

Détection d'Incohérences dans des Spécifications Formelles : Application à la Modélisation Spatiale et Temporelle en Archéologie

Nicolas GORSE – {ngorse@online.fr}

LOTOS Group – University of Ottawa, ON, Canada

Résumé

Cet article traite de la détection d'incohérences dans les spécifications formelles. Nous présentons les concepts de base de la logique des prédicats du premier ordre et de la résolution automatique s'y rattachant. Suit une présentation de nos travaux au sein du domaine de la téléphonie et une proposition d'extension de ces concepts au domaine de l'archéologie, plus particulièrement pour l'aide à la validation de chronologies.

1. INTRODUCTION

Un modèle se veut, par sa définition, une représentation générale d'un ensemble de caractéristiques, ou d'éléments, et des relations qui existent entre eux. La principale difficulté que présente l'élaboration d'un modèle est sa validation. Il va de soi qu'il se doit d'être cohérent quel que soit son champ d'application. La phase d'élaboration doit donc s'accompagner, ou tout

du moins être suivie, d'une phase de validation visant à s'assurer de la validité dudit modèle, c'est-à-dire, entre autres, de sa cohérence.

La validation d'un modèle s'avère souvent être longue et laborieuse, d'où l'intérêt, si possible, d'en automatiser une partie si ce n'est la totalité. Ici, le terme « automatique » fait référence à l'informatique, c'est-à-dire à l'utilisation d'algorithmes dédiés à l'analyse du modèle à valider¹. Bien entendu, il va de soi que l'utilisation d'outils informatiques implique que ce qui doit être validé soit représenté de façon formelle.

Nous traitons ici de l'application de la logique formelle, et des principes de résolution automatique, à la détection d'incohérences dans des spécifications formelles. Tout d'abord, nous présentons les concepts de base de la logique des prédicats du premier ordre ainsi que les principes de résolution automatique s'y rattachant. Suit une présentation de l'application de cette logique à la détection d'incohérences dans les spécifications de services téléphoniques. Dans le même ordre d'idées, nous élaborons quelques possibilités d'application à l'archéologie en vue de permettre la détection d'incohérences au sein de chronologies et d'ensembles d'éléments.

En conclusion, nous discutons des avantages et inconvénients des idées présentées et soumettons les possibilités de recherche future concernant l'intégration des concepts étudiés au prototype Porphyre² développé par Aurélien BÉNEL.

2. LOGIQUE DES PRÉDICATS DU PREMIER ORDRE

L'étude de la logique puise ses origines dans la civilisation grecque. De son évolution découle, entre autres, une méthode consistant à proposer des arguments pouvant être évalués par un ensemble de règles de raisonnement prédéfinies et acceptées. L'utilisation de cet ensemble de règles, conjuguées à un ensemble de faits reconnus comme étant *vrais*, permet l'évaluation des arguments qui se voient alors, tels des hypothèses, validés (vrai) ou invalidés (faux).

¹ E. KIT, *Software Testing in the Real World, Improving the Process*, Addison Wesley, 1996.

² A. BENEL, S. CALABRETTO, A. IACOVELLA, J-M. PINON, *Consultation de Documents et Sémantique, Application à des publications savantes*, CIFED 2000.

2.1. BASES

L'idée de base de cette logique repose sur le fait que les phrases représentent aussi bien des relations entre des objets que des qualités ou attributs d'objets. L'objet peut représenter des personnes, des objets physiques, des concepts, des idées etc. Les relations entre objets sont appelées *prédicats* et les objets forment les termes, ou arguments, de ces prédicats.

Plusieurs langages implémentent les concepts de cette logique, dont le bien connu Prolog³. Basé sur la logique du premier ordre, le Prolog a été inventé au début des années 70 par Alain Colmerauer, à l'université de Marseille, avec comme objectif de pouvoir faire du traitement de la langue naturelle (on s'apercevra par la suite que le champ d'application est en fait beaucoup plus large). En Prolog, tel que suggéré précédemment, les prédicats sont regroupés en deux catégories : les faits et les règles⁴.

2.1.1. Faits

Tel que mentionné précédemment, un fait est utilisé pour exprimer une relation entre des objets. Par exemple, pour modéliser un lien de parenté paternel entre deux personnes nous utilisons un prédicat pourvu de deux termes, l'un représentant le père et l'autre le fils. En Prolog, les noms de prédicats et les noms de constantes (ici, antoine, alain et anaïs) commencent toujours par une minuscule, les majuscules étant réservées pour la représentation des variables, ce qui permet de faire une distinction. Ainsi, le fait qu'Antoine soit le père d'Alain et le fait qu'Alain soit le père d'Anaïs seront représentés par les prédicats suivants :

```
pere(antoine, alain).  
pere(alain, anaïs).
```

2.1.2. Règles

Tout comme les faits, les règles sont des prédicats se composant de termes. Cependant, ils

³ P. ROUSSEL, *Prolog: Manuel de Référence et d'Utilisation*, Groupe d'Intelligence Artificielle, Marseilles, 1975.

⁴ W. F. CLOCKSIN, C.S. Mellish, *Programming in Prolog*, Springer IV, 1996.

diffèrent des faits puisqu'ils expriment non pas une relation établie comme l'expriment les faits mais une relation inférée, c'est-à-dire déduite à partir d'un ensemble de faits et de règles logiques. Une règle se présente donc sous la forme d'un fait auquel est associé un corps. Le corps est une équation booléenne se composant de prédicats et de variables. Cette équation définit la relation modélisée par la règle en question. Les règles sont utilisées pour définir des relations générales reposant sur des faits et des règles. Par exemple, au lieu de définir un ensemble de faits illustrant les relations entre les grands-pères et leurs petits-enfants, nous pouvons simplement définir la règle générale suivante :

```
grand_pere(X, Y) :-  
    pere(X, Z), (  
        pere(Z, Y)  
        ;  
        mere(Z, Y)  
    ).
```

La virgule et le point virgule représentent respectivement les opérations logiques *ET* et *OU*. Comme nous pouvons le constater, la règle exprime la relation suivante : X est le grand-père de Y s'il existe un Z dont X est le père et que ce même Z est, soit le père de Y, soit sa mère. De manière plus générale, nous considérons la règle *grand_pere(X, Y)* comme étant vraie s'il existe une valeur de X et une valeur de Z (variable intermédiaire) tel que le prédicat *pere(X, Z)* soit vrai et qu'il existe une valeur de Y tel que, soit le prédicat *pere(Z, Y)*, soit le prédicat *mere(Z, Y)*, soit vrai.

2.2. RESOLUTION AUTOMATIQUE

Les interpréteurs Prolog sont dotés d'un mécanisme d'évaluation des prédicats, le moteur d'inférence, leur permettant de traiter des requêtes simples concernant des faits. Il est donc

possible d'effectuer une requête sous la forme d'un fait. L'interpréteur en vérifiera alors l'existence dans sa base de connaissances (base de faits) et formulera un verdict en qualifiant la requête de vraie ou fausse suivant la présence ou l'absence dudit fait.

Admettons que les faits présentés précédemment aient été ajoutés à la base de connaissances de l'interpréteur. À la requête « Antoine est-il le père d'Alain », posée sous la forme « Antoine est le père d'Alain », sous-entendant « cette proposition est-elle vraie ? », l'utilisateur se verra répondre « oui », indiquant la véracité du fait, c'est-à-dire, son existence dans la base de faits de l'interpréteur. Réciproquement, la proposition inverse « Alain est le père d'Antoine » aura pour réponse « non ». La figure suivante illustre cet exemple.

```
?- pere(antoine, alain).  
Yes  
  
?- pere(alain, antoine).  
No
```

Plus intéressant encore, les interpréteurs Prolog sont dotés d'un mécanisme de résolution automatique offrant la possibilité d'évaluer et de compléter des requêtes dites partielles, c'est-à-dire dont un ou plusieurs termes sont des variables. Ce mécanisme est basé sur le principe d'initialisation automatique des variables avec pour but d'initialiser celles-ci avec une combinaison de valeurs conduisant à la véracité de la requête. En réponse à une requête partielle le verdict de l'interpréteur sera soit « non » s'il n'existe pas de combinaison de valeurs pouvant satisfaire la véracité de la requête, soit « oui » s'il en existe une ou plusieurs. Dans le second cas, l'interpréteur proposera à l'utilisateur la première combinaison trouvée et lui proposera d'afficher la prochaine.

Nous pouvons ainsi faire des requêtes du type « qui est le père d'Alain » comme l'illustre l'exemple suivant qui ne comporte qu'une solution possible.

```
?- pere(X, alain).  
  
X = antoine ;  
  
No
```

De la même façon, il est évidemment possible d'effectuer des requêtes, partielles ou non, sur des règles. Ainsi, la règle *grand_pere* définie précédemment en section 2.1.2, peut être utilisée pour formuler une requête visant à savoir le nom du grand-père d'Anaïs, tel qu'illustré par la figure suivante :

```
?- grand_pere(X, anais).  
  
X = antoine ;  
  
No
```

Et, bien évidemment, il est tout aussi possible de faire des requêtes à l'aide de prédicats ne comportant que des variables :

```
?- pere(X, Y).  
  
X = antoine  
Y = alain ;  
  
X = alain  
Y = anais ;  
  
No
```

3. ANALYSE DE SYSTÈMES TÉLÉPHONIQUES

Au cours du développement d'un système téléphonique (analyse, spécification, validation, implantation, etc.), l'utilisation de méthodes formelles permet d'assurer, de façon quasi automatique à différents niveaux du développement, la détection d'une grande partie de ces interactions. Une telle validation assure ainsi une réduction des interactions ainsi qu'une

meilleure qualité d'implantation⁵. Un grand nombre d'approches existent mais toutes n'agissent pas à tous les niveaux de développement et la majeure partie sont restreintes à certains types ou langages de spécifications formelles.

Dans le cadre de nos travaux au sein du groupe LOTOS de l'université d'Ottawa nous avons développé une approche basée sur une représentation formelle des services et sur un ensemble de règles permettant la détection des incohérences pouvant mener à des interactions entre les services considérés⁶. À cette détection s'ajoute la possibilité de générer des séries de tests de façon semi-automatique, permettant ainsi d'effectuer une validation aux différents niveaux de développement d'un système (analyse, spécification, implantation) avec une intervention minimale du concepteur.

Ces travaux ne relèvent en rien du domaine de l'intelligence artificielle. Notre but n'est pas, comme cela pourrait sembler, de bâtir un modèle de connaissances complet qui permettrait un apprentissage automatique des caractéristiques d'une interaction. Nous utilisons logique des prédicats du premier ordre de façon à pouvoir spécifier formellement les services téléphoniques ainsi qu'un ensemble de règles « méthodologiques » caractérisant les interactions. Nous utilisons ensuite les principes de résolution automatique dans le simple but d'identifier, automatiquement, la présence ou l'absence de problèmes dont les caractéristiques sont définies par les règles méthodologiques préétablies.

3.1. LE PROBLEME DES INTERACTIONS DE SERVICES

Les systèmes téléphoniques d'aujourd'hui sont définis de façon composite. Ils consistent en un noyau implémentant les fonctions de base du système auquel peuvent s'ajouter des fonctionnalités supplémentaires. Les fonctions de base consistent en l'établissement d'une

⁵ D. AMYOT, L. CHARFI, N. GORSE, T. GRAY, L. LOGRIPPO, J. SINCENNES, B. STEPIEN, T. WARE, *Feature Description and Feature Interaction Analysis with Use Case Maps and LOTOS*, Feature Interactions in Telecommunications and software systems VI, IOS Press, 2000, p. 274-289.

⁶ N. GORSE, *The Feature Interaction Problem: Automatic Filtering of Incoherences & Generation of Validation Test Suites at the Design Stage*, Master, University of Ottawa, 2000.

connexion entre deux usagers, la facturation d'un appel, sa terminaison, etc. Les fonctionnalités supplémentaires consistent en des services optionnels tels que, par exemple, le signal d'appel ou la boîte vocale.

La composition de certains services, c'est-à-dire leur utilisation simultanée présente des risques d'interactions. Ce que nous appelons une *interaction de service* se caractérise par l'engendrement d'un comportement non prévu, et généralement non désiré, gênant ainsi les utilisateurs ou pire, entravant le fonctionnement normal du système et mettant en danger sa stabilité.

3.1.1. Définition d'un Service Téléphonique

Le concept de service en téléphonie peut être défini de façon pragmatique par rapport aux besoins des usagers. Un service est une fonctionnalité offerte par le système et a pour objectif de satisfaire certaines intentions d'un usager dans le contexte d'un appel. Un service est vu comme une unité incrémentale optionnelle ajoutée au système. Un tel système se compose donc d'un noyau, autour duquel gravitent les services optionnels.

Appelons N le noyau du système et S_x un service où x représente le nom du service et définissons l'opérateur de composition : \oplus . La composition d'un service S_x avec le noyau est dénotée : $N \oplus S_x$. De même, un système composé d'un noyau et de n services sera dénoté : $N \oplus S_1 \oplus S_2 \oplus \dots \oplus S_n$. Bien évidemment, l'ajout d'un service au système influence son comportement, ainsi, $N \oplus S_1$ et $N \oplus S_1 \oplus S_2$ sont considérés comme deux systèmes différents. De nombreux services existent. Nous pouvons en citer quelques uns tels que :

- Le signal d'appel : *l'abonné est prévenu si quelqu'un tente de le joindre alors que sa ligne est occupée. Il a alors la possibilité, dans un délai limité, de décider de suspendre la communication courante de façon à pouvoir parler à l'utilisateur qui tente de le joindre.*
- La re-direction d'appels : *L'abonné a la possibilité de re-diriger à un numéro de téléphone différent les appels lui étant destinés.*

- La boîte vocale : *Tel un répondeur, la boîte vocale se déclenche lorsque la ligne est occupée ou que l'utilisateur ne répond pas. Elle offre alors à l'appelant la possibilité de laisser un message vocal.*

3.1.2. Définition et Caractéristiques d'une Interaction entre deux Services

Dans un système, les interactions de services peuvent avoir différentes causes. Nous identifions deux catégories principales :

Interactions Directes – Une interaction directe se caractérise par le fait que deux ou plusieurs services sont déclenchés en même temps par le même événement et mènent à des résultats différents ou contradictoires. Nous parlerons dans ce cas d'interactions caractérisées par un non-déterminisme.

Par exemple, un usager *X* ayant souscrit au signal d'appel est averti s'il reçoit un appel alors que sa ligne est occupée. Supposons que ce même usager ait aussi souscrit au service de boîte vocale, service étant déclenché lorsque la ligne est occupée et qu'un appel est reçu, ou bien si la ligne n'est pas occupée mais que l'abonné ne répond pas au bout d'un certain temps.

Ici, nous nous trouvons face à un non-déterminisme. Si la définition des services n'est pas attentivement étudiée, il se peut que l'un des deux, la boîte vocale par exemple, écrase les fonctionnalités de l'autre, le signal d'appel, sans pour autant que ce comportement soit désiré par l'utilisateur.

Interactions Transitives – Une interaction transitive, quant à elle, se caractérise par le fait que le résultat de l'exécution d'un service entraîne le déclenchement d'un autre service et que les résultats de ces deux derniers présentent une contradiction ou une boucle.

Supposons, par exemple, qu'un usager *X* soit abonné au service de re-direction d'appels et re-dirige tous les appels arrivant à un usager *Y*. Supposons que l'utilisateur *Y* soit lui-même abonné à un service de filtrage d'appels configuré pour bloquer tous les appels

en provenance d'un certain usager Z . Si Z appelle X , son appel sera alors re-dirigé à Y par le service de re-direction d'appels, avec l'intention d'atteindre Y . Cependant, l'appel arrivant à Y sera bloqué par le service de filtrage d'appels puisqu'il provient de Z .

Comme nous pouvons le constater, nous faisons ici face à une contradiction entre les résultats, ou intentions, des deux services.

3.2. REPRESENTATION FORMELLE ET VALIDATION DES SERVICES

D'une façon générale, il s'avère évident que les concepts présentés en section 2 peuvent être appliqués à la détection d'incohérences dans des spécifications formelles. En effet, si nous considérons le cas des services téléphoniques et des interactions que leur utilisation simultanée peut présenter, il est possible d'établir un modèle formel permettant la détection d'incohérences engendrées par la composition de deux services et ainsi d'identifier les interactions possibles.

En définissant les services de façon formelle à l'aide de prédicats, il devient alors possible d'élaborer des règles caractérisant les incohérences types. À l'aide d'un interpréteur Prolog, nous pouvons poser une requête de la forme « La composition de ces deux services peut-elle satisfaire la règle d'incohérence R ? ». Bien évidemment, le moteur d'inférence de notre interpréteur Prolog tentera de trouver une combinaison de valeurs qui permette d'initialiser les variables de façon à ce que la règle R soit satisfaite. Si une telle combinaison existe, la règle est satisfaite et nous pouvons conclure qu'une incohérence est présente. Dans le cas contraire, il apparaît évident que la combinaison des deux services ne présente pas d'incohérence, tout au moins par rapport à la règle donnée.

3.2.1. Représentation d'un Service

Un service téléphonique est une fonctionnalité du système pouvant être activée dans certains cas particuliers par un ou plusieurs événements déclencheurs. L'utilisation d'une telle fonctionnalité remplace ou s'ajoute à celles préalablement définies et conduit à un ensemble de résultats. De ce

fait, la représentation formelle d'un service est décomposée en trois groupes :

- Un ensemble de pré-conditions représentant l'état dans lequel le système doit se trouver pour que le service puisse être déclenché.
- Un ensemble d'évènement(s) déclencheur(s) représentant les actions entraînant l'activation du service.
- Un ensemble de résultat(s) représentant tout simplement les résultats engendrés par l'utilisation du service.

Nous considérons de surcroît un ensemble de contraintes utilisées pour définir certaines relations entre les variables des trois ensembles précédents. Ces relations permettent d'exprimer des contraintes telles que l'abonné et l'appelant doivent être deux usagers différents, autrement dit, que le service ne peut être déclenché si l'abonné s'appelle lui-même.

En guise d'exemple, considérons le service de re-direction d'appel si occupé présenté en section 3.1.1. La figure suivante illustre sa formalisation en Prolog sous forme d'une règle dont les termes sont des listes de prédicats contenant respectivement le nom du service, les pré-conditions, les évènements déclencheurs et les résultats. Le corps, quant à lui, contient la définition des contraintes relatives aux variables sous forme d'une équation booléenne.

<code>service(rao,</code>	<i>Nom du service</i>
<code> [abonne(B, rao), occupe(B)],</code>	<i>Pré-conditions</i>
<code> [appelle(A, B)],</code>	<i>Déclencheurs</i>
<code> [appelle(A, C), connexion(A, C)]</code>	<i>Résultats</i>
<code>) :-</code>	
<code> A \= B, A \= C, B \= C.</code>	<i>Contraintes</i>

3.2.2. Détection d'Incohérences

Plusieurs types d'incohérences directes et transitives sont possibles. À titre d'exemple, nous considérerons les incohérences directes caractérisées par le fait qu'un même usager soit abonné à deux services différents, que ces deux services puissent être déclenchés par les mêmes

événements et que leurs résultats diffèrent. En se basant sur les concepts présentés en section 2, nous pouvons formaliser les caractéristiques de ce groupe type d'incohérences. Une incohérence entre deux services S1 et S2 est définie par le fait que ces deux services existent (c'est-à-dire sont définis dans la base de faits de l'interpréteur) et que :

- Leurs pré-conditions ne présentent pas de contradiction.
- Le même usager est abonné aux deux services S1 et S2, ce qui signifie que les prédicats *abonne(X, S1)* et *abonne(X, S2)* font respectivement partie des ensembles de pré-conditions de S1 et de S2.
- Les services S1 et S2 sont différents l'un de l'autre.
- Les ensembles de déclencheurs sont identiques.
- Les ensembles de résultats diffèrent.

Ces critères sont représentés en Prolog comme suit :

```
service(Nom_1, Pre_conditions_1, Declencheurs_1, Resultats_1),
service(Nom_2, Pre_conditions_2, Declencheurs_2, Resultats_2),

not(contradiction(Pre_conditions_1, Pre_conditions_2),

est_inclus(abonne(X, S1), Pre_conditions_1),
est_inclus(abonne(X, S2), Pre_conditions_2),

S1 \= S2,

Declencheurs_1 = Declencheurs_2

Resultats_1 \= Resultats_2
```

L'interpréteur Prolog tentera d'initialiser les variables des deux services avec une combinaison de valeurs permettant de satisfaire à la règle qui caractérise les incohérences recherchées. Nous serons alors en mesure d'en certifier la présence ou l'absence en raison de la véracité de la règle.

3.3. RESUME

L'élaboration de notre théorie a conduit à l'implémentation d'un certain nombre de règles sous la forme d'un outil de détection des incohérences. L'application de notre technique et de notre prototype à un projet industriel ainsi qu'à un modèle de recherche nous a permis d'évaluer ses capacités de détection. La qualité des résultats obtenus et la généralité de cette méthode nous mènent à penser que les concepts sur lesquels elle se base sont extensibles à d'autres domaines.

4. PROPOSITION D'APPLICATION À L'ARCHÉOLOGIE

Les principes fondamentaux de cette approche semblent être applicables au domaine de l'archéologie. En effet, il semble que la formalisation des connaissances et l'élaboration de règles adéquates puisse permettre la détection automatique d'incohérences au sein de chronologies et d'ensembles d'éléments. Une telle détection, basée sur des règles établies par l'archéologue lui-même, permettrait alors, non pas de sanctionner son travail, mais de lui apporter une aide significative, dans notre cas, pour l'élaboration de chronologies.

Notre étude se base sur la chronologie de Ian Morris⁷. Nous présentons dans les sections qui suivent les possibilités d'application relatives à une chronologie ainsi qu'à un ensemble de poteries. Dans les sections qui suivent, les choix que nous faisons quant aux diverses représentations et incohérences possibles sont purement expérimentaux. Ils ne doivent pas être considérés comme définitifs mais plutôt comme des idées préliminaires susceptibles de changer suivant l'éventuelle application des concepts que nous présentons.

4.1. APPLICATION A UNE CHRONOLOGIE

Une chronologie est définie comme étant une liste d'événements par ordre de dates. Il s'agit en quelque sorte de la représentation formelle d'une succession d'évènements ou de périodes. De ce

⁷ I. MORRIS, *Burial and ancient society : the rise of the Greek city-state*, Cambridge University Press, 1987.

fait, une telle représentation peut être aisément traduite en Prolog, permettant ainsi l'établissement d'un certain nombre de règles caractérisant des incohérences types et fournissant en conséquence un mécanisme de détection desdites incohérences.

4.1.1. Représentation

La chronologie sur laquelle notre étude se base établit le classement d'un ensemble de périodes marquées par un style de poterie particulier. Dans le but de permettre une analyse automatique, nous devons tout d'abord représenter celle-ci en Prolog.

Nous proposons donc de découper la chronologie en nous basant sur les périodes, ou styles, qu'elle définit. Nous considérons qu'un style se caractérise par son nom, une abréviation correspondante, une date de début et de fin, et la région à laquelle il se rattache. Suivant ces considérations, nous proposons donc un prédicat Prolog se composant des champs énumérés précédemment. L'exemple qui suit définit le style « Submycenaean » :

Style('Submycenaean',	<i>Nom du style</i>
'SM',	<i>Abréviation correspondante</i>
-1125,	<i>Date de commencement</i>
-1050,	<i>Date de fin</i>
'Attic'	<i>Région à laquelle le style se rattache</i>
).	

Comme nous pouvons le constater, le prédicat en question est composé de cinq termes étant tous des constantes et ne contient pas de corps. Il s'agit tout simplement d'un fait.

4.1.2. Incohérences

La détection d'incohérences est basée sur les concepts présentés en section 3.2.2. Il nous faut tout d'abord identifier celles que nous voulons être en mesure de détecter. Bien évidemment, le concept d'incohérence reste flou, ce qu'une personne estime incohérent peut être considéré

comme étant tout à fait cohérent par une autre. Il est donc bien entendu que nous énonçons ici des propositions de règles à titre purement indicatif. Quelques incohérences possibles sont les suivantes :

- Chevauchement entre deux styles, caractérisé par le fait qu'un style se termine après le début d'un autre. Nous pouvons, par extension, considérer les cas où un style en chevauche deux autres en commençant avant la fin de l'un et finissant après le début de l'autre.
- Inclusion d'un ou plusieurs styles par un autre, caractérisée par le fait qu'un ou plusieurs styles sont définis entre le début et la fin d'un autre.
- Non-continuité d'un style, caractérisée par le fait qu'aucun style n'est défini avant le début ou après la fin du style considéré.
- Par similitude, les espaces vides, caractérisés par le fait qu'il y a un espace temporel entre la définition de deux styles, cet espace n'étant pas comblé par un autre style.

La figure suivante présente la règle de détection caractérisant un chevauchement :

```
chevauche(S1abrv, S2abrv):-  
    style(_, S1abrv, S1debut, S1fin, S1region),  
    style(_, S2abrv, S2debut, S2fin, S1region),  
  
    S1abrv \= S2abrv,  
  
    S1debut < S2debut,  
    S2debut < S1fin,  
    S1fin < S2fin.
```

La figure suivante présente trois styles faisant partie de la même chronologie illustrant deux des incohérences citées précédemment :

'Late Protoattic',	'Transitional',	'Orientalising',
'LPA',	'Tr',	'O',
-650,	-625,	-700,
-624,	-575,	-575,
'Attic'	'Attic'	'Attic'

Comme nous pouvons le voir, nous avons à faire à :

- Un chevauchement caractérisé par le fait que le style « Late Protoattic » se termine après le début du style « Transitional ». Dans ce cas, l'Archéologue devra déterminer s'il s'agit d'une erreur ou d'un chevauchement intentionnel.
- Une inclusion caractérisée par le fait que le style « Orientalising » englobe les styles « Late Protoattic » et « Transitional ». Dans ce deuxième cas, nous pouvons constater, d'après la chronologie, que cette erreur n'en n'est pas une puisque le style « Orientalising » définit en fait une période plus générale englobant plusieurs autres styles.

4.2. APPLICATION A UN ENSEMBLE D'ÉLÉMENTS

L'application à un ensemble d'éléments se base sur les même principes que l'application à une chronologie. Considérons un ensemble de poteries ; chaque poterie (ou élément) comporte un certain nombre de caractéristiques qui lui sont propres et qui doivent être identifiées de façon à pouvoir la formaliser ainsi que formaliser, par la suite, les incohérences types.

4.2.1. Représentation

Considérons que les champs qui caractérisent un élément soient les suivants : son nom, une clef ou numéro utilisé à des fins de classification, une classe dénotant l'utilité de l'objet considéré, le chantier de fouilles sur lequel il a été trouvé, l'âge qu'il est estimé avoir, le style artistique auquel

il correspond, et la forme qui le caractérise. Cette représentation est, bien sûr, arbitraire et donnée à titre d'exemple mais permet néanmoins d'obtenir un aperçu des possibilités de formalisation.

```

element('Assiette Céramique',      Nom donné à l'objet
        'V-AC-001',                Clef correspondante
        'Vaisselle',               Classe (utilité)
        site_14c,                   Chantier de fouilles
        -770,                       Estimation de l'age
        'MG II',                    Style correspondant
        forme_1a                     Code de la forme
    ).

```

4.2.2. Incohérences

Tout comme pour une chronologie, différentes incohérences sont possibles, parmi elles, nous pouvons identifier les suivantes :

- Éléments dont toutes les caractéristiques sont identiques mais dont les noms diffèrent.
- Éléments présentant les mêmes caractéristiques mais étant rattachés à des styles différents.
- Éléments de mêmes dates rattachés à des styles différents.

Une fois de plus, l'élaboration des règles se fait selon les mêmes principes que précédemment. À titre d'exemple de détection, nous pouvons considérer l'incohérence suivante :

'Assiette Céramique',	'Assiette Céramique',
'V-AC-001',	'V-AC-002',
'Vaisselle',	'Vaisselle',
site_14c,	site_14c,
-770,	-780,
'MG II',	'LG Ia',
forme_1a	forme_1a

Comme nous pouvons le constater, les deux objets présentent une incohérence au niveau de leur style. En effet, ils présentent les mêmes caractéristiques mais sont rattachés à des styles différents. Il s'agit ici d'une erreur, le deuxième élément affichant une date qui ne correspond pas aux dates mentionnées dans la chronologie. Il est à noter que ce type d'erreur peut révéler le fait que l'élément a été mal daté, qu'il s'agit d'une copie, ou tout aussi bien que la chronologie doit être revue.

4.3. APPLICATION CONJUGUEE

Les principes de détection d'incohérences au sein d'une chronologie peuvent être conjugués à ceux de détection d'incohérences au sein d'un ensemble d'éléments. Cela permet la vérification des liens entre un élément et la chronologie à laquelle il se rattache tel que :

- La cohérence entre le style et la date associés à l'élément et les dates de début et fin de ce style dans la chronologie.
- Le chantier de provenance d'un élément et la région correspondant au style attribué.

Nous pouvons ainsi détecter des incohérences telles que le fait que la date d'un élément ne correspond pas aux dates du style qui lui est attribué ou que ce même style ne correspond à aucun style existant dans la chronologie. De même, nous pouvons détecter le fait qu'un style attribué à un élément ne corresponde pas à la région de provenance de l'élément. La figure suivante illustre un exemple d'incohérence entre deux styles et un élément :

'Middle Geometric II', 'MG II', -800, -750, 'Attic'	'Late Geometric Ia', 'LG Ia' , -760, -750, 'Attic'	'Assiette Céramique', 'V-AC-002', 'Vaisselle', site_14c, -780 , 'LG Ia' , forme_1a
---	---	--

Comme nous pouvons le constater, l'objet fait référence à un style ne correspondant pas à la date attribuée, date selon laquelle il devrait faire référence au style de la première colonne.

4.3. RESUME

Nous avons proposé dans cette section quelques possibilités d'application des concepts de résolution automatique, propres à la logique du premier ordre, au domaine de l'archéologie. Nous avons présenté quelques idées de détection d'incohérences au sein d'une chronologie ainsi qu'au sein d'un ensemble d'éléments à caractère archéologique. Nous discuterons des avantages et inconvénients de cette méthode et évoquerons sa pertinence dans la section suivante.

5. CONCLUSION ET RECHERCHE FUTURE

Nous avons présenté les concepts de base de la logique des prédicats du premier ordre appliqués à la résolution automatique de problèmes. Dans le même ordre d'idées, nous avons présenté les principes d'application de cette logique au problème des interactions de services en téléphonie. Se basant sur les résultats concluants obtenus, nous pensons que ces mêmes principes pourraient éventuellement être appliqués à la détection d'incohérences au sein de chronologies et d'un ensemble d'éléments.

L'intérêt de permettre une telle analyse a pour but d'assister l'Archéologue dans l'établissement d'une chronologie. En effet, l'identification et la classification de périodes peuvent s'avérer laborieuses et des incertitudes ou des erreurs peuvent parfois être introduites. En possédant un outil d'analyse automatique rapide, l'Archéologue pourra alors être assisté dans la vérification du modèle sur lequel il travaille.

Le travail de vérification est semi-automatisé. L'algorithme identifie les problèmes potentiels en fonction des règles définies auparavant par l'Archéologue qui devra par la suite vérifier le poids des problèmes soulevés. L'intérêt de cette semi-automatisation est que,

contrairement à un outil totalement automatique qui déciderait seul, le verdict final concernant une incohérence revient ici à l'Archéologue. L'outil de vérification est alors tout simplement une sorte de filtre (on emploie d'ailleurs en téléphonie le terme « incoherences filtering »), suggérant un problème au chercheur et lui laissant le soin de décider de la validité de la suggestion.

Bien évidemment, l'utilisation d'une telle méthode au sein d'une communauté de chercheurs implique nécessairement l'adoption d'un certain nombre de règles devant être acceptées par tous. Cette nécessité peut malheureusement se révéler être un inconvénient dans la mesure où les incohérences caractéristiques peuvent varier d'un modèle à un autre, comme d'un point de vue à un autre.

Nous envisageons dans un futur proche de travailler avec Aurélien BÉNEL en vue de l'intégration des concepts que nous avons présentés au prototype Porphyre. Cette intégration aura pour résultat l'enrichissement de Porphyre, permettant ainsi une détection d'éventuelles incohérences au sein d'une classification, ajoutant donc aux couches de classification syntaxiques et sémantiques une couche de vérification syntaxique semi-automatique.