Caractérisation des transitions temporisées dans les logs de conversation de services Web (version longue)

Didier Devaurs, Fabien De Marchi, Mohand-Saïd Hacid

LIRIS, UMR 5205, CNRS / Université Claude Bernard Lyon 1
Bâtiment Nautibus, 8 boulevard Niels Bohr, F-69622 Villeurbanne, France
didier.devaurs@irisa.fr, {fabien.demarchi, mohand-said.hacid}@liris.cnrs.fr

Résumé

La connaissance du protocole de conversation d'un service Web est importante, pour les utilisateurs et les fournisseurs de services car il en modélise le comportement externe. Cependant, ce protocole n'est souvent pas spécifié pendant la conception. Notre travail s'inscrit dans une thématique d'extraction du protocole de conversation temporisé d'un service existant à partir de ses données d'exécution (logs). Nous en étudions un sous-problème important qui est la découverte des transitions temporisées, i.e. la recherche des changements d'états liés à des contraintes temporelles. Nous présentons un cadre formel regroupant différents concepts, dont les *expirations propres*, qui représentent pour les logs ce que sont les transitions temporisées dans le protocole de conversation. A notre connaissance, ceci représente la première contribution à la résolution de ce problème.

1 Introduction

Les services Web sémantiques constituent la nouvelle génération des technologies du Web pour l'intégration d'applications. Ils se situent à la convergence de deux domaines de recherche importants : les services Web et le Web sémantique. Selon la définition proposée par Tim Berners-Lee [5], le Web sémantique devrait constituer une extension du Web traditionnel, dans laquelle on donnerait à chaque information un sens bien défini, afin de permettre aux ordinateurs de traiter ces informations *de façon intelligente*, et d'améliorer ainsi la coopération entre les hommes et les machines.

La notion de service Web désigne un composant logiciel mis à disposition par un fournisseur, et invocable sur Internet par des clients (que ce soient des utilisateurs ou d'autres services), les échanges entre services Web se faisant au travers de communications asynchrones, sous la forme de messages. L'ambition portée par les services Web est de permettre une plus grande interopérabilité entre applications sur Internet. On envisage ainsi des services Web capables, de façon automatique, de se découvrir et d'être découverts, de négocier entre eux, ou de se composer en des services plus complexes. Le but ultime de cette vision consiste à créer des applications constituées uniquement de services Web interagissant entre eux. Dans une telle optique, peu importe où est déployé le service (en interne, chez un partenaire, ...) ; ce qui importe est alors que le service remplisse un rôle bien précis. L'une des sources de complexité provient de la capacité des services à se composer entre eux, et donner ainsi naissance, de façon transparente pour l'extérieur, à de nouveaux services.

Les technologies classiques d'intégration de systèmes d'informations hétérogènes, telles J2EE ou CORBA, nécessitent la conception de ponts logiciels *ad-hoc* coûteux et à fort couplage ; l'entretien de telles solutions requiert une forte cohésion entre les différents acteurs impliqués de part et d'autre (concepteurs et intégrateurs). Au contraire, les services Web permettent de réaliser une intégration à faible couplage et à moindre coût, du fait qu'ils utilisent des standards généralistes

et fortement répandus, tels XML ou HTTP. Cependant, cette souplesse d'intégration est possible uniquement si les utilisateurs éventuels d'un service savent comment interagir avec celui-ci. A un service doivent donc être associées des descriptions suffisamment riches pour permettre la compréhension de sa sémantique d'exécution. Progressivement, des standards s'imposent donc pour la publication homogène des services Web.

Le langage WSDL, par exemple, permet de spécifier l'interface d'un service : les opérations réalisées, les types de messages reçus et envoyés, les formats des entrées et des sorties. Toutefois, des travaux récents [4] ont montré que ces spécifications étaient insuffisantes dans l'optique d'une utilisation automatique des services Web, et ont proposé l'utilisation du protocole de conversation d'un service ou d'un groupe de services. Ces travaux mettent en avant l'importance que revêt la publication de la partie dynamique d'un service, en venant s'ajouter à la représentation de sa partie statique, assurée par l'interface WSDL. Au-delà de la description des messages envoyés, il s'agit de spécifier quelles sont les séquences ordonnées de messages (appelées conversations) qu'un service est en mesure d'émettre ou de recevoir au sein d'une même session. Le formalisme des automates finis permet de modéliser ces ensembles de conversations possibles, et de pratiquer un certain nombre de raisonnements automatiques et efficaces. Plus récemment, les mêmes auteurs [2, 3] ont également ajouté une notion de contrainte temporelle à leur modèle, rebaptisé de ce fait protocole de conversation temporisé.

Il apparaît clairement que l'utilisation d'une telle modélisation des services Web offre de nombreuses applications, principalement en ce qui concerne la vérification automatique de bon fonctionnement, de compatibilité, etc. Néanmoins, en pratique, un grand nombre de services ne possèdent pas une telle spécification. En effet, une technique classique pour concevoir un service Web consiste simplement à migrer une application existante afin de la rendre compatible avec les standards adéquats (SOAP, ...). Les contraintes de production conduisent souvent les développeurs à négliger ou remettre à plus tard la spécification du protocole de conversation. La question qui se pose alors est la suivante : est-il possible d'obtenir le protocole de conversation d'un service s'il n'a pas été défini lors de la phase de conception ?

Fournir le protocole d'un service à ses partenaires et clients est bien sûr l'application la plus directe de ce problème de découverte, mais il possède un intérêt bien plus grand pour l'ingénierie des services Web. Par exemple, un concepteur pourrait connaître le protocole "effectif" (réellement utilisé) d'un service, et savoir s'il correspond bien aux contraintes de conception ; il serait possible de faire évoluer un service plus facilement : l'utilisation d'un modèle visuel de son comportement (plutôt que la simple analyse de code) permettrait de faciliter l'ajout de nouvelles fonctionnalités, de nouvelles contraintes ou règles, etc.

L'extraction du protocole de conversation d'un service englobe de nombreux défis techniques. Le premier réside dans la façon de modéliser le protocole découvert : il est important de prendre en compte l'incertitude du résultat, et de proposer des indices de confiance et des critères de qualité, permettant d'évaluer sa pertinence. Cette incertitude provient principalement du fait que les logs d'exécution d'un service contiennent souvent des erreurs d'enregistrement (du "bruit"). Des outils sont donc nécessaires, pour analyser et nettoyer les données avant de les traiter. Il est également important de proposer à l'utilisateur des outils lui permettant de modifier et corriger le protocole découvert. Un autre point délicat est la corrélation des messages, à savoir l'identification et la séparation, dans les logs, des différentes conversations (qui peuvent se chevaucher). Il est aussi intéressant d'essayer de découvrir les conditions qui peuvent éventuellement être associées aux transitions dans le protocole.

Ce problème constitue en fait un cas particulier d'une problématique beaucoup plus large : la découverte d'un modèle à partir d'instances de celui-ci. De nombreux travaux, concernant des domaines applicatifs variés, sont liés à cette thématique ; on peut citer par exemple l'inférence grammaticale [1][12], la fouille de workflow [6][13], ou la fouille d'interactions de services Web [7]. En inférence grammaticale, le but consiste à trouver, à partir d'un ensemble de mots appartenant

ou pas à un langage donné, une grammaire permettant de générer ce langage. Dans le cadre de la fouille de workflow (également appelée découverte de processus), le problème réside dans la construction, à partir des logs d'exécution d'un processus, d'un modèle formel (automate fini, graphe orienté ou réseau de Pétri) permettant de représenter le fonctionnement de ce processus. En ce qui concerne la fouille des interactions de services Web, l'objectif est de découvrir, à partir des logs d'un ensemble de services, un workflow modélisant les interactions possibles entre ces services.

Dans le contexte applicatif des services Web [9, 10], il a été proposé une méthode d'extraction de protocoles à partir des archives de conversation entre services (fichiers "logs"). Ce travail porte sur plusieurs des enjeux mentionnés plus haut. Cependant, il ne considère pas les aspects temporels du protocole de conversation.

Contribution. Notre travail se place également dans le contexte de cette problématique. Nous traitons un sous-problème important qui est la découverte des transitions temporisées du protocole de conversation, i.e. des changements d'état liés non pas à l'émission d'un message mais à l'existence d'une contrainte de temps. Bien qu'une telle transition ne figure pas de façon explicite dans les logs du service, il est possible d'identifier les conséquences de son existence. Ce sont donc ces "traces" que nous formalisons et que nous extrayons des données. Pour cela, nous introduisons la notion d'*expiration propre*, qui représente dans les logs ce que la notion de transition temporisée représente dans le protocole de conversation.

Organisation de l'article. Afin de clarifier le contexte de ce travail, nous explicitons rapidement dans la suite ce qu'est le protocole de conversation temporisé d'un service Web, ainsi que ce que contiennent des logs de conversation. Puis, nous définissons de façon plus précise le problème, et présentons nos hypothèses de travail. Nous décrivons ensuite les différents concepts que nous avons définis pour résoudre notre tâche de découverte. Enfin, nous donnons les conclusions et perspectives de notre travail.

2 Préliminaires

Avant d'expliciter notre travail, nous allons tout d'abord présenter les notions sur lesquelles il s'appuie, à savoir le protocole de conversation et les logs d'exécution d'un service Web.

2.1 Protocole de conversation temporisé

Le langage WSDL étant une spécification trop "bas-niveau" du fonctionnement d'un service Web, il a été proposé dans [4] une nouvelle modélisation pour la sémantique d'exécution d'un service : il s'agit du protocole de conversation. Ce protocole est représenté par un automate à états fini déterministe, où les états représentent les différentes phases dans lesquelles le service peut se trouver lors de son exécution. Les transitions sont déclenchées lorsque le service émet ou reçoit des messages ; chacune est donc étiquetée, soit par un message entrant, soit par un message sortant. Un message correspond à l'invocation d'une opération du service, ou à l'envoi de son résultat. Une conversation est une séquence de messages échangés entre un service et un client donnés au cours d'une session.

L'objectif du protocole de conversation d'un service Web est principalement de spécifier l'ensemble des conversations supportées par ce service (i.e. l'ensemble des séquences d'échanges de messages valides). Il est à noter qu'un service peut être engagé simultanément dans plusieurs conversations avec des clients différents, et peut donc être caractérisé par plusieurs instanciations concurrentes du protocole de conversation. Le choix du déterminisme provient de l'observation suivante : autoriser la présence de plusieurs états cibles, atteignables à partir d'un état donné par une même transition, rendrait le protocole ambigu, dans le sens où le service pourrait passer dans un état sans que le client puisse savoir lequel.

Une fois les bases du modèle posées, il a été mis en avant, dans [2, 3], que les aspects temporels devaient être pris en compte dans la description du protocole de conversation. En effet, bien que la plupart des changements d'état soient dus à des invocations d'opérations (ou à des réponses) explicites, ce n'est pas toujours le cas. Ces transitions se produisant sans envoi d'un message explicite sont appelées des transitions implicites. La plupart sont liées à des contraintes temporelles (durée de validité, échéance à respecter, ...). Par exemple, un service peut permettre à ses clients de réserver une ressource ou d'effectuer une action dans un certain intervalle de temps, après lequel ces opérations ne sont plus permises. Afin de modéliser ce type de comportement, la notion de transition temporisée a été introduite ; il s'agit d'une transition qui se produit de façon automatique, après qu'un certain laps de temps se soit écoulé à partir du moment où la transition a été permise (i.e. le moment où l'état source de la transition est devenu l'état courant), ou bien après qu'une certaine date ou une certaine heure soit atteinte.

Il est important de noter que, puisque le protocole de conversation est modélisé grâce à un automate déterministe, un état ne peut admettre plusieurs transitions implicites comme transitions sortantes.

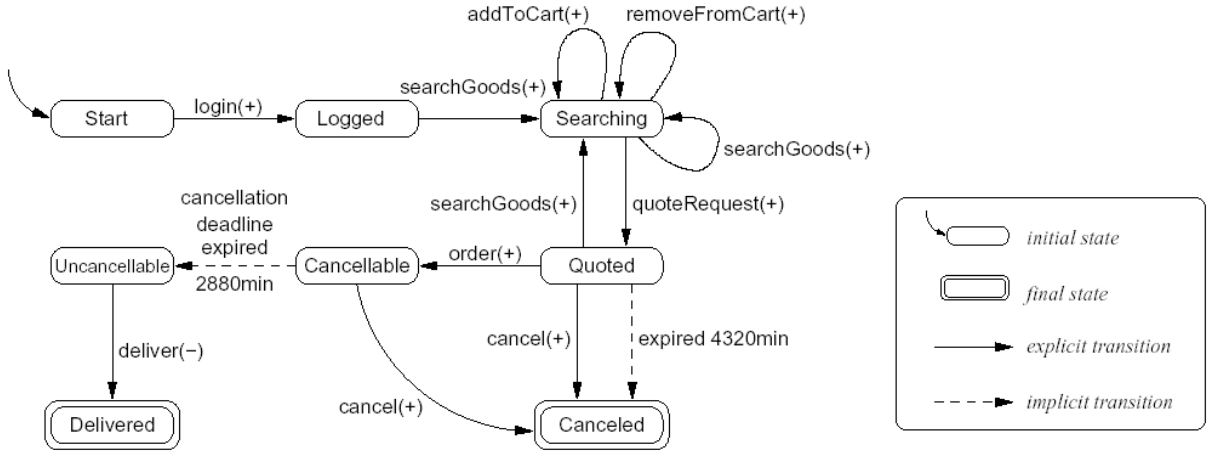


FIG. 1 – Exemple de protocole de conversation temporisé [2, 3].

Exemple 1 La figure 1 représente un protocole de conversation temporisé décrivant le comportement externe d'un service de commande de marchandises. Chaque transition explicite (en trait plein) est étiquetée par l'intitulé d'un message, suivi de sa polarité, qui précise s'il s'agit d'un message entrant (signe +) ou sortant (signe −). Chaque transition implicite (en pointillés) est étiquetée par la contrainte de temps qui lui est associée. Ce protocole spécifie que le client du service doit d'abord se connecter (opération *login*), puis chercher des produits (opération *searchGoods*). Ensuite, celui-ci peut ajouter ou enlever des produits de son caddie (opérations *addToCart* et *removeFromCart*), chercher d'autres marchandises (opération *searchGoods*), ou encore demander un devis (opération *quoteRequest*) qui sera valide seulement pendant 3 jours (i.e. 4320 minutes), laps de temps pendant lequel il peut commander les marchandises (opération *order*). S'il ne le fait pas, au bout des 3 jours la conversation se termine (par le biais de la transition implicite sortant de l'état *Quoted*), et la commande est annulée.

2.2 Logs de conversation

Les différentes façons de collecter les logs d'interaction d'un service ont été décrites dans [7]. En fonction de la façon dont les services sont implémentés et du type d'outils utilisés pour gérer leur exécution, différents types d'informations peuvent être présents dans les logs. Dans un scénario réaliste, les informations collectées sont en général, en plus du contenu du message, l'émetteur, le receveur et la date.

Notons que ces informations peuvent ne pas suffire pour identifier une conversation de façon unique. Ceci ne pose pas de problème dans le cas où un identifiant de chaque conversation est enregistré dans les logs, ce qui n'est malheureusement pas toujours le cas. Il est bien mis en avant dans [9] que le fait de fournir automatiquement un identifiant pour chaque conversation (s'il n'est pas présent par défaut) est un véritable problème en soi. Aussi, ces derniers ont supposé, pour mener à bien leur tâche de découverte du protocole de conversation, que cette information était présente dans les logs. Nous en ferons de même.

Les logs que nous allons traiter sont les enregistrements des messages émis ou envoyés par un service. Ces enregistrements sont effectués par le serveur hébergeant le service. Nous ne considérons pas les logs "internes" du service, qui peuvent être ajoutés au code par le concepteur. Les informations que nous retiendrons sont les intitulés des messages, ainsi que la date à laquelle chaque message a été reçu ou envoyé ; en effet, la connaissance de l'émetteur ou du receveur ne nous sera d'aucune utilité. Précisons également que nous ne tiendrons pas compte de la polarité des messages.

Exemple 2 Pour un service vérifiant le protocole de conversation représenté par la figure 1, on peut par exemple obtenir les conversations suivantes :
(login, 8:18) (searchGoods, 8:20) (addToCart, 8:21) (quoteRequest, 8:22) (cancel, 8:51) ;
(login, 9:03) (searchGoods, 9:04) (addToCart, 9:08) (searchGoods, 9:09) (quoteRequest, 9:15).

3 Spécification du problème

Dans la suite de cet article nous utiliserons les notations suivantes :

Notations : L'ensemble des intitulés des messages appartenant au protocole de conversation sera noté Msg. On notera L les logs de conversation dont on dispose. Formellement, L sera un multi-ensemble de conversations. Une conversation sera notée : $C = \langle M_1, M_2, ..., M_{n_C} \rangle$, où $\forall\, 1 \leq i \leq n_C$ (avec $n_C \in \mathbb{N}^*$), $M_i = (m_i, t_i)$, avec $m_i \in Msg$ (intitulé du message) et $t_i \in \mathbb{R}_+$ (instant où est enregistré le message), et $t_1 < t_2 < ... < t_{n_C}$; elle représentera une séquence d'occurrences de messages. On notera de plus : $\forall\, 1 \leq i \leq n_C$, $m_i = M_i.nom$ et $t_i = M_i.date$.

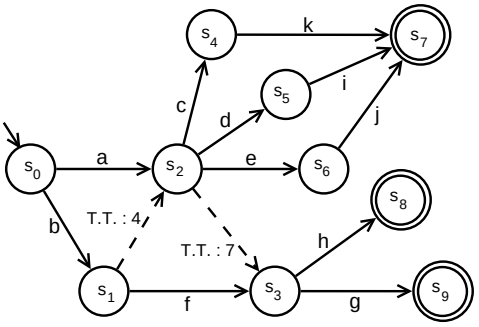
3.1 Tâche de découverte

Si l'on suppose que l'on ne connaît pas le protocole de conversation qui a permis de générer les logs L, et que l'on ne dispose que de ces logs, le problème consiste à exhiber le fait que ces données traduisent la présence de transitions temporisées dans le protocole. Notre méthode vise à examiner les séquences de messages dans les logs, afin de déterminer si, entre deux transitions explicites, a été déclenchée une transition implicite ou pas. Pour ce faire, nous allons calculer, pour chaque conversation, le laps de temps écoulé entre l'émission de deux messages consécutifs.

Exemple 3 Considérons le protocole P_1 représenté dans la figure 2, pour lequel nous avons proposé un exemple de logs possibles : L_1. Nous devons par exemple mettre en évidence le fait, qu'après l'émission du message a, les messages c, d et e peuvent être émis uniquement avant un certain laps de temps, et que les messages g et h peuvent être émis uniquement après ce laps de temps. Il est important de noter que les dates d'enregistrement des messages sont les dates relatives au début de chaque conversation.

3.2 Hypothèses de travail

Nous supposerons dans la suite que les protocoles associés aux logs que nous traiterons, comme dans l'exemple précédent, ne possèdent pas de transition temporisée menant dans un état



$C_1 = \langle (a,0),(c,1),(k,4) \rangle$
$C_2 = \langle (a,0),(c,3),(k,5) \rangle$
$C_3 = \langle (a,0),(d,2),(i,6) \rangle$
$C_4 = \langle (a,0),(d,5),(i,8) \rangle$
$C_5 = \langle (a,0),(e,4),(j,5) \rangle$
$C_6 = \langle (a,0),(e,6),(j,8) \rangle$
$C_7 = \langle (a,0),(h,8) \rangle$
$C_8 = \langle (a,0),(h,10) \rangle$
$C_9 = \langle (a,0),(g,15) \rangle$
$C_{10} = \langle (a,0),(g,16) \rangle$
$C_{11} = \langle (b,0),(f,1),(h,3) \rangle$
$C_{12} = \langle (b,0),(f,2),(g,4) \rangle$
$C_{13} = \langle (b,0),(f,3),(h,4) \rangle$
$C_{14} = \langle (b,0),(f,1),(g,5) \rangle$
$C_{15} = \langle (b,0),(c,6),(k,8) \rangle$
$C_{16} = \langle (b,0),(c,8),(k,9) \rangle$
$C_{17} = \langle (b,0),(d,7),(i,8) \rangle$
$C_{18} = \langle (b,0),(d,10),(i,11) \rangle$
$C_{19} = \langle (b,0),(e,8),(j,10) \rangle$
$C_{20} = \langle (b,0),(e,9),(j,10) \rangle$
$C_{21} = \langle (b,0),(h,13) \rangle$
$C_{22} = \langle (b,0),(h,15) \rangle$
$C_{23} = \langle (b,0),(g,14) \rangle$
$C_{24} = \langle (b,0),(g,15) \rangle$

FIG. 2 – Protocole P_1 et logs associés L_1.

final, bien que cela puisse se produire dans les cas réels. Le fait de se ramener à ce cas particulier - qui est déjà relativement complexe en soi - nous aidera par la suite à mieux appréhender le cas général. Nous supposerons également que les transitions du protocole de conversation sont étiquetées de façon unique, même si certaines correspondent à un même message. Si nous ne sommes pas dans un tel cas de figure, nous pourrons envisager de nous y ramener en effectuant un pré-traitement sur les données.

En ce qui concerne les logs, nous allons supposer qu'ils ne sont pas bruités, c'est-à-dire que les messages sont correctement enregistrés, et dans une séquence correcte. Ceci nous permettra, dans un premier temps, de proposer une méthode "complète". Une approche probabiliste pourra ensuite être envisagée pour pallier le fait que les logs puissent être bruités dans les cas réels. Nous supposerons également que les logs sont suffisamment "complets" pour retrouver les transitions temporisées, c'est-à-dire que tous les "chemins" du protocole de conversation ont été parcourus. Cette hypothèse concerne également tout procédé de découverte du protocole complet : il est impossible d'extraire des logs ce qui n'y figure pas.

4 Formalisation du problème

4.1 Episode

Afin de formaliser le fait que deux messages sont consécutifs dans le temps, nous introduisons la notion d'épisode, qui est directement inspirée de celle définie dans [8], ainsi que la notion d'occurrence d'un épisode, explicitées formellement par les définitions suivantes :

Définition 1 (Episode) Un épisode constitue une séquence de deux intitulés de messages : $\alpha = \langle m, m' \rangle$, avec $m, m' \in Msg$.

Définition 2 (Occurrence d'un épisode) Une occurrence de l'épisode $\alpha = \langle m, m' \rangle$ dans L est une séquence $\langle M, M' \rangle$ telle que : $\exists\, C \in L$ vérifiant

$$\begin{cases} M, M' \in C \\ M.nom = m \text{ et } M'.nom = m' \\ M.date < M'.date \\ \forall\, M'' \in C,\ M''.date < M.date \text{ ou } M''.date > M'.date \end{cases}$$

Si une telle séquence existe, on dira que l'épisode α se produit dans la conversation C.

Notation : On notera $Occ(\alpha)$ l'ensemble des occurrences de l'épisode α dans L.

Remarque : On dira que l'épisode α se produit dans les logs L si α se produit dans au moins une conversation C de L, i.e. si $Occ(\alpha) \neq \phi$.

Notation : On notera Ep l'ensemble des épisodes se produisant dans les logs L.

Proposition 1 *Soit, $\forall\, m \in Msg$, l'ensemble $P_m = \{\alpha \in Ep \mid \exists\, m' \in Msg,\, \alpha = \langle m, m' \rangle\}$. Alors, $\{P_m \mid m \in Msg\}$ est une partition de Ep.*

Cette proposition exprime le fait que l'on peut construire une partition de l'ensemble des épisodes, où chaque partie est constituée de l'ensemble des épisodes dont le premier élément est un message m donné. Nous ne donnons pas de preuve pour ce résultat, qui est relativement trivial. Cette propriété va nous permettre de décomposer notre tâche de découverte. En effet, au lieu d'examiner les épisodes dans leur ensemble, nous allons traiter chaque élément de cette partition séparément, ce qui est légitime car ces parties sont totalement indépendantes les unes des autres. Pour ce faire, nous allons définir la notion de durée d'occurrence d'un épisode et d'un ensemble d'épisodes.

Intuitivement, la durée d'une occurrence d'un épisode est la différence des dates des messages de l'épisode dans l'occurrence. A partir de ceci, on définit la durée d'occurrence minimale (respect. maximale) d'un épisode comme étant la plus petite (respect. la plus grande) durée de toutes les occurrences de cet épisode. L'intervalle de durée d'occurrence d'un épisode est l'intervalle qui englobe l'ensemble des durées d'occurrence de cet épisode.

Définition 3 (Durée d'occurrence d'un épisode)
- La durée de l'occurrence $\langle M, M' \rangle$ de l'épisode α est le réel : $M'.date - M.date$.
- La durée d'occurrence minimale de α dans L est donnée par la fonction $d_{min} : Ep \longrightarrow \mathbb{R}_+$ telle que : $\forall\, \alpha \in Ep$, $d_{min}(\alpha) = min\{M'.date - M.date \mid \langle M, M' \rangle \in Occ(\alpha)\}$.
- La durée d'occurrence maximale de α dans L est donnée par la fonction $d_{max} : Ep \longrightarrow \mathbb{R}_+$ telle que : $\forall\, \alpha \in Ep$, $d_{max}(\alpha) = max\{M'.date - M.date \mid \langle M, M' \rangle \in Occ(\alpha)\}$.
- On appelle intervalle de durée d'occurrence de α dans L, l'intervalle $[d_{min}(\alpha); d_{max}(\alpha)]$.

De la même façon, on définit la durée d'occurrence minimale (respect. maximale) d'un ensemble d'épisodes comme étant le minimum (respect. maximum) des durées d'occurrences minimales (respect. maximales) des épisodes appartenant à cet ensemble. L'intervalle de durée d'occurrence d'un ensemble d'épisodes est l'intervalle qui englobe l'ensemble des durées d'occurrence de tous les épisodes de cet ensemble.

Définition 4 (Durée d'occurrence d'un ensemble d'épisodes)
- La durée d'occurrence minimale d'un ensemble d'épisodes dans L est donnée par la fonction $D_{min} : \mathcal{P}(Ep) \setminus \{\phi\} \longrightarrow \mathbb{R}_+$ telle que : $\forall\, A \subseteq Ep$, $D_{min}(A) = min\{d_{min}(\alpha) \mid \alpha \in A\}$.
- La durée d'occurrence maximale d'un ensemble d'épisodes dans L est donnée par la fonction $D_{max} : \mathcal{P}(Ep) \setminus \{\phi\} \longrightarrow \mathbb{R}_+$ telle que : $\forall\, A \subseteq Ep$, $D_{max}(A) = max\{d_{max}(\alpha) \mid \alpha \in A\}$.
- On appelle intervalle de durée d'occurrence d'un ensemble d'épisodes A dans L, l'intervalle $[D_{min}(A); D_{max}(A)]$.

Exemple 4 Considérons les logs L_1 (cf. figure 2). Les intervalles de durée d'occurrence de $\{\langle a, c \rangle, \langle a, d \rangle\}$ et $\{\langle a, h \rangle\}$ sont $[1; 5]$ et $[8; 10]$. Ils sont disjoints et vérifient une relation de précédence sur l'échelle du temps. Bien évidemment, ceci est dû au fait qu'il existe une transition temporisée, entre les états s_2 et s_3, déclenchée automatiquement au temps 7 (après l'arrivée dans l'état s_2) si aucun des messages c, d ou e n'est émis avant.

Du fait qu'elle soit la conséquence de la présence d'une transition temporisée, la relation de précédence présentée dans cet exemple nous sera utile dans la suite. En effet, si l'on inverse le raisonnement, trouver qu'une telle relation est vérifiée par les données pourrait nous conduire à la découverte d'une transition temporisée. Nous formalisons donc cette relation.

4.2 Relation d'ordre sur les ensembles d'épisodes

Définition 5 (relation $\prec$) Soient $A, B \subset Ep$ $(A, B \neq \phi)$.
– On dira que A est **avant** B, ce que l'on notera $A \prec B$, si : $D_{max}(A) < D_{min}(B)$.
– On dira que A et B sont incomparables, ce que l'on notera $A \parallel B$, si : $A \not\prec B$ et $B \not\prec A$.

Intuitivement, l'expression $A \prec B$ traduit le fait que l'intervalle de durée d'occurrence de A est disjoint de celui de B et qu'il le précède sur l'échelle du temps. On dira également que B est après A.

Propriété 1 *La relation $\prec$ est une relation d'ordre strict sur $\mathcal{P}(Ep) \setminus \{\phi\}$.*

Preuve
– *Irréflexivité* :
Soit $A \subseteq Ep(A \neq \phi)$.
$D_{min}(A) \leq D_{max}(A) \Rightarrow D_{max}(A) \not< D_{min}(A) \Rightarrow A \not\prec A$.
Donc, $\forall A \subseteq Ep(A \neq \phi), A \not\prec A$.
– *Asymétrie (ou antisymétrie forte)* :
Soient $A, B \subset Ep$ tels que $A \prec B$.
Puisque $A \prec B$, on a : $D_{max}(A) < D_{min}(B)$.
Comme $D_{min}(A) \leq D_{max}(A)$ et $D_{min}(B) \leq D_{max}(B)$, on a : $D_{min}(A) < D_{max}(B)$.
Ainsi, $D_{max}(B) \not< D_{min}(A)$, i.e. $B \not\prec A$.
Donc, $\forall A, B \subset Ep, A \prec B \Rightarrow B \not\prec A$.
– *Transitivité* :
Soient $A, B, C \subset Ep$ tels que : $A \prec B$ et $B \prec C$.
On a : $A \prec B \Rightarrow D_{max}(A) < D_{min}(B)$ et $B \prec C \Rightarrow D_{max}(B) < D_{min}(C)$.
Puisque $D_{min}(B) \leq D_{max}(B)$, on a : $D_{max}(A) < D_{min}(C)$, i.e. $A \prec C$.
Donc, $\forall A, B, C \subset Ep$, on a : $(A \prec B$ et $B \prec C) \Rightarrow A \prec C$.

□

Exemple 5 Considérons les logs L_1 (présentés dans la figure 2). On obtient :
$\{\langle a, c\rangle\} \prec \{\langle a, g\rangle\}$, car $D_{max}(\{\langle a, c\rangle\}) = 3 < 15 = D_{min}(\{\langle a, g\rangle\})$;
$\{\langle a, c\rangle, \langle a, d\rangle\} \prec \{\langle a, g\rangle\}$, car $D_{max}(\{\langle a, c\rangle, \langle a, d\rangle\}) = 5 < 15 = D_{min}(\{\langle a, g\rangle\})$;
$\{\langle a, c\rangle, \langle a, d\rangle\} \prec \{\langle a, g\rangle, \langle a, h\rangle\}$, car $D_{max}(\{\langle a, c\rangle, \langle a, d\rangle\}) < 8 = D_{min}(\{\langle a, g\rangle, \langle a, h\rangle\})$;
$\{\langle a, c\rangle, \langle a, d\rangle, \langle a, e\rangle\} \prec \{\langle a, g\rangle, \langle a, h\rangle\}$; etc.

Proposition 2 *Soient $m \in Msg$, et $A, B \subset P_m$ $(A, B \neq \phi)$. S'il existe une transition temporisée, dans le protocole de conversation, entre l'état d'où sortent les transitions correspondant aux éléments de A et celui d'où sortent les transitions correspondant aux éléments de B, alors $A \prec B$.*

Preuve Soient $m \in Msg$, et $A, B \subset P_m$ $(A, B \neq \phi)$ tels que $A = \{\langle m, m'_1\rangle, \ldots, \langle m, m'_p\rangle\}$ et $B = \{\langle m, m''_1\rangle, \langle m, m''_2\rangle, \ldots, \langle m, m''_q\rangle\}$. Puisque les transitions du protocole de conversation sont étiquetées de façon unique, on a : $A \cap B = \phi$.
Supposons qu'il existe une transition temporisée, dans le protocole de conversation, entre l'état (noté s_1) d'où sortent les transitions étiquetées par m'_1, m'_2, …, m'_p, et celui (noté s_2) d'où sortent les transitions étiquetées par m''_1, m''_2, …, m''_q. A cette transition est associé un certain laps de temps t. Au cours d'une conversation, une fois l'état s_1 atteint, si aucun des messages m'_1, m'_2, …, m'_p, n'est émis avant le temps t, le service passe automatiquement dans l'état s_2. Autrement dit, d'un point de vue global à toutes les exécutions, on observe que : une fois l'état s_1 atteint, les messages m'_1, m'_2, …, m'_p, peuvent être émis uniquement avant ce temps t, tandis que les messages m''_1, m''_2, …, m''_q, peuvent être émis uniquement après ce temps t.

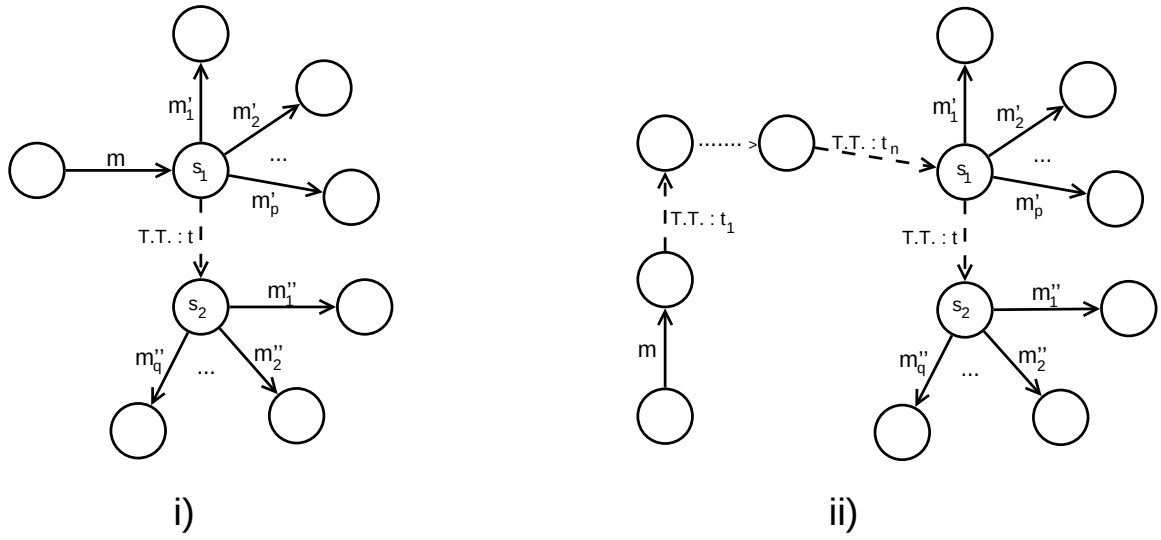


FIG. 3 – Différentes configurations faisant intervenir une transition temporisée de délai t.

– Si la transition étiquetée par m arrive sur l'état s_1 (cf. figure 3 i)) :
On a : $\forall\ 1 \leq i \leq p$, $d_{max}(\langle m, m'_i\rangle) < t$, et $\forall\ 1 \leq j \leq q$, $d_{min}(\langle m, m''_j\rangle) > t$.
Comme $D_{max}(A) = max_{1\leq i\leq p}\{d_{max}(\langle m, m'_i\rangle)\}$ et $D_{min}(B) = min_{1\leq j\leq q}\{d_{min}(\langle m, m''_j\rangle)\}$, on a : $D_{max}(A) < t < D_{min}(B)$.
Donc, $A \prec B$.
– Sinon (cf. figure 3 ii)) :
L'état où arrive la transition étiquetée par m est relié à s_1 par une succession de n (avec $n \geq 1$) transitions temporisées auxquelles sont associées les durées t_1, t_2, …, t_n.
On a alors : $D_{max}(A) < t + T < D_{min}(B)$, où $T = t_1 + t_2 + \cdots + t_n$.
Donc, $A \prec B$.

Remarque : On peut montrer également que : s'il existe une chaîne de n transitions temporisées (avec $n \geq 1$), dans le protocole de conversation, entre l'état d'où sortent les transitions correspondant aux éléments de A et celui d'où sortent les transitions correspondant aux éléments de B, alors $A \prec B$. Dans ce cas, si t_1, t_2, …, t_n, sont les laps de temps associés aux différentes transitions temporisées, la preuve est identique en posant $t = t_1 + t_2 + \cdots + t_n$.

□

Cette proposition nous donne une condition nécessaire à l'existence d'une transition temporisée. Notre problème de découverte serait résolu si cette condition était également suffisante : nous disposerions alors d'un objet équivalent à une transition temporisée. Ce n'est malheureusement pas le cas.

Remarque : La réciproque de la proposition 2 est fausse.

Contre-exemple On a $\{\langle a, c\rangle\} \prec \{\langle a, e\rangle\}$ (car $D_{max}(\{\langle a, c\rangle\}) = 3 < 4 = D_{min}(\{\langle a, e\rangle\})$), dans les logs L_1, alors que les transitions étiquetées par c et e sortent du même état (cf. fig. 2).

La proposition 2 et l'hypothèse de complétude des logs nous assurent que l'ensemble des expressions de la forme $A \prec B$ vérifiées par les logs englobe l'ensemble des transitions temporisées. Toutefois, ces expressions peuvent aussi nous donner, en plus, de fausses informations, sur des transitions inexistantes. Ceci vient du fait que la relation $\prec$ ne prend pas en compte l'ensemble des informations présentes dans les logs, induites par la présence d'une transition temporisée. C'est pourquoi nous définissons dans la suite une relation plus riche sur les ensembles d'épisodes.

4.3 Expiration

Définition 6 (Expiration) Soient $m \in Msg$, et $A, B \subset P_m$ $(A, B \neq \phi)$. On dira que les logs L satisfont l'expiration $E(m, A, B)$, ce que l'on notera $L \vDash E(m, A, B)$, si :

$$\begin{cases} A \prec B \\ \forall\, \alpha \in P_m,\ \{\alpha\} \nparallel A \text{ ou } \{\alpha\} \nparallel B \end{cases}$$

Si $A = \{\langle m, m'_1\rangle, \langle m, m'_2\rangle, \ldots, \langle m, m'_p\rangle\}$, si $B = \{\langle m, m''_1\rangle, \ldots, \langle m, m''_q\rangle\}$, et si $L \vDash E(m, A, B)$, par abus de notation, on écrira : $L \vDash E(m, \{m'_1, \ldots, m'_p\}, \{m''_1, \ldots, m''_q\})$.

Intuitivement, l'assertion $L \vDash E(m, A, B)$ traduit le fait que, d'après les logs L, les durées d'occurrence de tous les épisodes de A sont strictement inférieures aux durées d'occurrence de tous les épisodes de B, et qu'il n'existe pas d'épisode dont certaines occurrences auraient une durée appartenant à l'intervalle de durée d'occurrence de A et dont d'autres occurrences auraient une durée appartenant à l'intervalle de durée d'occurrence de B. A cette expiration peut être associé un délai, supérieur à la durée d'occurrence maximale de l'ensemble des épisodes de A (i.e. $D_{max}(A)$), et inférieur à la durée d'occurrence minimale de l'ensemble des épisodes de B (i.e. $D_{min}(B)$). Il existe donc une infinité de valeurs possibles pour un tel délai d'expiration : ce sont tous les réels de l'intervalle $]D_{max}(A); D_{min}(B)[$, que l'on appellera intervalle d'expiration de $E(m, A, B)$.

Exemple 6 Les logs L_1 (cf. fig. 2) satisfont les expirations, $E(a, \{c, d, e\}, \{g\})$, $E(a, \{c, d\}, \{g\})$, $E(b, \{f\}, \{c, d, e\})$, $E(b, \{c, d, e\}, \{g, h\})$, $E(b, \{f\}, \{g, h\})$, $E(b, \{f\}, \{c, d, e, g, h\})$, etc.
Par contre, $L_1 \nvDash E(a, \{c\}, \{e\})$, car $\{\langle a, d\rangle\} \parallel \{\langle a, c\rangle\}$ et $\{\langle a, d\rangle\} \parallel \{\langle a, e\rangle\}$.

Proposition 3 *Soient $m \in Msg$, et $A, B \subset P_m$ $(A, B \neq \phi)$. S'il existe une transition temporisée, dans le protocole de conversation, entre l'état d'où sortent les transitions correspondant aux éléments de A et celui d'où sortent les transitions correspondant aux éléments de B, alors $L \vDash E(m, A, B)$.*

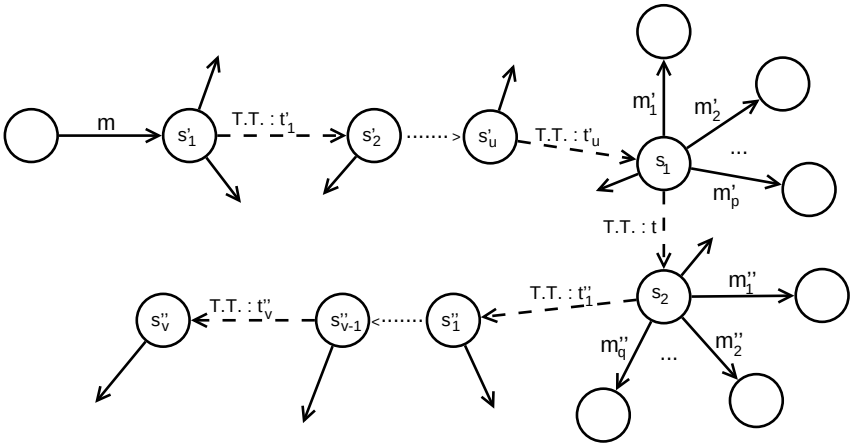


FIG. 4 – Configuration générale regroupant les transitions associées aux éléments de P_m.

Preuve Soient $m \in Msg$, et $A, B \subset P_m$ $(A, B \neq \phi)$ tels que $A = \{\langle m, m'_1\rangle, \ldots, \langle m, m'_p\rangle\}$ et $B = \{\langle m, m''_1\rangle, \langle m, m''_2\rangle, \ldots, \langle m, m''_q\rangle\}$. Supposons qu'il existe une transition temporisée, dans le protocole de conversation, entre l'état d'où sortent les transitions étiquetées par m'_1, m'_2, …, m'_p, et celui d'où sortent les transitions étiquetées par m''_1, m''_2, …, m''_q. La situation est représentée par la figure 4 (où u et v peuvent être nuls). D'après la proposition 2, on sait déjà que $A \prec B$. Soit $\alpha = \langle m, m'\rangle \in P_m$. Puisque $\alpha \in P_m$, m' est associé à une transition qui part, soit de l'état s_1, soit de l'état s_2, soit d'un des états s'_1, s'_2, …, s'_u, soit d'un des états s''_1, s''_2, …, s''_v.

– S'il s'agit de s_1, ou d'un des états s'_1, s'_2, ..., s'_u, on a :
$\forall\ 1 \leq j \leq q$, $d_{max}(\alpha) < t + \sum_{k=1}^{u} t'_k < d_{min}(\langle m, m''_j \rangle)$.
D'où : $D_{max}(\{\alpha\}) = d_{max}(\alpha) < min\{d_{min}(\langle m, m''_j \rangle) \mid 1 \leq j \leq q\} = D_{min}(B)$.
Ainsi, $\{\alpha\} \prec B$, et donc $\{\alpha\} \nparallel B$.
– S'il s'agit de s_2, ou d'un des états s''_1, s''_2, ..., s''_v, on a :
$\forall\ 1 \leq i \leq p$, $d_{min}(\alpha) > t + \sum_{k=1}^{u} t'_k > d_{max}(\langle m, m'_i \rangle)$.
D'où : $D_{min}(\{\alpha\}) = d_{min}(\alpha) > max\{d_{max}(\langle m, m'_i \rangle) \mid 1 \leq i \leq p\} = D_{max}(A)$.
Ainsi, $A \prec \{\alpha\}$, et donc $\{\alpha\} \nparallel A$.
– Conclusion : $\forall\, \alpha \in P_m$, $\{\alpha\} \nparallel A$ ou $\{\alpha\} \nparallel B$.
Puisque $A \prec B$, on a bien : $L \vDash E(m, A, B)$.

□

Remarque : La réciproque de la proposition 3 est fausse.

Contre-exemple L'expiration $E(b, \{f\}, \{g, h\})$ est satisfaite par les logs L_1, bien qu'il n'y ait pas de transition temporisée entre les états s_1 et s_3 du protocole P_1 (cf. figure 2). Par contre, il existe une chaîne formée de deux transitions temporisées, reliant ces deux états.

La proposition 3 et l'hypothèse de complétude des logs nous assurent que chaque transition temporisée du protocole de conversation peut être retrouvée par l'intermédiaire d'une certaine expiration satisfaite par les logs. Cependant, une expiration est satisfaite entre deux ensembles d'épisodes, aussi bien en présence d'une transition temporisée entre les états correspondant à ces ensembles d'épisodes, que d'une chaîne de transitions temporisées. Nous allons donc définir une classe d'expirations plus restreinte, afin d'éviter cette ambiguïté. Un autre problème lié aux expirations est qu'elles sont beaucoup plus nombreuses que les transitions temporisées.

Exemple 7 Dans le cas du protocole P_1 (cf. figure 2), la transition temporisée présente entre les états s_1 et s_2 entraîne la satisfaction, par les logs L_1, de l'expiration $E(b, \{f\}, \{c, d, e\})$, mais aussi de $E(b, \{f\}, \{c, d, e, g, h\})$; celle qui relie les états s_2 et s_3 entraîne la satisfaction de l'expiration $E(a, \{c, d, e\}, \{g\})$, mais aussi de $E(a, \{c, d\}, \{g\})$.

Cet exemple illustre le fait que plusieurs formes de "redondance" apparaissent. Or, nous voulons apporter le minimum d'informations nécessaires à l'utilisateur pour retrouver les transitions temporisées. C'est donc dans ce sens que nous définissons les *expirations propres*.

4.4 Expiration propre

Définition 7 (Expiration propre) Soient $m \in Msg$, et $A, B \subset P_m$ $(A, B \neq \phi)$. On dira que les logs L satisfont l'expiration propre $EP(m, A, B)$, ce que l'on notera $L \vDash EP(m, A, B)$, si :

$$\begin{cases} A \prec B \\ \forall\, \alpha \in P_m \setminus (A \cup B),\ \{\alpha\} \nparallel A \cup B \\ \forall\, X, Y \subset A\ (X, Y \neq \phi) \text{ tels que } X \cup Y = A,\ \text{on a} : X \not\prec Y \\ \forall\, X, Y \subset B\ (X, Y \neq \phi) \text{ tels que } X \cup Y = B,\ \text{on a} : X \not\prec Y \end{cases}$$

Si $A = \{\langle m, m'_1 \rangle, \langle m, m'_2 \rangle, \ldots, \langle m, m'_p \rangle\}$, si $B = \{\langle m, m''_1 \rangle, \ldots, \langle m, m''_q \rangle\}$, et si $L \vDash EP(m, A, B)$, par abus de notation on écrira : $L \vDash EP(m, \{m'_1, \ldots, m'_p\}, \{m''_1, \ldots, m''_q\})$.

Intuitivement, l'assertion $L \vDash EP(m, A, B)$ traduit le fait que, d'après les logs L, les durées d'occurrence de tous les épisodes de A sont strictement inférieures aux durées d'occurrence de tous les épisodes de B, qu'il n'existe pas d'épisode, en dehors de ceux de A et de B, ayant des occurrences dont la durée appartienne à l'intervalle de durée d'occurrence de $A \cup B$ (intervalle englobant les intervalles de durée d'occurrence de A et de B), et que, ni A, ni B n'est partitionnable en deux sous-ensembles ordonnés par la relation $\prec$. Il est clair qu'une expiration propre

est une expiration. Ce résultat est formalisé par la proposition suivante, dont la réciproque est fausse (cf. exemple 8).

Propriété 2 *Soient $m \in Msg$, et $A, B \subset P_m$ $(A, B \neq \phi)$. Si $L \vDash EP(m, A, B)$, alors $L \vDash E(m, A, B)$.*

Preuve Soient $m \in Msg$, et $A, B \subset P_m$ $(A, B \neq \phi)$.
Supposons que : $L \vDash EP(m, A, B)$. On a alors : $A \prec B$.
Soit $\alpha \in P_m$.
– Si $\alpha \in P_m \setminus (A \cup B)$, par hypothèse, on a : $\{\alpha\} \nparallel A \cup B$, i.e. $\alpha \prec A \cup B$ ou $A \cup B \prec \alpha$.
D'où : $d_{max}(\alpha) < D_{min}(A \cup B)$ ou $d_{min}(\alpha) > D_{max}(A \cup B)$.
Ainsi, $d_{max}(\alpha) < D_{min}(A)$ ou $d_{min}(\alpha) > D_{max}(A)$,
car $A \subset A \cup B \Rightarrow [D_{min}(A \cup B) \leq D_{min}(A)$ et $D_{max}(A \cup B) \geq D_{max}(A)]$.
Donc, $\{\alpha\} \prec A$ ou $A \prec \{\alpha\}$.
Par conséquent, $\{\alpha\} \nparallel A$.
– Si $\alpha \in A$, on a : $d_{max}(\alpha) \leq D_{max}(A)$.
Comme $D_{max}(A) < D_{min}(B)$ (car $A \prec B$), on a : $d_{max}(\alpha) < D_{min}(B)$.
Ainsi, $\{\alpha\} \prec B$, et donc $\{\alpha\} \nparallel B$.
– Si $\alpha \in B$, on a : $d_{min}(\alpha) \geq D_{min}(B)$.
Comme $D_{min}(B) > D_{max}(A)$ (car $A \prec B$), on a : $d_{min}(\alpha) > D_{max}(A)$.
Ainsi, $A \prec \{\alpha\}$, et donc $\{\alpha\} \nparallel A$.
– Conclusion : $\forall\, \alpha \in P_m$, $\{\alpha\} \nparallel A$ ou $\{\alpha\} \nparallel B$
Puisque $A \prec B$, on a donc : $L \vDash E(m, A, B)$.

□

expiration propre	intervalle d'expiration
$EP(a, \{c, d, e\}, \{h\})$	$]6; 8[$
$EP(a, \{h\}, \{g\})$	$]10; 15[$
$EP(b, \{f\}, \{c, d, e\})$	$]3; 6[$
$EP(b, \{c, d, e\}, \{g, h\})$	$]10; 13[$

TAB. 1 – Expirations propres satisfaites par les logs L_1.

Exemple 8 Les expirations propres satisfaites par les logs L_1 (cf. figure 2) sont reportées dans le tableau 1 (rappelons que les ensembles P_a et P_b sont traités indépendamment l'un de l'autre). On vérifie également que :
– $L_1 \nvDash EP(b, \{f\}, \{g, h\})$, car $\{\langle b, c\rangle\} \parallel \{\langle b, f\rangle, \langle b, g\rangle, \langle b, h\rangle\}$;
– $L_1 \nvDash EP(b, \{f\}, \{c, d, e, g, h\})$, car $\{\langle b, c\rangle, \langle b, d\rangle, \langle b, e\rangle\} \prec \{\langle b, g\rangle, \langle b, h\rangle\}$;
– $L_1 \nvDash EP(a, \{c, d\}, \{g\})$, car $\{\langle a, e\rangle\} \parallel \{\langle a, c\rangle, \langle a, d\rangle, \langle a, g\rangle\}$.

Proposition 4 *Soient $m \in Msg$, et $A, B \subset P_m$ $(A, B \neq \phi)$. S'il existe une transition temporisée, dans le protocole de conversation, entre deux états s_1 et s_2 tels que les ensembles A et B correspondent respectivement aux ensembles de transitions sortant de s_1 et s_2, alors il existe $A' \subseteq A$ et $B' \subseteq B$ $(A', B' \neq \phi)$ tels que $L \vDash EP(m, A', B')$.*

Preuve Soient $m \in Msg$, et $A, B \subset P_m$ $(A, B \neq \phi)$. Supposons qu'il existe une transition temporisée, dans le protocole de conversation, entre deux états s_1 et s_2 tels que les deux ensembles de transitions sortant de s_1 et de s_2 soient en bijection avec A et B respectivement.
Soit $\{A_1, A_2, \ldots, A_x\}$ la partition (pouvant être réduite à un seul élément) de A telle que :
– $A_1 \prec A_2 \prec \cdots \prec A_x$ (sachant que $\prec$ est une relation d'ordre), et
– $\forall\, 1 \leq k \leq x$, $\forall\, X, Y \subset A_k$ $(X, Y \neq \phi)$ tels que $X \cup Y = A_k$, on ait : $X \not\prec Y$.
De même, soit $\{B_1, B_2, \ldots, B_y\}$ la partition de B telle que :

– $B_1 \prec B_2 \prec \cdots \prec B_y$ (sachant que $\prec$ est une relation d'ordre), et
– $\forall\, 1 \leq k \leq y$, $\forall\, X, Y \subset B_k$ $(X, Y \neq \phi)$ tels que $X \cup Y = B_k$, on ait : $X \not\prec Y$.
Notons m'_1, m'_2, …, m'_p, les éléments de A_x, et m''_1, m''_2, …, m''_q, les éléments de B_1.
La situation est représentée par la figure 4 (où u et v peuvent être nuls).
D'après la proposition 3, on sait déjà que $L \vDash E(m, A_x, B_1)$, et donc que $A_x \prec B_1$.
Soit $\alpha = \langle m, m' \rangle \in P_m \setminus (A_x \cup B_1)$. m' est associé à une transition qui part, soit de l'état s_1, soit de l'état s_2, soit d'un des états s'_1, s'_2, …, s'_u, soit d'un des états s''_1, s''_2, …, s''_v.
– S'il s'agit d'un des états s'_1, s'_2, …, s'_u, on a :
$\forall\, 1 \leq i \leq p$, $d_{max}(\alpha) < \sum_{k=1}^{u} t'_k < d_{min}(\langle m, m'_i \rangle)$; et
$\forall\, 1 \leq j \leq q$, $d_{max}(\alpha) < \sum_{k=1}^{u} t'_k < d_{min}(\langle m, m''_j \rangle)$.
D'où : $d_{max}(\alpha) < min(\{d_{min}(\langle m, m'_i \rangle) \mid 1 \leq i \leq p\} \cup \{d_{min}(\langle m, m''_j \rangle) \mid 1 \leq j \leq q\})$,
c'est-à-dire que $D_{max}(\{\alpha\}) < D_{min}(A_x \cup B_1)$.
Ainsi, $\{\alpha\} \prec A_x \cup B_1$, et donc $\{\alpha\} \nparallel A_x \cup B_1$.
– S'il s'agit d'un des états s''_1, s''_2, …, s''_v, on a :
$\forall\, 1 \leq i \leq p$, $d_{min}(\alpha) > t + t''_1 + \sum_{k=1}^{u} t'_k > d_{max}(\langle m, m'_i \rangle)$; et
$\forall\, 1 \leq j \leq q$, $d_{min}(\alpha) > t + t''_1 + \sum_{k=1}^{u} t'_k > d_{max}(\langle m, m''_j \rangle)$.
D'où : $d_{min}(\alpha) > max(\{d_{max}(\langle m, m'_i \rangle) \mid 1 \leq i \leq p\} \cup \{d_{max}(\langle m, m''_j \rangle) \mid 1 \leq j \leq q\})$,
c'est-à-dire que $D_{min}(\{\alpha\}) > D_{max}(A_x \cup B_1)$.
Ainsi, $A_x \cup B_1 \prec \{\alpha\}$, et donc $\{\alpha\} \nparallel A_x \cup B_1$.
– S'il s'agit de l'état s_1 (ce qui sous-entend que $x \geq 2$) :
Puisque $\alpha \in P_m \setminus A_x$, on a : $\alpha \in A_l$, avec $1 \leq l \leq x - 1$.
D'où : $d_{max}(\alpha) \leq D_{max}(A_l) < D_{min}(A_x)$ (car $A_l \prec A_x$, par transitivité).
De plus, $\forall\, 1 \leq j \leq q$, $d_{max}(\alpha) < t + \sum_{k=1}^{u} t'_k < d_{min}(\langle m, m''_j \rangle)$.
Ainsi, $d_{max}(\alpha) < D_{min}(B_1)$, et donc $D_{max}(\{\alpha\}) = d_{max}(\alpha) < D_{min}(A_x \cup B_1)$.
Par conséquent, $\{\alpha\} \prec A_x \cup B_1$, et donc $\{\alpha\} \nparallel A_x \cup B_1$.
– S'il s'agit de l'état s_2 (ce qui sous-entend que $y \geq 2$) :
Puisque $\alpha \in P_m \setminus B_1$, on a : $\alpha \in B_h$, avec $2 \leq h \leq y$.
D'où : $d_{min}(\alpha) \geq D_{min}(B_h) > D_{max}(B_1)$ (car $B_1 \prec B_h$, par transitivité).
De plus, $\forall\, 1 \leq i \leq p$, $d_{min}(\alpha) > t + \sum_{k=1}^{u} t'_k > d_{max}(\langle m, m'_i \rangle)$.
Ainsi, $d_{min}(\alpha) > D_{max}(A_x)$, et donc $D_{min}(\{\alpha\}) = d_{min}(\alpha) > D_{max}(A_x \cup B_1)$.
Par conséquent, $A_x \cup B_1 \prec \{\alpha\}$, et donc $\{\alpha\} \nparallel A_x \cup B_1$.
– Conclusion : $\forall\, \alpha \in P_m \setminus (A_x \cup B_1)$, $\{\alpha\} \nparallel A_x \cup B_1$.
Par hypothèse, on a également :
– $\forall\, X, Y \subset A_x$ $(X, Y \neq \phi)$ tels que $X \cup Y = A_x$, on a : $X \not\prec Y$;
– $\forall\, X, Y \subset B_1$ $(X, Y \neq \phi)$ tels que $X \cup Y = B_1$, on a : $X \not\prec Y$.
Donc, $L \vDash EP(m, A_x, B_1)$, avec $A_x \subseteq A$ et $B_1 \subseteq B$ $(A_x, B_1 \neq \phi)$.

□

Remarque : La réciproque de la proposition 4 est fausse.

Contre-exemple On a $L_1 \vDash EP(a, \{h\}, \{g\})$, alors que les transitions étiquetées par h et g sortent du même état (cf. figure 2). La satisfaction de cette expiration propre s'explique par le fait que, dans les logs L_1, après que le message a ait été émis, le message g est toujours plus long à émettre que le message h.

D'après la proposition 4 et l'hypothèse de complétude des logs, puisque chaque transition temporisée engendre la satisfaction d'une expiration propre dans les logs, il est possible de toutes les retrouver. Toutefois, on peut découvrir plus d'expirations propres qu'il n'y a de transitions temporisées, dans le cas où certains messages sont toujours plus long à envoyer (ou à recevoir) que tous les messages associés aux autres transitions du même état. Le théorème suivant exprime le fait que ceci constitue le seul cas d'erreur possible.

Théorème 1 *Soient $m \in Msg$, et $A, B \subset P_m$ $(A, B \neq \phi)$. Si $L \vDash EP(m, A, B)$, alors il existe dans le protocole de conversation :*

- *ou bien deux états s_1 et s_2 tels que s_2 soit relié à s_1 par une transition temporisée, A corresponde à un sous-ensemble des transitions sortant de s_1, et B corresponde à un sous-ensemble des transitions sortant de s_2,*
- *ou bien un état s tel que $A \cup B$ corresponde à un sous-ensemble des transitions sortant de s, et les messages de B soient toujours plus longs à émettre que les messages de A.*

Preuve Soient $m \in Msg$, et $A, B \subset P_m$ $(A, B \neq \phi)$. Supposons que $L \vDash EP(m, A, B)$, i.e. :

$$\begin{cases} A \prec B & (1) \\ \forall\, \alpha \in P_m \setminus (A \cup B),\ \{\alpha\} \nparallel A \cup B & (2) \\ \forall\, X, Y \subset A\ (X, Y \neq \phi) \text{ tels que } X \cup Y = A, \text{ on a : } X \not\prec Y & (3) \\ \forall\, X, Y \subset B\ (X, Y \neq \phi) \text{ tels que } X \cup Y = B, \text{ on a : } X \not\prec Y & (4) \end{cases}$$

Notons s_m l'état du protocole de conversation dans lequel entre la transition étiquetée par m. Puisque $A, B \subset P_m$, les éléments de $A \cup B$ correspondent à des transitions pouvant sortir, soit de s_m, soit d'un état relié à s_m par une transition temporisée, ou une chaîne de transitions temporisées.

- Si ces transitions sortent toutes du même état (que ce soit s_m ou un autre) :
  Puisque $A \prec B$, on sait que les messages de B sont toujours plus longs à émettre que les messages de A.

- Si ces transitions sortent de deux états distincts s_1 et s_2 :
  s_1 et s_2 sont reliés par une chaîne de p transitions temporisées, où $p \geq 1$ (si $p = 1$, il s'agit d'une unique transition temporisée). On peut supposer, quite à renommer les états, que s_1 est au "début" de la chaîne, et s_2 à la "fin". Notons E_1 et E_2 les ensembles respectifs d'épisodes de P_m correspondant aux transitions sortant respectivement de s_1 et de s_2.
  On a : $A \cup B \subseteq E_1 \cup E_2$, $(A \cup B) \cap E_1 \neq \phi$, et $(A \cup B) \cap E_2 \neq \phi$.
  Supposons que : $E_1 \cap A = \phi$.
  On a : $A \subseteq E_2$, et donc $E_1 \cap B \neq \phi$.
  Supposons que : $E_2 \cap B \neq \phi$.
  On a : $(E_1 \cap B) \cup (E_2 \cap B) = B$.
  Puisque s_2 est relié à s_1 par une chaîne de transitions temporisées, on a :
  $E_1 \cap B \prec E_2 \cap B$, ce qui contredit (4).
  Ainsi, $E_2 \cap B = \phi$, et donc $B \subseteq E_1$.
  Comme $A \subseteq E_2$, et que s_2 est à la fin de la chaîne de transitions temporisées, on a :
  $B \prec A$, ce qui contredit (1).
  Donc, $E_1 \cap A \neq \phi$.
  Supposons que $E_2 \cap A \neq \phi$.
  On a : $(E_1 \cap A) \cup (E_2 \cap A) = A$.
  Puisque s_2 est relié à s_1 par une chaîne de transitions temporisées, on a : $E_1 \cap A \prec E_2 \cap A$, ce qui contredit (3).
  Donc, $E_2 \cap A = \phi$.
  Ainsi, $A \subseteq E_1$, et donc $E_2 \cap B \neq \phi$.
  Comme on ne peut avoir $E_1 \cap B \neq \phi$, on a : $B \subseteq E_2$.
  Par conséquent, A et B correspondent respectivement à des ensembles de transitions sortant de s_1 et de s_2.

  Supposons que $p \geq 2$ (i.e. la chaîne comporte plusieurs transitions).
  Il existe au moins un état s_3 "entre" s_1 et s_2.
  Notons E_3 l'ensemble des épisodes de P_m correspondant aux transitions sortant de s_3.
  Soit $\alpha \in E_3$.
  On a : $d_{min}(\alpha) < D_{max}(B) = D_{max}(A \cup B)$ (car $A \prec B$), et donc $A \cup B \not\prec \{\alpha\}$.

De plus, $d_{max}(\alpha) > D_{min}(A) = D_{min}(A \cup B)$ (car $A \prec B$), et donc $\{\alpha\} \not\prec A \cup B$.
Donc, $\{\alpha\} \parallel A \cup B$, ce qui contredit (2), car $\alpha \in P_m \setminus (A \cup B)$.
Par conséquent, $p = 1$, i.e. s_2 est relié à s_1 par une unique transition temporisée.

– Si ces transitions sortent de n états s_1, s_2, ..., s_n (avec $n \geq 3$) :
Notons $E_1, E_2, \ldots, E_n$, les ensembles respectifs d'épisodes de P_m correspondant aux transitions sortant respectivement de s_1, s_2, ..., s_n.
On a : $A \cup B \subseteq \bigcup_{i=1}^{n} E_i$, et $\forall\ 1 \leq i \leq n$, $(A \cup B) \cap E_i \neq \phi$.
Supposons que : $\exists\ i \in [\![1; n]\!]$ tel que $A \subseteq E_i$.
Si $I = [\![1; n]\!] \setminus \{i\}$, on a : $card(I) \geq 2$, et $\forall\ k \in I$, $B \cap E_k \neq \phi$.
Donc, $\exists\ j \in I$ tel que $B \cap E_j \prec \bigcup_{k \in I \setminus \{j\}} (B \cap E_k)$, ce qui contredit (4).
Donc, $\nexists\ i \in [\![1; n]\!]$ tel que $A \subseteq E_i$.
Si $J = \{k \in [\![1; n]\!] \mid A \cap E_k \neq \phi\}$, on a : $card(J) \geq 2$.
Donc, $\exists\ j \in J$ tel que $A \cap E_j \prec \bigcup_{k \in J \setminus \{j\}} (A \cap E_k)$, ce qui contredit (3).
Donc, on ne peut pas avoir $n \geq 3$.

□

Bien que l'on ne puisse établir une correspondance totale entre ces objets, les expirations propres représentent, en pratique, le meilleur équivalent possible des transitions temporisées. En effet, le théorème 1 nous assure que, si l'on découvre une expiration propre dans les logs, alors il existe une transition temporisée dans le protocole de conversation, ou alors on est en présence de messages plus longs à émettre que d'autres, sachant que les logs seuls ne permettent pas de déceler si l'on se trouve dans un tel cas de figure. Ce résultat justifie la pertinence de la mise en place d'une méthode de découverte des transitions temporisées basée sur la recherche des expirations propres satisfaites par les logs.

La méthode de découverte "naïve" consiste à générer toutes les expirations propres possibles, et à tester pour chacune d'entre elles si les conditions de la définition 7 sont vérifiées. Cette méthode est cependant doublement exponentielle car, d'une part le nombre d'expirations propres possibles est exponentiel, et d'autre part pour chaque couple (A, B) de sous-ensembles de P_m comparables grâce à la relation $\prec$, il est nécessaire de vérifier que tous les sous-ensembles de A et de B sont incomparables. Aussi, travaillons-nous actuellement à la définition d'une caractérisation des expirations propres conduisant à un algorithme de découverte polynomial. Cette caractérisation est basée sur la construction d'une partition de chaque sous-ensemble P_m d'épisodes. Elle est actuellement en cours de démonstration.

Rappelons que les éléments de la partition $\{P_m \,|\, m \in Msg\}$ de Ep sont traités séparément. A chaque partie P_m va donc correspondre un ensemble d'expirations propres qui lui sont associées. Ce procédé peut sembler redondant, dans le sens où, si deux transitions étiquetées respectivement par les messages a et b arrivent sur un même état, d'où sort une transition temporisée, nous allons trouver que deux expirations propres différentes sont satisfaites dans les logs (une pour P_a et une autre pour P_b), et les interpréter comme étant deux transitions temporisées différentes. Il est possible de résoudre ce problème en faisant des recoupements entre les différents ensembles d'expirations propres. Ceci permettra également de rejeter certaines expirations propres qui ne peuvent correspondre à des transitions temporisées.

Exemple 9 Considérons les expirations propres satisfaites par les logs L_1 (cf. tableau 1). L'expiration propre $EP(a, \{h\}, \{g\})$ (associée à P_a) ne peut correspondre à une transition temporisée, car h et g interviennent ensemble dans l'expiration propre $EP(b, \{c, d, e\}, \{g, h\})$ (associée à P_b). D'après le théorème 1, on sait que h et g correspondent à des transitions sortant du même état. Finalement, on trouve deux transitions temporisées : l'une correspondant à $EP(a, \{c, d, e\}, \{h\})$ et $EP(b, \{c, d, e\}, \{g, h\})$, et l'autre à $EP(b, \{f\}, \{c, d, e\})$.

5 Conclusions et perspectives

Notre travail se situe dans le contexte de l'extraction du protocole de conversation temporisé d'un service Web à partir de ses logs d'exécution. Il traite de la découverte des transitions temporisées, et constitue, à notre connaissance, la première contribution apportée à la résolution de ce problème. Notre apport consiste en un cadre formel aboutissant à la définition de la notion d'expiration. Nous avons montré que l'ensemble des expirations propres satisfaites par les logs constitue une caractérisation de l'ensemble des transitions temporisées présentes dans le protocole de conversation d'un service.

Du fait qu'il concerne les aspects temporels du protocole de conversation, notre résultat s'inscrit en complément des travaux existants. Nous envisageons d'intégrer l'algorithme de découverte des transitions temporisées sur lequel nous travaillons à la méthode d'extraction du protocole de conversation (non temporisé) proposée par [9, 10, 11], au sein d'une plateforme commune de gestion de services Web. Ceci nous permettra de pouvoir effectuer des tests à grande échelle de notre méthode.

Signalons également que nous avons comme objectif d'élargir le cadre formel présenté ici. Nous envisageons pour cela d'essayer de relâcher les contraintes fixées au départ (par exemple le fait que les transitions du protocole de conversation soient étiquetées de façon unique, ou qu'il n'existe pas de transition temporisée menant dans un état final). La solution permettant de pallier ces limites pourrait être d'effectuer un pré-traitement sur les données, afin de se ramener au cas particulier défini par nos hypothèses de travail. Il serait également intéressant de prendre en compte le "bruit" présent dans les données.

Ce travail s'inscrit dans le cadre du projet *ServiceMosaic*[1] regroupant plusieurs équipes de recherche. ServiceMosaic est un prototype de plateforme visant à permettre la modélisation, l'analyse et la gestion de services Web. Les principaux objectifs du projet sont la définition de modèles pour la description, l'orchestration et la composition de services, la spécification d'une algèbre pour une analyse de haut niveau, ainsi que la création d'outils de développement et de gestion de services.

References

[1] Dana Angluin and Carl H. Smith. Inductive inference: Theory and methods. *ACM Computing Surveys*, 15(3):237–269, Sep 1983.

[2] Boualem Benatallah, Fabio Casati, Julien Ponge, and Farouk Toumani. Compatibility and replaceability analysis for timed web service protocols. In Véronique Benzaken, editor, *Proceedings of BDA 2005*, Saint-Malo, France, Oct 2005.

[3] Boualem Benatallah, Fabio Casati, Julien Ponge, and Farouk Toumani. On temporal abstractions of web services protocols. In *Proceedings of CAiSE Forum 2005*, Porto, Portugal, Jun 2005. Springer.

[4] Boualem Benatallah, Fabio Casati, and Farouk Toumani. Analysis and management of web service protocols. In Paolo Atzeni, Wesley W. Chu, Hongjun Lu, Shuigeng Zhou, and Tok Wang Ling, editors, *Conceptual Modeling - ER 2004*, pages 524–541, Shanghai, China, Nov 2004. Springer.

[5] Tim Berners-Lee, James A. Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.

[6] Jonathan E. Cook and Alexander L. Wolf. Discovering models of software processes from event-based data. *ACM Transactions on Software Engineering and Methodology*, 7(3):215–249, Jul 1998.

[1] http://servicemosaic.isima.fr/

[7] Schahram Dustdar, Robert Gombotz, and Karim Baïna. Web services interaction mining. Technical Report TUV-1841-2004-16, Technical University of Vienna, Vienna, Austria, Sep 2004.

[8] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3):259–289, Sep 1997.

[9] Hamid Motahari, Boualem Benatallah, and Régis Saint-Paul. Protocol discovery from imperfect service interaction data. In Junho Shim and Fabio Casati, editors, *Proceedings of the VLDB 2006 Ph.D. workshop*, Seoul, Rep. of Korea, Sep 2006.

[10] Hamid Motahari, Régis Saint-Paul, Boualem Benatallah, and Fabio Casati. Protocol discovery from imperfect service interaction logs. In *Proceedings of ICDE'07*, Istanbul, Turkey, Apr 2007.

[11] Hamid Motahari, Régis Saint-Paul, Boualem Benatallah, Fabio Casati, Julien Ponge, and Farouk Toumani. Servicemosaic: Interactive analysis and manipulations of service conversations. In *ICDE'07*, Istanbul, Turkey, Apr 2007. Demonstration.

[12] Rajesh Parekh and Vasant Honavar. Grammar inference, automata induction, and language acquisition. In Robert Dale, Hermann Moisl, and Harold Somers, editors, *Handbook of natural language processing*, pages 727–764. Marcel Dekker, Inc., New York, USA, 2000.

[13] Wil van der Aalst, Ton Weijters, and Laura Maruster. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, Sep 2004.

Summary

The knowledge of the business protocol of a Web service is very important, for both clients and providers, because it represents a model of its external behaviour. However, it is often not specified during the design phase. The context of our work is the discovery of the timed business protocol of an existing service from its execution data. We consider an important subproblem: the discovery of the timed transitions (i.e. the state changes related to temporal constraints). We present a formal framework where we define the *proper timeouts*, which represent in the logs an equivalent of what the timed transitions are in the business protocol. To the best of our knowledge, it is the first contribution to this problem.