



HAL
open science

Des outils pour la vérification des Graphes Sémantiques

Mahdi Gueffaz, Sylvain Rampacek, Christophe Nicolle

► **To cite this version:**

Mahdi Gueffaz, Sylvain Rampacek, Christophe Nicolle. Des outils pour la vérification des Graphes Sémantiques. JMFB (la Jeunesse Musulmane de France en Bourgogne), Jun 2010, Dijon, France. pp.24. hal-00620997

HAL Id: hal-00620997

<https://hal.science/hal-00620997v1>

Submitted on 9 Sep 2011

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.

Des outils pour la vérification des Graphes Sémantiques

Mahdi Gueffaz¹, Sylvain Rampacek¹, Christophe Nicolle¹,

¹ LE2I, UMR CNRS 5158,
Université de Bourgogne
21000 Dijon, France
{Mahdi.Gueffaz, Sylvain.Rampacek, Christophe.Nicolle,
}@u-bourgogne.fr

Résumé. Le langage le plus utilisé pour représenter les graphes sémantiques est RDF (standard du W3C). Les graphes RDF sont généralement stockés dans une base de données relationnelle et manipulés en utilisant le langage SQL ou les langages dérivés comme SPARQL. Malheureusement, cette solution, bien adaptée pour les petits graphes RDF n'est pas bien adaptée pour les grands graphes RDF. Pour ces grands graphes RDF, nous proposons une nouvelle approche en utilisant la vérification formelle. Pendant nos recherches, on a développé deux outils RDF2SPIN et RDF2NuSMV qui permettent de convertir les graphes RDF vers un modèle et les vérifier avec l'outil SPIN et l'outil NuSMV respectivement (deux model-checker très puissant) afin de vérifier la cohérence des données décrit dans les grands graphes RDF. Pour illustrer notre proposition, nous avons utilisé des graphiques RDF provenant de fichiers représentant le modèle numérique 3D de bâtiment. Notre objectif final est de vérifier la cohérence de ces fichiers qui sont faites à partir d'une coopération des sources d'information hétérogènes (plombiers, architectes, électriciens...).

Mots clés: Graphes Sémantique (RDF), model-checking, SPIN, NuSMV, logique temporelle.

1 Introduction

Grâce aux efforts de la communauté du W3C¹, les informations disponibles sur le web peuvent être traitées automatiquement par des machines, et non par les humains. L'idée est de rendre le Web intelligent, où les informations ne seront plus stockées mais comprises par les machines afin de fournir aux utilisateurs des réponses pertinentes.

Plusieurs langages ont été développés dans le cadre du Web sémantique et la plupart de ces langages sont basés et utilise la syntaxe XML² [1]. Le OWL³ [2] et RDF⁴ [3] sont les langages les plus importants du web sémantique, ils sont basés sur XML. RDF augmente la facilité de traitement automatique des ressources Web. Le RDF est le premier standard du W3C pour l'enrichissement des ressources sur le Web avec des descriptions détaillées. Les descriptions peuvent être caractéristiques des ressources, telles que l'auteur ou le contenu d'un site web. Ces descriptions sont des métadonnées. Enrichir le Web avec des métadonnées permet le développement de ce qu'on appelle le Web Sémantique. Le RDF est aussi utilisé pour représenter les graphes sémantiques correspondant à une modélisation des connaissances spécifiques.

SPARQL⁵ [4] est le langage standard pour interroger les graphes sémantiques, cette solution n'est pas adaptée aux grands graphes sémantiques. Dans cet article, nous proposons une nouvelle méthode en utilisant la vérification formelle, qui consistent en la transformation de graphes sémantiques dans un modèle et de les vérifier avec des model-

¹ World Wide Web Consortium

² eXtensible Markup Language

³ Ontology Web Language

⁴ Resource Description Framework

⁵ Langage d'interrogation pour RDF

checker. Nous présenterons deux outils RDF2SPIN et RDF2NuSMV qui transforment les graphes sémantiques dans un modèle écrit en langage PROMELA⁶ [5] et NuSMV [6]. Après cette transformation, SPIN et NuSMV vérifient l'exactitude du modèle écrit avec la logique temporelle afin de vérifier la cohérence des données décrivant les graphes sémantiques.

2. Le Web Sémantique

Le Web sémantique vise à organiser et à structurer d'énorme quantité d'informations présentes sur le Net. Il s'agit d'un langage semi-structuré basé sur XML. La figure 1 montre l'une des versions de l'organisation en couches proposées par le W3C. Chaque couche est construite sur les couches en dessous. Ainsi, l'ensemble des couches utilise la syntaxe XML. Cela permet de profiter de toutes les technologies développées autour de XML: XML Schema, les outils d'exploitation des ressources XML (bibliothèques Java, etc), bases de données XML. XML provient du langage SGML⁷, mais contrairement à HTML, la structure et la présentation des documents XML sont conceptuellement séparées. XML est un langage qui utilise des balises comme format de représentation universelle des données. Un document XML contient en même temps les données et les indications sur le rôle que jouent ces données. XML est la pierre angulaire des échanges d'informations sur le Web. Malheureusement, XML est insuffisant pour décrire toute la sémantique nécessaire dans le Web.

⁶ Process Meta Language

⁷ Standard Generalized Markup language

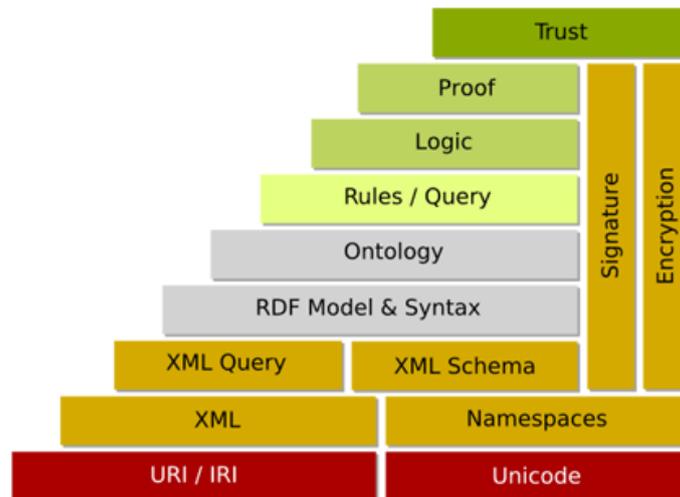


Figure 1. Pile des langages pour le Web Sémantique.

RDF est un langage développé par le W3C pour mettre une couche sémantique au Web [7]. Il permet la connexion des ressources Web à l'aide d'arc dirigé et étiqueté. La structure des documents RDF est complexe. Un document RDF est un ensemble de triplets $\langle \textit{sujet}, \textit{prédicat}, \textit{objet} \rangle$ comme indiqué dans la Figure 2. En outre, le prédicat (également appelé propriété) relie le sujet (ressource) à l'objet (valeur). Ainsi, le sujet et l'objet sont des nœuds du graphe reliés par une arête dirigée du sujet vers l'objet. Les nœuds et les arcs appartiennent à des types "ressources". Une ressource est identifiée par une URI⁸ [8].

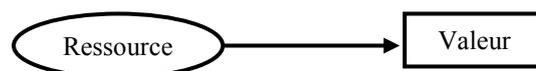


Figure 2. Un triplet RDF.

Les déclarations peuvent également être représentée comme un graphe, les nœuds pour les ressources et les valeurs et les arcs représentent les propriétés. Les ressources sont représentées dans le graphique par des cercles, les propriétés sont représentées par des arcs

⁸ Uniform Resource Identifier

orientés et les valeurs par une boîte (un triangle). Les valeurs peuvent être des ressources si elles sont décrites par des propriétés supplémentaires, par exemple si une valeur est une ressource dans un autre triplet dans ce cas, la valeur est représentée par un cercle.

3. Le Model-checking

Les méthodes formelles [9] offrent un grand potentiel pour une intégration précoce de vérification dans le processus de conception, l'audit technique est plus efficace et réduit le temps de vérification. Les méthodes formelles sont une technique hautement recommandée pour le développement de logiciels. Les méthodes formelles ont conduit au développement de certaines techniques de vérification très prometteuses qui facilitent la détection des défauts. Deux types de méthodes de vérification formelle peuvent être distingués: les méthodes basées sur la preuve du théorème et les méthodes basées sur des modèles. Voir Figure 3.

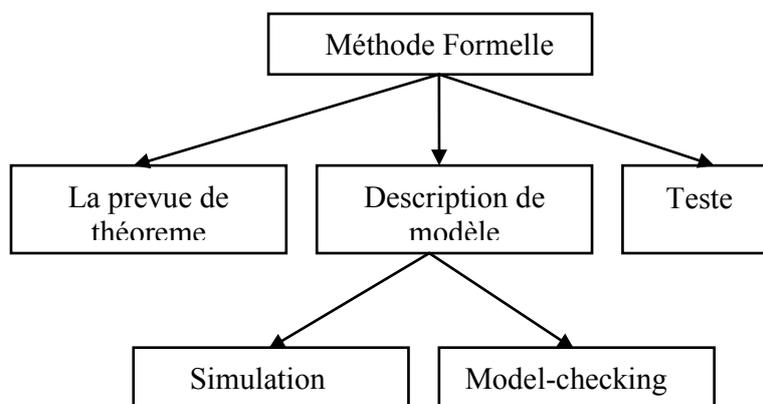


Figure 3. Les méthodes formelles.

Les méthodes fondées sur la preuve du théorème vérifient l'exactitude des systèmes par des propriétés d'une théorie mathématique. Ces propriétés ont fait leurs preuves avec la plus grande

précision possible en utilisant des outils tels que les démonstrateurs et les vérificateurs de preuves. La preuve de théorèmes est également appelée assistants de preuve.

Les méthodes basées sur des modèles décrivant le comportement du système possible d'une manière mathématique précise et sans ambiguïté. Les modèles du système sont accompagnés par des algorithmes qui explorent systématiquement tous les états du modèle du système. Ceci fournit la base pour toute une gamme de techniques de vérification allant de l'exploration exhaustive "model-checking" à des expériences avec un jeu restrictif de scénario dans le modèle "Simulation". La simulation permet à l'utilisateur d'étudier le comportement du système. Il est moins adapté pour détecter les erreurs, car il est difficile de générer tous les scénarios possibles du système et de les simuler. Le model-checker est une technique de vérification qui explore tous les états possibles du système. De cette façon, il peut être démontré qu'un modèle de système donné répond véritablement à une certaine propriété.

Les techniques de vérification formelle comme la simulation et le model-checker sont basées sur la description de modèle qui peut générer tous les états possibles du système, la technique de vérification basé sur les tests est appliqué dans le cas où c'est difficile ou impossible d'obtenir le modèle qui représente le système.

4. Notre approche

Notre approche consiste dans l'utilisation des méthodes formelles. Pour cela, nous développons deux outils: RDF2SPIN et RDF2NuSMV qui transforment les graphes sémantiques en langage PROMELA et NuSMV pour être vérifiés par les model-checker SPIN et NuSMV

respectivement, l'objectif est de vérifier les graphes sémantiques représentant les données dans le secteur du bâtiment.

Nous utilisons SPIN et NuSMV comme outils de vérification des modèles de graphes sémantiques transformés. SPIN est un outil logiciel pour la vérification de modèles. Le système est décrit dans un langage de modélisation appelé PROMELA. NuSMV est l'amélioration du Model-checker SMV [10]. SPIN vérifie l'exactitude des propriétés exprimées en logique temporelle linéaire, par contre NuSMV vérifie les propriétés dans la logique temporelle linéaire et la logique temporelle arborescente.

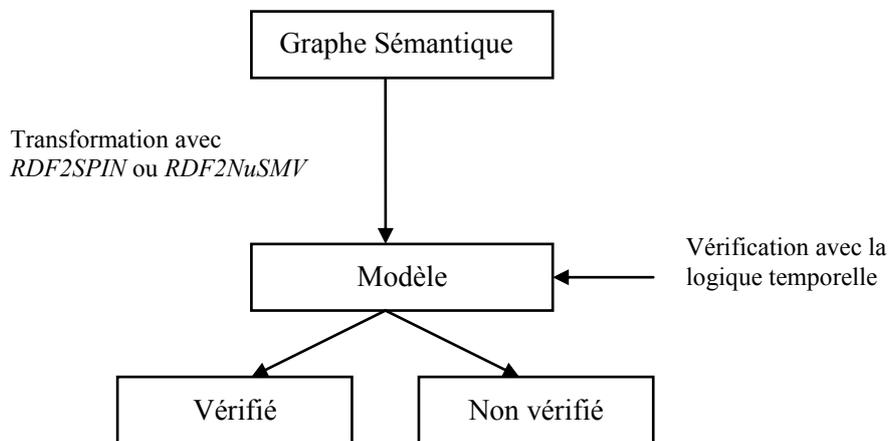


Figure 4. L'architecture de notre approche.

Dans la figure 4, nous avons présenté l'architecture de notre approche. Dans cette architecture, il y a deux phases, la première phase est de transformer le graphe sémantique dans un modèle, cette transformation se fera en trois étapes, la première étape est d'explorer tous le graphe RDF afin d'obtenir la table des triplets, La deuxième étape consiste à déterminer une racine pour le graphe, et la dernière étape sert à écrire le modèle qui représente le graphe sémantique à vérifier dans le langage PROMELA pour *RDF2SPIN* ou dans le

langage NuSMV pour RDF2NuSMV. La deuxième phase consiste à vérifier ce modèle avec les formules de la logique temporelle.

Les graphes RDF considérés sont représentés sous forme de fichiers XML, dans lequel les informations ne sont pas stockées de façon hiérarchique (on l'appelle graphe du point de vue). Ces graphes RDF ne sont pas nécessairement liés, ce qui signifie qu'ils peuvent ne pas avoir un sommet racine à partir duquel tous les sommets de l'autre sont accessibles. La transformation de graphe RDF dans un modèle est articulée en trois étapes: l'exploration du graphe RDF, l'élection du sommet racine, et la génération du modèle du graphe sémantique que l'on veut vérifier.

Nous avons testé nos deux outils RDF2SPIN et RDF2NuSMV sur plusieurs graphes RDF qui représente des bâtiments comme le montre la figure 6, pour calculer le temps nécessaire pour les convertir en modèle.

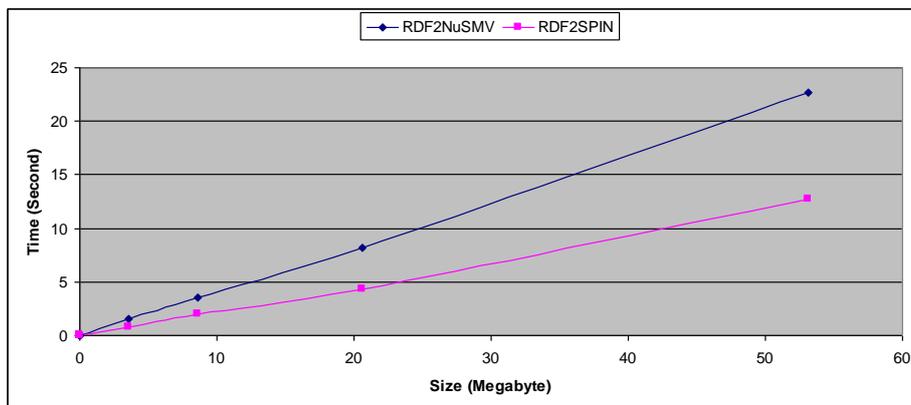


Figure 5. Le temps de conversion.

Nous remarquons dans la figure 5, que le temps de conversion croît d'une façon polynomiale avec la taille des graphes RDF. Les deux outils sont très rapide à convertir de gros graphes, comme on peut le constater, avec un graphe RDF dépassant les 50 Méga octet, on a

presque 15 secondes avec RDF2SPIN et presque 25 secondes pour RDF2NuSMV.

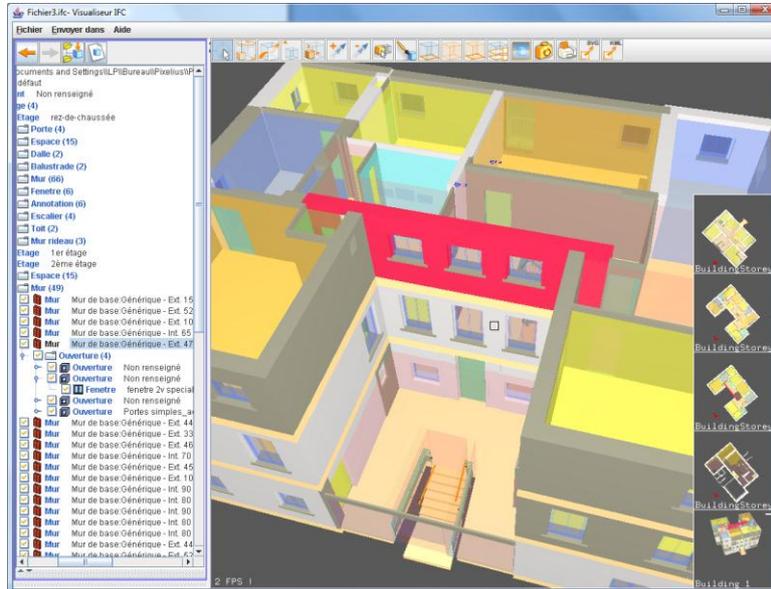


Figure 6. Un bâtiment en 3D.

5. Conclusion

Dans ces travaux de recherches, nous avons présenté deux outils RDF2SPIN et RDF2NuSMV pour convertir les graphes sémantiques en langage PROMELA et en langage NuSMV respectivement afin d'être vérifiés avec les model-checker SPIN et NuSMV. La vérification avec le model-checker sera effectuée avec des formules de la logique temporelle. L'avantage de NuSMV est que la vérification peut être faite avec les deux logiques temporelles LTL et CTL, par contre lorsque on utilise l'outil SPIN, la vérification est seulement avec la logique temporelle LTL. Nous avons comparé les deux outils RDF2SPIN et RDF2NuSMV.

Références

- [1] T. Bray, J. Paoli, C. Sperberg-McQueen. M., Maler, E., Yergeau, F., Cowan, J.: Extensible Markup Language (XML) 1.1 (second edition) W3C recommendation, <http://www.w3.org/TR/2006/REC-xml11-20060816/>. (2006)
- [2] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneijder, L. Andrea Stein, OWL Web Ontology Language Reference, World Wide Web Consortium (W3C), <http://www.w3.org/TR/owl-ref/>, (2004).
- [3] D. Becket, B. McBride: RDF/ XML Syntax Specification (Revised). W3C recommendation. <http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/>. (2004)
- [4] E. Prud'hommeaux, A. Seaborne: SPARQL Query Language for RDF. W3C recommendation, <http://www.w3.org/TR/rdf-sparql-query/> (2008)
- [5] M. Ben-Ari: Principles of the SPIN Model Checker. Springer. ISBN: 978-1-84628-769-5. (2008)
- [6] A. Cimatti, E. Clarke, F. Giunchiglia, M. Roveri: NuSMV: a new symbolic model checker. (2000)
- [7] J. J. C. G. Klyne: Resource Description Framework (rdf): Concepts and abstract syntax. Tech. rep., W3C. (2004)
- [8] V. Bönström, A. Hinze, H. Schweppe: Storing RDF as a graph. Latin American WWW conference, Santiago, Chile. (2003)
- [9] J. P. Katoen: The principiopl of Model Checking. University of Twente. (2002)
- [10] K. L. McMillan: The SMV System. Last update Novembre 2000.