From Team Plan to Individual Plans: a Petri Net-based Approach

Olivier Bonnet-Torres
ENSAE-Supaero & Onera-CERT / DCSD
2, avenue E. Belin
31055 Toulouse cedex 4 FRANCE
olivier.bonnet@onera.fr

Catherine Tessier
Onera-CERT / DCSD
2, avenue E. Belin
31055 Toulouse cedex 4 FRANCE
catherine.tessier@onera.fr

ABSTRACT

This paper focuses on a framework for representing a team plan and its projections on individual agents. The team plan is represented with a coloured Petri net. Using the implicit place reduction rule an agenticity hierarchy is deduced: each transition bearing two or more output places corresponds to splitting the (sub)team into (sub)subteams; a two-input-place transition merges subteams. The reduction rule is extended to support the notion of transfer of an agent from one subteam to another. These notions of splitting, merging and transfer are basic team management structures which describe the dynamic team hierarchical organisation. At each level of agenticity a plan is derived from the team plan reduction. Controlling an agent individually requires extracting individual information, such as activities involving the agent as well as interacting agents or subteams at each level of agenticity. The agent-projected plan encompasses for each level of agenticity an activity plan and a list of cooperating agents or subteams.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems*; D.2.2 [**Software Engineering**]: Design Tools and Techniques—*Petri nets*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Plan execution, formation, and generation*

Keywords

Hierarchical Petri nets, object Petri nets, replanning, teamwork

1. INTRODUCTION

In the agent world activity planning has been widely studied. The increasing complexity of the jobs assigned to agents has led to using groups of agents. The groups, when organised and aware of their organisation, are called teams. The problem of team planning is considered difficult (state-space size of $(2^m - 1)^k k! \prod_{j=1}^{k} u_j$, with m the number of agents, k the number of goals, u_j the number of recipes for the jth goal).

Hierarchical task networks (HTN) [11] consist of decomposing tasks into subtasks until elementary tasks. A set of methods to achieve each task is then organised into an agent plan. In the wake of HTN, Grosz *et al.* [12] base the *SharedPlan* approach on the hierarchical decomposition of shared plans into a sequence of recipes to be applied by a team of agents. Their work also inherits from the logics of beliefs and intentions [6, 7, 18]. Tambe *et al.* [20, 21] have focused on team behaviour in STEAM. The planning module in STEAM uses rules to produce team reactions to external events.

From a different standpoint the representation of the plan itself tends to make use of the automata theory and the Petri net formalism (see Appendix A). For instance El Fallah *et al.* have modified Petri nets [10] to refine actions, to be compared to task decomposition. The multiagent aspect consists in merging individual plans. Another approach [9] uses hybrid automata to formalise and execute agent plans. The automata are synchronised so as to merge the plans. However, in the domain of individual planning, operational use of Petri nets is appearing for representing an itinerary and controlling the execution of the subsequent plan [4] or even as a task planning and scheduling tool compatible with Petri net design and analysis environments [15].

This paper aims at presenting a Petri net-based model for plans that eases plan information management during plan execution. The next section introduces the notion of agenticity to denote the team organisation. Section 3 formalises team plan Petri nets and their relations to team organisation. Finally section 4 exposes a way to extract individual plans from the team plan using a projection operator.

2. MISSION, AGENTS AND TEAM ORGANISATION

The general framework is a mission specified in terms of objectives: agents are implemented to carry out the mission and are hierarchically organised in a team.

2.1 Mission and Goals

The mission is characterised by an *objective* to be reached by the agent team. The objective is decomposed into mission *goals*, which are in turn decomposed into subgoals until reaching elementary goals.

DEFINITION 1. *(Agent) an* agent *is a physical entity equipped with resources (sensors, actuators, communication devices) that is implemented to achieve some goals within the mission, therefore contributing to the achievement of the objective. An* elementary agent *is an indivisible entity (e.g. a robot, a drone) whereas a* composite agent *is a set of agents that may themselves be organised as composite agents.*

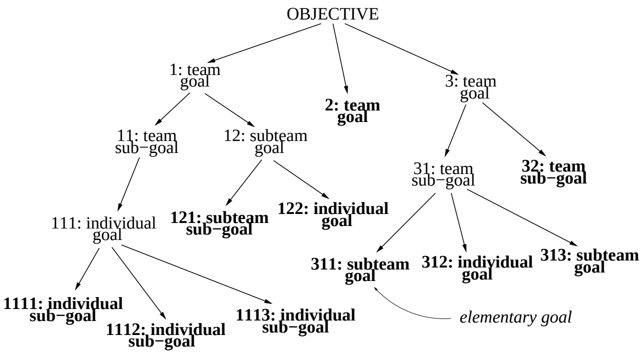


Figure 1: Decomposing the objective into a hierarchy of goals

Following Shoham's remark that a group of closely interacting agents can be considered as an agent in itself [19] a team of agents is equivalent to a composite agent.

DEFINITION 2. *(Goal) for an agent a goal corresponds to a possible state of the environment such that the actions of the agent tend to bring the environment to that state.*

The decomposition of the objective gives a hierarchy of goals that must be carried out [20] (fig. 1). Some goals involve elementary agents, other involve composite agents, *i.e.* subteams or even the team itself.

DEFINITION 3. *(Recipe) a recipe [12] is the specification of a course of actions to be performed by an agent, either composite or individual, resulting in the achievement of a goal.*

DEFINITION 4. *(Elementary goal) an elementary goal is such that there exists a known recipe to achieve it (fig. 1).*

Several recipes may be available to achieve one elementary goal. The team plan is extracted by organising a subset of the set of recipes. The initial plan is attached a possible organisation of the team.

2.2 Agenticity

When an agent is involved in a group of agents, some characteristics of the group are inherited by the agent. In particular if the group is involved in some activity, each individual agent is committed to that activity and to the interaction with its fellow agents [7]. To make use of this property we suggest to consider a team as an *agenticity hierarchy*, whose leaves are elementary agents and whose nodes are subteams, *i.e.* composite agents. Each node has for children nodes the agents that compose the subteam it represents (fig. 2). One can notice that there is no requirement that an individual agent be represented only once.

More formally the team X is composed of elementary agents $\{x_1, x_2, \dots, x_n\}$. It is hierarchically organised and each node in the hierarchy $\mathcal{H}_X$ is considered as an agent a_i [19]. Let $A = \{a_1, a_2, \dots, a_m\}$ be the set of agents in team X. Preliminary properties are that:

1. the team is an agent: $X \in A$, *i.e.* $\exists p \in [1, \dots, m], X = a_p$,
2. and each individual has a counterpart in the agent set A: $x_i \in X \Rightarrow \exists j \in [1, \dots, m], x_i = a_j$.

The father of agent a_i is denoted $father(a_i)$. $child(a_i)$ is the set of children of a_i. $child(.)$ and $father(.)$ are functions and

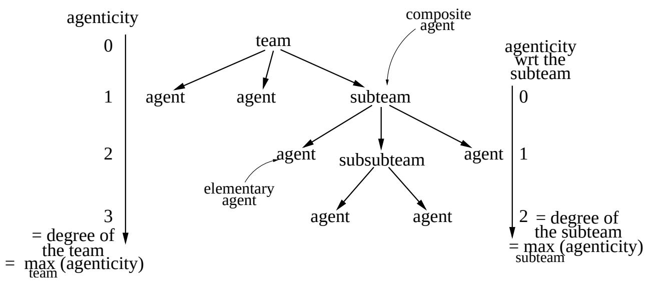


Figure 2: Agenticity Hierarchy

as such can be composed. The hierarchy $\mathcal{H}_X$ is an application: $\begin{cases} A \setminus \{x_1, x_2, \dots, x_n\} \rightarrow A \\ a_i \mapsto child(a_i) \end{cases}$.

DEFINITION 5. *(Agenticity) the agenticity of agent a_i with regards to team X is its depth in the hierarchy $\mathcal{H}_X$ whose root is the team: $Ag_X(a_i) = depth(a_i, \mathcal{H}_X) = (u|father^u(a_i) = X)$.*
The agenticity of agent a_i with regards to any subteam $a_j, a_i \subset a_j$ is its depth in the hierarchy $\mathcal{H}_{a_j}$ whose root is the considered subteam: $Ag_{a_j}(a_i) = depth(a_i, \mathcal{H}_{a_j}) = (u|father^u(a_i) = a_j)$.

DEFINITION 6. *The father agent of agent a_j is agent $a_k = father(a_j)$ corresponding to the father node in the hierarchy $\mathcal{H}_X$. The father's agenticity is less than the child's by 1:*
$a_j \subset child(a_k) \Rightarrow Ag_X(a_j) = Ag_X(a_k) + 1$.

Examples

1. The agenticity of an agent pertaining to no subteam is 1 with regards to the team: $X = a_p, x_i = a_j, \nexists k \in [1, \dots, m] \setminus \{j, p\} : a_j \subset a_k \Rightarrow Ag_X(x_i) = 1$ (fig. 3).
2. If all agents belong to the same team, the agenticity of the team is 0 with regards to the agent population:
$\forall i \in \{1, \dots, n\}, \exists j \in \{1, \dots, n\} : a_i = x_j \Rightarrow Ag_A(X) = 0$,
$A = \{x_i, i \in \{1, \dots, n\}\} \cup \{X\}$.

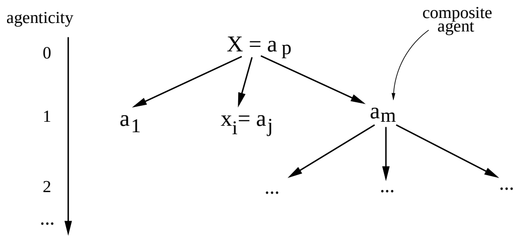


Figure 3: Example 1

DEFINITION 7. *(Degree) the degree of an agent is the highest agenticity of the individual agents that belong to this agent: $deg(a_j) = \max(Ag_{a_j}(x_i, i \in \{1, \dots, n\}, x_i \in a_j)$. An elementary agent has a null degree: $deg(x_i) = 0$.*

Example

3. If two elementary agents compose the only subteam of a given team, the team has a degree of 2: $a_j = \{x_{i_1}, x_{i_2}\}, A = \{X\} \cup \{a_j\} \cup \{x_i, i \in \{1, \dots, i_1, \dots, i_2, \dots, n\}\} \Rightarrow deg(X) = 2$ (fig. 4). The father a_j of the two agents x_{i_1} and x_{i_2} is the composite agent representing the subteam.

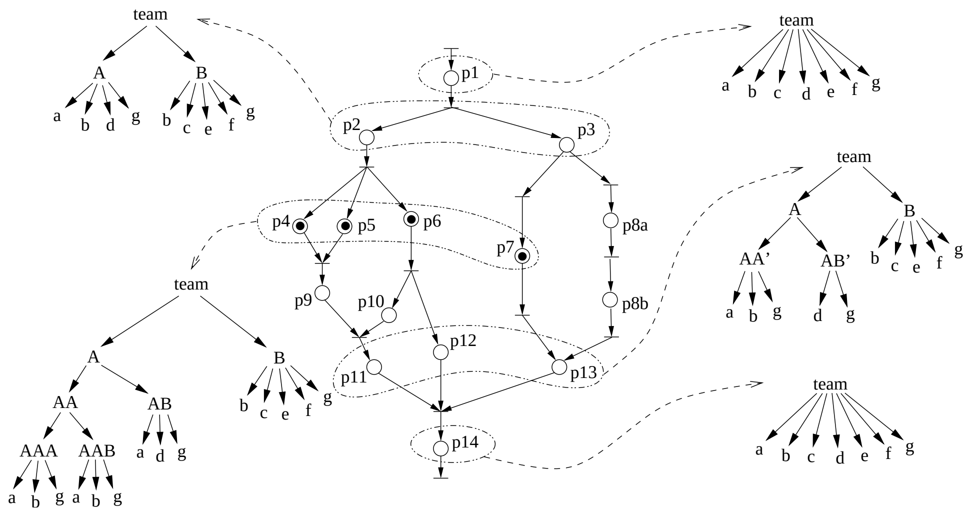


Figure 5: Team plan with some agenticity hierarchies

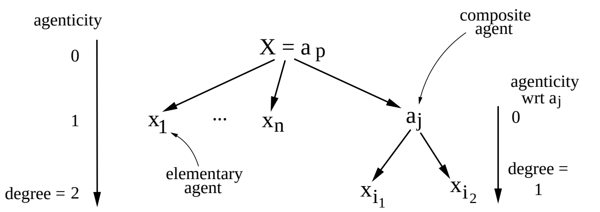


Figure 4: Example 3

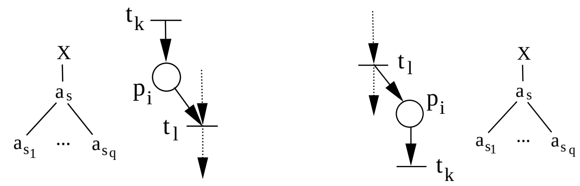


Figure 6: Source and sink structures and their associated agenticity hierarchies

3. TEAM PLAN REPRESENTATION

3.1 Team Plan Definition

The team plan is designed in terms of a detailed sequence of tasks, represented as a Petri net.

Let $\mathcal{P}_X$ be the detailed team plan. $\mathcal{P}_X$ is a coloured Petri net [14]: $\mathcal{P}_X = (P, T, S, N, \mathcal{C}, F)$, such that:

1. P is a finite set of places p_i, each place p_i represents the activity associated to an elementary goal;
2. T is a finite set of transitions t_j;
3. S is a finite set of arcs s_k;
4. N is a node function from S to $P \times T \cup T \times P$;
5. $\mathcal{C}$ is the colour set;
6. F is a colour function from P into $\mathcal{C}$.

$F : \begin{cases} P \rightarrow \mathcal{C} \\ p_i \mapsto \mathcal{H}_X(p_i) \end{cases}$. The set of token colours $\mathcal{C}$ is the set of agenticity hierarchies. The colour of a given token in a given place p_i, $F(p_i)$, is the branch in the agenticity hierarchy corresponding to the activity associated to the place: $F(p_i) = \mathcal{H}_X(p_i) = \{father^k(X|_{p_i}), k \in \{0, \ldots, deg(X|_{p_i})\}\}$, where the elementary agents involved in p_i are $X|_{p_i}$ (team restricted to the agents involved in p_i). Hence each reachable marking $\mathcal{M}$ corresponds to an agenticity hierarchy $\mathcal{H}_X(\mathcal{M})$ of the whole team X.

For example, in figure 5, reachable marking $\{p_4, p_5, p_6, p_7\}$ is associated to an agents' hierarchy with two subteams A and B, two subteams AA and AB of A and two subteams AAA and AAB of AA. Individual agents are the leaves of the hierarchy. Place p_4 is associated subgraph $(X, A, AA, AAA, leaves)$, p_5 subgraph $(X, A, AA, AAB, leaves)$, p_6 subgraph $(X, A, AB, leaves)$, p_7 subgraph $(X, B, leaves)$.

3.2 Analysing the Team Plan

The team plan bears some typical structures identified as modifications of the team organisation. Let us recall the notations $^\circ t_j$ for the input places of t_j, $t_j{}^\circ$ for its output places, $^\circ p_i$ for the input transitions of p_i and $p_i{}^\circ$ for its output transitions. $^\circ$ is readily composable: for instance $^{\circ\circ} p_i$ designates the set of input places of all input transitions of p_i.

DEFINITION 8. *(Source, fig. 6) Let* `source` *be the structure represented by a place p_i and a transition t_k such that $^\circ p_i = t_k$ and $^\circ t_k = \emptyset$.*
The hierarchy born by the structure has an agenticity of 1 with respect to the team: $\mathcal{H}_X(p_i)$[1]$= a_s$ and $child(a_s) = \{a_{s_1}, \ldots, a_{s_q}\}$.

The `source` structure allows the introduction of q agents into the team. It is worth noticing that t_k cannot bear two or more output places because this would mean that a group of agents is introduced in the team and immediately split. Common sense does not allow this, all the more since Petri net transitions are considered indivisible and instantaneous.

DEFINITION 9. *(Sink, fig. 6) Let* `sink` *be the structure represented by a place p_i and a transition t_k such that $p_i{}^\circ = t_k$ and $t_k{}^\circ = \emptyset$.*

[1]The reference to place p_i designates the agent associated to p_i.

The hierarchy born by the structure has an agenticity of 1 with respect to the team: $\mathcal{H}_X(p_i) = a_s$ and $child(a_s) = \{a_{s_1}, \dots, a_{s_q}\}$.

The `sink` structure allows the withdrawal or the abduction of q agents from the team. It is worth noticing that t_k cannot bear two or more input places because it would mean that several unsynchronised (since not pertaining to the same subteams) groups of agents withdraw from the team at the same time. Common sense does not allow it, all the more since Petri net transitions are indivisible and instantaneous.

DEFINITION 10. *(Fork, fig. 7) Let* `fork` *be the structure based on transition t_j such that ${}^\circ t_j = p_i$ and $t_j{}^\circ = \{p_{k_1}, p_{k_2}, \dots, p_{k_m}\}$. Firing transition t_j inserts before the individual level – for which $Ag = deg(\mathcal{H}_X(p_i))$ – a level of agenticity whose (composite) agents share out the individual agents among themselves: $deg(\mathcal{H}_X(t_j{}^\circ)) = deg(\mathcal{H}_X({}^\circ t_j)) + 1$. If in p_i, $child(a_a) = \{a_{b_1}, \dots, a_{b_q}\}$, in $p_{k_p}, p \in \{1, \dots, m\}$, $child(a_a) = \{a_{c_1}, \dots, a_{c_m}\}$ and $\cup_{s=1}^{m} child(a_{c_s}) = \{a_{b_1}, \dots, a_{b_q}\}$.*

The `fork` structure allows creating from a subteam m subteams whose levels of agenticity are increased by 1.

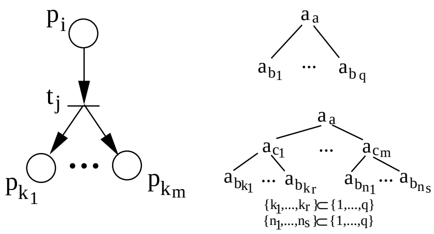


Figure 7: Fork structure and its associated agenticity hierarchies

DEFINITION 11. *(Merge, fig. 8) Let* `merge` *be the structure based on transition t_j such that ${}^\circ t_j = \{p_{i_1}, p_{i_2}, \dots, p_{i_m}\}$ and $t_j{}^\circ = p_k$.*
Firing transition t_j suppresses the level of agenticity before the individual level. It thus fuses the composite agents of the truncated level: $deg(\mathcal{H}_X(t_j{}^\circ)) = deg(\mathcal{H}_X({}^\circ t_j)) - 1$. If in $p_{i_p}, p \in \{1, \dots, m\}$, $child(a_a) = \{a_{c_1}, \dots, a_{c_m}\}$ and $\cup_{s=1}^{m} child(a_{c_s}) = \{a_{b_1}, \dots, a_{b_q}\}$, in p_k, $child(a_a) = \{a_{b_1}, \dots, a_{b_q}\}$.

The `merge` structure allows fusing m subteams to form a single subteam whose level of agenticity is decreased by 1.

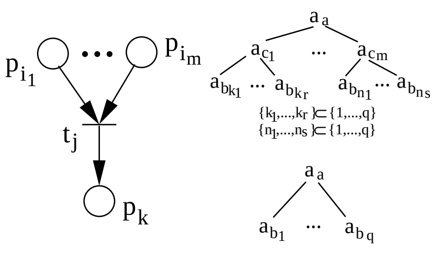


Figure 8: Merge structure and its associated agenticity hierarchies

DEFINITION 12. *(Reorganise, fig. 9) Let* `reorganise` *be the structure based on transition t_j such that ${}^\circ t_j = \{p_{i_1}, p_{i_2}, \dots, p_{i_m}\}$ and $t_j{}^\circ = \{p_{k_1}, p_{k_2}, \dots, p_{k_p}\}$.*
Combining characteristics of the two preceding structures, firing transition t_j modifies the composition and possibly the number of agents at level $deg(\mathcal{H}_X({}^\circ t_j)) - 1$. However it does not affect the degree of the subteam: $deg(\mathcal{H}_X({}^\circ t_j)) = deg(\mathcal{H}_X(t_j{}^\circ))$.
If in $p_{i_s}, s \in \{1, \dots, m\}$, $child(a_a) = \{a_{c_1}, \dots, a_{c_m}\}$ and $\cup_{u=1}^{m} child(a_{c_u}) = \{a_{b_1}, \dots, a_{b_q}\}$, in $p_{k_s}, s \in \{1, \dots, p\}$, $child(a_a) = \{a_{d_1}, \dots, a_{d_p}\}$ and $\cup_{u=1}^{p} child(a_{d_u}) = \{a_{b_1}, \dots, a_{b_q}\}$.

The `reorganise` structure allows fusing m subteams to form p new subteams, all of them bearing the same level of agenticity.

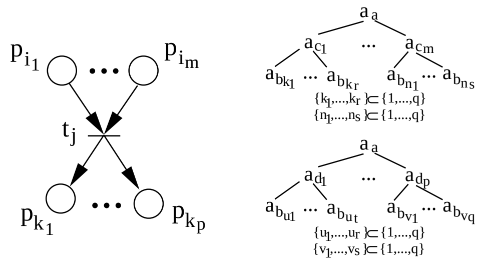


Figure 9: Reorganise structure and its associated agenticity hierarchies

DEFINITION 13. *(Transfer, fig. 10) Let* `transfer` *be the structure based on a place p_t such that $t_{j_1} = p_{i_1}{}^\circ = {}^\circ p_{k_1} = {}^\circ p_t$ and $t_{j_2} = {}^\circ p_{k_2} = p_{i_2}{}^\circ = p_t{}^\circ$.*
It modifies the composition but does not change the number of agents at level $deg(\mathcal{H}_X({}^\circ t_{j_{1,2}})) - 1$: there always remains two of them. The places in the structure correspond to the following agents:

- $p_{i_1} \to a_r = \{a_{r_u}, u \in \{1, \dots, m\}\}$;
- $p_{i_2} \to a_s = \{a_{s_u}, u \in \{1, \dots, p\}\}$;
- $p_t \to a_t = \{a_{t_u}, u \in \{1, \dots, q\}\}$;
- $p_{k_1} \to a'_r = \{a_{r_u}, u \in \{1, \dots, m\}\} \setminus \{a_{t_u}, u \in \{1, \dots, q\}\}$;
- $p_{k_2} \to a'_s = \{a_{s_u}, u \in \{1, \dots, p\}\} \cup \{a_{t_u}, u \in \{1, \dots, q\}\}$.

The `transfer` structure allows transferring q agents from the activity associated to p_{i_1} to that associated to p_{k_2}. This is equivalent to collocating a `source` structure and a `sink` structure where p_t represents the withdrawing agents on one side and the arriving agents on the other.

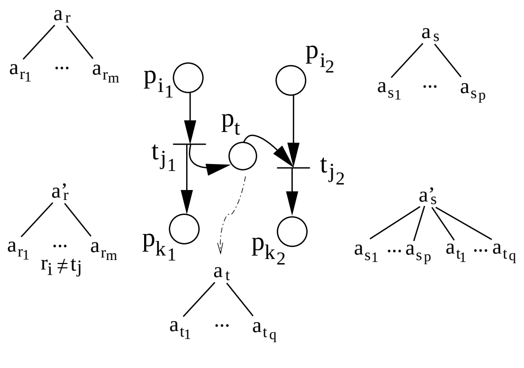


Figure 10: Transfer structure and its associated agenticity hierarchies

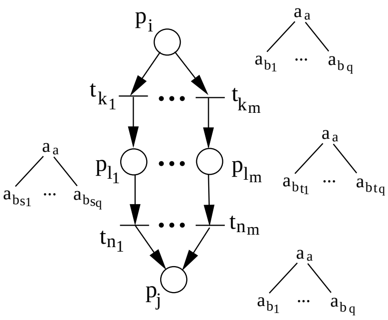


Figure 11: Choice structure and its associated agenticity hierarchies

DEFINITION 14. *(Choice, fig. 11) Let* `choice` *be the structure located between two places p_i and p_j such that*
$p_i{}^\circ = \{t_{k_1}, t_{k_2}, \ldots, t_{k_m}\}$ *and* $\forall u \in \{1, \ldots, m\}$,
$t_{k_u}{}^\circ = p_{l_u}$, $p_{l_u}{}^\circ = t_{n_u}$ *and* $t_{n_u}{}^\circ = p_j$.
The hierarchy is not modified by the structure:
$\mathcal{H}_X(p_i) = \mathcal{H}_X(p_j) = \mathcal{H}_X(p_{l_u}), \forall u \in \{1, \ldots, m\}$.

The `choice` structure allows proposing m possible activities for the considered subteam.

3.3 Abstracting the Team Plan

Representing a team plan using hierarchical coloured Petri nets [13, 16] – or modular coloured Petri nets [5, 16] – allows for more flexibility than coloured Petri nets and reduces the amount of duplicated information.

The net $\mathcal{P}_X$ can be abstracted so as to represent the activities at each level of agenticity. To build this information we extend the ordinary Petri net reduction rules. The Petri net is reduced according to the semantics of basic team management structures, namely `source`, `sink`, `fork`, `merge`, `reorganise`, `transfer` and `choice`.

RULE 1. — **Reduction of late arrival:** *(fig. 12) If t_k and p_i constitute a source structure,* i.e. $^\circ p_i = t_k$, $^\circ t_k = \emptyset$, $p_i{}^\circ = t_l$ *and* $\exists j \neq i, p_j{}^\circ = t_l$*, they become a single place p_{j,i^*}.*

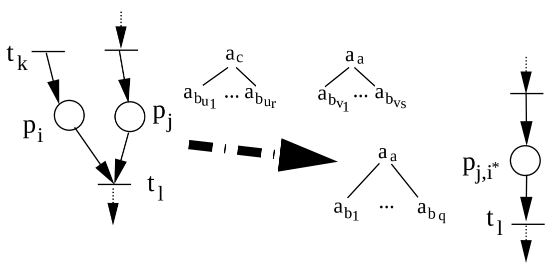


Figure 12: Rule 1 and its effect on hierarchy

Rule 1 preserves the level of agenticity. However the token is modified so as to encompass the newly introduced (individual or composite) agent.

RULE 2. — **Reduction of early withdrawal:** *(fig. 13) If p_i and t_k constitute a sink structure,* i.e. $p_i{}^\circ = t_k$, $t_k{}^\circ = \emptyset$, $^\circ p_i = t_l$ *and* $\exists j \neq i, {}^\circ p_j = t_l$*, they become a single place $p_{j^*,i}$.*

Rule 2 preserves the level of agenticity. However the token is modified so as to encompass the leaving (individual or composite) agent.

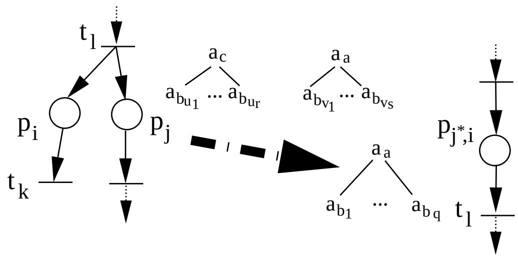


Figure 13: Rule 2 and its effect on hierarchy

RULE 3. — **Fusion of consecutive activities:** *(fig. 14) If p_{i_1}, $p_{i_2}, \ldots, p_{i_k}$ are k consecutive places,* i.e. $^\circ p_{r+1} = p_r{}^\circ, \forall r \in \{i_1, \ldots, i_{k-1}\}$*, they are substituted by a unique place $p_{i_1,i_2,\ldots,i_k}$.*

Rule 3 is a transposition of the substitution rule for consecutive places in ordinary Petri nets. It preserves the level of agenticity: the token is not modified.

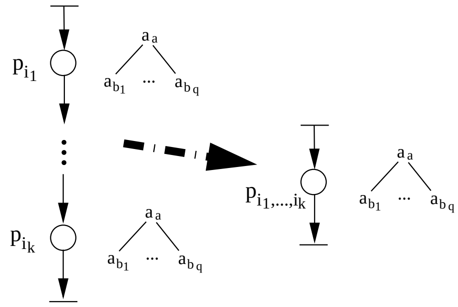


Figure 14: Rule 3 and its effect on hierarchy

RULE 4. — **Fusion of choice between activities:** *(fig. 15) If $p_{i_1}, p_{i_2}, \ldots, p_{i_k}$ are k possible places,* i.e. $^{\circ\circ}p_r = {}^{\circ\circ}p_s$, $p_r{}^{\circ\circ} = p_s{}^{\circ\circ}$, $^\circ p_r \neq {}^\circ p_s$, $p_r{}^\circ \neq p_s{}^\circ$, $\forall (r,s) \in \{i_1, \ldots, i_k\}, r \neq s$*, they are fused into a single place $\tilde{p}_{i_1,\ldots,i_k}$.*

Rule 4 preserves the level of agenticity. However the token is modified so as to bear, if needed, the different possible agenticity subhierarchies. The agent will be tagged as encompassing multiple possible organising structures.

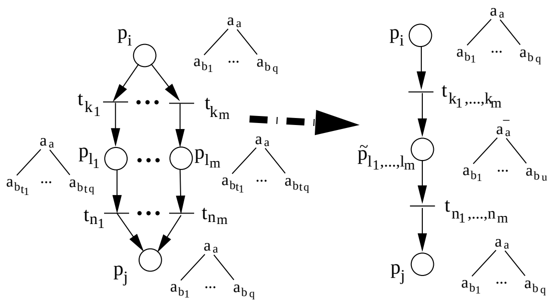


Figure 15: Rule 4 and its effect on hierarchy

RULE 5. — **Fusion of parallel activities:** *(fig. 16) If p_{i_1}, $p_{i_2}, \ldots, p_{i_k}$ are k places in parallel,* i.e. $^\circ p_r = {}^\circ p_s$, $p_r{}^\circ = p_s{}^\circ, \forall (r,s) \in \{i_1, \ldots, i_k\}$)*, they are replaced by a single place $p_{i_1,i_2,\ldots,i_k}$.*

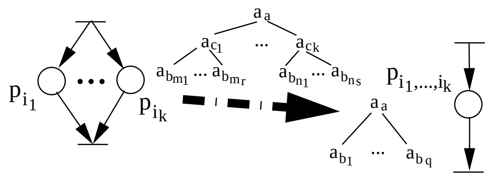


Figure 16: Rule 5 and its effect on hierarchy

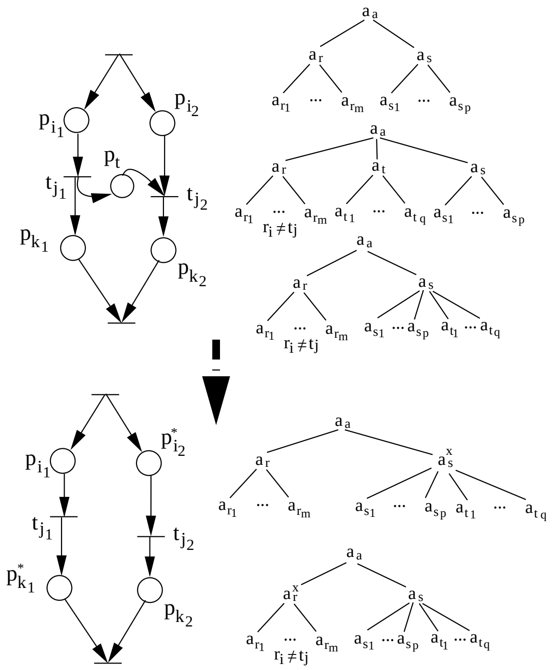


Figure 17: Rule 6 and its effect on hierarchy

Rule 5 is derived from the implicit place suppression rule in ordinary Petri nets. It decreases the level of agenticity by 1: the structure born by the token is shifted upwards.

RULE 6. — **Reduction of agent transfer:** *(fig. 17) If p_{i_1}, p_{i_2}, p_{k_1} and p_{k_2} are the four places of a transfer structure through p_t,* i.e. $p_{i_1}{}^\circ = {}^\circ p_{k_1} = {}^\circ p_t$ *and* ${}^\circ p_{k_2} = p_{i_2}{}^\circ = p_t{}^\circ$*, they are reduced into two separate branches with p_{i_1},$p^*_{k_1}$ and $p^*_{i_2}$, p_{k_2}.*

Rule 6 does not decrease the level of agenticity but modifies the contents of the structure: the structure born by the token is transformed so that the transferred agents are passed on. The father agents corresponding to each branch are tagged as operating a transfer. In fact the reduction is performed by splitting the transfer place p_t and then simultaneously applying rule 1 and 2 on the two separate branches of the structure.

The rules are iteratively applied, thus building the dynamic agenticity hierarchy. Rules 1 (source) and 2 (sink) allow to get rid of late-arriving and early-withdrawing agents before starting the iterative part of the algorithm. Iterations begin with testing and, if possible, applying rule 5 (parallel activities) until it can no longer be applied. Then the iteration breaks so as to skip to rule 3 (consecutive activities) to compress all sequences that have appeared in the iteration. If rule 5 cannot be applied at all, it is replaced by rule 4 (choice) which in turn, when not applicable, gives way to rule 6 (transfer). The process ends when the net is reduced to a single place. More details can be found in [3]. The resulting plan then consists of a hierarchical Petri net whose levels correspond to the levels of agenticity in the team. Each place develops into a sub-net of higher agenticity. The tokens in the sub-net hold the agents performing the activities corresponding to the marked places as well as the children of these agents (fig. 18).

4. REDUCTION AND PROJECTION: FROM TEAM PLAN TO INDIVIDUAL PLANS BY THE EXAMPLE

Figure 18 shows an example of a hierarchical team plan. The plan that appears in figure 5 is gradually reduced in order to yield a single-place Petri net. Let us consider the marking in greater details. p_4 and p_5 are parallel activities. Their tokens are similar and bear hierarchies respectively AAA and AAB and their children a, b and g. They are reduced into $p_{4,5}$ according to rule 5. The resulting token bears the piece of hierarchy AA with its children AAA and AAB. At the next level several reductions are possible. First rule 3 is applied on two sequences. $p_{8a} - p_{8b}$ is reduced into p_8 and $p_{4,5} - p_9$ becomes $p_{4,5,9}$. Then on one hand $p_{4,5,9}$, p_6, p_{10}, p_{11} and p_{12} show a transfer structure: they are reduced into $p^* = p^*_{4,5,6,9,10,11,12}$ according to rule 6. In that structure tokens bear from left to right AA and its children AAA and AAB, and AB and its children a, d and g. On the other hand p_8 is an alternative to p_7. They are reduced into $\tilde{p}_{7,8}$ according to rule 4. The token is not changed while moving through the sequence and bears b, c, e, f and g. Sequences $p_2 - p^*$ and $p_3 - \tilde{p}_{7,8} - p_{13}$ are aggregated into respectively $p_{2,*}$ and $p_{3,\tilde{7,8},13}$ according to rule 3. At this stage the structure resulting from all previous reductions bears these two parallel activities. The structure is reduced into $p_{2,\ldots,13}$ using rule 5. The last stage of the reduction concatenates the sequence $p_1 - p_{2,\ldots,13} - p_{14}$ into a single place p_m that represents the mission. The token in the sequence is composed of the hierarchy $team$ and its children. For p_1 the children are a, b, c, d, e, f and g. For $p_{2,\ldots,13}$ they are A and B.

The Petri net in figure 5 in fact corresponds to the *detailed global plan* built from the leaf-places of the hierarchical team plan in figure 18.

The hierarchical structure of the team plan now allows the agents' individual plans to be derived. This is done through a projection mechanism.

DEFINITION 15. *(Projection) the projection of the team plan on agent a_i is an agent plan whose hierarchy of places has been cut to level $Ag_X(a_i)$ and the hierarchies of agenticity are cut to level $Ag_X(a_i)$: $deg(\mathcal{H}_X(a_i)) = Ag_X(a_i)$.*

DEFINITION 16. *(Agent plan) the plan of agent a_i consists in the path of a_i's token in the team plan and all levels above. The corresponding activities all involve a_i or its ancestors in the agenticity hierarchies.*

The projection of the team plan on an agent consists in isolating the places of the corresponding level of agenticity in which the agent is involved and extracting the hierarchies of places and of agenticity associated to the places.

Let us unfold the previous example. Figure 19 shows the agent plan for the elementary agent d. At each level the team plan Petri net has been pruned so that the remaining places involve d or its ancestors. One can notice that the same operation can be performed locally for agent AA. Locality is a consequence of the fugacity of AA due to its being a composite agent. Modifying the team organisation according to the activity creates local cooperation groups. For instance in marking $\{p_4, p_5, p_6\}$, a, b and g are collaborating

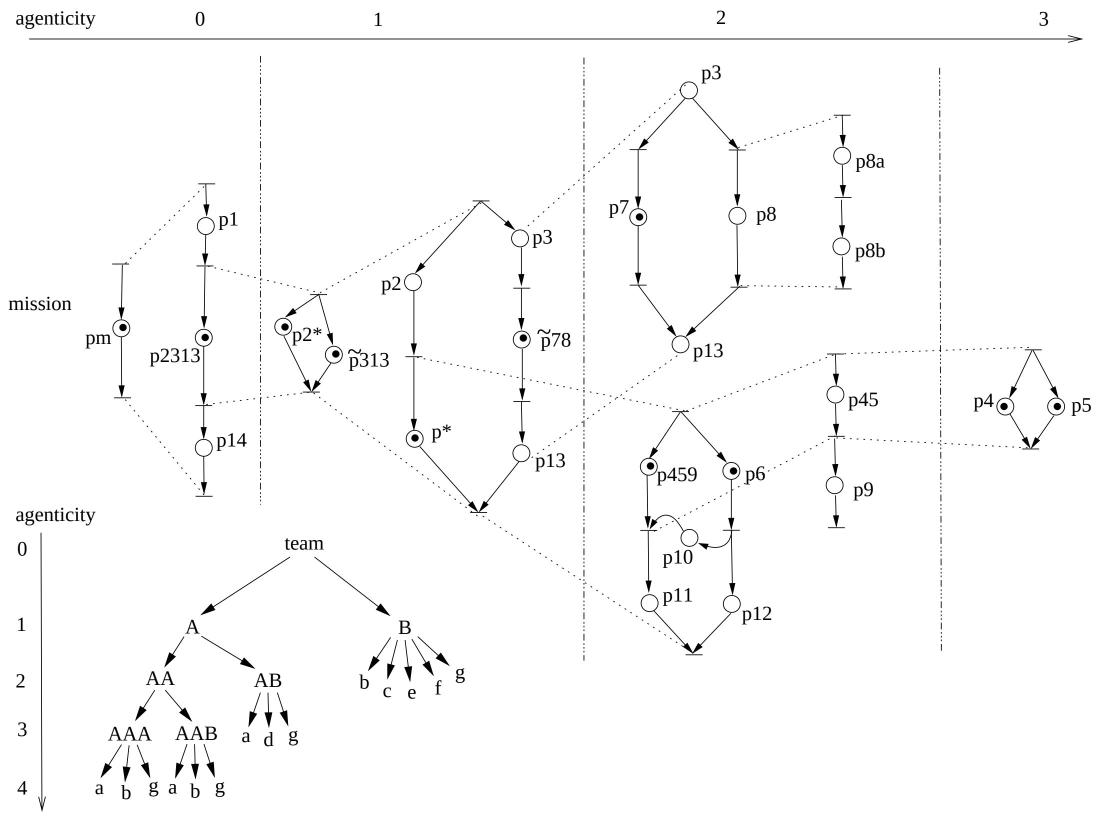


Figure 18: A hierarchical team plan with agenticity hierarchy tokens

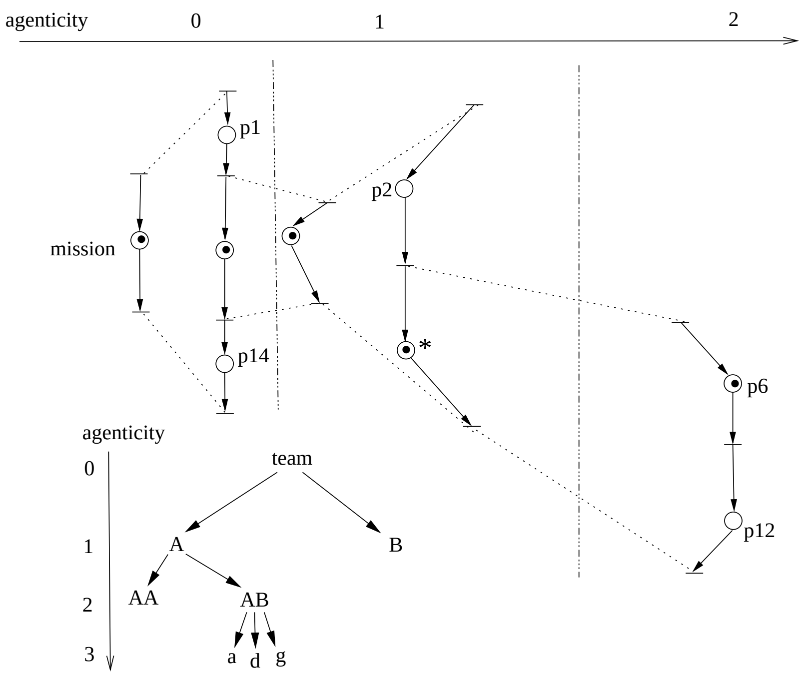


Figure 19: Projection of the team plan on agent d

for p_4 and p_5 in resp. AAA and AAB, whereas b is not individually involved with AB: their interface with AB is AA and AB does not need to know the specifics of the activity of each individual agent.

5. CONCLUSION

In the general framework of agents carrying out a mission specified in terms of objectives a Petri net-based representation of team plans is presented. In this approach agents are hierarchically organised in a team. Each node in the agenticity hierarchy can be regarded as an agent. The plan itself is represented by a hierarchical Petri net whose places are agents' activities. The organisation of the team dynamically changes as the marking in the net evolves. The team plan is designed for team activity monitoring.

From the team plan a projection operator allows to derive individual plans so that each elementary agent knows which agents may interact with it for any activity. The interaction information is held in the tokens of its plan as an agenticity sub-hierarchy, whereas the (hierarchical) marking gives the current activities at all levels of granularity/agenticity. The conjunction of individual plans permits distributed team coordination.

The distribution of the information at all levels of agenticity in each agent may facilitate team management. In particular, in the context of teams of robots, it may help in dynamically responding to an unforeseen event, such as a failure or an external action. A modification to the initial plan — a repair — will be provided, involving agents at an individual (robot) or global (team) level. Current and future works concern the development of EΛAIA, a Petri net-based decision architecture for local replanning within the team [1, 2].

6. REFERENCES

[1] O. Bonnet-Torrès. Local replanning within a team of cooperative agents. In *AAMAS'05 Doctoral Mentoring*, Utrecht, The Netherlands, July 2005. University of Utrecht.

[2] O. Bonnet-Torrès. Local replanning within a team of cooperative agents. In *ICAPS'05 Doctoral Consortium*, Monterey, CA, June 2005.

[3] O. Bonnet-Torrès and C. Tessier. From multiagent plan to individual agent plans. In *ICAPS'05 Workshop on Multiagent Planning and Scheduling*, Monterey, CA, June 2005.

[4] E. Chanthery, M. Barbier, and J.-L. Farges. Integration of mission planning and flight scheduling for unmanned aerial vehicles. In *ECAI'04 - Workshop on Planning and Scheduling: Bridging Theory to Practice*, Valencia, Spain, 2004.

[5] S. Christensen and L. Petrucci. Towards a modular analysis of coloured Petri nets. In *ATPN'92*, pages 113–133, Sheffield, UK, 1992.

[6] P. Cohen and H. Levesque. Intention is choice with commitment. *Artificial intelligence*, 42:213–261, 1990.

[7] P. Cohen and H. Levesque. Teamwork. *Noûs*, 25(4):487–512, 1991.

[8] R. David and H. Alla. *Discrete, continuous and hybrid Petri nets*. Springer-Verlag, 2005.

[9] A. El Fallah-Seghrouchni, I. Degirmencyan-Cartault, and F. Marc. Modelling, control and validation of multi-agent plans in dynamic context. In *AAMAS'04*, New York, 2004.

[10] A. El Fallah-Seghrouchni and S. Haddad. A recursive model for distributed planning. In *ICMAS'96*, Kyoto, Japan, 1996.

[11] K. Erol, J. Hendler, and D. Nau. HTN planning: complexity and expressivity. In *AAAI'94*, pages 1123–1128, Seattle, WA, 1994.

[12] B. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.

[13] P. Huber, K. Jensen, and R. Shapiro. Hierarchies in coloured Petri nets. In *ATPN'89*, pages 192–209, Bonn, Germany, 1989.

[14] K. Jensen. *Coloured Petri nets. Basic concepts, analysis methods and practical use*. Monographs in Theoretical Computer Science. Springer-Verlag, 2nd edition, 1997.

[15] L. Kristensen. Using coloured Petri nets to implement a planning tool, 2003. 4th Advanced Course on Petri Nets.

[16] C. Lakos. From coloured Petri nets to object Petri nets. In *ATPN'95*, pages 278–297, Turin, 1995.

[17] T. Murata. Petri nets: properties, analysis and applications. In *Proc. of the IEEE*, volume 77-4, pages 541–580, 1989.

[18] A. Rao and M. Georgeff. BDI agents: from theory to practice. In *ICMAS'95*, San Francisco, CA, 1995.

[19] Y. Shoham. Agent-oriented programming. *Artificial intelligence*, 60:51–92, 1993.

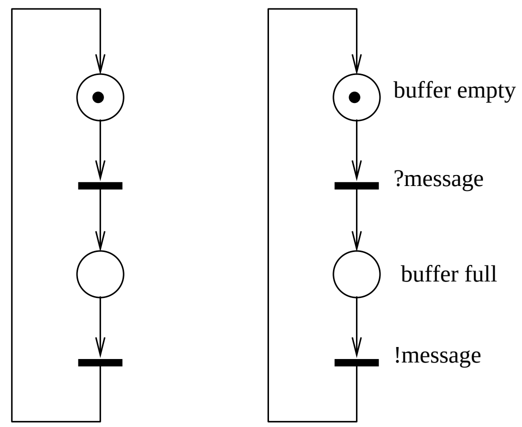
[20] M. Tambe. Teamwork in real-world, dynamic environments. In *ICMAS'96*, Kyoto, Japan, 1996.

[21] M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.

APPENDIX

A. A PETRI NET REMINDER

A *Petri net* $< P, T, F, B >$ is a bipartite graph with two types of nodes: $P = \{p_1, ..., p_i, ..., p_m\}$ is a finite set of places; $T = \{t_1, ..., t_j, ..., t_n\}$ is a finite set of transitions [17, 8] (fig. 20(a)). Arcs are directed and represent the forward incidence function $F : P \times T \rightarrow \mathbb{N}$ and the backward incidence function $B : P \times T \rightarrow \mathbb{N}$ respectively. An *interpreted Petri net* is such that conditions and events are associated with places and transitions respectively (fig. 20(b)). When the conditions corresponding to some places are satisfied, tokens are assigned to those places and the net is said to be marked. The evolution of tokens within the net follows transition firing rules. Petri nets allow sequencing, parallelism and synchronization to be easily represented.



(a) Standard Petri net (b) Interpreted Petri net

Figure 20: A model for a one-stage communication buffer