On Time-Stressed Team Collaboration

(Extended Abstract)

Xiaocong Fan
School of Engineering, The Behrend College
The Pennsylvania State University
Erie, PA 16563, USA
xfan@psu.edu

ABSTRACT

We extend the concept of timed Petri nets to explicitly capture timing constraints and collaboration points along a team process. Several properties of the model are examined. It can be employed as a collaboration mechanism for agents operating in a dynamic, time-stressed environment.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, Multiagent systems*

General Terms

Theory

Keywords

Timed Petri Nets, Teamwork, Collaboration

1. COLLABORATION NETS

Timed transition Petri nets (TTPT) [1] are extended Petri Nets with timing constraints on the firings of transitions. A timed transition Petri net is a tuple $\langle P, T, A, f_0, \gamma \rangle$, where P is a finite set of token places; T is a finite set of transitions; $A \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs connecting places and transitions; f_0 is the initial marking (specifying the number of tokens in each place $p \in P$); and γ is a mapping that associates each $t \in T$ with a time duration. The input places of a transition t, denoted by $\bullet t$, are the places from which an arc runs to t; the output places of t, denoted by $t\bullet$, are the places to which arcs run from t. Similarly, $\bullet p$ and $p\bullet$ denote p's input and output transitions, respectively.

We extend timed Petri nets with concepts for modeling multi-agent collaboration.

Definition 1 (Token Types). *A local token θ is a token that flows only within the net where it is originated; A sync token π is a token received from another peer agent through a synchronization message; A team token ω_G with respect to a group G of agents is a meta token which contains exactly one token slot for each agent $a \in G$.*

Cite as: On Time-Stressed Team Collaboration, (Extended Abstract), Xiaocong Fan, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 1353–1354
Copyright © 2009, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org), All rights reserved.

Let Θ, Π, and Ω be the universe of local tokens, sync tokens, and team tokens, respectively. Let $\omega_G(a)$ return a reference to the token in the slot for agent a in ω_G. $\omega_G(a) = null$ if the slot for a has no token.

Definition 2 (Fulfillment). *A team token ω_G is partially fulfilled when $\exists a \in G \cdot \omega_G(a) = null$. ω_G is fulfilled iff (a) $\forall a \in G \cdot \omega_G(a) \neq null$, (b) $\exists! b \in G \cdot \omega_G(b) \in \Theta$, and (c) $\forall a \neq b \in G \cdot \omega_G(a) \in \Pi$.*

That is, a team token is fulfilled iff there is exactly one token from the local agent and one sync token from each of the group members through synchronization messages.

Definition 3 (Node Types). *A local node is a Petri net place that consumes local tokens only. A team node with respect to a group of participating agents is a Petri net place that consumes local tokens (when a transition is fired) and sync tokens (when a message is received) to produce a team token and manages its fulfillment.*

A transition can trigger the execution of an activity. An activity can be performed by agents individually, or jointly performed by a group of agents. In the case of a joint activity, we assume there is a *team policy* regulating its execution. In the simplest form, a team policy should allow a group of agents to be aware of the teamwork progress.

We assume that each activity α is associated with a role constraint $\rho(\alpha)$, which specifies the capability requirements on doers. We use $CanDo(a, \alpha)$ to indicates that agent a satisfies $\rho(\alpha)$. We also assume that each activity α is guarded with a collection of first-order expressions, which is denoted by $pre(\alpha)$. An agent a cannot perform α if $pre(\alpha)$ is false when it is evaluated relative to a's mental state.

Because a group of agents need to synchronize their activities, we assume there is a global timing mechanism in the system and define timers in terms of global time points.

Definition 4 (Timer). *A timer ξ is a tuple $\langle s, e \rangle$, where s and e are the starting and ending global time points, respectively.*

Let $s(\xi)$ and $e(\xi)$ return the starting and ending points of ξ, respectively, and Ξ be the universe of timers.

Definition 5 (Collaboration Net). *A collaboration net is an extended TTPT $\langle L, M, T, A, f_0, \gamma, \delta \rangle$, where*

- *L is a finite set of local nodes;*
- *M is a finite set of team nodes. For each $m \in M$, let $G(m)$ returns the group of participating agents (which can be dynamically assigned in implementation);*

- *T is a finite set of transitions, each of which is associated with an activity. For each transition $t \in T$, let $\alpha(t)$ return the associated activity;*
- *$A \subseteq ((L \cup M) \times T) \cup (T \times (L \cup M))$ is a set of directed arcs connecting places and transitions;*
- *f_0 is the initial marking;*
- *$\gamma : T \rightarrow \Xi$. $\gamma(t)$ returns the timer associated with transition t; and*
- *$\delta : M \rightarrow \Xi$, where $\delta(p)$ returns the timer associated with team node p.*

We say that a group of agents share a collaboration net iff each agent has a local copy of the net and they all agree on the global timing constraints as given by γ and δ.

Definition 6 (Semantics). *When firing a transition t, an agent a will (i) immediately remove the tokens from its input places $\bullet t$; (ii) idle if $s(\gamma(t))$ has not come; (iii) when $s(\gamma(t))$ comes, start executing the associated activity $\alpha(t)$; and (iv) once $e(\gamma(t))$ comes, stop the activity, deposit one token in each of the output places $t\bullet$, and for any $p \in M \cap t\bullet$, the slot of the team token for the local agent a is filled, and the agent sends a synchronization message with a sync token to each agent peer in $G(p)$ excluding itself.*

Hence, firing a transition takes a fixed, finite amount of time, and any processing tasks associated with the transition can be performed during the allocated time interval.

2. TIME AND TEAM STRICTNESS

Definition 7 (Transition Enableness 1). *A transition t of a collaboration net is T-enabled w.r.t. agent a iff (i) $CanDo(a, \alpha(t))$ holds; (ii) $pre(\alpha(t))$ is evaluated by a as true; (iii) for any $p \in \bullet t$, there is a token in p; and (iv) for any $p \in \bullet t \cap M$, $e(\delta(p))$ has just come.*

Definition 8 (Time-strict Net). *A time-strict net is a collaboration net where a transition fires as soon as it is T-enabled.*

T-enableness is useful when the timeliness of a process is a concern. However, in the case that a team token is only partially fulfilled when $e(\delta(p))$ comes, some of the expected agents might not be ready for the activities associated with the subsequent transitions. As a consequence of using such a transition enableness, the quality of service could be sacrificed, because it might be the case that not all the allocated agents could fully contribute to the subsequent activities within the corresponding time intervals.

Definition 9 (Transition Enableness 2). *A transition t of a collaboration net is R-enabled w.r.t. agent a iff (i) $CanDo(a, \alpha(t))$ holds; (ii) $pre(\alpha(t))$ is evaluated by a as true; (iii) for any $p \in \bullet t$, there is a token in p; and (iv) for any $p \in \bullet t \cap M$, the team token in p is fulfilled.*

Definition 10 (Team-strict Net). *A team-strict net is a collaboration net where a transition fires as soon as it is R-enabled.*

Property 1. *An agent following a time-strict net (or a team-strict net) can be blocked (no enabled transitions) due to lack of condition or capability.*

An agent can wait when lack of condition happens. It typically indicates that the net is inappropriately designed when an agent can get blocked due to lack of capability.

Property 2. *A team-strict net containing a team node is sensitive to communication loss.*

Property 3. *Given a transition t of a team-strict net, $p \in M$ is t's input place only, and $\bullet t = \{p\}$. Assume communication is reliable, and $\forall a \in G(p) \cdot CanDo(a, \alpha(t))$ holds. If $e(\gamma(t))$ has not come, then regardless of whether $\alpha(t)$ is an individual or team activity, all agents in $G(p)$ who evaluate $pre(\alpha(t))$ as true will be involved in performing $\alpha(t)$.*

One issue regarding team-strict nets is that the timing constrains on a team node are completely ignored. What if, due to communication delay or some other reasons, $e(\delta(p))$ is already passed when the associated team token is fulfilled? To accommodate situations like this, we distinguish two types of team-strict nets.

Definition 11 (Time-greedy Team Net). *A time-greedy team net is a team-strict net, with the semantics of transition firing (Def. 6) expanded by (v) if the current time falls within $\delta(p)$, immediately fill the slot of p's team token reserved for the local agent, and send a synchronization message with a sync token to each agent peer in $G(p)$ excluding itself; (vi) if the current time falls within $\gamma(t')$, skip all the transitions between t and t' (including t), then start executing the associated activity $\alpha(t')$ if $pre(\alpha(t'))$ is true, and idle if $pre(\alpha(t'))$ is false; and (vii) when there exist multiple paths from t (i.e., in a net containing parallel branches), p or t' in rules (v) and (vi) is determined by following any one path where $a \in G(p')$ holds for any team node p' on the path and $CanDo(a, \alpha(t''))$ for any transition t'' on the path.*

Definition 12 (Time-adaptable Team Net). *A time-adaptable team net is a team-strict net with the semantics of transition firing (Def. 6) expanded by (v') if the current time falls beyond $\gamma(t)$, skip the activity, deposit one token in each of the output places $t\bullet$, and for any $p \in M \cap t\bullet$, fill the slot of the team token for the local agent, and send a synchronization message with a sync token to each agent peer in $G(p)$ excluding itself.*

A time-greedy team net allows an agent to advance immediately to a point where the unbroken timing constraints can be most likely honored. This is important when timeliness is critical. A time-adaptable team net allows an agent to conduct minimal chores (e.g. synchronization, belief update) before proceeding forward. Since all the activities with broken timers are skipped, a delayed agent could catch up with the other peer agents while still honoring the operation sequence. A time-adaptable team net offers a compromised solution when both timeliness and team coherence are the keys to teamwork success.

3. REFERENCES

[1] C. Ramchandani. Analysis of asynchronous concurrent systems by timed petri nets. Technical Report MAC-TR-120, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1974.