Coordination by design and price of autonomy[1]

Adriaan ter Mors [b] Chetan Yadati [a] Cees Witteveen [a] Yingqian Zhang [a]

[a] *Delft University of Technology, Delft*
[b] *Almende, Rotterdam*

A multi-agent planning problem refers to a set of activities that has to be performed by several autonomous agents. In general, due to the possible dependencies between agents' activities or interactions during execution of those activities, allowing agents to plan individually may lead to a very inefficient or even infeasible solution of the multi-agent planning problem. This is exactly where *planning coordination method* comes into play.

Due to the self-interested nature of the autonomous agents, in this paper, we concentrate on the *coordination by design* approach, which does not need to assume that agents are willing or able to communicate to achieve a coordinated result. We aim at the development of coordination techniques that *(i)* let each agent construct its plan completely independent from the others thereby *(ii)* guaranteeing that the joint combination of their plans always is coordinated.

The contribution of this paper is twofold. Firstly, we will point out that in general there exist at least two ways to achieve coordination by design: one called *concurrent decomposition* and the other *sequential decomposition*. We will briefly discuss the applicability of these two methods, and then illustrate them with two specific coordination problems: coordinating tasks and coordinating resources usage. Secondly, instead of focusing only on the feasibility of the resulting plans, we will investigate the additional *costs* incurred by the coordination by design method, that means, we propose to take into account the *price of autonomy*: the ratio of the costs of a solution obtained by coordinating selfish agents versus the costs of an optimal solution. We also show the price of autonomy of the two coordination methods respectively.

Coordination by design: a general framework

In a (task-based) multi-agent planning problem, a set of tasks is given to a set of agents. These agents have to complete such tasks by making a joint plan to complete them. Usually there is also a (common) set of resources needed to complete such tasks. Both tasks and resources are subject to some sets of constraints. To model such a multi-agent task-based planning problem, we consider a tuple $\Theta = (\mathcal{A}, \mathcal{T}, R, C, \phi, \psi)$, where $\mathcal{A} = \{A_1, \ldots, A_n\}$ denotes a set of agents, $\mathcal{T} = \{t_1, \ldots, t_m\}$ is a set of tasks, $R = \{r_1, \ldots, r_p\}$ is a set of resources and C is a non-empty set of constraints on tasks and resources. Usually, the set of constraints C is partitioned into $C = C_{\mathcal{T}} \cup C_R$ where $C_{\mathcal{T}}$ is the set of constraints on the tasks and C_R is the set of constraints on the resources. The functions $\phi : \mathcal{A} \to 2^{\mathcal{T}}$ and $\psi : \mathcal{T} \to 2^R$ are the *task-assignment* function and the *task-resource* function, respectively. A *solution* to a multi-agent planning problem Θ is a specification of a *task-resource plan* (or plan, for short). Such a plan is a tuple $P = (\mathcal{A}, \mathcal{T}, R, C^P, \phi, \psi)$, where C^P is a non-empty set of *plan constraints* refining C, that is, if all constraints in C^P are satisfied, then C is satisfied too. Coordination by design requires that from the specification of a multi-agent planning problem Θ, we are able to derive a set $\{\Theta_i\}_{i=1}^n$ of single agent planning problems that can be solved independently. Using the framework, we say that a multi-agent problem Θ *induces* a single agent planning problem $\Theta_i = (T_i, R, C_i, \psi)$ for agent A_i, where $T_i = \phi(A_i)$ is the set of tasks assigned to A_i, and $C_i \subseteq C$ is the set of constraints *restricted* to tasks occurring only in T_i. Like the solution for the overall planning problem, the solution of the single agent planning problem Θ_i is a single agent plan $P_i = (T_i, R, C_i^P, \psi)$ where C_i^P is a set of plan constraints that refine C_i.

[1]The full version of this paper appears in the Journal of Autonomous Agents and Multi Agent Systems and is available at http://dx.doi.org/10.1007/s10458-009-9086-9

A problem Θ is *decomposable* if the set of local plans (individual solutions) $\{P_1, \ldots, P_n\}$, can be combined into a global plan $P = \sqcup P_i = (\mathcal{T}, R, \bigcup_{i=1}^{n} C_i^P, \phi, \psi)$, which is a solution to the original problem. In general, however, a multi-agent planning problem Θ will not be decomposable and we will need a *coordination mechanism* to ensure that the local plans can be combined into a feasible global plan. Such a coordination (by design) mechanism M is an algorithm that, given as input a multi-agent planning problem Θ, an agent A_i, and a set $\mathcal{P}_{-i}$ of plans of other agents, not including the plan of agent A_i, returns a single agent planning problem Θ_i^M such that: (1)$\Theta_i^M = (T_i, R, C_i^M, \psi)$ is an individual planning problem for agent A_i, where $T_i = \phi(A_i)$ and (2) A_i is allowed to come up with any solution P_i^M for Θ_i^M, but is ensured that the combination of all such solutions is always a solution to the original problem Θ. We call a mechanism M *feasible* if circular dependencies do not occur. Clearly, if a mechanism M is feasible, then there exists at least one ordering of agents that allow them to plan independently from the others.

Concurrent decomposition. In concurrent decomposition the coordination mechanism M is able to specify—given the original problem Θ and an agent A_i—the set of local problems Θ_i directly without knowing the plans of the other agents. It is immediate that this mechanism is feasible for every enumeration of agents. Hence, each of the agents is able to make its plan independently from and concurrently with the others. Concurrent decomposition can be applied if *(i)* the task allocation ϕ induces a partitioning of the tasks, i.e., no task is assigned to more than one agent, and *(ii)* there is no resource dependency between tasks assigned to different agents.

We apply the idea of concurrent decomposition to *coordinate tasks* . Here we determine a set of interval constraints that decompose a given autonomous scheduling problem. Here instead of local plans, each agent develops a local schedule. Based on the depth and height of a task in the given task graph we develop a set of interval constraints which specify the earliest and the latest times any task can be scheduled at. We develop algorithms that compute these intervals for both agents with unbounded capacity and for agents with bounded capacity. The algorithm for the unbounded capacity results in optimal make span whereas the algorithm for bounded capacity case gives a makespan which is in its worst case twice that of the optimal.

Sequential decomposition. On the other hand, in sequential decomposition, there is an enumeration $\langle A_1, A_2, \ldots, A_n \rangle$ of the agents such that for all i, $M(A_i, \Theta, \mathcal{P}_{-i}) = M(A_i, \Theta, \{A_1, \ldots, A_{i-1}\}) = \Theta_i^M$. That is, all plans of the agents A_j preceding A_i in the enumeration are needed to guarantee that agent A_i can plan independently. The idea is that for every i, the results of all the plans P_j of the agent A_j $(j < i)$ are translated, by the coordination mechanism M, into a suitable set of constraints, in such a way that plans of agents $A_i, A_{i+1}, A_{i+2}, \ldots, A_n$ never can invalidate the plans P_j of agent A_j $(j < i)$. Sequential decomposition can be applied if there is an overlap in the tasks assigned to different agents or there are some dependencies on the resources that have to be used by two agents in the system.

We apply it to coordinate the usage of *shared resources*. In this case, each agent plan for resource usage is a specification of which reseource is being used and in which time interval. Based on the concept of free-time window reachability, we have designed an algorithm that returns the shortest-time, conflict-free route in the infrastructure and runs in $O(|F|log(|F|) + |F||R|)$ time, where F is the set of free time windows and R is the set of resources.

Price of autonomy

In both sequential and concurrent decomposition, additional constraints are imposed upon the agents to ensure *feasibility* of the resulting plans. Besides feasibility, however, coordination methods should also be evaluated based on the *efficiency* of the resulting joint plan produced by the agents. This price of autonomy ρ_M of using coordination method M w.r.t. make span efficiency is the worst-case ratio of make soan length of the joint plan obtained by letting selfish agents make their own plan as efficient as possible, versus the length of make span of the most efficient plan that could be obtained as the solution of the multi-agent problem. It can be proved that the price of autonomy of the concurrent decomposition approach for bounded capacity agents is 2 and that of sequential decomposition approach is $|A|$.

Future work It would be indeed a challenge to look at instances where both task and resource constraints exist. A naive mixture of both these approaches could potentially lead to a price of autonomy which is unacceptable. Another issue we have to address is that of fairness.In concurrent decomposition, it would be desirable to apply coordination constraints so that all agents have similar amounts of autonomy.