



HAL
open science

Vers une prise en charge approfondie des phénomènes itératifs par TimeML

Julien Lebranchu, Yann Mathet

► **To cite this version:**

Julien Lebranchu, Yann Mathet. Vers une prise en charge approfondie des phénomènes itératifs par TimeML. Actes de la 18ème conférence Traitement Automatique des Langues Naturelles (TALN 2011), Jun 2011, Montpellier, France. 6 p. hal-01069658

HAL Id: hal-01069658

<https://hal.science/hal-01069658>

Submitted on 29 Sep 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Vers une prise en charge approfondie des phénomènes itératifs par TimeML

Julien Lebranchu, Yann Mathet

Université de Caen Basse-Normandie, UMR 6072 GREYC, F-14032 Caen, France
Prénom.Nom@unicaen.fr

Résumé. Les travaux menés ces dernières années autour de l'itération en langue, tant par la communauté linguistique que par celle du TAL, ont mis au jour des phénomènes particuliers, non réductibles aux représentations temporelles classiques. En particulier, une itération ne saurait structurellement être réduite à une simple énumération de procès, et du point de vue de l'aspect, met en jeu simultanément deux visées aspectuelles indépendantes. Le formalisme TimeML, qui a vocation à annoter les informations temporelles portées par un texte, intègre déjà des éléments relatifs aux itérations, mais ne prend pas en compte ces dernières avancées. C'est ce que nous entreprenons de faire dans cet article, en proposant une extension à ce formalisme.

Abstract. The work that has recently been done concerning the iterative phenomena in language, which was performed by the linguistic and TAL communities, has illuminated specific phenomena, not reducible to classical time representations. In particular, an iteration can not structurally be reduced to a simple listing of process, and involves simultaneously two independent referred aspectual. The TimeML formalism, which aims to annotate temporal information of a given text, includes already relative elements to iterations but does not take into account recent advances. That is the reason why in this paper, we propose to extend this formalism.

Mots-clés : TimeML, discours, sémantique, phénomènes itératifs.

Keywords: TimeML, discourse, semantics, iterative phenomena.

1 Introduction

Le travail que nous menons actuellement se place dans le contexte de la sémantique temporelle, et plus précisément dans celui des phénomènes itératifs en corpus. Si plusieurs études linguistiques se sont attachées à décrire les mécanismes itératifs autour de la proposition ou de la phrase, nos travaux se positionnent quant à eux au niveau du discours (repérage des données itératives au sein d'un texte, détermination de l'étendue de chaque itération, etc., cf. (Lebranchu, 2009)). C'est dans ce contexte que nous nous attachons à déterminer le plus précisément les informations temporelles et itératives d'un texte, et que nous portons un intérêt particulier à TimeML.

L'itération en langue est un phénomène menant à la construction de différents procès répartis dans le temps et considérés comme étant la répétition d'un même événement. De façon typique, ce phénomène peut se manifester à partir d'une seule proposition telle que « *Nous sommes allés 7 fois à la montagne.* » où le procès « nous aller à la montagne » est itéré par l'adverbe de quantification « 7 fois ». Elle peut être circonscrite à une seule proposition, mais peut aussi s'étaler sur plusieurs propositions ou phrases, successives ou non, comme dans l'exemple qui suit.

Chaque lundi matin, le brocanteur qui logeait sous l'allée étalait par terre ses ferrailles. Vers midi, au plus fort du marché, on voyait paraître sur le seuil un vieux paysan de haute taille. Peu de temps après, c'était Liébard, le fermier de Touques, [...]

Ainsi, le circonstanciel *chaque lundi matin* déclenche l'itération du procès « le brocanteur étaler par terre ses ferrailles », à raison d'une fois par semaine sur une certaine période. Les deux propositions « Vers midi, ..., on voit paraître », et « Peu de temps après, ce être Liébard » sont elles des constituants de cette itération, qu'elles viennent successivement enrichir. Nous noterons en revanche que la subordonnée (*qui logeait sous l'allée*) n'est pas incluse dans l'itération.

Afin de représenter les informations liées aux phénomènes itératifs, nous avons opté pour ISO-TimeML (ci-après TimeML), un formalisme XML défini par Lee *et al.* (2007), dont l'objectif est d'annoter des textes avec l'ensemble de leurs informations temporelles. Il permet d'annoter les unités linguistiques réalisant des procès, les expressions temporelles, les relations aspectuelles, temporelles ou de subordination qui peuvent exister entre événements et expressions temporelles ainsi que les marqueurs de ces relations.

TimeML offre cependant une prise en charge assez limitée des phénomènes itératifs, et l’objet de cet article est donc d’enrichir ce formalisme en conséquence. Nous exposons dans la section 2 les entités nécessaires à la représentation dans notre approche des phénomènes itératifs. La section 3 présente une introduction à TimeML, ses enjeux, sa façon actuelle de formaliser des itérations, et en étudie les limites. Enfin nous exposons à la section 4 nos propositions d’extension de ce formalisme.

2 Tour d’horizon des phénomènes itératifs en langue

La littérature sur les phénomènes itératifs est principalement orientée vers la source de ces phénomènes. Citons notamment les travaux de Condamines (1990) sur les subordonnées temporelles qui font une distinction entre la phrase simple et la phrase complexe à subordonnée temporelle. Plus récemment, Lim (2002) a étudié la notion de fréquence en français, en présentant de façon approfondie les différents mécanismes mis en place dans les phénomènes itératifs. Nous orientons nos travaux selon deux approches complémentaires, une approche aspectuelle et une approche orientée objet, que nous allons brièvement présenter.

L’approche aspectuelle se fonde sur le modèle de la sémantique de la temporalité (ci-après « *SdT* ») en français de Gosselin (1996). Ce modèle néo-Reichenbachien intègre les différents paramètres explicatifs de la temporalité verbale en français. Il repose sur une représentation du temps linguistique au moyen d’une ligne temporelle et quatre intervalles temporels, dont en particulier l’intervalle de référence qui en constitue la pierre angulaire. Il propose un ensemble de règles permettant de rendre compte des phénomènes linguistiques liés à la temporalité, en se basant sur deux concepts linguistiques : le temps qui situe le procès dans le passé, le présent ou le futur, et l’aspect qui précise la façon dont est présenté le procès. Le traitement des phénomènes itératifs se fonde sur les mêmes principes mais un nouveau concept a été introduit pour pouvoir représenter les itérations : la série itérative. Elle est conçue comme un macro-procès englobant une série d’occurrences d’un même élément itéré, ce dernier étant considéré comme « *modèle* » (ou prototype) de chacun des itérés. Un aspect important de ce dispositif est qu’il permet de rendre compte du fait que l’itération peut faire l’objet, simultanément, de deux visées aspectuelles éventuellement différentes, comme dans « *Depuis deux mois, il mangeait en dix minutes.* », où la visée est aoristique sur le procès modèle, et inaccomplie sur la série (d’où la compatibilité de circonstanciers apparemment contradictoires : *en dix minutes* évalue la durée du procès modèle, tandis que *depuis deux mois*, porte sur le début de la série et le moment considéré). Cette approche a fait l’objet d’adaptations à d’autres langues et notamment l’anglais.

La seconde approche, qui a une visée cognitive, propose une Modélisation Objet des Phénomènes Itératifs (MOPI), cf. (Mathet, 2007). En particulier, la notion de modèle itératif (comprenant un ou plusieurs procès modèles) servant de support à l’itération, et le processus menant à l’instanciation multiple de ce modèle en différents itérés (la série itérative évoquée *supra*) ont été modélisés de façon orientée objet. Par ailleurs, la notion d’Itérateur rend compte des éléments déclencheurs d’une itération (comme par exemple un circonstanciel), et sont répartis en 4 classes : les itérateurs calendaires (tous les jours, le jeudi, ...), les itérateurs fréquentiels (souvent, rarement, parfois, ...), les itérateurs quantificationnels (3 fois, à cinq reprises, ...), et les itérateurs événementiels qui correspondent aux subordonnées temporelles.

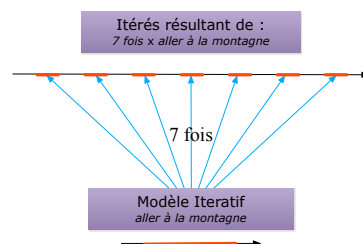


FIGURE 1: Le modèle itératif *aller à la montagne* et ses itérés.

Parallèlement, en plus de l’itération, le modèle distingue une seconde structure itérative : la sélection. Les sélections, via un sélecteur (adverbe de fréquence, ...), viennent sélectionner certains itérés d’une itération afin d’enrichir ou de modifier ces derniers. Dans « *Nous sommes allés 7 fois à la montagne. Parfois, des amis nous ont accompagnés* », le procès modèle *aller à la montagne* est répété par le déclencheur quantificationnel. Puis s’opère une sélection, rattachée à l’itération du procès précédent, qui se voit enrichie du procès modèle *des amis nous accompagner*. Notons qu’une sélection constitue elle-même une itération au sens où elle est constituée d’itérés provenant d’un même modèle.

À la lumière de ces études, nous souhaitons donc que soient prises en charge les informations itératives suivantes :

- le modèle de l’itération (l’itérant) ;
- la série associée (les itérés) ;
- la visée aspectuelle propre au modèle d’une part, et à la série d’autre part ;
- les itérateurs et la catégorie à laquelle ils appartiennent ;
- la notion de sélection opérant sur une itération.

La Figure 2 présente autour d’un exemple un certain nombre de ces informations.

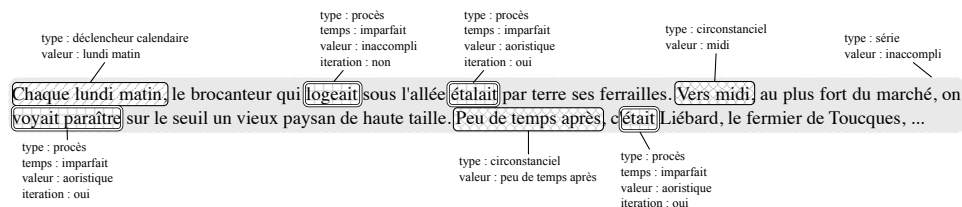


FIGURE 2: Structures de traits relatives aux informations temporelles.

3 TimeML

Présentation

TimeML est le fruit d’un groupe de travail dirigé par James Pustejovsky qui vise à standardiser les annotations sémantiques ayant trait à la temporalité dans un texte en langage naturel. Originellement, il fut conçu pour répondre à quatre problèmes liés à l’annotation des événements et des expressions temporelles : l’identification d’un événement et son ancrage dans le temps, l’ordonnancement des événements les uns par-rapport aux autres (ordonnancement lexical ou discursif), le raisonnement sur des expressions temporelles en prenant en compte le contexte (*la semaine dernière*) et le raisonnement sur la persistance des événements (combien dure un événement). Le guide d’annotation issu des premiers travaux est tourné vers l’anglais, mais les éléments du standard sont *a priori* indépendants de la langue considérée. Il existe notamment des travaux sur l’italien ou encore sur le coréen, et, plus récemment, un guide d’annotation pour le français a été élaboré (Bittar, 2010).

Les entités de TimeML

Actuellement, afin de couvrir l’essentiel des informations temporelles que l’on peut associer à un texte, le standard prévoit les fonctionnalités suivantes : l’annotation des événements, l’étiquetage des expressions temporelles et la normalisation de leurs valeurs, ainsi que la mise en évidence des relations aspectuelles, temporelles ou de subordination qui peuvent exister entre ces deux types d’entités temporelles. Le standard propose trois types d’entités annotables : les adverbiaux (TIMEX3), les éventualités (EVENT), et les signaux (SIGNAL).

Les TIMEX3 représentent les expressions temporelles, et sont ordonnées selon quatre classes : les dates (*le 23 mars 2011, 23/03/2011, ...*), les heures (*11H29, le matin, ...*), les durées (*4 jours, deux ans et demi, ...*) et les ensembles (*annuellement, le jeudi, ...*), correspondant respectivement aux types DATE, TIME, DURATION et SET. Il est possible d’annoter les expressions temporelles comme *l’année prochaine, le lundi précédent* dont les valeurs sont spécifiées par rapport un autre point de référence. Les EVENT correspondent aux événements d’un énoncé dans une acception large puisque intégrant les états. On retrouve cette acception dans différents travaux sous la notion d’éventualité (Vendler, 1967), ou encore procès que nous reprenons à notre compte. Ces procès sont marqués textuellement soit par un verbe soit par un nom, et sont classés selon une ontologie prédéfinie composée de 7 classes : occurrence (la plupart des événements), état, action ou état intensionnel (ceux-ci supposent un second événement qui ne se réalise pas nécessairement, comme dans *je voudrais qu’il vienne*), action d’énonciation ou de perception (dire, révéler, entendre, voir, ...) et les aspectuels (commencer, finir de, continuer, ...). Le formalisme permet de distinguer l’événement lui-même de sa réalisation en attribuant un identifiant à l’événement et un à l’instance. L’entité EVENT permet également de représenter les informations telles que le temps grammatical, la forme (nom, verbe, adjectif), l’aspect (Progressive, Imperfective, Perfective, ...) ou encore la présence d’une négation. Enfin, la balise SIGNAL sert à identifier une forme (généralement une préposition) marquant une relation temporelle entre événements et localisations temporelles, telle que *après* dans l’exemple *Il viendra après le 3 avril*.

Le standard prévoit, en plus des entités, la formalisation des relations existantes entre ces entités. Ces relations sont de trois types : TLINK, représentant les relations temporelles entre des événements, des marqueurs temporels ou entre l’un et l’autre (ces relations sont issues des travaux d’Allen (1983) sur les intervalles, entre autres la *succession*, l’*antériorité* et le *recouvrement*) ; ALINK, représentant les relations entre les verbes aspectuels et

les événements auxquels ils réfèrent ; et SLINK, introduisant les relations entre deux instances d'événements. L'exemple suivant présente une annotation TimeML de la phrase « *Bill veut enseigner le lundi* »

```
Bill <EVENT eid="e1" eiid="ei1" class="I_STATE" pos="VERB" tense="PRESENT" aspect="NONE"
polarity="POS">veut</EVENT> <EVENT eid="e2" eiid="ei2" class="OCCURRENCE" pos="VERB"
aspect="NONE" tense="NONE" vform="INFINITIVE" polarity="POS"> enseigner</EVENT>
<TIMEX3 tid="t1" type="SET" value="XXXX-WXX-1"> le lundi</TIMEX3>
<TLINK eventInstanceID="ei2" relatedToTime="t1" relType="IS_INCLUDED"/>
<SLINK eventInstanceID="ei1" relType="INTENSIONAL" subordinatedEventInstance="ei2"/>
```

Les itérations dans TimeML

Actuellement, le standard représente les itérations au niveau d'un événement, c'est-à-dire via l'entité EVENT en renseignant l'attribut CARDINALITY. De plus, il est également possible de créer l'ensemble des instances en créant n entités. Le standard formalise les sélections d'une sous-série d'événements en utilisant les cardinalités respectives des entités EVENT et une relation TLINK de type INCLUDES ou IS_INCLUDED. Quant aux déclencheurs, il est uniquement possible d'annoter des « quantificateurs », via l'entité EVENT, et des « calendaires », par l'entité TIMEX. Dans ce dernier cas, l'entité TIMEX représente l'ensemble des unités temporelles décrivant l'expression temporelle. L'exemple ci-dessus présente l'annotation d'une itération dont le déclencheur calendaire est « *le lundi* », représenté par le patron « *XXX-WXX-1* » associé à TIMEX.

Les limites actuelles

Compte-tenu de ce que nous venons de présenter, les limites de TimeML relativement aux phénomènes itératifs sont de deux ordres.

Premièrement, il n'y pas de distinction entre l'instance d'un événement et l'instance d'une itération, hormis l'attribut *cardinality*. Or, nous avons montré qu'il est possible d'affecter des valeurs aspectuelles différentes à la série itérative et au procès. De plus, il est nécessaire de déterminer si le rattachement d'un procès se fait au niveau de l'itération (*J'allais souvent à la mer. C'était des moments magnifiques.*) ou de l'événement (*Marie tomba plusieurs fois en tâchant d'atteindre la barrière, puis parvint à la franchir.*).

Deuxièmement, le standard, par choix des concepteurs, ne permet pas de prendre en compte les adverbes de fréquences (« rarement », « souvent », ...) ou bien les conjonctions de subordination (« quand », « lorsque », « à chaque fois que », ...), c'est-à-dire les itérateurs de type « fréquentiel » et « événementiel ». De plus, le formalisme de représentation des expressions temporelles n'est pas complet. Il ne permet pas de couvrir des expressions telles que « Un mardi sur deux », « tous les troisièmes mardis du mois », « trois jours ouvrables », ou encore « le 4ème des 6 jours de campagne ».

4 Extension de TimeML aux phénomènes itératifs

Nous allons proposer dans cette section un certain nombre d'extensions au standard TimeML relatives aux phénomènes itératifs. Pour distinguer du standard TimeML les extensions que nous proposons, nous avons choisi d'utiliser l'espace de nom « *iteration* » qui précède les entités que nous introduisons. Nous proposons donc l'entité ITERATION:STRUCTURE pour les structures itératives, et l'entité ITERATION:TRIGGER pour les déclencheurs.¹

4.1 Les structures itératives

Nous proposons de représenter les structures itératives via une entité STRUCTURE dans laquelle nous intégrons un attribut obligatoire TYPE, permettant de distinguer une itération et une sélection. Pour rendre compte de l'ensemble des informations nécessaires, nous avons les attributs suivants :

- l'attribut STID attribue un identifiant à l'entité STRUCTURE ;
- l'attribut TYPE permet de distinguer l'itération de la sélection (« itération », « sélection ») ;
- l'attribut TRIGGERID permet d'identifier le déclencheur associé à la structure ;
- l'attribut ASPECT renseigne la valeur aspectuelle de la structure (aoristique, accompli, inaccompli et prospectif) ;
- l'attribut STRUCTUREINSTANCEID permet de représenter les liens entre deux structures itératives, notamment dans les cas d'imbrication d'itérations et de sélections.

Nous avons montré préalablement qu'une structure itérative est composée de procès. Nous conservons la représentation existante, c'est-à-dire l'entité EVENT, en lui ajoutant un attribut (nommé STRUCTUREINSTANCEID) qui permet d'identifier la structure itérative à laquelle elle est rattachée. L'exemple qui suit illustre la représentation

1. Par souci de lisibilité, dans la suite de l'article, nous ne mentionnons pas l'espace de noms.

des structures itératives, où nous retrouvons le déclencheur de l'itération *7 fois* (dont nous détaillons la balise TRIGGER dans la suite de cet article), et l'événement itéré *aller à la montagne*. Puis le déclencheur *parfois* qui opère une sélection de certains itérés, pour les enrichir du procès modèle *des amis nous accompagner*.

Nous

```
<EVENT eid="e1" eiid="ei1" aspect="aoristique" structureInstanceID="st1">sommes allés</EVENT>
<TRIGGER tid="t1"...>7 fois</TRIGGER> à la montagne.
<STRUCTURE type="iteration" stid="st1" triggerID="t1" aspect="aoristique"/>
<TRIGGER tid="t2"...>Parfois,</TRIGGER> des amis nous <EVENT eid="e2" eiid="ei2"
aspect="aoristique" structureInstanceID="st2">ont accompagnés</EVENT>.
<STRUCTURE type="selection" stid="st2" triggerID="t2" aspect="aoristique"
structureInstanceID="st1"/>
```

4.2 Les déclencheurs

Nous proposons de créer une nouvelle entité pour couvrir l'ensemble des déclencheurs d'itérations tels que nous avons pu les définir en présentant le modèle MOPI. Nous représentons donc ces derniers par l'entité TRIGGER à laquelle est adjoit un ensemble de couples d'attributs et valeurs pour notamment distinguer les différents types de déclencheurs. L'entité est donc composée d'un identifiant TRID, d'un attribut TYPE et d'un attribut VALUE contenant la valeur portée par le déclencheur analysé qui sont tous trois obligatoires, et en fonction du déclencheur, nous avons un certain nombre d'attributs optionnels que nous détaillons.

Les déclencheurs aspectuels

Les déclencheurs de type aspectuel, tels que *encore*, présupposent l'existence d'autres instances de l'événement. Nous proposons de spécifier le nombre d'itérés au moyen d'un patron simple, correspondant à un nombre minimum de répétitions présupposées suivi du symbole "+". Dans l'exemple suivant, nous avons donc au minimum deux instances de *casser une vitre*.

```
Il a <TRIGGER tid="t1" type="ASP" value="2+">encore</TRIGGER> cassé une vitre.
```

Les déclencheurs quantificationnels

Les déclencheurs de type quantificationnel, pour rappel, correspondent à des syntagmes tels que « à trois reprises », « 5 fois », « quelques fois ». L'attribut TYPE se voit affecter la valeur QUANT, et l'attribut VALUE contient la valeur de la répétition portée par le quantifieur. Dans les cas où la sémantique associée au quantifieur n'est pas précise, comme pour « plusieurs fois », la valeur VALUE est *unspecified*.

```
Marie tomba <TRIGGER trid="n1" value="3" type="QUANT">à trois reprises</TRIGGER>.
Nos <TRIGGER trid="n2" value="2" type="QUANT">2 voyages</TRIGGER> à Hawaii furent merveilleux.
```

Les déclencheurs événementiels

Les déclencheurs événementiels correspondent aux conjonctions introduisant des subordinées temporelles telles que *lorsque*, *à chaque fois que*, *dès que*. Nous attribuons la valeur EV à l'entité TYPE tandis que l'attribut VALUE spécifie la relation temporelle existant entre la subordinée introduite par le marqueur événementiel et la proposition principale. L'exemple suivant illustre une représentation du marqueur *lorsque* qui spécifie une relation de simultanéité entre les deux propositions, sur l'énoncé *Lorsque John enseigne à l'IUT, il dépose d'abord les enfants à l'école*. Remarquons qu'il existe une contradiction apparente entre la valeur de l'introducteur et la principale qui s'explique par le fait que le procès principal peut dans certains cas, comme ici avec « *John enseigne* », être en fait une partie seulement d'un scénario itéré plus large inferé à partir de ce procès.

```
<TRIGGER trid="ev1" type="EV" value="SIMULTANEOUS">Lorsque</TRIGGER> John
<EVENT eid="e1" eiid="ei1" structureInstanceID="scil" aspect="aoristique">enseigne</EVENT>
à l'IUT, <STRUCTURE type="iteration" structureID="scil" triggerID="ev1" aspect="inaccompli"/>
il <EVENT eid="e2" eiid="ei2" structureInstanceID="scil" aspect="accompli" >dépose</EVENT>
<signal sid="s2">d'abord</signal> les enfants à l'école.
<TLINK eiid="ei2" relatedToEvent="ei1" relType="BEFORE" signalID="s2" />
```

Les déclencheurs calendaires

Les déclencheurs calendaires sont actuellement représentés par l'entité TIMEX, permettant de formaliser des formes telles que *chaque lundi*, *tous les jeudis*. Cependant certaines expressions ne sont pas exprimables dans ce formalisme, notamment pour des exemples tels que *un mardi sur deux*, *tous les troisièmes mardis du mois*. Par souci d'homogénéité, nous représentons également ces déclencheurs par l'entité TRIGGER, en spécifiant le type par la valeur CAL et rendant la valeur SET de TIMEX obsolète, dans le cadre de notre extension. Afin de couvrir tous les expressions calendaires, nous utilisons la formalisation algébrique développée par P. Enjalbert et G. Becher (À paraître) qui possède un ensemble de mécanismes permettant de calculer les séries associées à une expression calendaire itérative. La fonction permet d'obtenir la série correspondant à la valeur de l'attribut VALUE, et nous obtenons les représentations suivantes pour les expressions associées :

- un mardi sur deux \Rightarrow SUR(1, 2, MARDI)
- tous les troisièmes mardis du mois \Rightarrow TOUSLESN(3, MARDI)

Par ailleurs, il est possible de se référer temporellement au déclencheur calendaire soit à un niveau global (*J'allais à la piscine tous les mardis. Maintenant, j'y vais le jeudi.*), soit au niveau modèle (*Tous les jeudis, John enseigne pendant 3 heures puis il va manger au RU.*). Afin de distinguer les deux niveaux, nous ajoutons un identifiant au niveau modèle, TMID, auquel il pourra être fait référence.

```
<TRIGGER trid="t1" tmid="tm1" type="CAL" value="TousLesN(jeudi)">Tous les jeudis</TRIGGER>,
des habitués venaient faire une partie de boston.
```

Les déclencheurs fréquentiels

Nous finissons par la représentation des déclencheurs fréquentiels (*rarement, parfois, ...*), qui s'avère être plus complexe que pour les déclencheurs précédents. En effet, le calcul de la fréquence dépend de nombreux paramètres tels que l'agent mis en œuvre dans l'énoncé, le contexte ou bien encore l'action réalisée. Par exemple, dans les deux exemples suivants : *Elizabeth T. s'est souvent mariée* et *Elizabeth T. a souvent tourné des films*, l'adverbe *souvent* aura une fréquence différente. Dans notre premier exemple, la fréquence est de huit mariages au cours de sa vie tandis que le second fait référence à 70 tournages. Ce calcul de la fréquence dépasse cependant le cadre de cet article. Nous proposons donc de représenter la valeur sémantique associée par une fonction. Dans nos deux exemples précédents, nous représentons donc la fréquence par la fonction *often* comme suit :

```
E.T. a<TRIGGER trid="trl" type="FREQ" value="often()">souvent</TRIGGER> tourné des films.
```

5 Conclusion

Une extension² du langage TimeML s'est avérée nécessaire pour une annotation enrichie des phénomènes itératifs. Premièrement, pour rendre compte de la double visée aspectuelle, de la série itérative et du procès ; deuxièmement, pour capturer l'ensemble des structures itératives, itérations et sélections ; et troisièmement, pour représenter les différents déclencheurs existants. De fait, nous avons élaboré deux nouvelles entités STRUCTURE et TRIGGER représentant respectivement les structures itératives et les déclencheurs. Dans la perspective d'une meilleure intégration, il nous reste à définir une représentation des déclencheurs de type conflits comme dans l'exemple « *Depuis deux mois, il mangeait en dix minutes.* » que nous avons présenté ci-dessus. Actuellement, notre annotation s'est limitée à quelques articles du journal Le Monde, notre prochaine étape se consacrera donc à réaliser une annotation sur un corpus plus conséquent.

Références

- ALLEN J. F. (1983). Maintaining knowledge about temporal intervals. *Commun. ACM*, **26**(11), 832–843.
- BITTAR A. (2010). *Building a TimeBank for French : A Reference Corpus Annotated According to the ISO-TimeML Standard*. PhD thesis, Université Paris Diderot (Paris 7).
- CONDAMINES A. (1990). Les conjonctions de subordination temporelle en français. *Cahiers de Grammaires*, **15**, 25–54.
- ENJALBERT P. & BECHER G. (à paraître). *Itérations*, chapter L'itération dans le modèle SdT : une formalisation algébrique.
- GOSSELIN L. (1996). *Sémantique de la temporalité en français : un modèle calculatoire et cognitif du temps et de l'aspect*. Duculot.
- LEBRANCHU J. (2009). Vers une analyse automatique des phénomènes itératifs en corpus. In G. WILLIAMS, Ed., *Actes des 6èmes Journées de la Linguistique de Corpus*, Lorient, France : Université de Bretagne-Sud.
- LEE K., PUSTEJOVSKY J., BUNT H., BOGURAEV B. & IDE N. (2007). *Language resource management - Semantic annotation framework (SemAF) - Part 1 : Time and events*. Rapport interne, International Organization for Standardization, Geneva, Switzerland.
- LIM J.-H. (2002). *La fréquence et son expression en français*. Honoré Champion.
- MATHET Y. (2007). Une approche cognitive de l'itération et sa modélisation objet. *RTE*.
- VENDLER Z. (1967). *Linguistics in Philosophy*. Cornell U. P.

2. La syntaxe complète associée à l'extension que nous venons de présenter est décrite au sein d'une DTD accessible à l'adresse : <http://jlebranc.perso.info.unicaen.fr/TimeML/timeml-iter.dtd>.