
Diagnostic pour les systèmes distribués dynamiques partiellement observables

Thomas Chatain

*IRISA/ENS Cachan-Bretagne
Campus de Beaulieu
F-35042 Rennes cedex, France
Thomas.Chatain@irisa.fr*

RÉSUMÉ. De plus en plus de composants des systèmes distribués émettent des alarmes ou des messages d'erreur lorsqu'une situation particulière se produit. Or la masse d'alarmes enregistrées oblige à construire des outils automatiques qui se basent sur un modèle du système pour remonter aux causes premières des anomalies. Nous nous intéressons au problème du diagnostic basé sur des modèles de vraie concurrence. Notre approche consiste à construire un dépliage d'un modèle du système supervisé en sélectionnant les histoires qui expliquent les alarmes observées. Dans cet article nous étendons la notion de dépliage symbolique de réseaux de Petri de haut niveau à un modèle de réseaux dynamiques que nous définissons. Ce modèle se rapproche des réseaux de Petri de haut niveau mais permet en plus de modéliser des systèmes dynamiques. Enfin nous montrons comment utiliser les dépliages symboliques des réseaux dynamiques dans le cadre du diagnostic.

ABSTRACT. More and more components of distributed systems emit alarms or error messages when some particular situation occurs. But the huge number of registered alarms forces to build automatized tools to find the cause of the failures. We are interested in the problem of diagnosis based on true concurrency models. Our approach consists in building an unfolding of a model of the supervised system, that selects the histories that explain the observed alarms. In this paper we extend the notion of symbolic unfolding of high-level Petri nets to a model of dynamic systems that we define. This model is close to high-level Petri nets and allows to model dynamicity. Finally we explain how to use symbolic unfoldings of dynamic nets for the diagnosis application.

MOTS-CLÉS : systèmes distribués dynamiques, diagnostic, réseaux de Petri de haut niveau, dépliages symboliques, observation partielle

KEYWORDS: dynamic distributed systems, diagnostic, high-level Petri nets, symbolic unfoldings, partial observation

1. Introduction

Dans les systèmes distribués actuels, comme les réseaux de télécommunications, de plus en plus de composants émettent des messages d'erreurs ou alarmes lorsqu'une situation particulière est rencontrée localement. Enregistrer ces alarmes permet d'aider à retrouver les causes des anomalies. C'est ce problème que nous appelons le diagnostic. La nature largement distribuée de certains systèmes suggère fortement de traiter ce problème en utilisant des modèles de vraie concurrence. Les travaux [BEN 03] sont basés sur les dépliages de réseaux de Petri [ENG 91]. Mais les réseaux de Petri ne sont pas facilement utilisables pour modéliser de vrais systèmes. Il convient donc de s'intéresser à leurs extensions. Les dépliages symboliques définis dans [CHA 04] permettent de faire du diagnostic sur des systèmes distribués modélisés par des réseaux de Petri de haut niveau [JEN 95]. Ce modèle convient pour la représentation de systèmes dont la structure est statique. Dans cet article nous étendons cette approche pour traiter aussi les systèmes distribués dynamiques. Nous adaptons les réseaux de Petri de haut niveau pour définir un modèle de réseaux dynamiques. Les extensions apportées s'inspirent des modèles décrits dans [BUS 01]. De plus, ils font intervenir des arcs de lecture [BAL 01, VOG 97, JAN 95] pour modéliser l'accès non destructif à des données. Ces arcs de lecture jouent aussi un rôle important dans la représentation de la dynamique. Nous proposons une notion de dépliage symbolique pour ce modèle.

Dans la première partie de l'article nous présentons le modèle de réseaux dynamiques. Puis nous définissons leurs dépliages symboliques. Enfin nous montrons comment les dépliages symboliques apportent une solution au problème du diagnostic.

2. Le modèle de réseaux dynamiques

Pour un ensemble X , on note 2^X l'ensemble des parties de X , X^\oplus les multi-ensembles finis sur X . On représente les multi-ensembles avec les délimiteurs $\{\cdot\}$ et on note \oplus et \ominus la somme et la différence des multi-ensembles. On utilise des conversions implicites entre les ensembles et les multi-ensembles lorsqu'il n'y a pas d'ambiguïté. On note $X \mapsto Y$ l'ensemble des applications de X dans Y .

Un *réseau dynamique* est un quadruplet $N \stackrel{\text{def}}{=} (Tok, PAR, \iota, W)$, où Tok est un ensemble de jetons, PAR est un ensemble de valeurs servant de paramètres et ι et W sont des applications telles que :

- $\iota \in Tok \mapsto (2^{Tok} \times Tok^\oplus \times PAR) \mapsto \mathbf{bool}$
- $W \in Tok \mapsto (2^{Tok} \times Tok^\oplus \times PAR \mapsto Tok)^\oplus$

Un marquage est un multi-ensemble de jetons $M \in Tok^\oplus$, qui représente l'état du système. Contrairement aux réseaux de Petri de haut niveau, les jetons ne sont pas disposés dans des places. Ceci ne restreint pas la généralité puisqu'on peut simuler l'utilisation de places en faisant correspondre à la présence d'un jeton de couleur c dans la place p d'un réseau de Petri de haut niveau, la présence d'un jeton de valeur (c, p) dans un réseau dynamique. Le fait de s'affranchir des places résout la difficulté de représenter dans un réseau de Petri de haut niveau une transition qui consommerait ou écrirait des jetons dans un ensemble de places non figé.

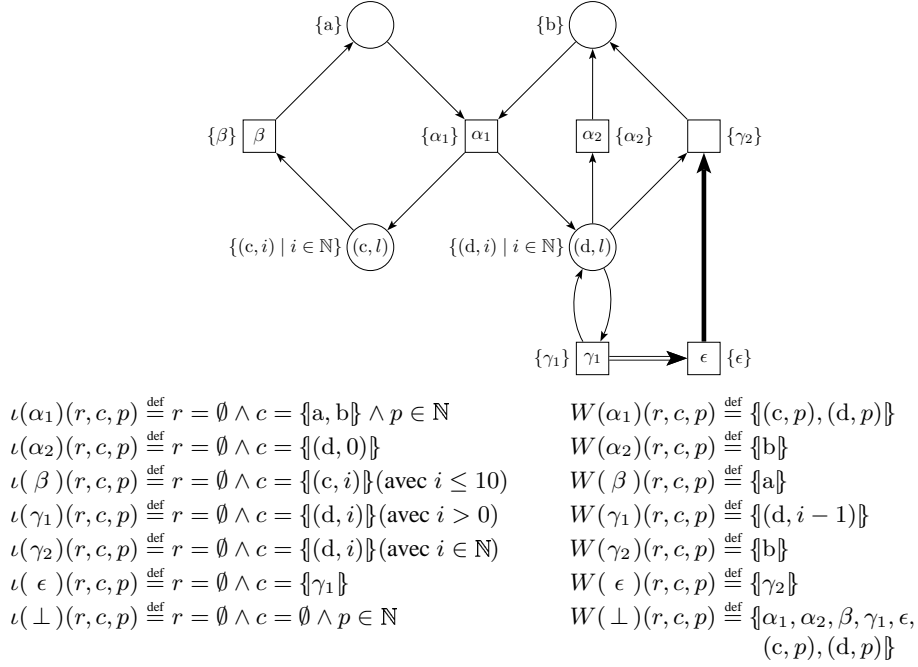


Figure 1. Un exemple de système distribué dynamique. Le système est représenté juste après une panne de sévérité l .

L'originalité de ce modèle est de considérer les transitions comme des jetons particuliers. Ainsi, elles font partie du marquage et peuvent être créées ou consommées. Or une transition doit être présente dans le marquage pour tirer. On peut donc modéliser des systèmes dynamiques dans lesquels des composants sont ajoutés ou supprimés au cours de l'exécution, en faisant apparaître ou disparaître des transitions.

Un jeton $t \in Tok$ est appelé *transition* s'il existe $r \in 2^{Tok}$, $c \in Tok^\oplus$ et $p \in PAR$, tels que $\iota(t)(r, c, p)$. Si de plus $(\{t\} \cup r) \oplus c$ est inclus dans M , alors la transition t peut tirer avec le paramètre p , en lisant les jetons de r et en consommant les jetons de c . Lorsque t tire, on obtient le marquage $M' \stackrel{\text{def}}{=} M \ominus c \oplus W(t)(r, c, p)$ (on note par commodité $W(t)(r, c, p)$ au lieu de $\{w(r, c, p) \mid w \in W(t)\}$).

On suppose que toutes les transitions consomment au moins un jeton à chaque fois qu'elles tirent. On ajoute seulement une transition initiale, notée \perp qui ne lit et ne consomme aucun jeton et sert à amorcer le système. Elle n'est pas considérée comme un jeton et n'apparaît jamais dans le marquage.

L'ensemble des transitions est noté T .

La figure 1 montre un système distribué dynamique. Sa description formelle est écrite en bas de la figure. Pour la représentation graphique, nous avons partitionné les jetons : chaque transition est isolée et représentée par un carré. Les ronds représentent chacun un sous-ensemble des jetons qui ne sont pas des transitions. Les jetons qui sont présents dans le réseau sont représentés à l'intérieur des ronds et des carrés. Dans la

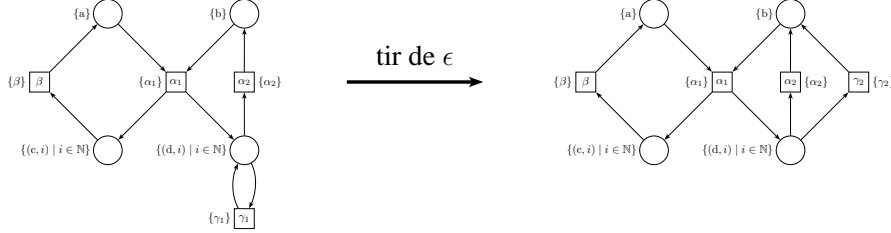


Figure 2. Le système avant et après le changement de structure. Le tir de ϵ supprime la transition γ_1 et crée la transition γ_2 .

figure on trouve les jetons (c, l) , (d, l) , α_1 , α_2 , β , γ_1 et ϵ . On note que la transition γ_2 est absente dans le marquage représenté. Les flèches représentent la consommation et la création de jetons par les transitions. Une flèche reliant une transition t à une transition t' représente soit la consommation de t par t' (flèche large vide), soit la création de t' par t (flèche large pleine).

Le système représenté fait intervenir trois composants A , B et C . Chaque action de A émet une alarme α , B émet des alarmes β , et C des alarmes γ . Les alarmes émises par les composants A et B sont observées par le même capteur 1. Le capteur 2 observe les alarmes émises par le composant C . Le composant principal A est susceptible de tomber en panne (transition α_1) avec une sévérité $l \in \mathbb{N}$ variable. Il émet alors une alarme α , qui ne donne aucune information sur la sévérité. Dans le marquage représenté, le système vient de tomber en panne avec une sévérité l , d'où la présence des jetons (c, l) et (d, l) . Pour notre exemple de diagnostic, nous considérons que le réseau démarre juste après une panne de sévérité inconnue.

Après une panne, la réparation est possible et fait intervenir les deux autres composants. Le composant B doit faire une action (transition β), possible uniquement si la sévérité est inférieure à 10. Le composant C doit intervenir aussi. Initialement, il lui faut l actions γ_1 pour réparer une panne de sévérité l , et ces actions doivent être complétées par l'action α_2 du composant A . Mais le composant C est susceptible d'être amélioré, pour qu'il puisse réparer la panne en une seule action γ_2 , et sans que A n'ait à intervenir ensuite. L'amélioration est représentée par la transition ϵ , qui supprime γ_1 et crée γ_2 . Ce changement n'est pas observable. La figure 2 montre les transitions (autres que ϵ) présentes dans le réseau avant et après ce changement.

3. Dépliage symboliques

Le dépliage symbolique représente toutes les histoires d'un système distribué dynamique. La figure 3 montre un préfixe du dépliage du système de la figure 1.

Pour un réseau dynamique $N \stackrel{\text{def}}{=} (Tok, PAR, \iota, W)$, on définit inductivement l'ensemble E d'événements par :

- l'événement initial $e_{\perp} \in E$, et on note $\tau(e_{\perp}) \stackrel{\text{def}}{=} \perp$ et $e_{\perp} = \bullet e_{\perp} \stackrel{\text{def}}{=} \emptyset$;
- pour tout $t \in T$ et pour tous $b, r_1, \dots, r_n, c_1, \dots, c_m$ de la forme (f, w) avec $f \in E$ et $w \in W(\tau(f))$, $e = (t, b, \{r_1, \dots, r_n\}, \{c_1, \dots, c_m\}) \in E$, et on note

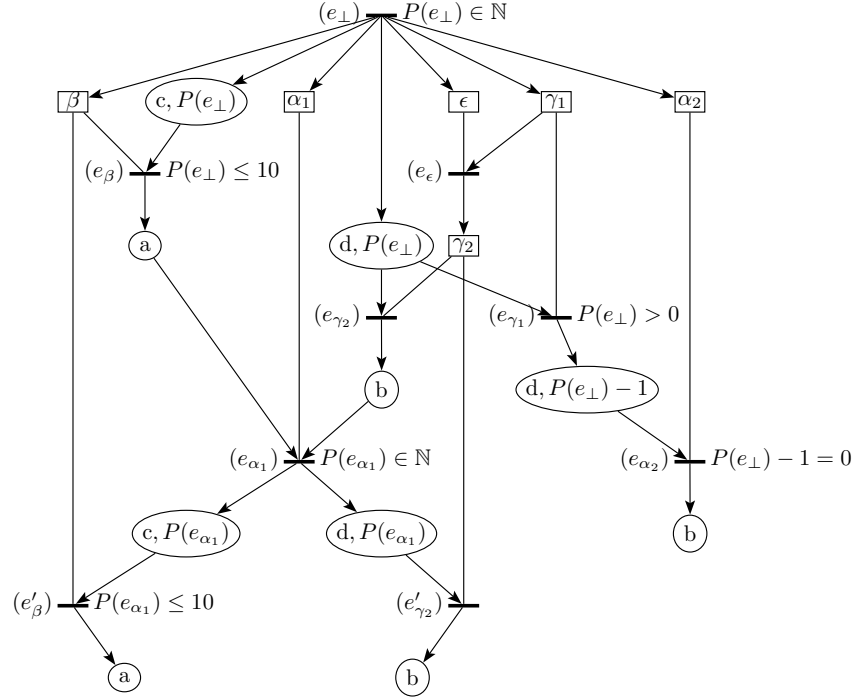


Figure 3. Un préfixe du dépliage symbolique du réseau de la figure 1.

$\tau(e) \stackrel{\text{def}}{=} t, \underline{e} \stackrel{\text{def}}{=} \{b, r_1, \dots, r_n\}$ et $\bullet e \stackrel{\text{def}}{=} \{c_1, \dots, c_m\}$.

Un couple (e, w) avec $e \in E$ et $w \in W(\tau(e))$ est appelé *condition* et représente la présence d'un jeton à un certain moment de l'histoire (entre sa création et sa consommation). Chaque événement $e \in E$ crée un ensemble des conditions défini par $e^\bullet \stackrel{\text{def}}{=} \{(e, w) \mid w \in W(\tau(e))\}$.

Un événement $e = (t, b, R, C)$ représente le tir de la transition t (présente dans la condition b), en lisant les conditions de R et en consommant les conditions de C . Chaque événement e est représenté par une barre noire. Les conditions créées par e sont dessinées et des flèches indiquent cette création. De même des flèches indiquent les conditions $\bullet e$ consommées par e et des lignes indiquent les conditions \underline{e} lues par e .

L'événement initial est placé en haut. Les flèches qui en sortent pointent vers des conditions correspondant aux jetons qui ont été créés par la transition initiale. Certains de ces jetons sont des transitions; nous avons représenté les conditions correspondantes par des rectangles. Les autres conditions sont ovales.

On définit les relations \rightarrow , \rightsquigarrow et \nearrow sur E comme suit :

- $e \rightarrow e'$ ssi $e^\bullet \cap (\underline{e'} \cup \bullet e') \neq \emptyset$
- $e \rightsquigarrow e'$ ssi $\underline{e} \cap \bullet e' \neq \emptyset \vee (e \neq e' \wedge \bullet e \cap \bullet e' \neq \emptyset)$
- $e \nearrow e'$ ssi $e \rightarrow^+ e' \vee e \rightsquigarrow e'$

Pour tout événement $e \in E$ on définit le passé de e : $[e] \stackrel{\text{def}}{=} \{f \in E \mid f \rightarrow^* e\}$, où \rightarrow^* représente la fermeture réflexive transitive de \rightarrow .

Pour tout $F \subseteq E$, on définit $\lceil F \rceil \stackrel{\text{def}}{=} \bigcup_{f \in F} \lceil f \rceil$.

Dans le cas de l'événement initial de notre exemple, les jetons créés dépendent de la sévérité de la panne qui s'est produite juste avant le début du diagnostic. Plus généralement, les jetons créés par un événement e dépendent du paramètre avec lequel la transition correspondante a tiré et des jetons lus et consommés par la transition, qui dépendent eux-mêmes des paramètres des événements du passé de e . On écrit dans les conditions des expressions qui représentent la valeur du jeton en fonction de ces paramètres. Ces expressions sont les $val(e, w, P)$ définies comme suit.

Considérons une condition (e, w) . Si l'on dispose d'une application P qui associe à chaque événement $f \in [e]$ une valeur $P(f) \in PAR$, on peut définir le jeton $val(e, w, P) \in Tok$ que l'événement e a créé sur sa sortie w si les événements de $[e]$ ont eu lieu avec les paramètres donnés par P .

Pour e_{\perp} , $val(e_{\perp}, w, P) \stackrel{\text{def}}{=} w(\emptyset, \emptyset, P(e_{\perp}))$.

Pour $e = (t, b, R, C)$, $val(e, w, P) \stackrel{\text{def}}{=} w(\{val(e', w', P) \mid (e', w') \in R\}, \{\!\!\{val(e', w', P) \mid (e', w') \in C\}\!\!\}, P(e))$

Si la panne initiale a eu lieu avec une sévérité inférieure à 10, la transition β peut tirer en consommant le jeton $(c, P(e_{\perp}))$ et en créant un jeton a . On place donc un événement e_{β} . Une flèche de la condition correspondant à $(c, P(e_{\perp}))$ vers l'événement e_{β} indique la consommation du jeton, et un trait (arc de lecture) de la condition correspondant à la transition β vers l'événement e_{β} indique que cet événement ne peut avoir lieu que si la transition β est bien présente. Enfin, la condition $P(e_{\perp}) \leq 10$ est attachée à l'événement e_{β} pour rappeler que e_{β} n'est pas possible dans tous les cas. L'événement e_{γ_1} est placé de façon analogue. Les conditions attachées aux événements sont les $loc_pred(e, P)$ définis plus loin.

L'événement e_{ϵ} est placé aussi car la transition ϵ peut tirer après la transition initiale. Cette action consomme la transition γ_1 , et crée la transition γ_2 . La flèche de γ_1 à e_{ϵ} empêche de placer dans le futur de l'événement e_{ϵ} , un événement correspondant au tir de la transition γ_1 . De plus, si les événements e_{ϵ} et e_{γ_1} apparaissent tous les deux dans la même histoire, alors e_{γ_1} apparaît nécessairement avant e_{ϵ} . D'autre part, il est maintenant possible de placer un événement correspondant au tir de γ_2 .

Pour un événement e et une application P , on définit le booléen $loc_pred(e, P)$ qui décide si e peut avoir lieu avec le paramètre qui lui est associé. Les valeurs des jetons en entrée de e sont connues grâce à la fonction val .

$loc_pred(e_{\perp}, P) \stackrel{\text{def}}{=} \iota(\perp)(P(e_{\perp}))$;

si $e = (t, (f, w), R, C)$, $loc_pred(e, P) \stackrel{\text{def}}{=} \begin{cases} val(f, w, P) = t \\ \wedge \iota(t)(\{val(e', w', P) \mid (e', w') \in R\}, \{\!\!\{val(e', w', P) \mid (e', w') \in C\}\!\!\}, P(e)) \end{cases}$

Notons que le dépliage permet de représenter plusieurs histoires. On dit qu'un ensemble d'événements sont en conflit s'ils ne peuvent pas appartenir à la même histoire. Un conflit apparaît entre autres lorsque deux événements consomment la même condition, comme e_{β_1} et e_{β_2} . Par contre la lecture concurrente d'une condition par plusieurs conditions est possible sans conflit.

Les événements de l'ensemble $F \subseteq E$ sont en *conflit*, noté $\#(F)$, si :

$$\begin{cases} \exists e_0, e_1, \dots, e_n \in [F] \quad e_0 \nearrow e_1 \nearrow \dots \nearrow e_n \nearrow e_0 & \text{(conflit structurel)} \\ \vee \exists P \in [F] \mapsto PAR \quad \bigwedge_{f \in [F]} loc_pred(f, P) & \text{(conflit non structurel)} \end{cases}$$

On appelle *dépliage symbolique* du réseau dynamique N , l'ensemble des événements qui ne sont pas en auto-conflit : $U \stackrel{\text{def}}{=} \{e \in E \mid \neg\#(\{e\})\}$.

4. Diagnostic pour les systèmes dynamiques partiellement observables

Dans notre approche, nous considérons que plusieurs capteurs enregistrent chacun une séquence d'alarmes émises par les composants du système. Un superviseur collecte alors les séquences d'alarmes de tous les capteurs. Dans notre exemple, les séquences d'alarmes enregistrées sont (β, α) et (γ) . On considère qu'aucune alarme n'est perdue et que l'ordre d'observation des alarmes par un capteur ne contredit pas les causalités définies dans le modèle.

L'observation est partielle pour différentes raisons. D'une part les causalités entre les événements qui ont produit les alarmes ne sont pas connues. D'autre part, il arrive que plusieurs actions émettent des alarmes indistinguables. Certaines actions sont même silencieuses. Enfin, tous les paramètres des actions ne sont pas observables. C'est le cas de la sévérité de la panne dans notre exemple.

Nous avons montré comment le dépliage symbolique permet de représenter les différentes histoires d'un système. Dans le cadre du diagnostic nous calculons une structure similaire, mais dans laquelle ne figurent que les histoires qui expliquent les observations. Cela se fait en contraignant au préalable le modèle en rajoutant des aspects correspondant à l'observation. Ce mécanisme est présenté dans [CHA 04] dans le cadre des réseaux de Petri de haut niveau.

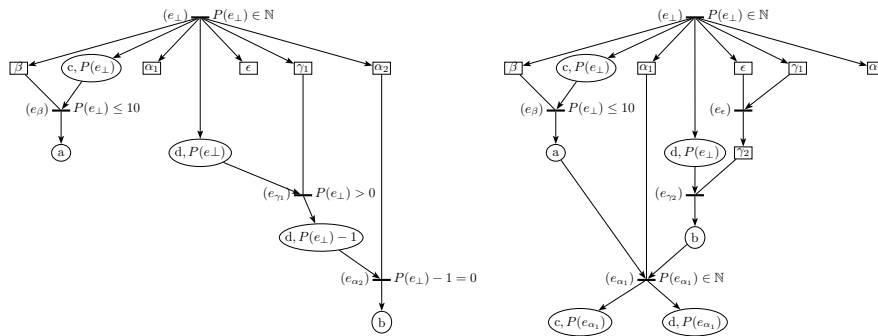


Figure 4. Les deux histoires qui expliquent les observations (β, α) et (γ) dans le système représenté dans la figure 1.

On a présenté dans la figure 4 les deux histoires qui expliquent les alarmes observées. Si c'est la première histoire qui a eu lieu, alors l'alarme γ a été émise par la transition γ_1 , et l'alarme α par la transition α_2 . De plus les deux événements correspondant sont causalement liés. Par contre les transitions β et α_1 ont tiré de manière concurrente. D'autre part, la confrontation des contraintes associées aux événements nous apprend que la panne d'origine avait une sévérité $P(e_{\perp}) = 1$.

Si c'est la deuxième explication qui est la bonne, alors les transitions β et γ_2 ont tiré de manière concurrente. La transition ϵ a tiré, puisque γ_2 est apparue. Puis le système est encore tombé en panne (transition α_1). Dans ce scénario, on ne connaît pas la sévérité des pannes ; on sait seulement que la première panne avait une sévérité inférieure à 10.

5. Conclusion

Dans cet article, nous avons présenté une méthode pour le diagnostic dans les systèmes distribués dynamiques partiellement observables. Pour cela nous avons adapté la notion de dépliage symbolique introduite dans [CHA 04] à un nouveau modèle de réseaux dynamiques que nous définissons. Ce modèle permet de modéliser les aspects dynamiques et les paramètres des systèmes. Dans le cadre du diagnostic, on montre comment l'utiliser pour contraindre le comportement d'un système en fonction d'une observation partielle, dans le but de construire un dépliage symbolique représentant exactement toutes les histoires qui expliquent les observations. Nous travaillons à la réalisation d'un prototype permettant d'expérimenter notre méthode.

6. Bibliographie

- [BAL 01] BALDAN P., CORRADINI A., MONTANARI U., « Contextual Petri Nets, Asymmetric Event Structures, and Processes », *Inf. Comput.*, vol. 171, n° 1, 2001, p. 1–49.
- [BEN 03] BENVENISTE A., HAAR S., FABRE E., JARD C., « Distributed Monitoring of Concurrent and Asynchronous Systems », *CONCUR*, 2003, p. 1–26.
- [BUS 01] BUSCEMI M. G., SASSONE V., « High-Level Petri Nets as Type Theories in the Join Calculus », *FoSSaCS*, 2001, p. 104–120.
- [CHA 04] CHATAIN T., JARD C., « Symbolic Diagnosis of Partially Observable Concurrent Systems », *FORTE*, 2004, p. 326–342.
- [ENG 91] ENGELFRIET J., « Branching Processes of Petri Nets », *Acta Inf.*, vol. 28, n° 6, 1991, p. 575–591.
- [JAN 95] JANICKI R., KOUTNY M., « Semantics of Inhibitor Nets », *Inf. Comput.*, vol. 123, n° 1, 1995, p. 1–16.
- [JEN 95] JENSEN K., *Coloured Petri nets: basic concepts, analysis methods and practical use*, Springer-Verlag, 1995.
- [KHO 03] KHOMENKO V., KOUTNY M., « Branching Processes of High-Level Petri Nets », *TACAS*, 2003, p. 458–472.
- [VOG 97] VOGLER W., « Partial Order Semantics and Read Arcs », *MFCS*, 1997, p. 508–517.