Enabling Agility through Coordinating Temporally Constrained Planning Agents

J. Renze Steenhuisen
Delft University of Technology
J.R.Steenhuisen@tudelft.nl

Mathijs M. de Weerdt
Delft University of Technology
M.M.deWeerdt@tudelft.nl

Cees Witteveen
Delft University of Technology
C.Witteveen@tudelft.nl

ABSTRACT

In crisis response, hierarchical organizations are being replaced by dynamic assemblies of autonomous agents that promise more agility. However, these autonomous agents might cause a decrease in effectiveness when individually constructed plans for moderately-coupled tasks are not jointly feasible. Existing coordination techniques can be applied in the pre-planning phase to guarantee feasible joint plans for partially-ordered tasks. Temporal relations in crisis response are often more complex than the simple precedence relations in current work. Therefore, we analyze whether temporal information can be dealt with by a conversion to partially-ordered tasks with only precedence constraints. Time windows and two temporal constraints (overlaps and during) can be rewritten in such a way that the task remains partially-ordered. When other temporal constraints (meets, starts, finishes, and equals) are used, tasks become tightly-coupled, requiring coordination in the execution phase as well. This work shows the applicability of pre-planning coordination as an enabling technology for the effective formation of agile organizations.

Keywords

Multi-agent system, planning, qualitative temporal constraints, coordination.

INTRODUCTION

Crisis response organizations are shifting from hierarchical structures to dynamic assemblies of agents with local autonomy (Harrald, 2006). This local autonomy allows the system, by means of a single agent, to quickly respond to local issues. For example, not only information can be missing, possibly becoming available over time, but resources (e.g., communication) might also be unavailable (for a certain period). Despite these uncertainties, an effective and highly efficient response is required in which it is important to have joint plans that are robust enough to guarantee completion of the set of tasks. The agents should be able to replan their actions without causing harm to the global goal (effectiveness), when new information becomes available that allows them to increase efficiency (e.g., saving additional lives). Consider the following example which illustrates these issues.

Example 1. On a highway across a river, numerous cars have crashed into each other and some cars are known to be on fire. However, the first firefighters to arrive at the scene come in from the other end and are yet unable to extinguish the fire. A police officer, who arrived earlier, has already found some people that are trapped in a car that is almost falling down the bridge into the water. Because the fire is out of reach and it is of no immediate threat, the firefighters decide to first secure this car and save those peoples lives.

As can be seen from the above example, agents can use their autonomy to change their plans locally in unforeseen circumstances without endangering the timely completion of the global goal. This is exactly what we envision the role of coordination to be in crisis response. We believe that coordination is one of the most important goals of any information system for crisis response.

Because it is not unlikely that communication is lost during a crisis situation,[1] we propose to apply a pre-planning coordination technique. Consequently, during the crisis, the agents know what they can do without causing conflicts, while minimally restricting the agents' behavior. Of course, the agents can still negotiate with other agents about their agreements (e.g., when changes in the environment obsolete a task), if communication is possible between these agents.

The advantage of enabling agility through coordinating agents is threefold. First, the joint behavior of the agents is robust because their actions are coordinated. Second, agents can jointly change their agreements opportunistically to optimize their plans. Third, each individual agent is able to respond to local incidents and changes, as long as no agreements with other agents are broken.

The remainder of this paper is structured as follows. We start to describe the context for coordinating multi-agent planning problems. We then construct a framework for describing crisis response situations. Next, we show how time windows and two temporal constraints (overlaps and during) can be rewritten in a partial order, and show that four other temporal constraints (meets, starts, finishes, and equals) turn the problem into a set of tightly-coupled tasks. Thereafter, our approach to coordinating temporally constrained is related to other work. Finally, we summarize our findings and give some directives for future research.

INTERDEPENDENCE OF CRISIS RESPONSE TASKS

When a set of inter-dependent agents want to guarantee that their joint plan or schedule is feasible, some form of coordination is required. This coordination problem is very complex and arises in domains as disaster rescue, patrolling, urban reconnaissance, and crisis response (Zlot and Stentz, 2006).

It is clear that connected *tasks* cause agents to become inter-dependent which in turn results in the need for coordination. However, some tasks are stronger connected than others, which cause a higher level of coordination to be required. Therefore, it is common to classify sets of tasks with respect to the level of coordination they require (Kalra, Stentz and Ferguson, 2004). This classification is based on the relation of subtasks to their abstract tasks in the hierarchical decomposition.

First, a set of tasks are called *loosely coupled* when they can independently be assigned to the agents. Here, each agent is able to construct an independent plan for its subset of tasks, and coordination reduces to solving the *task allocation* problem (i.e., who will do which tasks). Examples of this category are tasks that are totally unrelated to each other, such as searching for casualties in different parts of a city.

Second, a higher degree of coordination is required, when abstract tasks are decomposed into a set of tasks with precedence constraints or time windows. For example, if tasks are partially-ordered and assigned to multiple agents, agents have to coordinate their plans in order not to break that partial order. Such partially-ordered tasks are called *moderately coupled*. Typical problems in this category are *(i)* pre-defined action plans, and *(ii)* discipline such as supporting the neck before freeing a person from a wrecked car.

Third, an even higher degree of coordination is required when the tasks are stronger constrained (e.g., temporal and geographical constraints). For example, the execution of tasks might require synchronization of begin and/or end points (e.g., *simultaneity constraint*) as in jointly lifting a table. These tasks require coordination while executing the tasks, and are said to be *tightly coupled*. Two examples of tightly-coupled tasks are *(i)* extinguishing of a large fire that requires simultaneous action of multiple firefighters from different angles, and *(ii)* simultaneously lifting a patient onto a bed.

In crisis response, tightly-coupled tasks involve synchronization between the actions of multiple agents, which can best be solved during execution. Loosely-coupled tasks are trivially solved due to expertise (e.g., firefighters extinguish fires) and the fact that they are (almost) independent. Moderately-coupled tasks, however, occur in the form of disaster plans and discipline, and need coordination to guarantee effectiveness and enable local autonomy for the agents. Therefore, the focus of this paper is on coordinating moderately-coupled tasks.

[1] No communication was available in New Orleans after hurricane Katrina had left.

FRAMEWORK FOR COORDINATING CRISIS RESPONSE TASKS

In order to do pre-planning coordination, we need to construct a model for describing crisis response situations. We consider a set of agents $\mathcal{A} = \{A_1, \ldots, A_n\}$ representing the different actors that are involved in completing a complex task $\mathcal{T}$ (e.g., clear the highway while minimizing casualties). Such a complex task can be decomposed into a set $T = \{t_1, \ldots, t_k\}$ of *elementary tasks* t_i (e.g., extinguish fire, apply first aid, and secure car) and a set of *ordering constraints* π between these elementary tasks, inducing a partial order (T, π). An ordering constraint $t \, \pi \, t'$ means that the execution of t has to precede the execution of t' (e.g., a trapped person needs to be freed before an ambulance can bring him/her to a hospital).

Without loss of generality, we assume that the set T is partitioned into n partitions where each partition $T_i \subseteq T$ is the set of tasks to be executed by agent A_i. Furthermore, we assume that the assignment of the partitions to agents is such that each agent is able to complete its set of tasks.

Example 2. In Figure 1, a situation is depicted where three agents $\{A_1, A_2, A_3\}$ are assigned two tasks each. The six tasks $\{t_1, \ldots, t_6\}$ together constitute a complex task that is partially ordered with a precedence relation π (depicted by arrows between tasks).

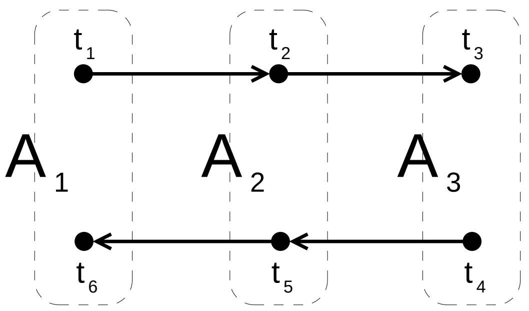


Figure 1. Problems can occur when planning autonomously.

Now, each agent A_i has to plan the complex subtask (T_i, π_i) which is generated by the set of tasks T_i assigned to it. In order to complete such a complex subtask, the agent has to construct a plan for it (i.e., extend the partial order). We assume that each agent is an autonomous planner and is able to find a suitable plan for (T_i, π_i) . Then making such a plan means that an agent A_i is free to order the tasks occurring in T_i for execution as long as the precedence constraints π_i are respected. Although agents are assumed to have complete freedom in choosing their own (possibly domain-specific) planning tools, we do not need to bother about the specific details of their plans. We simply assume that the outcome (an individual plan of an agent A_i) can be represented as a partial order of the set of tasks T_i assigned to it.

It is not difficult to find examples where, due to autonomous planning, the simple joining is not feasible, because it is not acyclic. In Figure 1, consider agent A_2. It is clear that if A_2 makes a plan where $t_2 \, \pi \, t_5$ then there exists a plan for agent A_1 that creates a cycle (if agent A_1 chooses a plan where t_6 precedes t_1). On the other hand, if A_2 creates a plan where $t_5 \, \pi \, t_2$ then agent A_3 can make a plan that creates a cycle. We call such a situation, where inter-agent cycles can be created, *uncoordinated*.

Since we want to enable agility by giving agents local autonomy, they need to be able to (*i*) individually develop their plans, and such that (*ii*) the joining of every set of these individually developed plans is feasible. Intuitively, it is clear that these requirements might easily conflict and hence that we face a *coordination problem*.

We define the coordination problem as follows: Given a coordination instance $(T, \pi, \mathcal{A})$ and a partitioning $T = \mathrm{Y}_{i=1}^{n} T_i$ of T such that agent A_i can create a plan for the complex subtask (T_i, π_i), find a set of ordering constraints $\Delta = \mathrm{Y}_{i=1}^{n} \Delta_i$ (called a *coordination set*) that ensures that every joining of locally determined plans respecting $\pi_i \cup \Delta_i$ is a feasible plan for executing T.

Example 3. Consider Figure 1 which was already shown to be uncoordinated. We need to prevent *every* possible inter-agent cycle from occurring by adding intra-agent constraints. We could, for instance, prevent the cycle t_1 - t_2 - t_5 - t_6 - t_1 by having $\Delta_1 = \{t_1 \pi t_6\}$. If we also take $\Delta_3 = \{t_4 \pi t_3\}$, the problem instance is coordinated while A_2 is still free to choose the order in which it wishes to plan t_2 and t_5.

In previous work (Steenhuisen, Witteveen, Mors and Valk, 2006), we have investigated this problem formally and we have shown that it can be solved by adding such a minimal coordination set. It turns out that this general coordination problem for finding a minimal coordination set is $\sum_2^p$ -complete, and that it is NP-complete when the number of agents is bounded.

INCORPORATING TEMPORAL CONSTRAINTS

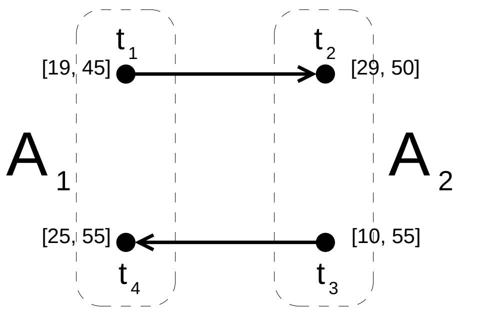
Previous work (Valk, 2005) on coordinating moderately-coupled tasks only considered precedence constraints. In crisis response, however, temporal relations are often more complex than these simple precedence relations, and additional temporal information is often needed (e.g., estimated durations) (Windhouwer, Klunder and Sanders, 2005). First, we show that time windows can seamlessly be integrated in these previously developed coordination techniques. Second, we show that two qualitative temporal constraints (**during** and **overlaps**) can be used in moderately-coupled tasks, but that incorporating four other constraints (**meets**, **starts**, **finishes**, and **equals**) makes it a tightly-coupled task.

Time Windows

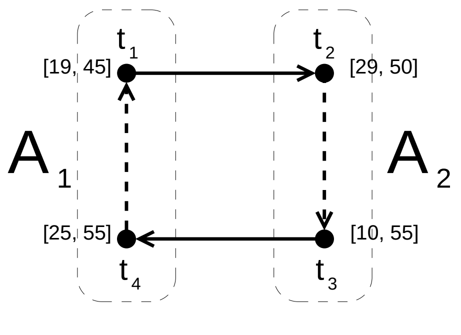
In many domains, like crisis response, the execution of tasks is constrained by earliest finishing times or deadlines. In general, a time window can be assigned to each task to define when that task must be executed (i.e., started and finished).

Although scheduling techniques can be used to find feasible schedules in such situations, they fail to allow agents to impose additional ordering constraints on their sets of tasks. As discussed earlier, these agents need to add these precedence constraints *before* coordinating their scheduling activities. The problem is that if we allow agents to make such decisions autonomously, there is a possibility that they will never be able to find a coordinated schedule for the total set of tasks. To see this, consider the following example.

Example 4. There are two agents A_1 and A_2, each having to complete two tasks with time windows (see Figure 2(a)). Now, suppose that each of the agents decides autonomously to add the additional precedence constraints, indicated by dashed arrows in Figure 2(b), and each agent tightens its set of time windows correspondingly (i.e., when $t_i \pi t_j$ then tighten the time windows to $[lb(t_i), \min(ub(t_i), ub(t_j))]$ and $[\max(lb(t_i), lb(t_j)), ub(t_j)]$ for t_i and t_j, respectively). With the local information, agent A_1 tightens the time windows for t_1 to [25,45] and for t_4 to [25,45], while agent A_2 can only tighten the time window for t_3 to [29,55]. Since the precedence constraints are cyclic, it is not difficult to see that there cannot exist a feasible joint schedule for this set of tasks.



(a) A problem instance.



(b) A broken partial order.

Figure 2. Autonomous planning of agents might cause failing of coordinated scheduling.

In general, we can state the plan+schedule coordination problem as follows: Given a coordination problem $(T, \prec, I, \mathcal{A})$ with time intervals I, how to ensure that a feasible joint schedule can be found for all the tasks in T if every agent A_i is free to compose its own ordering $\prec_i$ on the tasks T_i assigned to him, while respecting the precedence relation $\prec$ and the given time intervals?

This problem closely resembles the plan-coordination problem we have investigated earlier (Steenhuisen et al., 2006). The difference lies in the incorporation of time intervals for tasks. We show how our problem with time intervals can be solved by reducing it to the plan-coordination problem.

The idea is to solve our plan+schedule coordination problem by first reducing an instance with time intervals to a plan-coordination instance without time intervals. After the plan-coordination problem is solved by identifying and adding a set of suitable constraints, the agents are allowed to plan autonomously. Finally, the schedule-coordination problem needs to be solved, which can be done by applying a pre-scheduling mechanism (Hunsberger, 2002).

Here, it suffices to fully describe the reduction of the coordination problem with time intervals into a coordination problem without time intervals. The idea is the following: Given a coordination instance with time intervals $(T, \prec, I, \mathcal{A})$, we first collect all the begin and end points of the intervals $I(t)$. More precisely, for each task t, the time interval $I(t) = [lb(t), ub(t)]$ is coded into two values s_{lb} and s_{ub}, where $s_{lb} = lb(t) - 1$ and $s_{ub} = ub(t) + 1$. We collect the total set of all these values in the set $S = \{s_1, \ldots, s_p\}$ and we may assume that $s_1 < s_2 < \ldots < s_p$, without loss of generality. Now, we construct the following coordination instance $(T', \prec', \mathcal{A}')$, where

1. the set of tasks T' is the set T extended with the set of tasks S: $T' = T \cup S$,
2. the precedence relation $\prec'$ is the relation $\prec$ extended with the total ordering imposed on S. Moreover, for each task t, associated with time points s and s', two additional precedence constraints $s_{lb} \prec t$ and $t \prec s_{ub}$ are constructed. The result then is $\prec' = \prec \cup \{(s_i, s_j) \mid 1 \leq i < j \leq p\} \cup \{(s_{lb}, t), (t, s_{ub}) \mid t \in T, s_{lb} = lb(t) - 1, s_{ub} = ub(t) + 1\}$, and
3. we add a special agent A_S taking care of the set of virtual tasks S: $\mathcal{A}' = \mathcal{A} \cup \{A_S\}$.

If we determine a coordination set Δ for this instance, we can simply add the resulting set of constraints to the original instance with time intervals. The agents are free to form their own plans as long as they respect these constraints and, if they succeed in finding locally feasible plans, we obtain a cycle-free instance of the schedule-coordination problem.

After applying plan coordination, earliest starting times and latest finishing times can be determined by considering the maximum and minimum possible parallelism, respectively. Others have already reported on great gains in practice when determining these tighter bounds (Sultanik, Modi and Regli, 2006). Finally, pre-scheduling coordination techniques (Hunsberger, 2002) can be used to let agents make the remaining scheduling decisions autonomously.

Example 5. Consider the problem instance of Figure 2(a). After reduction, the result of the reduction to the plan-coordination instance is depicted in Figure 3. By adding the constraint $t_3 \prec t_2$ it is impossible to create an inter-agent cycle.

Qualitative Temporal Constraints

Besides precedence constraints there are six other qualitative temporal constraints (Allen, 1983). In this section, we describe how qualitative temporal constraints can be represented in our framework such that they can be used in the process of coordinating a planning instance. First, we describe a more fine-grained way of representing tasks by splitting each task into two time points. Second, we show that the binary constraints **before**, **overlaps**, and **during** (and their inverses) can flawlessly be integrated into a framework based on precedence constraints alone. Third, the question is addressed whether the binary relations **meets**, **starts**, **finishes**, and **equals** (and their inverses) can be rewritten such that the existing coordination techniques for moderately-coupled tasks can deal with them.

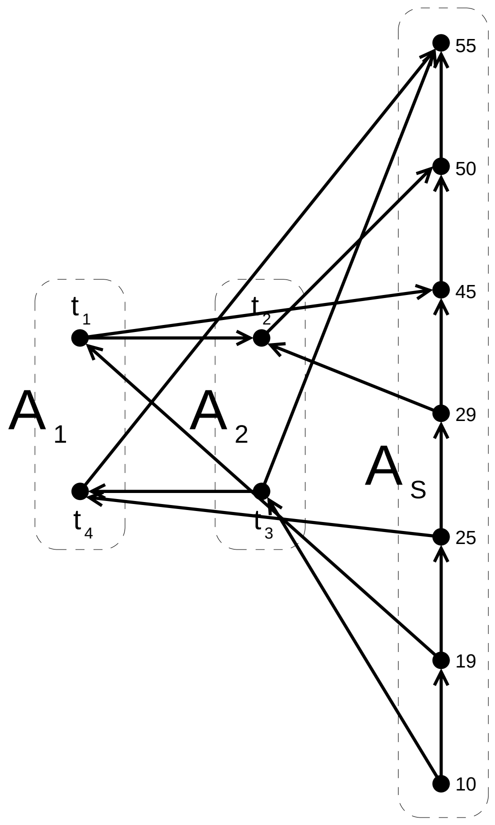


Figure 3. Reducing a temporal coordination problem to a plan-coordination problem.

Moderately-Coupled Tasks

When representing the constraints **overlaps** and **during**, it should be noted that these relations are defined on time intervals (i.e., instead of time points). This is completely in line with Allen's interval algebra (Allen, 1983) in which **overlaps** and **during** are part of the seven qualitative temporal constraints that can be defined on time intervals. Actually, it is quite intuitive to represent tasks as intervals instead of single time points, because tasks tend to take up some limited amount of time.[2] More formally, we denote each task t by two time points t_s (starting time) and t_e (ending time), that are constrained by $t_s \pi\ t_e$. For a graphical representation of tasks as time intervals, see Figure 4. Note that t_s and t_e are the time points on which the execution of task t actually starts and ends, respectively. Contrary to this, the time points s_{lb} and s_{ub} represent the lower and upper bounds, respectively, between which the execution of the associated task needs to be scheduled (and executed).

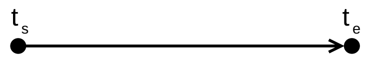
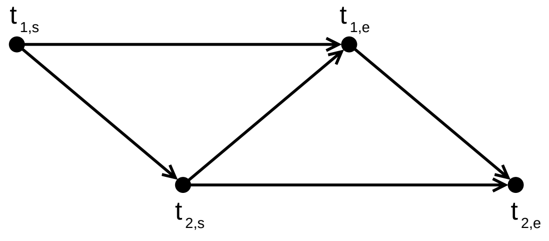


Figure 4. Tasks represented as time intervals.

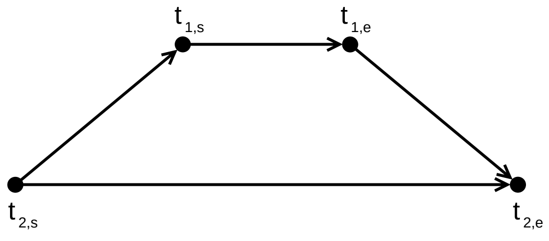
Note that no changes are needed for the **before** (precedence) relation (and its inverse **after**). For the remaining qualitative temporal constraints, we will not explicitly show how to deal with the inverse relations because they are handled similarly.

[2] Although some tasks seem to take for ever.

With the **before**, **overlaps**, and **during** constraints the end points of the time intervals are not allowed to coincide. We recall that the **overlaps**-relation expresses that the execution of two tasks partially overlap (like running in steeple chase), and that the **during**-relation constrains a task to be executed between the starting and ending of another task's execution. Obviously, these relations can be represented by constraining the end-points of the time intervals by using only precedence constraints. For both constraints we start by splitting up tasks t_1 and t_2 into time intervals with end points $t_{1,s}$, $t_{1,e}$ and $t_{2,s}$, $t_{2,e}$, respectively, and constrained by $t_{1,s} \pi\ t_{1,e}$ and $t_{2,s} \pi\ t_{2,e}$. Now, we can rewrite t_1 **overlaps** t_2 as $t_{1,s} \pi\ t_{2,s} \pi\ t_{1,e} \pi\ t_{2,e}$ (see Figure 5(a)), and change t_1 **during** t_2 into $t_{2,s} \pi\ t_{1,s} \pi\ t_{1,e} \pi\ t_{2,e}$ (see Figure 5(b)). Therefore, we can rewrite the constraints **overlaps** and **during** to a representation with only precedence constraints which allows us to apply existing pre-planning coordination techniques for moderately-coupled tasks (Valk, 2005).



(a) t_1 overlaps t_2.

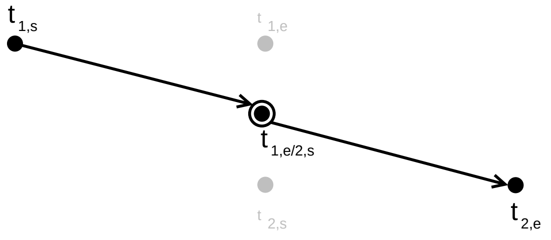


(b) t_1 during t_2.

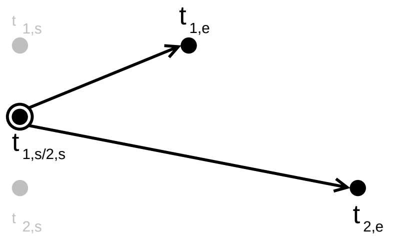
Figure 5. Rewriting the temporal constraints overlaps and during.

Tightly-Coupled Tasks

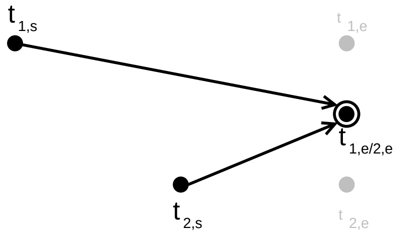
Contrary to the previous constraints, end points can coincide with the **meets**, **starts**, **finishes**, and **equals** constraints. Therefore, we cannot rewrite these constraints to a set of time points with only precedence constraints between them in the same way as above. However, this problem can be resolved by merging the coinciding time points into one (see Figure 6).



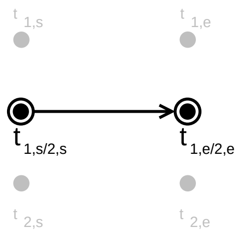
(a) t_1 meets t_2.



(b) t_1 starts t_2.



(c) t_1 finishes t_2.

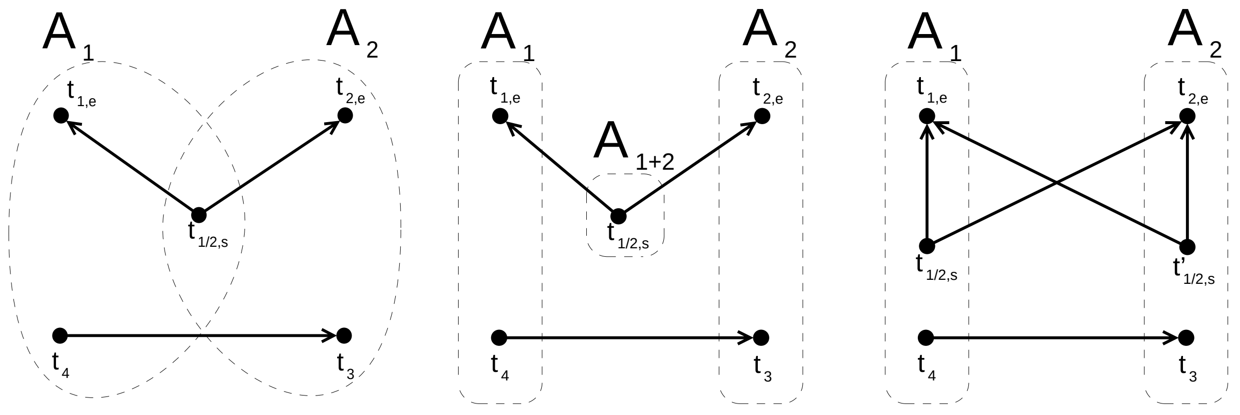


(d) t_1 equals t_2.

Figure 6. Rewriting the temporal constraints meets, starts, finishes, and equals.

Unfortunately, this representation introduces the problem that multiple agents share one time point. Of course, this makes sense because they have to synchronize the execution of some of their tasks, and these synchronization points are shared. Therefore, the problem lies in the assignment (or responsibility) for these synchronization points. Recall that coordination mechanisms for moderately-coupled tasks require tasks to be assigned to different agents (i.e., a task can be assigned to only one agent). However, shared time points cannot be assigned to a single agent because they are shared. In fact, it turns out that introducing a virtual agent, representing the agent coalition, does not even solve the problem.

In Figure 7, a problem instance is given that illustrates that introducing an additional agent does not help in coordinating instances with shared time points. Here, four tasks need to be completed by two agents, where tasks t_1 and t_4 are assigned to agent A_1 and tasks t_2 and t_3 are assigned to A_2. Moreover, the execution of the tasks is constrained such that t_4 π t_3 and t_1 **starts** t_2, as is illustrated by Figure 7(a). The introduction of a virtual agent A_{1+2}, as depicted in Figure 7(b), results in a coordinated instance (because no inter-agent cycle can be constructed by adding intra-agent arcs). However, it is clear from Figure 7(c) that the agents cannot plan independently, because an infeasible joint plan results when A_1 plans $t_{1,e}$ π t_4 and A_2 plans t_3 π $t'_{1/2,s}$.



(a) Tightly-coupled task. **(b) Coordinated.** **(c) Uncoordinated.**

Figure 7. An additional agent does not help in coordinating instances with shared time points.

As we have seen previously, shared time points are introduced when tasks are constrained by **meets**, **starts**, **finishes**, or **equals**. Clearly, mechanisms for coordinating moderately-coupled tasks are not capable of coordinating problem instances with shared time points. Therefore, when the execution of tasks is constrained with **meets**, **starts**, **finishes**, or **equals** it becomes a tightly-coupled task that needs to be resolved during execution.

RELATED WORK

In crisis response, it should be possible to do planning during execution in order to let agents use local autonomy to respond to local issues. Coordination is needed to guarantee that using this local autonomy does not cause conflicts to the global goal. In coordination, a distinction can be made between *pre*, *interleaved*, and *post*-planning coordination. However, both interleaved and post-planning coordination assume communication to be available during and after planning and thus during execution. Recall that it is not unlikely that communication is lost in a crisis situation, which makes interleaved and post-planning coordination not suitable for crisis response in general. Therefore, we use a pre-planning coordination approach.

Although post-planning coordination techniques do not suffice, they can be used in case communication is available. In addition, these other techniques can form a source of inspiration for future research in pre-planning coordination. For instance, the *Partial-Order Causal-Link* framework (Cox and Durfee, 2005), that is used in post-planning coordination, allows symmetric *concurrency* and *non-concurrency* relations to constrain tasks. These constraints are very useful for describing, for instance, disaster plans. Note that the concurrency relation is equal to a shared time point in our framework.

Pre-planning (Valk, 2005) and post-planning (Cox et al., 2005) approaches coordinate partially-ordered tasks, but existing work has a limited expressiveness concerning temporal constraints. Schedule-coordination approaches often use a framework that allows more temporal information to be used.

In the *Temporal Constrained Satisfaction Problem* (TCSP) (Dechter, Meiri and Pearl, 1991) framework, it is possible to represent arbitrary intervals of temporal distances between time points. This is a rich framework for representing problem instances for which temporal information is available, such as time windows and task durations. A restricted version of the TCSP framework is the *Simple Temporal Network* (STN) (Dechter et al., 1991), in which the number of intervals between two time points is limited to (at most) one. On these STNs, the *Temporal Decoupling Problem* (TDP) (Hunsberger, 2002) is defined, which can be classified as a pre-scheduling coordination approach.

However, planning is different from scheduling and it could, therefore, be argued that this approach is too restrictive when the agents are not allowed to plan the order in which they wish to execute their tasks. This means that solving the TDP restricts the agents much more than is required, which is an undesired property in crisis response. Finally, note that we could call pre-planning coordination the *Plan Decoupling Problem*, because it resolves all interdependencies between agents on a plan level instead of a schedule level, which is the case with the TDP. Our work is an attempt to combine the advantages of using pre-planning coordination and using an expressive framework for temporal information.

CONCLUDING REMARKS

In this paper, we argued that local autonomy of the agents is a necessary precondition for forming agile organizations that are desired in crisis response, and that this can be provided by coordination. More precisely, we showed that pre-planning coordination is required to guarantee effectiveness in crisis response situations. Furthermore, we addressed the problem that previous work on pre-planning coordination uses precedence constraints, while crisis response requires the ability to reason about other temporal relations as well. We extended the applicability of existing pre-planning coordination techniques for coordinating partially-ordered tasks, which we called moderately-coupled tasks.

First, it was shown that a set of partially-ordered tasks with time intervals could be represented in a partial order without time intervals. This representation uses one additional (virtual) agent to represent time that allows the time intervals to be rewritten as a partial order. In practice, this means that existing coordination techniques can be used to coordinate these problem instances with time intervals.

Second, we showed how two qualitative temporal constraints (**overlaps** and **during**) can be rewritten by using only precedence constraints. Hereby, we extended the applicability of existing pre-planning coordination techniques for coordinating moderately-coupled tasks, and showed that tasks with time intervals and constraints **overlaps** and **during** are moderately-coupled.

Third, we showed that four other qualitative temporal constraints (**meets**, **starts**, **finishes**, and **equals**) require the use of shared time points. These shared time points make it impossible to rewrite the problem instance as a partial order, thereby labeling tasks as tightly-coupled tasks that need to be dealt with during execution on the spot, when these qualitative temporal constraints are used.

Some challenging problems remain for future work. It would be interesting to develop pre-planning coordination mechanisms that minimize the latest finishing time, instead of finding a smallest coordination set. Another challenging problem that remains for future work is whether it is possible to do pre-planning coordination of tightly-coupled tasks, or to model these problems as a simple temporal network.

ACKNOWLEDGMENTS

J. Renze Steenhuisen is supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024. The ICIS project is hosted by the DECIS Lab, the open research partnership of Thales Nederland, the Delft University of Technology, the University of Amsterdam, and the Netherlands Foundation of Applied Scientific Research (TNO).

REFERENCES

1. Allen, J. F. (1983) Maintaining knowledge about temporal intervals, *Communications of the ACM*, 26, 11, 832-843.
2. Cox, J. S., Durfee, E. H. (2005) An efficient algorithm for multiagent plan coordination, *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, 828-835.
3. Dechter, R., Meiri, I., Pearl, J. (1991) Temporal constraint networks, *Artificial Intelligence*, 49, 1-3, 61-95.
4. Harrald, J. R. (2006) Agility and discipline: Critical success factors for disaster response, *The Annals of The American Academy of Political and Social Science*, 604, 256-272.
5. Hunsberger, L. (2002) *Group Decision Making and Temporal Reasoning*, PhD thesis, Harvard University, Cambridge, MA, USA.
6. Kalra, N., Stentz, A., Ferguson, D. (2004) *Hoplites: A market framework for complex tight coordination in multi-agent teams*. Technical Report CMU-RI-TR-04-41, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA.
7. Steenhuisen, J. R., Witteveen, C., ter Mors, A. W., Valk, J. M. (2006) Framework and complexity results for coordinating non-cooperative planning agents, *Proceedings of the 4th German conference on Multi-Agent System Technologies*, Lecture Notes in Artificial Intelligence, 4196, 98-109.
8. Sultanik, E. A., Modi, P. J., Regli, W. C. (2006) Constraint propagation for domain bounding in distributed task scheduling, *Proceedings of Principles and Practice of Constraint Programming*, Lecture Notes in Computer Science, 4204, 756-760.
9. Valk, J. M. (2005) *Coordination among Autonomous Planners*, PhD thesis, Delft University of Technology, Delft, The Netherlands.
10. Windhouwer, C. J., Klunder, G. A., Sanders, F. M. (2005) Decision support system emergency planning, creating evacuation strategies in the event of Flooding, *Proceedings of 2nd International Conference on Information Systems for Crisis Response and Management*, 171-180.
11. Zlot, R. M., Stentz, A. (2006) Market-based multirobot coordination for complex tasks, *International Journal of Robotics Research, Special Issue on the 4th International Conference on Field and Service Robotics*, 25, 1, 73-101.