Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 39–45
ISBN 978-83-60810-14-9
ISSN 1896-7094

Planning and Re-planning in Multi-actors Scenarios by means of Social Commitments

Antonín Komenda, Michal Pěchouček, Jiří Bíba, Jiří Vokřínek
Gerstner Laboratory – Agent Technology Center
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University
Technická 2, 16627 Praha 6, Czech Republic
{antonin.komenda, michal.pechoucek, jiri.biba, jiri.vokrinek}@agents.felk.cvut.cz

***Abstract*—We present an approach to plan representation in multi-actors scenarios that is suitable for flexible replanning and plan revision purposes. The key idea of the presented approach is in integration of (i) the results of an arbitrary HTN (hierarchical task network) -oriented planner with (ii) the concept of commitments, as a theoretically studied formalism representing mutual relations among intentions of collaborating agents. The paper presents formal model of recursive form of commitments and discusses how it can be deployed to a selected hierarchical planning scenario[1].**

I. INTRODUCTION

COOPERATION between intelligent agents is usually established by means of negotiation resulting in a set of obligations for the participating agents that lead onwards to achievement of a common goal agreed to by the agents. Wooldridge and Jennings formalize the obligations by describing the cooperative problem solving by means of *social commitments* [1]—the agents commit themselves to carry out actions in the social plan leading onwards to achievement of their joint persistent goal [2].

The problem of distributed planning (DP) has been often discussed in the AI planning and multi-agent research communities recently (e.g. [3], [4], [5], [6]). Distributed planning has been viewed as either (*i*) planning for activities and resources allocated among distributed agents, (*ii*) distributed (parallel) computation aimed at plan construction or (*iii*) plan merging activity. The classical work of Durfee [3] divides the planning process into five separate phases: task decomposition, subtask delegation, conflict detection, individual planning and plan merging.

The distributed planning approach proposed in this paper does not provide constructive algorithms for dealing with either of the phases. Instead we propose a special mechanism for plan execution in distributed, multi-actor environment. As such it will affect all the phases of the Durfee's distributed planning architecture.

While classical planning algorithms produce a series of partially ordered actions to be performed by individual actors, we propose an extension of the product (but also an object) of the planning process so that it provides richer information about the context of execution of the specific action. The context shall be particularly targeted towards mutual relation between the actions to be performed by individual actors and shall be used mainly for replanning and plan repair purposes.

The planning problem we are trying to deal with can be informally understood as the task of solving a classical HTN (hierarchical task network) planning problem, defined by an initial partially ordered (causally connected) series of goals, by a set of admissible operators (defined by their preconditions and effects) and methods suggesting a decomposition of a goal into a lower-level planning problem. The plan can be sought for by an individual actor or in collaboration of multiple actors (sharing knowledge and resources). The product of planning is a set of partially ordered terminal actions, allocated to individual actors who agreed to implement the actions under certain circumstances. These circumstances are expressed by specific commitments including the following pieces of information:

- *commitment condition* that may be (*i*) a specific situation in the environment (such as completion of some precondition) or (*ii*) a time interval in which the action is 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 recommit from the commitment once the task is finished (e.g. notification) or once the task cannot be completed (e.g. a failure)

For long, multi-agent research community has been providing interesting results in the formal work in the field of agents' social commitment, as specific knowledge structures detailing agents individual and mutual commitments. The presented research builds on and extends this work.

The article is structured as follows. In the section II, the formal description of commitments by Wooldridge is extended, a recurrent notation formalizing the commitments is presented and its use for distributed planning purposes is shown using a scenario for verification. The section III gives a brief overview of the most relevant works to our approach. Finally, the last section concludes the paper.

---

[1] The presented research has been supported by I-GLOBE (http://i-globe.info/), the US ARMY, CERDEC project (no.: N62558-06-P-0353). The project has been performed in cooperationwith University of Edinburgh, Artificial Intelligence Application Institute

II. COMMITMENTS FOR PLANNING AND RE-PLANNING

As stated in the introduction, a social commitment is a knowledge structure describing an agent's obligation to achieve a specific goal, if a specific condition is made valid and how it can drop the commitment if it cannot be achieved. The commitment does not capture description how the committed goal can be achieved. Individual planning for a goal achievement, plan execution and monitoring is a subject of agents internal reasoning processes and is not represented in the commitment.

In the context of the planning problem defined in the Introduction, we understand the agent's specific goal (to which it commits) as an individual action, a component of the plan, which resulted from the given planning problem. While typical action in a plan contains only a precondition and an effect, in this paper we will describe how its representation can be extended so that the commitment-related information is included.

Michael Wooldridge in [7] 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 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 valid. A decommitment is allowed provided that γ_i is made true. A formal definition in modal logic (working with the models of mental attitudes like Believes, Desires, Intentions, [8], and temporal logic where the operator AG denotes an the inevitability and operator $\curvearrowleft$ denotes the temporal until) follows as defined in [7]:

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

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.

One of the goals of the research described in this paper was to provide a formalism for networked commitments to be used for replanning. As clearly stated in the introduction, the commitment conditions can represent variable bindings among preconditions and effects of the individual commitments achieved either by monitoring the environment status or by inter-agent communication (e.g. reception of a specific trigger message). Such representation would be very inflexible in practical applications as it would either need the agents to do nothing and wait for an inhibiting event to happen or risk that once an inhibiting event happens the agent will be busy performing other commitments. Therefore the agents may want to engage in booking and the commitment's precondition would contain fixed time when the commitment is supposed to be adopted. The most flexible approach would be a combination of both—inhibition event and preliminary booked time window, specifying when the inhibiting event is likely to happen. Let us assume that this is the case in the remainder of the paper.

In the distributed plan execution a failure may occur. The indirect impact of this failure may be e.g. a situation where the arranged inhibition event will not happen in the preliminary booked time window. Such occurrence may invoke replanning and allow some agents to e.g. drop unnecessary commitments. This is the reason why the commitments shall not be linked one with other not only via preconditions but also by means of variable bindings among individual agent's decommitment rules. Using these bindings, we can describe the causal sequentiality of the commitments and requests for particular decommitments—Fig. 1.

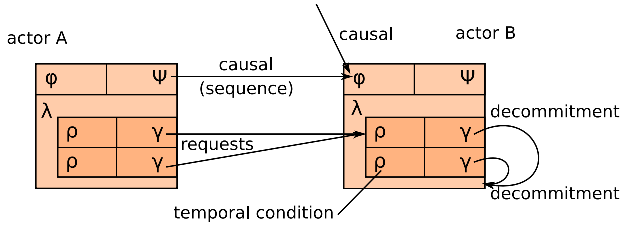

Fig. 1. Commitments and bindings—the actor A's commitment influences the actor B's commitment using the causal (sequential) link, the link is described using the ψ and φ clauses (e.g. $\psi=$ `building-is-ready(B)` and $\varphi=$ `ready(B)`). The actor B's commitment is influenced by external causality too. The actor B's commitment can be decommitted in two cases: either the *temporal condition* ρ becomes true or one of the actor A's rules *requests* the decommitting. The decommitment request is triggered by one of the actor A's ρ conditions.

While we will be generalizing on the process of decommitment later in the paper, let us work for now with the specific particular decommitment case suggested in the previous paragraph. Let us assume one agent A forcing decommitment of the other agent's B commitment by means of setting a value of a variable contained in the other agent's commitment. The agent A contains a commitment with a decommitment rule in the form $\langle\rho,v\rangle$ and the agent B contains a commitment with a decommitment rule in the form $\langle v,\mathtt{decommit}(B)\rangle\in\lambda_A$.The request is started by ρ precondition of the actor A (e.g. decommitting the A's commitment). Thus the actor A intends to make the variable v valid. This causes the agent B to intend to decommit by intending the variable `decommit`(B) to be valid (see Fig. 1).

This clear example uncovers two needed extensions of the classical social commitment model: (*i*) recurrence of the commitment form—enabling a possibility to disable (decommit) a decommitment request and (*ii*) explicit termination condition—describing termination without any intentional part.

A. Commitment Recurrence

The original Wooldridge definition of a commitment makes a clear distinction between the commitment subject (φ) and the mini-goals set in the commitment convention (γ). While there is a mechanism for the agent to drop φ, a once adopted mini-goal γ cannot be decommitted. Due to high dynamism and uncertainty of the target scenario, we assume the re-planning and plan repair mechanisms to be substantially more complex. We require that the mechanism would allow the agent to try out several different decommitment alternatives, based on the current properties of the environment. The set λ, allows listing various different decommitment rules, while no mechanism have been specified how different decommitment alternatives are tried out.

That is why we propose generalization of the commitment so that each goal in the commitment structure can be treated equally. Let us introduce the recursive form of a commitment, which enables the nesting of the commitments—Fig. 2:

$$\begin{aligned}&(\mathsf{Commit}\ A\ \psi\ \varphi\ \lambda^*), \lambda^* = \\ &\quad \{(\mathsf{Commit}\ x_1\ \rho_1\ \gamma_1\ \lambda_1^*), \\ &\quad (\mathsf{Commit}\ x_2\ \rho_2\ \gamma_2\ \lambda_2^*), \ldots, \\ &\quad (\mathsf{Commit}\ x_k\ \rho_k\ \gamma_k\ \lambda_k^*)\}.\end{aligned} \tag{3}$$

The formula 3 extends the definition in 2 not only by inclusion of a set of decommitment rules in each of the individual decommitment rules. It also allows the newly adopted commitments to be assigned to different actors. The delegation kind of decommitment between two agents A and B would have the following form:

$$(\mathsf{Commit}\ A\ \psi\ \varphi\ \{(\mathsf{Commit}\ B\ \rho\ \varphi\ \emptyset)\}), \tag{4}$$

representing that agent A can drop the commitment towards φ provided that ρ is valid and provided that B accepts a commitment towards φ on A's behalf.

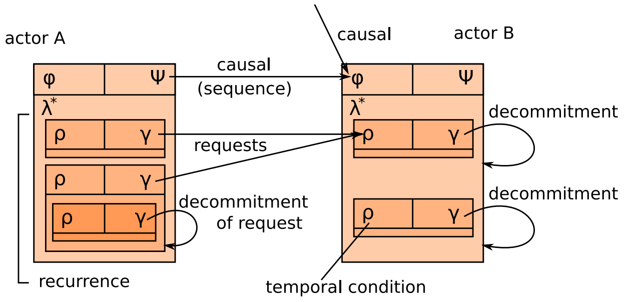

Fig. 2. Commitment and its λ^* commitments—the Fig. 1. is extended by one *decommitment of request* which can be decommitted if the most inner ρ condition becomes true. Decommitting of the request causes the actor B's commitment cannot be decommited by the actor A's convention goal any more. Here the recursive form enables the nesting of the inner commitment.

This form is very expressive in the sense of the description of exceptional states. It allows us to have a branched chain of individual nested commitments for each individual situation. The recursive nature allows us to describe an arbitrarily complex protocol using only one knowledge base structure—a recursive form of the commitment. The recursive form of the commitment is thus defined as:

$$\begin{aligned}&(\mathsf{Commit}\ A\ \psi\ \varphi\ \lambda^*) \equiv \\ &\quad ((\mathsf{Bel}\ A\ \psi) \Rightarrow \mathrm{A}((\mathsf{Int}\ A\ \varphi) \wedge \bigwedge_j \lambda_j^*) \curvearrowleft \bigvee_i \gamma_i).\end{aligned} \tag{5}$$

B. Termination Condition

We have explained in Section II that if the agent complies with the commitment, the formula 2 is always valid. However, this implication is not bidirectional. If we use this commitment definition for writing a computer program, running the behavior of an agent, we would need that all the runs that can be implemented by the formula 2 implement agent's correct commitment. In order to do this we need to show how a termination condition can be modeled by means of the social commitment. Let us assume we wanted to implement e.g. the blind commitment. According to [7] the blind commitment is defined as

$$(\mathsf{Commit}\ A\ \varphi) \equiv \mathsf{AG}(\mathsf{Int}\ A\ \varphi) \curvearrowleft (\mathsf{Bel}\ A\ \varphi) \tag{6}$$

Here the term $\mathsf{Bel}(A\ \varphi)$ is the simplest example of a termination condition. The termination condition here would be described using the λ^* commitment as follows:

$$(\mathsf{Commit}\ A\ \mathtt{false}\ (\mathsf{Bel}\ A\ \varphi)\ \emptyset). \tag{7}$$

A general termination condition t in the commitment model can be defined as follows:

$$(\mathsf{Commit}\ A\ \mathtt{false}\ t\ \emptyset). \tag{8}$$

The condition ($\mathtt{false}$) will never trigger the intention towards the termination condition—t. Termination condition of the rule plays an important role here as it will be added to the *until*-part of the commitment and allows the commitment to be valid even if the intention is dropped provided that the termination condition t is valid. Therefore we can extend the set of decommitment rules with a set of termination conditions $\mathcal{T}$ as follows:

$$\begin{aligned}&(\mathsf{Commit}\ A\ \psi\ \varphi\ \{\lambda^* \cup \mathcal{T}\}), \mathcal{T} = \{t_1, \ldots, t_k\} \\ &(\mathsf{Commit}\ A\ \psi\ \varphi\ \{\lambda^* \cup \mathcal{T}\}) \equiv \\ &\quad ((\mathsf{Bel}\ A\ \psi) \Rightarrow \mathrm{A}((\mathsf{Int}\ A\ \varphi) \wedge \bigwedge_j \lambda_j^*) \curvearrowleft \bigvee_i \gamma_i \bigvee_k t_k).\end{aligned} \tag{9}$$

C. 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 are satisfied. Similarly, agents, when they negotiate about who will accept which commitment, 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. However, with the lack of information about the dynamics of the environment, we will be only able to partially order the decommitment rules by assigning them to different types. Let us introduce four different types of decommitment rules:

- *Termination conditions* (TC)—as described in the Section II-B. These are obviously the most preferred decommitment rules as no further action is required for dropping the particular commitment.
- *Individual commitments* (IC)—commitments that do not involve other agent than the agent itself. These commitments shall be used if the impact of a failure within the multi-agent community shall be minimized. Individual commitments shall represent several other ways how an agent can accomplish a given task.
- *Delegation* (D)—by using this type of commitments 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.
- *Joint commitments* (JC)—these commitments provides mutually linked commitments (of several agents) via decommitment rules. In a replanning situation the joint commitments proactively assure that the cost of the failure is minimized. An example of the use of a joint commitment is decommitting another agent's linked commitment as explained in the Section II
- *Minimal social commitment* (MSC)—is the classical type of decommitment, where the agent is required to notify the members of the team about its inability to achieve the commitment.
- *Relaxation* (R)—is a special decommitment, where the original commitment is replaced with a new commitment with relaxed condition and/or goal. The new commitment must be consistent with all other bound commitments. Provided that the bound commitment is of other agent, the relaxation must be negotiated. The asked agent tries to fit the requested relaxed commitment into its knowledge base and eventually use some other decommitment rules of other commitments to change it and fulfill the request.

During the replanning process, the preference relation over the commitments is TC $\succ$ IC $\succ$ D $\succ$ JC $\succ$ MSC. The preference of R can be arbitrary managed by the agent in consideration of current circumstances.

D. Commitment Graph

Using the extended form of the social commitment we can propose a graph notation of the commitments. The mutual bindings and commitments form a commitment graph—Fig. 3. The commitment graph describes the same properties of the mutual decommitting as the logical notation.

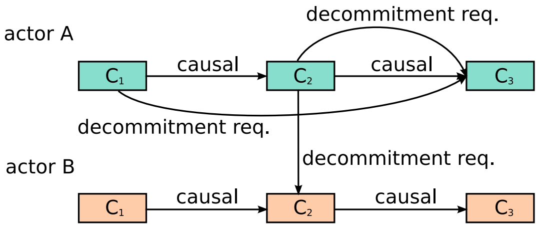

Fig. 3. Commitment graph—the *causal* links define the sequentiality of the commitments of each actor. The commitment C_3 of the actor A can be decommitted by both C_1 and C_2 commitments. The C_2 commitment of the actor B can be decommitted by actor A using the decommitment request.

The graph notation can be used to describe the process of the successive solving of the exceptional states. The process is based on the traversing through the commitment graph. The traversing starts with the first violated commitment. One of the decommitment rules is triggered (according to the violation type). As the decommitment rule is a commitment it starts an intetnion of the agent to terminate the commitment. In the case, that the intention is a decommitment request, the process crosses on the requested commitment (decommitment rule respectively) and starts one of the decommitment rules on the side of the requested commitment. Provided that the decommitment rule terminates the commitment wihout a need to request other decommitments, the process ends here and the violation is fixed.

E. Deployment Scenario

The approach presented in this paper is being verified on a realistic simulation scenario—Fig. 4. The scenario is based on an island inspired by Pacifica Suite of Scenarios[2]. On the island, there are cities and a net of roads connecting them, but off-road movement is also enabled. There are also several seaports and airports. The scenario actors are several unit types (ground, armored, aerial or sea units), civilians and non-friendly units.

There are ground units, which are *Transporters* (can provide faster transportation of other unit(s), material or civilians), *Construction* (can repair damages or assemble/disassemble stationary units) and *Medical* (provides medical care for other units or some rescue operations). The *Armored* units for protection of other units or secure an area or convoy. The *Aerial*—the UAVs with an extended visibility range and *Sea* units for transportation over the water.

The scenario simulates limited information visibility and sharing. Due to this, the environment provides non-deterministic behavior from the single unit point of view. There are heterogenous independent self-interested units in the scenario that commit to the shared/joint goals. To fulfill the desired strategic goals in such environment, the units provide

---

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

Fig. 4. Scenario island screenshot

complex cooperative actions on several levels of planning and control.

Planning and control of activities of individual units and actors in the scenario is loosely structured into three levels of detail. We recognize several layers of coordination and control:

- **Strategic layer:** The actors use aggregated meta-information from the tactical layer. This layer provides an overall strategic plan for middle and long term time horizon. High level planning and peer-to-peer coordination among the actors is possible (while non-transparent to the tactical level).
- **Tactical layer:** On this layer, the units use aggregated information from the individual layer, the information obtained through communication with each other and the information obtained from the strategic layer. The units and actors use classical planning and cooperation methods and can create new goals or adapt the goals received from the strategic level.
- **Individual layer:** On this layer, the units should perform reactive behavior based on obtained information and current goals.

The suggested coordination is hierarchical with respect of type of unit, area of operation and visibility. Three-layer architecture enables to separate middle- and long-term strategic planners from the real-time planning and control on the tactical and individual level. The strategic planner can utilize advanced planning methods with using aggregated meta-data from the whole system. On the other hand, the tactical planner has to provide real-time response and it uses limited information provided by individual layer of respective unit. On the tactical level local cooperation and information sharing of the field units is provided.

Each layer produces particular commitments and these commitments define the plan.

The strategic layer uses the HTN I-X planner [9] and a distributed resource allocation algorithm. The planner uses an abstract sub-domain derived from the scenario domain and produces an abstract plan. This plan is instantiated using negotiation about the resources—Fig. 5.

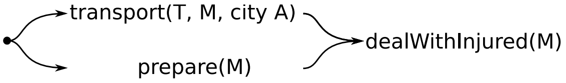

Fig. 5. Instantiated strategic plan—the medic unit M was requested by the commander agent to fulfill a task: deal with the injured in city A, and it negotiated the transport with the transport unit T.

The instantiated plan is converted into commitments—Fig. 6. The conversion process creates a commitment according to the particular plan action ($\varphi = a$) and according to forward causality links of the plan.

The commitments of the tactical layer are based on strategic commitments. The layer uses negotiation to form the most suitable mutual commitments. The constraints for the negotiation respects the particular needs of the agents. The tactical commitments also define recommitments to the strategic layer and they can additionally refine some strategic commitment too. They are much more refined than the strategic commitment in the sense of spatio-temporal constraints, and particular world-states. The tactical commitments are most enriched by the λ^* commitments. Thus, the most important part of the decommitting / replanning process is done by this layer.

An example of the tactical negotiation can be: A transport unit T is planning the tactical commitment $moveto(l_1, l_2)$, it can find out it needs support from another unit. In this case, a negotiation process must find an appropriate support unit S_p that proposes the most complying commitment (e.g. in terms of temporal constraints). If such a unit is found the JC is established, planned, and connected to other commitments in the knowledge base.

And finally, the individual layer plans commitments for later execution. These commitments copy the tactical commitments, but some of these can be omitted (e.g. $atPosition$ in the Fig. 6). Each individual commitment contains a decommitment request only to its parent commitment (from the tactical layer).

During the execution of the plan the commitments are processed. The commitment can evolve (Section II-C) according to the plan or due to unexpected environment interactions.

The 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.

One of the response to the unexpected situation can be relaxation. For instance, if a truck T commits itself C_1 to move to position l_1 exactly at time t_1 and it faces an unknown risk combat zone the commitment has to be decommitted (because the time t_1 cannot be satisfied). So, T tries to relax the commitment and thus changes the time constraint to t_2 (it plans a new route to l_1). And because the next bound commitment C_2 is constrained by time interval $\langle t_{min}, t_{max} \rangle$ where $t_2 > t_{min}$ and $t_2 < t_{max}$ the C_2 has not to be decommitted.

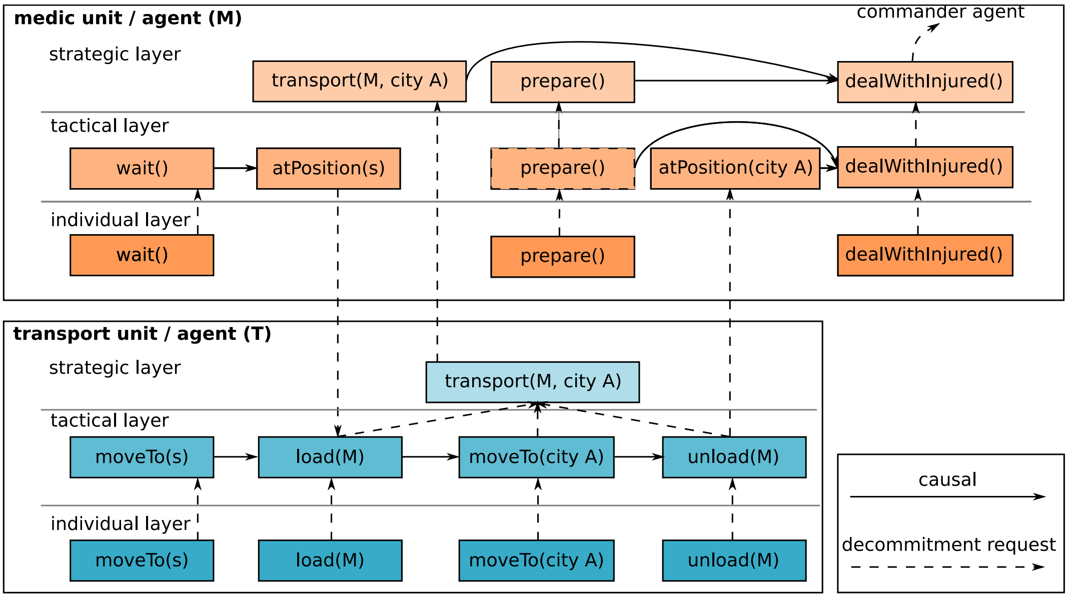

Fig. 6. Commitment bindings of multi-layer architecture for two units—the medic unit M is committed to fulfill a task: deal with the injured in city A, and the transport unit T is committed to transport the medic unit M to city A. The figure shows the directions of the potential decommitment propagation among the layers of the actors.

Another example is the delegation of the commitment using negotiation. The current agent A has to find a replacement agent B. If B is found the agent's commitment passes on to the B. B must integrate the commitment into its plan in the sense of the λ^* commitments. The process of choosing B is based on a measure of necessity to modify the current commitments of the proposing agent. For instance, let us have three trucks T_1, T_2 and T_3 and two builders B_1 and B_2. T_1 and T_2 commit B_1 and B_2 to move them to a location l. Let us assume that during the transport a problem occurs and as a result of it T_1 is no longer able to fulfill the commitment. In this situation, the commitment can be passed on either to T_2 or T_3. Since T_3 is idle, it is more appropriate to pass the commitment on to T_3 rather than T_2. T_2 would have to replan the current transportation commitment and all its successors.

The last example can be used to describe the usage of the nested commitments (commitment recurrence) too. In the case that T_3 cannot be accidently used, the T_1 cannot delegate the task to the T_3. This fact can be described using decommitting of the delegation decommitting rule in the T_1's commitment base. Formally:

$$\begin{aligned}
&(\mathsf{Commit}\ T_1\ \texttt{true}\ \texttt{is-transported(}B_1\texttt{)}\ \{\\
&\quad(\mathsf{Commit}\ T_1\ \texttt{immobile(}T_1\texttt{)}\ \texttt{delegated-to(}T_3\texttt{)}\ \{\\
&\qquad(\mathsf{Commit}\ T_1\ \texttt{immobile(}T_3\texttt{)}\ \texttt{true}\ \emptyset)\ \},\\
&\quad\dots \mathit{other\ decommitment\ rules}\\
&\})).
\end{aligned}\tag{10}$$

An agent can make a decision whether it is more suitable to re-run the strategic planning (which can be very costly and can lead to replanning of all plans of all other agents) or relax the commitment on its own (which would be probably a much less expensive operation).

III. 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 when describing obligations the agents adopt. Fasli [10] distinguishes two classes of obligations—general and relativized—and adoption of a social commitment by an agent is described as an adoption of a role. Thus, the agent promises its coherence with a (behavior) norm defined by the commitment. The framework extends BDI into a many-sorted first order modal logics to add concepts of obligations, roles and social commitments while it also uses branching temporal components from Computational Tree Logics (CTL) [11]. Besides strategies for adoption of social commitments by the agents the framework also defines strategies regarding conditions for a successful de-commitment from the agent's obligations.

Another formal representation of commitments considering temporal account has been introduced in [12]. CTL [11] has been extended to capture features not being usually considered in common approaches (but relevant for realistic environments), namely time intervals considered in commitments satisfaction, "maintenance" manner of commitments next to "achieve" manner of commitments and vague specification of time. Commitments have been formally defined using Backus-Naur Form as an n-tuple $(\mathsf{Commit}\ id, x, y, p)$ where the commitment identified uniquely by its id and the interpretation is that x commits to y to make the condition p become true. The formal framework uses event calculus and defines operations $create(x, C)$, $cancel(x, C)$, $release(y, C)$, $assign(y, z, C)$, $delegate(x, z, C)$ and $discharge(x, C)$ above the commitments as well as new predicates $satisfied(C)$ and $breached(C)$ which evaluate the status of the commitments.

The past is considered linear while the future is branching. When created, the commitment is neither satisfied nor breached (the satisfaction of commitments is applied three-value logics). A commitment once satisfied or breached remains satisfied or breached once and for ever since the time.

Evolution of commitments in teamwork has been studied by Dunin-Keplicz [13]. Teamwork is explicitely represented using BDI framework by introducing a concept of a collective intention resulting in a plan-based collective commitment established within a group of agents adopting it. The teamwork consists of four consecutive stages—*(i) potential recognition*, *(ii) team formation*, *(iii) plan formation* and *(iv) team action*. The collective commitment based on a social plan (the collective intention) splits into sub-actions expressed as pairwise social commitments between agents. Establishment of the collective commitment consists in a consecutive execution of social actions defined at the particular stages: *(i)* `potential-recognition` → *(ii)* `team-formation` → *(iii)* `plan-generation` executed as `task-division` → `means-end-analysis` → `action-allocation` and *(iv)* `team-action` implemented as execution of respective actions allocated to each agent in the former stage. Naturally, the above-mentioned social actions are hierarchically bound from the first to the last stage. Dynamically evolving environment may cause unfeasibility of the allocated actions during the team action which results in a need for evolution of the collective commitment accordingly. In such a case, the maintenance of the collective commitment is achieved by invoking reconfiguration at the `action-allocation` level progressing upwards to higher levels of the hierarchy of social actions, possibly up to the `potential-recognition`. Finally, the collective commitment is adapted (another potential for the teamwork recognized) or dropped. The hierarchical manner of the reconfiguration allows for minimization of changes necessary to perform in order to adapt the collective commitment. The communication necessary for the reconfiguration is explicitly involved and formalized in the framework. Adaptation of the commitment is motivated by persistency of the joint intention which differs given a chosen intention strategy (*blind*, *single-minded* and *open-minded*). For the sake of not making the presented multi-modal logical framework even more complex and less tractable, temporal aspects of the cooperation are assumed to be expressed in a procedural way rather than by employing temporal and dynamic elements among the modalities used.

IV. CONCLUSION AND FUTURE WORK

This paper dealt with the problem of distributed planning used for replanning and plan repair processes. The classical work on commitments has been extended towards commitment recurrence for flexible and more expressive representation of replanning alternatives. Similarly, the termination condition has been defined as a specific type of commitment. The various types of commitments were classified according to the impact they may have on the other collaborating actors. This classification enables the agents to perform the right decision during the decommitment process.

This contribution represents only a starting point towards a more complex research effort that will be performed with social commitments within the context of distributed planning. We need to go beyond classification of the commitments to basic types and we need to design metrics and mechanisms that would allow agents to assign costs to each of the commitments. This will facilitate further research in design of scalable negotiation mechanisms allowing agents to negotiate the best commitments for their and social welfare perspectives.

Further integration of the HTN planning mechanism and the social commitments knowledge structure will be a critical research challenge we want to address. As described in the paper, we assume that the commitments resulted from the agent-oriented programming process and are uploaded from the agent's knowledge base. We plan to develop and design mechanisms for runtime creation of commitments from the hierarchical task networks, defining the planning problem and from the known hierarchy and knowledge of competency and abilities of the agents.

REFERENCES

[1] M. Wooldridge and N. Jennings, "Cooperative problem solving," *Journal of Logics and Computation*, vol. 9, no. 4, pp. 563–594, 1999.

[2] H. Levesque, P. Cohen, and J. Nunes, "On acting together," in *AAAI-90 Proceedings, Eighth National Conference on Artificial Intelligence*, vol. 2. Cambridge, MA, USA: MIT Press, 1990, pp. 94–99.

[3] E. H. Durfee, "Distributed problem solving and planning," in *A Modern Approach to Distributed Artificial Intelligence*, G. Weiß, Ed. San Francisco, CA: The MIT Press, 1999, ch. 3.

[4] M. E. DesJardins and M. J. Wolverton, "Coordinating a distributed planning system," *AI Magazine*, vol. 20, no. 4, pp. 45–53, 1999.

[5] E. Ephrati and J. S. Rosenschein, "A heuristic technique for multiagent planning," *Annals of Mathematics and Artificial Intelligence*, vol. 20, no. 1–4, pp. 13–67, 1997. [Online]. Available: http://www.springerlink.com/openurl.asp?genre=article&id=doi:10.1023/A:1018924209812

[6] M. M. de Weerdt, A. Bos, J. Tonino, and C. Witteveen, "A resource logic for multi-agent plan merging," *Annals of Mathematics and Artificial Intelligence, special issue on Computational Logic in Multi-Agent Systems*, vol. 37, no. 1–2, pp. 93–130, Jan. 2003.

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

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

[9] A. Tate, "Intelligible ai planning," in *Research and Development in Intelligent Systems XVII (Proc. 20th ES)*, M. Bramer, A. Preece, and F. Coenen, Eds. Springer, 2000, pp. 3–16.

[10] M. Fasli, "On commitments, roles, and obligations," in *CEEMAS 2001*, ser. LNAI, B. Dunin-Keplicz and E. Nawarecki, Eds., vol. 2296. Springer-Verlag Berlin Heidelberg, 2002, pp. 93–102.

[11] 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*, ser. LNAI, vol. 354. Springer-Verlag, 1988, pp. 123–172.

[12] A. Mallya, P. Yolum, and M. Singh, "Resolving commitments among autonomous agents," in *Advances in Agent Communication. International Workshop on Agent Communication Languages, ACL 2003*, ser. LNCS, F. Dignum, Ed., vol. 2922. Springer-Verlag, Berlin, Germany, 2003, pp. 166–82.

[13] B. Dunin-Keplicz and R. Verbrugge, "Evolution of collective commitment during teamwork," *Fundamenta Informaticae*, vol. 56, no. 4, pp. 563–592, August 2003.