



RDF2SPIN: Mapping Semantic Graphs to SPIN Model Checker

Mahdi Gueffaz, Sylvain Rampacek, Christophe Nicolle

► **To cite this version:**

Mahdi Gueffaz, Sylvain Rampacek, Christophe Nicolle. RDF2SPIN: Mapping Semantic Graphs to SPIN Model Checker. International Conference on Digital Information and Communication Technology and its Applications, Jun 2011, France. pp.519-598. hal-00617983

HAL Id: hal-00617983

<https://hal-univ-bourgogne.archives-ouvertes.fr/hal-00617983>

Submitted on 31 Aug 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.

RDF2SPIN: Mapping Semantic graphs to SPIN Model Checker

Mahdi Gueffaz¹, Sylvain Rampacek¹, Christophe Nicolle¹,

¹ LE2I, UMR CNRS 5158

University of Bourgogne,

21000 Dijon, France

{Mahdi.Gueffaz, Sylvain.Rampacek, Christophe.Nicolle}@u-bourgogne.fr

Abstract. The most frequently used language to represent the semantic graphs is the RDF (W3C standard for meta-modeling). The construction of semantic graphs is a source of numerous errors of interpretation. The processing of large semantic graphs is a limit to the use of semantics in current information systems. The work presented in this paper is part of a new research at the border between two areas: the semantic web and the model checking. For this, we developed a tool, RDF2SPIN, which converts RDF graphs into SPIN language. This conversion aims checking the semantic graphs with the model checker SPIN in order to verify the consistency of the data. To illustrate our proposal we used RDF graphs derived from IFC files. These files represent digital 3D building model. Our final goal is to check the consistency of the IFC files that are made from a cooperation of heterogeneous information sources.

Keywords: Semantic graph, RDF, Model-Checking, Temporal logic, SPIN, IFC, BIM.

1 Introduction

The increasing development of networks and especially the internet has greatly developed the heterogeneous gap between information systems. In glancing over the studies about interoperability of heterogeneous information systems we discover that all works tend to the resolution of semantic heterogeneity problems. Now, the W3C¹ suggest norms to represent the semantic by ontology. Ontology is becoming an inescapable support for information systems interoperability and particularly in the Semantic. Literature now generally agrees on the Gruber's terms to define an ontology: explicit specification of a shared conceptualization of a domain [1]. The physical structure of ontology is a combination of concepts, properties and relationships. This combination is also called a semantic graph.

Several languages have been developed in the context of Semantic Web and most of these languages use XML² as syntax [2]. The OWL³ [3] and RDF⁴ [4] are the most

¹ World Wide Web Consortium

² eXtensible Markup Language

³ Web Ontology Language

important languages of the semantic web, they are based on XML. OWL allows representing the ontology, and it offers large capacity machines performing web content. RDF enhances the ease of automatic processing of Web resources. The RDF (Resource Description Framework) is the first W3C standard for enriching resources on the web with detailed descriptions. The descriptions may be characteristics of resources, such as author or content of a website. These descriptions are metadata. Enriching the Web with metadata allows the development of so-called Semantic Web [5]. The RDF is also used to represent semantic graph corresponding to a specific knowledge modeling. For example in the AEC⁵ projects, some papers used RDF to model knowledge from heterogeneous sources (electricians, plumbers, architects...). In this domain, some models are developed providing a common syntax to represent building objects. The most recent is the IFC⁶ [6] model developed by the International Alliance of Interoperability. The IFC model is a new type of BIM⁷ and requires tools to check the consistency of the heterogeneous data and the impact of the addition of new objects into the building.

As the IFC graphs have a large size, their checking, handling and inspections are a very delicate task. In [7] we have presented a conversion from IFC to RDF. In this paper, we propose a new way using formal verification, which consists in the transformation of semantic graphs into a model and verifying them with a model checker. We developed a tool called “RDF2SPIN” that transforms semantic graphs into a model represented in SPIN [8] language. After this transformation, SPIN verifies the correctness of the model written in PROMELA⁸ language with temporal logic in order to verify the consistency of the data described in the model of the huge semantic graphs.

The rest of this paper is organized as follows. In Section 2 we present an overview of the semantic graphs, especially the structure of the RDF graphs and the model checking. Then, in section 3, we describe the mapping of the semantic graphs into models and our approach is defined in section 4. Finally, we end with the conclusion.

2 An overview of Semantic graph and Model Checking

The RDF is also used to represent semantic graphs corresponding to a specific knowledge modeling. It is a language developed by the W3C to bring a semantic layer to the Web [9]. It allows the connection of the Web resources using directed labeled edges. The structure of the RDF documents is a complex directed labeled graph. An RDF document is a set of triples <subject, predicate, object> as shown in the Figure 1. In addition, the predicate (also called property) connects the subject (resource) to the object (value). Thus, the subject and the object are nodes of the graph connected by an

⁴ Resource Description Framework

⁵ Architecture Engineering Construction

⁶ Industrial Foundation Classes

⁷ Building Information Model

⁸ Process Meta Language

edge directed from the subject towards the object. The nodes and the edges belong to the “resource” types. A resource is identified by an URI⁹ [10, 11].

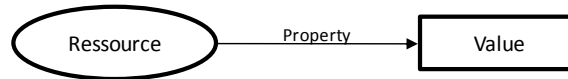


Figure 1. RDF triplet.

The declarations can also be represented as a graph, the nodes as resources and values, and the arcs as properties. The resources are represented in the graph by circles; the properties are represented by directed arcs and the values by a box (a rectangle). Values can be resources if they are described by additional properties. For example, when a value is a resource in another triplet, the value is represented by a circle.

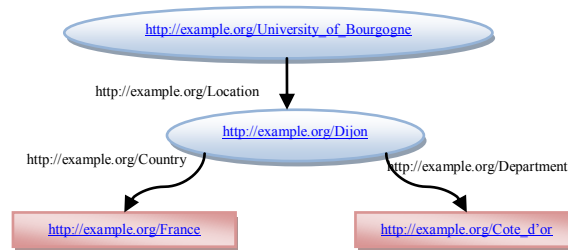


Figure 2. Example of a partial RDF graph.

The RDF graph in the Figure 2 defines a node “University of Bourgogne” located at “Dijon”, having as country “France” and as department “Cote d’Or”. RDF documents can be written in various syntaxes, e.g., N3 [12], N-Triple [13], and RDF/XML. Below, we present the RDF/XML document corresponding to Figure 2.

```

<rdf:Description rdf:about="http://example.org/university of
Bourgogne">
  <ex:Location>
    <rdf:Description rdf:about="http://example.org/Dijon">
      <ex:Country> France</ex:Country>
      <ex:Department>Cote d'or</ex:Department>
    </rdf:Description>
  </ex:Location>
</rdf:Description>
  
```

The model checking [14] described in Figure 3 is a verification technique that explores all possible system states in a brute-force manner. Similar to a computer chess program that checks all possible moves, a model checker, the software tool that performs the model checking, examines all possible system scenarios in a systematic manner. In this way, it can be shown that a given system model truly satisfies a certain property. Even the subtle errors that remain undiscovered using emulation, testing and simulation can potentially be revealed using model checking.

⁹ Uniform Resource Identifier

To make a rigorous verification possible, properties should be described in a precise unambiguous way. It is the temporal logic that is used in order to express these properties. The temporal logic is a form of modal logic that is appropriate to specify relevant properties of the systems. It is basically an extension of traditional propositional logic with operators that refer to the behavior of systems over time.

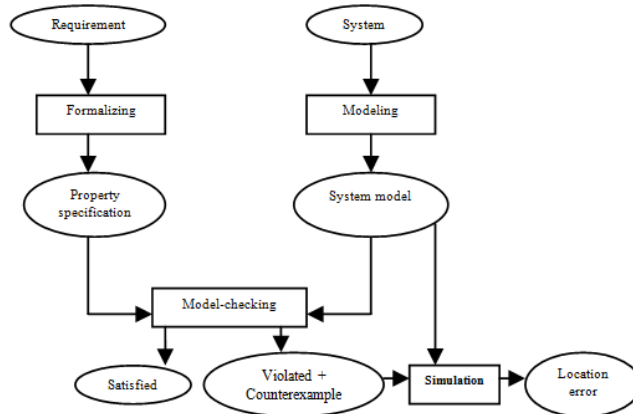


Figure 3. Model Checking approach

The following algorithm explains the way that the model checking works. First we put in the stack all the properties expressed in the temporal logic. All of them are verified one by one in the model and if a property does not satisfy the model, it is whether the model or the property that we must refine. In case of a memory overflow, the model must be reduced. Whereas formal verification techniques such as simulation and model checking are based on model description from which all possible system states can be generated, the test, that is a type of verification technique, is even applicable in cases where it is hard or even impossible to obtain a system model.

```

Algorithm: Model-checking
Begin
While stack  $\neq$  nil do
  P := top (stack);
  while  $\neg$  satisfied (p) then
    Refine the model, or property;
  Else if satisfied (p) then
    P := top (stack);
  Else // out of memory
    Try to reduce the model;
End
End
  
```

3 The mapping

This section speaks about our approach which consists in the transformation of semantic graphs into model in order to verify them with the model-checker. For this, we developed "RDF2SPIN" tool that transform semantic graph into PROMELA [8] language for the Model-checker SPIN.

The RDF graphs considered here are represented as XML verbose files, in which the information is not stored hierarchically (so-called graph point of view). On the one hand, these RDF graphs are not necessarily connected, meaning they may have no root vertex from which all the other vertices are reachable. On the other hand, the N μ SMV language manipulated by the verification tools of N μ SMV always have a root vertex, which corresponds to the initial state of the system whose behavior is represented by the N μ SMV language. The RDF graph transformation into N μ SMV language is articulated in three steps: exploring the RDF graph, determining a root vertex and, final step, generating the Model of the RDF graph. The third step is divided into three sub-steps. First and second one consists in generating two tables (triplets table and resources and values table). The last one consists in producing PROMELA language.

Table of triplets - Going through the RDF graph by graph traversal algorithms, we will create a table consisting of resources, properties and values. In our RDF graph, the resource is a vertex, the property represents the edge and the value is the successor vertex corresponding of the edge of the vertex. The table of triples of RDF graph is useful for the next step to create the table of resources and values.

Table of resources and values - Browsing the table triples seen in the previous step, we attribute for each resource and for each value a unique function. These functions are *proctype* type. We combine all these functions in a table called table of resource and values as you can see in the example in section 3.4.

PROMELA language - In this last step, we will write the PROMELA file corresponding to the RDF graph that we want to check. For this step, we will start by writing the function of the main root of the graph and for each property of the root, we call the function of the corresponding value. We will do the same for all "*resource*" functions defined in the table resources and values. In the other ones, all the function "value" we'll just display their contents. [15]

4 The verification with the Model Checker

As we saw in section 2, the model checker needs properties in order to check the model of semantic graphs. These properties are expressed in temporal logic. The concepts of temporal logic used for the first time by Pnueli [16] in the specification of formal properties are fairly easy to use. The operators are very close in terms of natural language. The formalization in temporal logic is simple enough although this apparent simplicity therefore requires significant expertise. Temporal logic allows representing and reasoning about certain properties of the system, so it is well-suited for the systems verification. There are two main temporal logics, that is linear time and branching time. In linear time temporal logic, each execution of the system is

independently analysed. In this case, a system satisfies a formula f , if f holds along every execution. The branching time combines all possible executions of the system into a single tree. Each path in the tree is a possible representation of the system execution.

This section details our approach which consists in transforming semantic graphs into models in order to be verified by the model-checker. For this, we have developed a tool called “RDF2SPIN” that transforms semantic graphs into SPIN language.

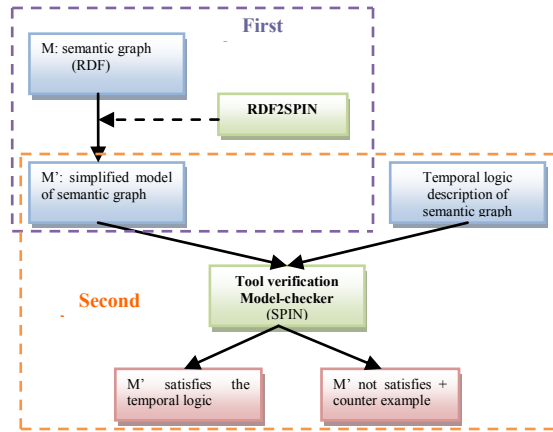


Figure 4. Our architecture.

The architecture of the Figure 4 is divided into two phases. The first phase concerns the transformation of the semantic graph into a model using our tool “RDF2SPIN”, as described in section 3. The second phase concerns the verification of the properties expressed in temporal logic on the model using the model-checker SPIN.

To illustrate our approach, we take an RDF graph represented in the Figure 5 and a temporal logic expressed in the table 1 to verify if the BIM “b1” contains a floor.

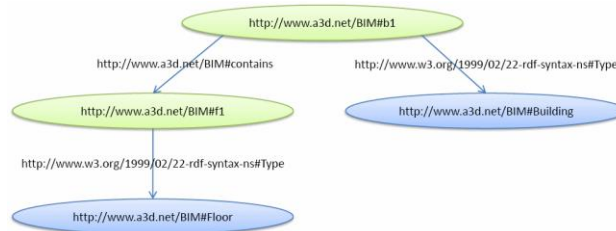


Figure 5. Example of partial RDF graph.

TABLE 1. Temporal logic formula.

Temporal logic	Meaning	Result
Eventually (b1 → Next Next floor)	Is there a floor after two states starting from the state b1	True

We tested several RDF graphs on our tool “RDF2SPIN”, graphs representing buildings as shown in Figure 6, using a machine that runs on a processor with a capacity of 2.4 GHz and 4 GB of RAM, calculating the time of conversion as shown in Figure 7. Note that the RDF2SPIN tool is faster in converting semantic graphs. We have almost 12 seconds for a graph of 53 MB size. The transformation tool follows a polynomial curve.

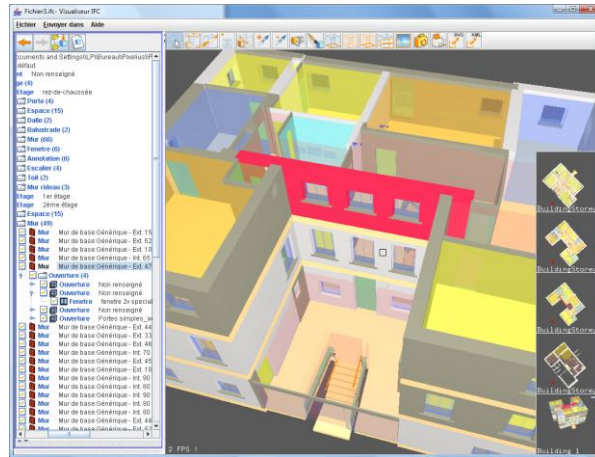


Figure 6. The 3D view of an IFC file.

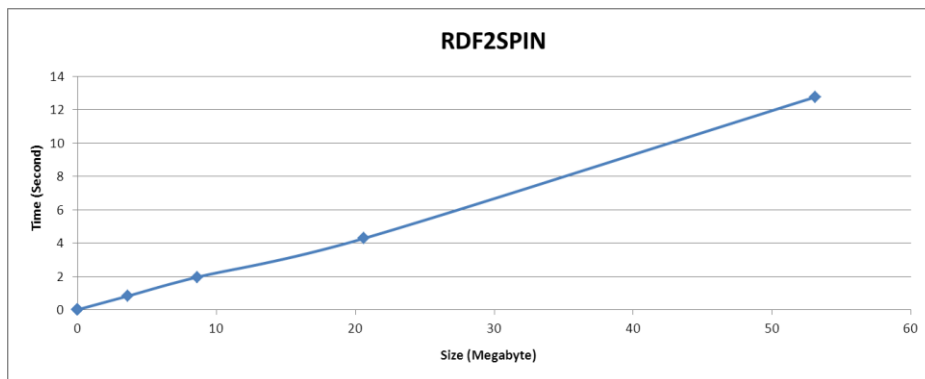


Figure 7. Time conversion of semantic graphs.

5 Conclusion

This paper presents how to transform a semantic graph into a model for verification by using a powerful formal method, that is the “model checking”. Knowing that the model-checker does not understand the semantic graphs, we developed a tool RDF2SPIN to convert them into SPIN language in order to be verified with the temporal logics. This transformation is made for the purpose of classifying large

semantic graphs in order to verify the consistency of IFC files representing 3D building.

References

1. Gruber, T. R.: Toward principles for the design of ontologies used for knowledge sharing. Presented at the Padua workshop on Formal Ontology, later published in *International Journal of Human-Computer Studies*, Vol. 43, Issues 4-5, November 1995, pp. 907-928. March 1993.
2. Bray, T., Paoli, J., Sperberg-McQueen, C. 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)
3. Bechhofer, S., van Harmelen, F., Hendler J., Horrocks, I., McGuinness, D., Patel-Schneijder, P., Andrea Stein, L., OWL Web Ontology Language Reference, World Wide Web Consortium (W3C), <http://www.w3.org/TR/owl-ref/>, (2004).
4. Becket, D., McBride, B.: RDF/ XML Syntax Specification (Revised). W3C recommendation. <http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/>. (2004)
5. Berners-Lee, T., Hendler, J., and Lassila, O. The Semantic Web. *Scientific American*. pp. 34-43. (2001)
6. IFC Model, Industrial Foundation classes, International Alliance for interoperability, <http://www.buildingsmart.com/> (2008)
7. Vanland, R., Nicolle, C., Cruz, C.: IFC and Buildings Lifecycle Management, *Journal of Automation in Construction*, Elsevier, (2008).
8. Ben-Ari, M.: Principles of the SPIN Model Checker. Springer. ISBN: 978-1-84628-769-5. (2008)
9. Klyne, J. J. C. G.: Resource Description Framework (rdf): Concepts and abstract syntax. Tech. rep., W3C. (2004)
10. Bönström, V., Hinze, A., Schweppe, H.: Storing RDF as a graph. Latin American WWW conference, Santiago, Chile. (2003)
11. Berners-Lee, T. W3C recommendation. <http://www.w3.org/DesignIssues/ HTTP-URI>. (2007)
12. Berners-Lee, T., Connolly, D.: Notation3 (N3): A readable RDF syntax. W3C recommendation, <http://www.w3.org/TeamSubmission/n3/>. (2008)
13. Becket, D., McBride, B.: RDF test cases. W3C Working draft. <http://www.w3.org/TR/rdf-testcases/> (2004)
14. Katoen, J. P. The principal of Model Checking. University of Twente. (2002)
15. Gueffaz, M., Rampacek, S., Nicolle, C. SCALESEM: Evaluation of Semantic graph based on Model Checking. The 7th International Conference on Web Information Systems and Technologies, WEBIST. Noordwijkerhout, Hollande, May 2011.
16. Pnueli, A. The temporal logic of programs. In *proc. 18th IEEE Symp. Foundations of Computer Science (FOCS'77)*, Providence, RI, USA. Pages 46-57. (1977)