition and can not affect other commitments). The second part reflects Definition 3.2.

The t_s, t_e are the parameters of the commitment negotiated and fixed at the planning (contracting) time. The estimates t_s^{est} and t_e^{est} are continuously updated and can vary over time.

3.2 Delegation

By using this type of the decommitment rule the agent shall be able to find some other agent who will be able to complete its commitment on the original agent's behalf. It is possible that such a commitment will contain unbound variables representing the need to search for an agent suitable for delegation. The basic idea is to find an agent that is able to undertake the commitment under circumstances when the decommitment condition (which is true in case of the original agent) became false, so the new agent is able to fulfill the commitment. The delegated commitment can contain a new set of decommitment rules.

Formally we can use the same variables as in full decommitment, but we are using

$$\begin{aligned}&(\mathsf{Commit}\ A\ \psi\ \varphi\ \lambda),\\ &\quad (t_s^{est} > t_s \Rightarrow update(t_s, t_s^{est}), false) \in \lambda,\\ &\quad (t_e^{est} > t_e, (\mathsf{Commit}\ B\ \psi\ \varphi\ \lambda)) \in \lambda,\\ &\quad t_e, B \in \varphi,\end{aligned} \tag{13}$$

where B is the other agent undertaking the commitment.

3.3 Relaxation

Relaxation is a special decommitment, where the original commitment is replaced with a new commitment with relaxed condition and/or goal. In the scope of this text we are focusing on the relaxation of the commitment time interval for the sake of simplicity. The commitment time interval is usually captured by the commitment subject φ and specifies the time frame booked for the commitment execution. The temporal uncertainty can be a part of the commitment subject definition (and thus the whole commitment has to be renegotiated in case of any change) or, more preferable, it can be included in the commitment as an instance of a decommitment rule.

According to Definition 3.4, the decommitment rule can be then described as:

$$\begin{aligned}&(\mathsf{Commit}\ A\ \psi\ \varphi\ \lambda),\\ &\quad (t_s^{est} > t_s \Rightarrow update(t_s, t_s^{est}), false) \in \lambda,\\ &\quad ((t_s^{est} < t_s) \wedge (t_s^{est} \in T_s^{rlx}),\\ &\qquad (\mathsf{Commit}\ A\ \psi\ \varphi\ \lambda) \wedge update(t_s, t_s^{est}) \in \lambda,\\ &\quad ((t_e^{est} > t_e) \wedge (t_e^{est} \in T_e^{rlx}),\\ &\qquad (\mathsf{Commit}\ A\ \psi\ \varphi\ \lambda) \wedge update(t_e, t_e^{est}) \in \lambda,\\ &\quad t_s, t_e \in \varphi,\end{aligned} \tag{14}$$

where T_s^{rlx} and T_e^{rlx} are the agreed relaxation intervals (negotiated relaxation conditions) for the start and end time. The T_s^{rlx} and T_e^{rlx} is an extended set of parameters negotiated and fixed at the planning time and the $update(t_s, t_s^{est}$ part changes the temporal parameters of the newly forming commitment to relaxed values.

3.4 Impact of Decommitment Rules

The impact of the particular rules is discussed in Section 4.2.1. We assume the complex combination of the decommitment rules provides non-trivial behavior and should improve the performance of the commitments' execution in non-deterministic environments under stress conditions (the system is overloaded). Let us postulate the following hypothesis:

HYPOTHESIS 3.1. *A proper combination of the three defined decommitment rules, i.e. relaxation stated in Definition 3.4, delegation stated in Definition 3.3, and full decommitment stated in Definition 3.2 improves the commitment execution stability in the non-deterministic environment and preserves the utilization of resources and should increase the commitment's execution success rate.*

Figure 1: Scenario island screenshot

Each of the presented rules provides different impact on the agent's current state. For example, relaxation helps to maintain the commitment execution, delegation effectively unblocks the agent's resources and full decommitment releases the agent's resources by dropping the commitment. We expect a combination of the decommitment rules to emerge in a self-adaptation pattern that should lead to some sort of a real-time commitment execution optimization.

The decommitment rules introduced in this chapter have been implemented in the commitment-based planning system and experimentally evaluated. The next section describes the experimental scenario and discusses the influence of the rules on stability of the commitments execution and decommitment flexibility.

4. EXPERIMENTS

The decommitment rules temporally formalized in Section 3 have been deployed in a realistic simulation scenario based on an island inspired by the Pacifica Suite of Scenarios[1]– Fig. 1. The scenario simulates limited information visibility and information sharing. Due to this, the environment provides non-deterministic behavior from a single unit's point of view. There are heterogenous independent self-interested units in the scenario that commit to the goals. During the execution of a plan the commitments are processed. The commitment can evolve according to the plan or due to unexpected environment interactions. Monitoring of the commitments is triggered by a change of the world, e.g. a tick of the world timer, movement of a unit, a change of a world entity state, etc. The process evaluates all commitments in the actor's knowledge base. The value of the commitment defines the commitment state and can start the decommitting process.

For the experimental evaluation purposes we have designed a multi-actor transport scenario, where individual agents provide non-accurate estimates of the transportation time (the execution time may differ because of unexpected events such as unit breakdowns, path changes, etc.). We combine decommitment strategies introduced in Section 3. The influence of the selection of strategies and ordering is analyzed by a series of experiments.

In the experimental scenario, there is a set of *resource agents* able to provide a unified resource to the *requestor agent*. The requestor agent introduces a set of tasks and allocates it to the resource agents. The allocation is done by the planning process that takes tasks one by one and finds the best resource agent for its execution. The planning pro-

[1]http://www.aiai.ed.ac.uk/oplan/pacifica

cess is based on the well-known contract-net-protocol [10] and provides an almost even distribution of tasks across the resource agents. During planning, the appropriate decommitment rules are set according to the experiment settings (see Section 4.2).

The experiments have been evaluated by the simulation, where the results have been aggregated from 10 runs for each experiment setting. For each run, random values of configuration variables have been generated. Each agent recomputes the parameters for the next ongoing commitment according to the current state and executes decommitment rules when necessary.

The decommitment rules execution differs for each commitment according to the experiment setting. The decommitment rules have been set according to the definitions in Section 3. A detailed description of the implementation follows:

Basic decommitment rules – according to Equation 11, we can use the temporal model formerly proposed; the goal φ is achieved if the commitment execution is finished not later than t_e and the commitment is violated if the execution is finished later than t_e (the goal φ cannot be achieved and the resource time frame is wasted).

Full decommitment – according to Equation 12; the t_e^{est} is computed during the simulation. If this rule applies the commitment is removed from the plan.

Relaxation – according to Equation 14; this rule can be applied before the commitment execution (relaxation of the start of the commitment t_s within T_s^{relax}) or during the commitment execution (relaxation of the end of the commitment t_e within T_e^{relax}).

Delegation – according to Equation 13; the t_e^{est} is computed during the simulation. If this rule applies the agent tries to find another agent which is able to undertake the commitment (the original commitment is delegated with no decommitment rules except the basic ones). The delegation is based on negotiation between agents, where each agent bids for the commitment undertaking. If there is no winning bidder the new commitment is not formed and the decommitment condition ρ remains true.

The delegation algorithm is based on contract-net-protocol. Each agent prepares the bid for the commitment delegation based on it's current state and commitment parameters. The bid computation is the following:

1. Let t be the current simulation time and ${}^{D}t_s$, ${}^{D}t_e$, ${}^{D}t_d$ parameters of the commitment D that has to be delegated.
2. If there is an active current commitment C in time t, compute
$$ {}^{D}t_e^{est} = {}^{C}t_e^{est} + {}^{D}t_d, $$
else set
$$ {}^{D}t_e^{est} = t + {}^{D}t_d. $$
3. If there is a next commitment N in the plan, compute
$$ t_{limit} = {}^{N}t_s^{est}, $$
else set
$$ t_{limit} = \textit{positive infinity}. $$
4. If the next commitment N contains a relaxation rule, recompute the estimate as
$$ t_{limit} = \max(T_s^{rlx}). $$
5. Compute the bidding value
$$ bid = t_{limit} - {}^{D}t_e^{est}. $$
6. If $bid < 0$ reject delegation, else send the bidding value *bid*.

The negative bidding value means that commitment D cannot be inserted to the agent's plan without breaking consequent commitments. Only the potential relaxation of the first consequent commitment is taken into account when estimating impact of delegation on the agent's plan. This approach doesn't affect the risk of subsequent commitments violation and has linear complexity, but it reduces the possibility of the delegation with comparison to the more complex methods operating with the subsequent commitments' rule sets.

4.1 Scenario Setup

The experiments show the influence of the selected decommitment rules and their order on flexibility, robustness and execution stability in the non-deterministic stressed environment.

In the experiments, the environment dynamics is simulated by non-deterministic prolonging of the activities. This dynamics is not taken into account by agents during the planning process. The prolonging events are generated for each agent individually using uniform distribution with mean value $\bar{e} = 15000$ time units and variance $\sigma^2 = 8333$ time units. Each experiment has been performed on a sequence of 10 randomly generated runs. The duration of the prolonging event varies from 0 to 14000 to evaluate the system behavior under different stress conditions and it is referred to as *repair time* t_r[2]. The system is critically overloaded when $t_r = 15000$, where the repair time is equal to prolonging events' meantime and the execution of commitments fails.

To enable the possibility of delegation rule execution we introduce vacant resources - the agents with no plans that are joining the system during execution phase and they are able to undertake delegated commitments. The number of *vacant agents* is set to 5 which produces 10% of overall free resources.

Each agent has a plan containing 100 commitments. The duration of the commitment execution t_d (ideally with no prolonging events) is randomly generated with uniform distribution from 5000 to 15000 time units. The start time of the commitment t_s is set to the earliest possible time of the winning resource agent and the end time is set to

$$ t_e = t_s + t_d, \tag{15} $$

which makes 100% load of the agents in ideal conditions (with no prolonging events taken into account). The relaxation intervals are set to

$$ T_s^{rlx} = \langle 0.7 \times t_s, 1.3 \times t_s \rangle, T_e^{rlx} = \langle 0.7 \times t_e, 1.3 \times t_e \rangle \tag{16} $$

[2] The individual agent load can be computed as $(1 + \frac{t_r}{\bar{e} - t_r}) \times 100\%$ and is varying from 0 to 1500%.

that makes 30% relaxation intervals. The commitment execution is non-interruptible, so if the decommitment rule applies after the commitment execution is started the resource is blocked for the whole t_d.

When we enable the prolonging events the overall system performance is very stressed. In the experiments we focus on the qualitative results of presented decommitment strategies rather than fine tuning commitment parameters according to current experimental settings. The evaluation and discussion is presented in the next chapter.

4.2 Results

This chapter summarizes the results of experiments based on the experimental setting described above. First, we will discuss the influence of individual decommitment rules used separately. Next, we will show the influence of the mixed strategies. We measure the number of executed decommitment rules and the number of successfully achieved commitments. Due to the over-stressed system, the utilization of agents is 100%, thus this parameter is not evaluated.

4.2.1 Single Rule

The first experiment provides the results of influence to the commitment execution for single rules usage. The delegation (D), relaxation (R) and full decommitment (Fd) rules have been used separately. For comparison we also measured the empty decommitment set noted as basic. For $t_r = 0$ there are zero rule executions and 100 successful commitments. The individual agent stress experiment results are the following (see Figures 2 and 3):

Basic – the number of successful commitments varies from 0 to 2 in the whole range. No decommitment rules are executed.

Full decommitment – the number of rule executions grows with the increasing t_r. The curve converges to the maximum number of commitments for a critically overloaded system. The number of successful commitments decreases with increasing t_r from 18 and converges to 0 for the critically overloaded system.

Delegation – the number of rule executions corresponds to possibility of delegation to the vacant agents. When vacant agents are saturated the delegation uses agents freed by the delegation of longer commitments. The number of successful commitments goes from 19 to 10. The variance between agents starts to be significant when $t_r > 10000$ so the robustness of this rule is decreasing.

Relaxation – the relaxation rule provides the best stability. It is executed for every commitment and provides no violations when the system is overloaded below the relaxation interval limitations. When the relaxation interval fails, the number of executed rules goes to 0 very fast and so does the number of successful commitments.

The delegation rule execution is also affected by the number of vacant agents. The relaxation and full decommitment rules are obviously not affected by the number of vacant agents. The second experiment examines this dependency. Figure 4 shows a typical curve shape for 5 vacant agents and

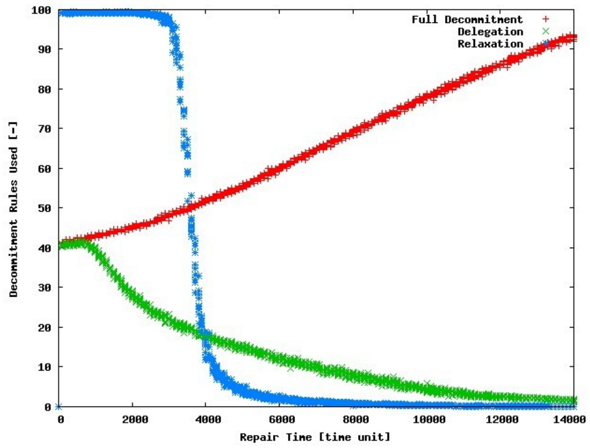

Figure 2: Number of decommitment rules execution for different t_r in single rule setting.

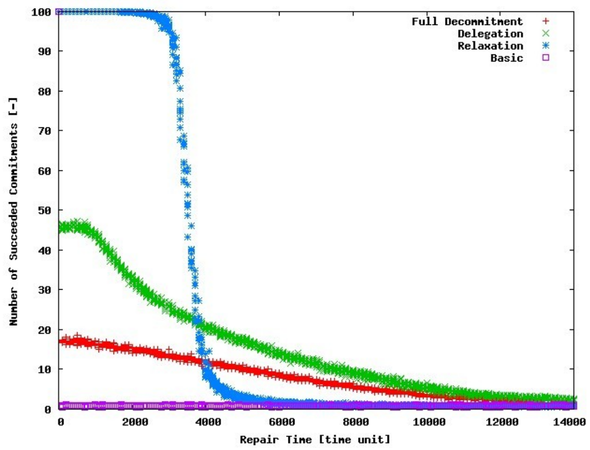

Figure 3: Number of succeeded commitments for different t_r in single rule setting.

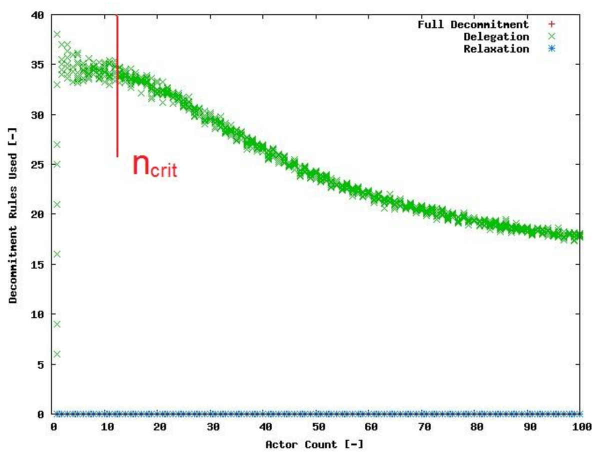

Figure 4: Number of delegation rule execution for different number of resource agents. The number of vacant agents is 5 and $t_r = 2000$

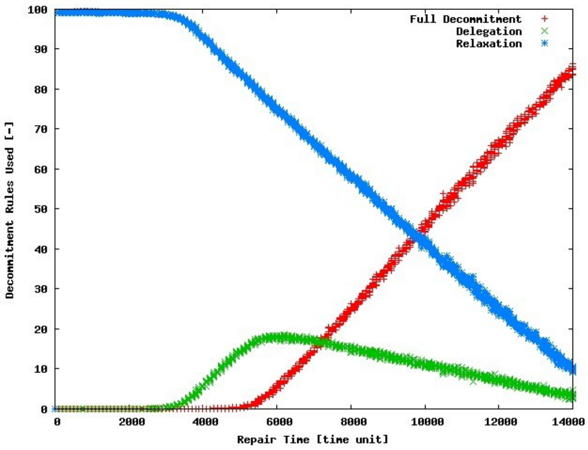

Figure 5: Number of rules execution for varying t_r in combined rule R-D-Fd setting.

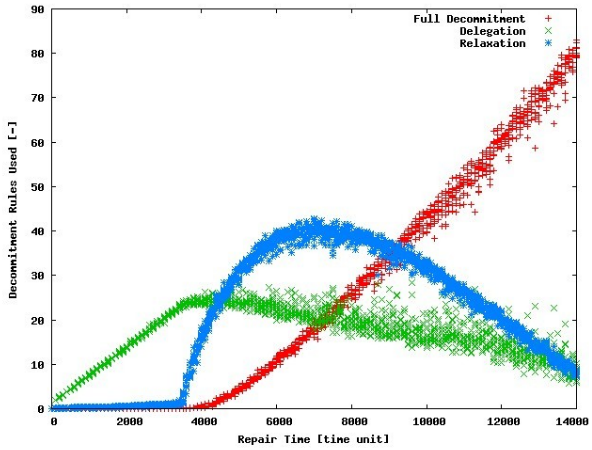

Figure 6: Number of rules execution for varying t_r in combined rule D-R-Fd setting.

$t_r = 2000$. This shape remains the same even for a different setting of the parameters. The number of decommitment rule executions is constant until the number of agents reaches the critical value n_{crit}. After this point the number of executions decreases because of saturation of the vacant agents. At this stage, the commitments are delegated mainly to other resource agents (the influence of vacant agents is decreasing). The position of the n_{crit} point depends on the experiment setting. With an increasing number of vacant agents the n_{crit} shifts to the right and with an increasing t_r it shifts slightly to the left. It's position corresponds to the relative number of vacant agents in the system and the individual agent's stress (the vacant agents are saturated sooner with increasing t_r).

4.2.2 Combined Rules

This set of experiments inspects the influence of decommitment rules combinations and their ordering. The main focus is on the two ordering scenarios – R-D-Fd for

$$relaxation \succ delegation \succ fulldecommitment$$

and D-R-Fd for

$$delegation \succ relaxation \succ fulldecommitment$$

that provide the most significant results. The full decommitment rule is ordered as the last one, because of its nature – no decommitment rule can be applied after its application.

The combination of rules provides complex results. The number of individual rules execution can be seen in Figures 5 and 6. The number of the full decommitment rule executions is similar in both cases but with different impact on the number of successful commitments. In the first part of the chart, when the system is *slightly overloaded* ($t_r < 3000$), only the first rule applies. In the range of an *overloaded* system ($t_r \in \langle 3000, 12000 \rangle$) the second and the third rule starts to apply. When the system is close to the *critical overload* ($t_r > 12000$) the number of rules executed starts to provide increasing deviation across the experiments runs and the robustness of the execution is reduced.

The number of successful commitments is compared for R-D-Fd and D-R-Fd with D-Fd, R-Fd, R-D and D-R sets and presented in Figure 7.

4.3 Discussion

The experiments show that for an overloaded system there is an increasing number of dropped commitments using full decommitment rule. The full decommitment rule applies when all preceding rules fail. This rule effectively release resources originally booked for dropped commitments. This causes a bigger chance for delegation of commitments and space for relaxation. Delegation rule provides the ability of real-time re-allocation of commitments according to current agents performance. The experimental results shows the ability of the system to adapt to the overload and thus to increase the number of succeeded commitments with increasing size of decommitment rule set and keep high utilization of available resources (execution time of the commitments compared to free time of resources excluding prolonging events).

As shown in Figure 7, the number of successful commitments is reaching 50% for R-D-Fd (D-R-Fd) for $t_r = 7500$ (7000) that corresponds to system load of 200% (187%). At this point, the R-D-Fd (D-R-Fd) method is able to utilize 100% (94%) of the overall system resources available. For $t_r = 7500$, the single rule settings (including basic rule set with no decommitment rules) reach maximum of 15% of succeeded commitments for delegation rule (Figure 3), which is 30% of utilization of available resources. Combined rule sets composed from decommitment rule pairs reach maximum of 30% of succeeded commitments, which is 60% of utilization of available resources.

This experimental observations prove Hypothesis 3.1 for any combination of rules. The best performance provides the biggest sets of decommitment rules. The R-D-Fd set has the biggest success rate of commitments execution until $t_r = 7000$. For bigger t_r the higher success rate can be observed for D-R-Fd, but with lower stability (higher variation of experiment runs). The best success rate for near critical load ($t_r > 10000$) can be reached with D-R set, but with minimal stability (most of the experiment runs provide worse results then both R-D-Fd and D-R-Fd).

The sets containing Fd provides generally better results, but may not be suitable in all application domains because of commitment drop-out by this rule.

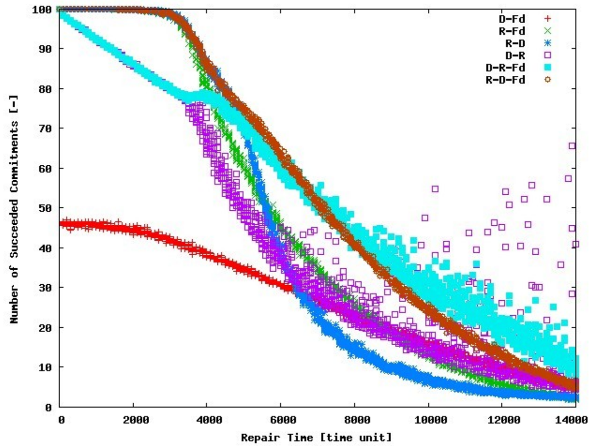

Figure 7: Number of succeeded commitments for varying t_r for different rules settings.

5. CONCLUSION

Based on the well-known social commitment representation we have defined basic decommitment rules for open-minded commitments representation. We have formalized three decommitment strategies (relaxation, delegation, and full decommitment) and showed how they affect application of the commitments in non-deterministic stressed environment. The evaluation of the presented approach has been made on an experimental realistic scenario and deployed in a multi-agent system for commitment-based distributed planning.

The relaxation decommitment rule provides the best performance in the limits of an estimated relaxation model. The delegation rule produces a relatively high amount of violations (even for small t_r) because of the usage of the non-optimal algorithm. Further improvement of this algorithm (e.g. by involving future commitments' decommitment rule sets) may lead to reduction of the number of violations and all the results affected by delegation rules may scale down. The full decommitment rule significantly reduces the number of violated commitments under high load, but may not be suitable for all applications (because of commitment drop-out).

The combinations of particular rules provides a complex decommitment behavior and significantly improves commitment execution performance and stability. The success rate of commitment execution and available resources utilization significantly increases with the size of the decommitment rule set. Different rule combinations have to be chosen for different application scenarios. We have identified, evaluated and discussed the strong and weak points of the presented combinations of decommitment rules.

6. ACKNOWLEDGMENTS

Presented effort was sponsored by Czech Ministry of Education grant 6840770038, supported by the European Research Office of the US Army under grant number W911NF-08-1-0041, and by Internal Grant of Czech Technical University under number CTU0802913.

The authors' organizations and research sponsors are authorized to reproduce and distribute reprints and on-line copies for their purposes notwithstanding any copyright annotation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of other parties.

7. REFERENCES

[1] A. Barrett and D. S. Weld. Partial-order planning: Evaluating possible efficiency gains. *Artificial Intelligence*, 67(1):71–112, May 1994.

[2] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

[3] E. Emerson and J. Srinivasan. Branching time temporal logic. In *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency, School/Workshop*, volume 354 of *LNAI*, pages 123–172. Springer-Verlag, 1988.

[4] M. Fasli. On commitments, roles, and obligations. In B. Dunin-Keplicz and E. Nawarecki, editors, *CEEMAS 2001*, volume 2296 of *LNAI*, pages 93–102. Springer-Verlag Berlin Heidelberg, 2002.

[5] H. Levesque, P. Cohen, and J. Nunes. On acting together. In *AAAI-90 Proceedings, Eighth National Conference on Artificial Intelligence*, volume 2, pages 94–99, Cambridge, MA, USA, 1990. MIT Press.

[6] S. M. Majercik and M. L. Littman. MAXPLAN: A new approach to probabilistic planning. In *Artificial Intelligence Planning Systems*, pages 86–93, 1998.

[7] A. Mallya, P. Yolum, and M. Singh. Resolving commitments among autonomous agents. In F. Dignum, editor, *Advances in Agent Communication. International Workshop on Agent Communication Languages, ACL 2003*, volume 2922 of *LNCS*, pages 166–82. Springer-Verlag, Berlin, Germany, 2003.

[8] D. J. Musliner, E. H. Durfee, and K. G. Shin. CIRCA: A cooperative intelligent real time control architecture. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(6):1561–1574, - 1993.

[9] M. P. Singh, A. S. Rao, and M. P. Georgeff. *Multiagent Systems A Modern Approach to Distributed Artificial Intelligence*, chapter Formal Methods in DAI: Logic Based Representation and Reasoning, pages 201–258. MIT Press, Cambridge, MA., 1999.

[10] R. G. Smith. The contract net protocol: High level communication and control in a distributed problem solver. *In IEEE Transactions on Computers*, C-29(12):1104–1113, 1980.

[11] M. Veloso and P. Stone. Flecs: Planning with a flexible commitment strategy. *Journal of Artificial Intelligence Research*, 3:25–52, 1995.

[12] M. Wooldridge. *Reasoning about Rational Agents.* Intelligent robotics and autonomous agents. The MIT Press, 2000.

[13] M. Wooldridge and N. Jennings. Cooperative problem solving. *Journal of Logics and Computation*, 9(4):563–594, 1999.

[14] P. Xuan and V. R. Lesser. Incorporating uncertainty in agent commitments. In *In: Proc. of ATAL-99*, pages 221–234. Springer-Verlag, 1999.