Science of Computer Programming 95 (2014) 93–111

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

An automated tool for semantic accessing to formal software models

Hai H. Wang[a,*], Danica Damljanovic[b], Jing Sun[c]

[a] School of Engineering and Applied Science, Aston University, Birmingham, UK
[b] Department of Computer Science, University of Sheffield, UK
[c] Department of Computer Science, The University of Auckland, New Zealand

ARTICLE INFO

Article history:
Received 27 March 2013
Received in revised form 24 January 2014
Accepted 17 February 2014
Available online 12 March 2014

Keywords:
Semantic Web
Ontology
Formal method
Software engineering

ABSTRACT

Many software engineers have found that it is difficult to understand, incorporate and use different formal models consistently in the process of software developments, especially for large and complex software systems. This is mainly due to the complex mathematical nature of the formal methods and the lack of tool support. It is highly desirable to have software models and their related software artefacts systematically connected and used collaboratively, rather than in isolation. The success of the Semantic Web, as the next generation of Web technology, can have profound impact on the environment for formal software development. It allows both the software engineers and machines to understand the content of formal models and supports more effective software design in terms of understanding, sharing and reusing in a distributed manner. To realise the full potential of the Semantic Web in formal software development, effectively creating proper semantic metadata for formal software models and their related software artefacts is crucial. This paper proposed a framework that allows users to interconnect the knowledge about formal software models and other related documents using the semantic technology. We first propose a methodology with tool support is proposed to automatically derive ontological metadata from formal software models and semantically describe them. We then develop a Semantic Web environment for representing and sharing formal Z/OZ models. A method with prototype tool is presented to enhance semantic query to software models and other artefacts.

1. Introduction

Formal methods are defined as mathematically based techniques for the specification, development and verification of software and hardware systems [1]. The well-defined semantics and syntax of formal specification languages make them suitable for precisely capturing and formally verifying system requirements.

[*] Corresponding author.
E-mail addresses: h.wang10@aston.ac.uk (H.H. Wang), D.Damljanovic@dcs.shef.ac.uk (D. Damljanovic), j.sun@cs.auckland.ac.nz (J. Sun).

http://dx.doi.org/10.1016/j.scico.2014.02.027

Due to the complex mathematical nature of the formal methods and the lack of tool support, many software engineers have found that it is difficult to understand, incorporate and use different formal models consistently, especially for large and complex systems. It is highly desirable to have different models and their related software artefacts systematically connected and collaboratively used in the process of software development [2], rather than in isolation. For a software system, the knowledge of its application domain, its different models and the related software artefacts should be sharable, linked and reusable in a consistent manner. However, the challenge is to have an open and flexible environment to support such cooperation.

With the advent of the World Wide Web (WWW), its wide accessibility and open distributed nature have provided an important infrastructure for a promising environment for formal software specification and design. By using the Internet, software engineers can share, search, reuse and collaboratively develop software models more effectively. Formal methods such as the CafeOBJ system [3] have included an environment for supporting formal specification over the Internet. Others [4–6,2] proposed to use Web browsers to display and navigate formal Z [7] models. Although the current Web environment has been successful in presenting information on the Internet, the lack of content information and the overburdened use of the display tags have made the efficient retrieval and exchange of information content difficult. The Semantic Web [8], emerged as the next generation of the Web, proposed the idea of having data on the Web defined and linked in a way that it can be understood by machines and used for automation and integration. This is achieved primarily by *annotating* information on the Web using semantic *metadata*. The success of the Semantic Web can have profound impact on the Web environment for formal specifications [9].

By using the Semantic Web technology, we can potentially transform the formal software models and their domain application documentations into a conceptually organised and semantically interlinked knowledge space that incorporates data from multiple software artefacts produced during the process of the software development, e.g., forum posting, requirement documents, source code, configuration files, database and etc. The semantically enriched information can then be used to add novel functionalities to Web-based documentation of the software concerned, providing the software engineers with new and powerful ways to comprehend and reuse software models. The realisation of this vision mainly depends on the fact whether proper *metadata* to semantically annotate formal software models and related software artefacts could be effectively created. Without a systematic methodology and proper tool support, annotating formal software models could be a very tedious and expensive process, which carries a definite risk of failure. There is an urgent need to provide strategies and tools that allow the users to automatically or semi-automatically annotate formal models.

In this paper, we present a methodology and a prototype to derive ontological metadata from formal software models and semantically describe them. Firstly, a framework that allows different formal software models to be described ontologically is defined. We then propose a methodology for interlinking the knowledge about the formal models, the application domain and other related software artefacts by semantically *annotating* them. We aim to automate the mechanical tasks during the annotating process. In this paper, the formal Z/Object-Z(OZ) notations is used to illustrate the approach. Z is a formal specification language based on set theory and predicate logic [10]. It has been widely used in both industry and academic research for the specification and verification of software systems. Object-Z (OZ) is an object-oriented extension of Z [11,12].

This work is part of the *Transitioning Applications to Ontologies (TAO)* project.[1] In TAO, which is in the European Sixth Framework Program, an open source infrastructure to aid transitioning of legacy applications to ontologies, through automatic ontology bootstrapping, semantic content augmentation, and generation of Semantic Web service descriptions, is created. The work is grounded in the tool – *TAO Suite*. All the related materials related to the tools introduced in this paper (e.g., TAO Suite software, source code, manuals, demos and deliverables) are publicly available at http://www.tao-project.eu.

The main contribution of this paper include:

- a Semantic Web environment for representing and sharing formal Z/OZ models,
- a methodology to automatically derive ontological metadata from formal software models,
- a methodology to semantically annotate formal models using ontology, and
- a set of tools to support the whole annotation process.

The remainder of this paper is organised as follows. Section 2 briefly introduces the background material in the areas of the Z/OZ notations and Semantic Web, and the case study used in this paper. Section 3 presents a Semantic Web environment for representing software models. Section 4 introduces a methodology that can bootstrap the domain ontology knowledge model automatic from the formal models and the related software artefacts for an application. Section 5 presents how to automatically generate ontology metadata for representing the knowledge about the formal model and its application domain. Section 6 shows how the derived ontology can be used to create semantically augmented formal software models and related artefacts. Section 7 presents a text-based tool which allows users to query the semantic knowledge of formal software models. The evaluation of the work is presented in Section 8. Section 9 concludes the paper and discusses the future work.

[1] http://www.tao-project.eu.