

Protos: A Cross-Organizational Business Modeling Tool (Demonstration)

Anup K. Kalia, Pankaj R. Telang, Munindar P. Singh

Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA

{akkalia,prtelang,singh}@ncsu.edu

ABSTRACT

Traditional approaches to cross-organizational business modeling use low-level abstractions such as data and control flow. These approaches result in rigid models that over-constrain business execution. Further, because such approaches ignore the underlying business relationships that drive process execution, they lack the notion of business level correctness.

Telang and Singh [5] propose a high-level business modeling approach based upon (social) commitments to address these shortcomings. The high-level model captures the business relationships in terms of commitments between the participants. Telang and Singh [5] develop a method for verifying if a low-level interaction model satisfies a high-level business model. They propose a top-down methodology in which a Business analyst first develops a high-level business model. An IT analyst then develops UML 2.0 sequence diagrams, and verifies if they satisfy the high-level model.

Protos is an Eclipse-based tool that implements Telang and Singh's [5] methodology. It enables: (a) the development of a high-level business model using reusable patterns, (b) the development of UML 2.0 sequence diagrams, as a low-level operational representation, and (c) the automated verification of the UML 2.0 sequence diagrams with respect to the high-level business model.

Categories and Subject Descriptors

I.2.11 [Distributed Artificial Intelligence]: Multiagent systems

Keywords

Software engineering, Commitment, Verification

1. ARCHITECTURE

1.1 Conceptual Architecture

Figure 1 shows the conceptual architecture of Protos. A business analyst starts with a desired cross-organizational scenario. The analyst selects an appropriate set of patterns from a pattern library, and composes them to develop a business model. Computation tree logic (CTL) specifications formalize each of the business pattern in the library. The union of the CTL specifications for all the patterns occurring in a business model constitute a formalization of the model.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

Copyright © 2012, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

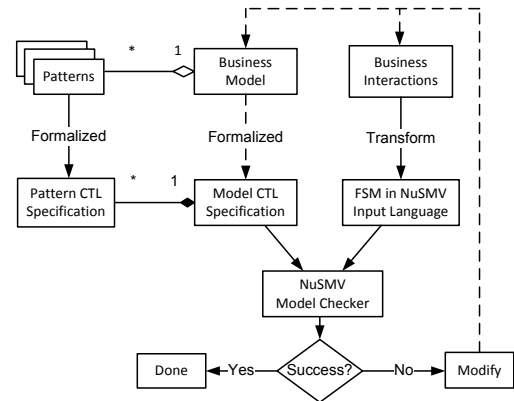


Figure 1: Conceptual architecture (verbatim from [5]).

An IT analyst starts from a business model to develop UML sequence diagrams (operational model). The analyst employs the standard operators of UML 2.0, such as alt(ernate) and opt(ion) with appropriate guards to specify the sequence diagrams. The sequence diagrams are transformed into a finite state machine in the NuSMV input language. The IT analyst runs the NuSMV model checker to verify if the sequence diagrams satisfy the business model. If the model-checker returns success, then the sequence diagrams satisfy the business model. Otherwise, IT or Business analyst inspect the NuSMV counterexample to identify the cause of the failure. They make appropriate changes to either or both of the models and rerun NuSMV.

1.2 Tool Architecture

Figure 2 shows the architecture of the Protos tool. The Protos tool consists of five key components: business modeler, UML sequence diagram modeler, Protos engine, Protos parser, and NuSMV model-checker.

The Business Modeler is implemented as an Eclipse plugin using the Eclipse Modeling Framework (EMF) [1] and the Graphical Modeling Framework (GMF) [2]. The concepts and constraints of the business metamodel are specified in an EMF ECore model. The graphical aspects of the tool such as the concept icons, connectors, and menus are specified in the GMF model. An Eclipse plugin of the tool is then generated using the GMF framework. The business modeler enables saving a business model as an ECore model instance file.

Figure 3 shows a screenshot of the business modeler presenting a model of a real-life Quote-to-Cash business process.

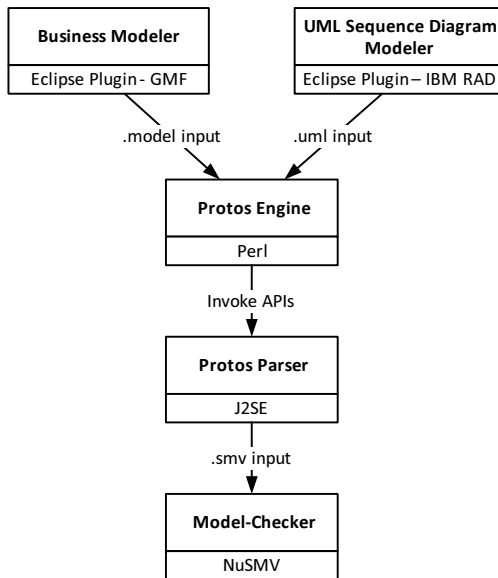


Figure 2: Protos tool architecture.

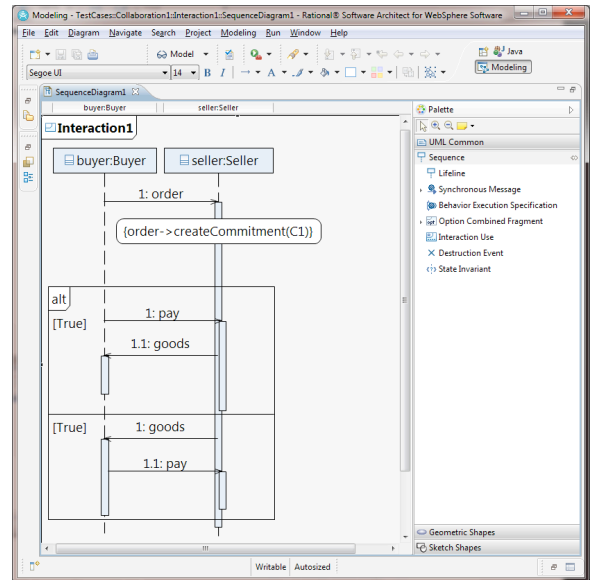


Figure 4: UML sequence diagram modeler.

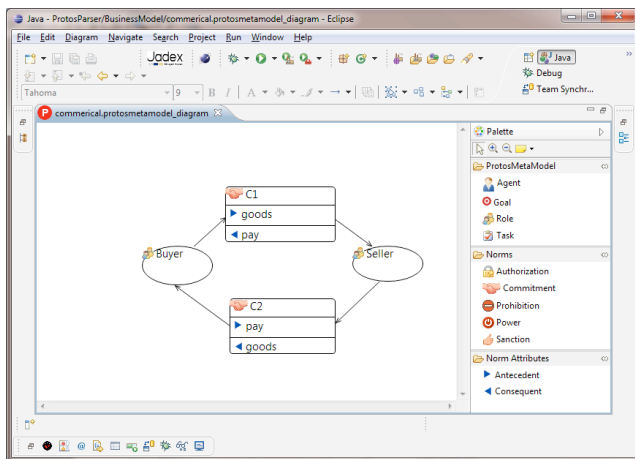


Figure 3: Protos business modeler.

The **UML Sequence Diagram Modeler** is part of IBM's Rational Software Architect (RSA) version 8.0 [3]. IBM RSA is an Eclipse-based tool that supports developing UML 2.0 sequence diagrams. The UML sequence diagram modeler can output a sequence diagram in a standard format.

Figure 4 shows a screenshot of the UML sequence diagram modeler from IBM RSA, presenting a sequence diagram with buyer and seller lifelines. We capture the meaning of a message as an annotation: for example, the accept message means the creation of the commitment $C1 = C(\text{Buyer}, \text{Seller}, \text{goods}, \text{pay})$.

NuSMV [4] is a well-known tool for model-checking. It verifies if a model specified as a finite state machine satisfies a given set of temporal logic formulae. It support computational tree logic (CTL), which we use. Protos employs CTL formulas to formalize a business model. If the finite state machine fails to satisfy a temporal logic formula, NuSMV returns a counterexample which indicates the cause of the failure.

The **Protos engine** is a Perl script that (a) invokes the Protos parser APIs to generate a NuSMV input file, (b) invokes the NuSMV model checker on the generated file, and (c) parses NuSMV's output to generate user-friendly output showing the result.

The **Protos Parser** is the heart of the Protos tool. The parser implements the algorithms from [5] to generate the CTL specifications from the business model ECore file, and to generate finite state machine in NuSMV input language from the UML model file.

2. CONCLUSION AND FUTURE WORK

This paper describes the architecture of Protos, a novel cross-organizational business modeling tool. In future, we plan to enhance usability of the tool by adding features such as drag and drop pattern selection, ability to invoke the Protos Perl script from within Eclipse, and improved user guidance in case of verification errors. We plan to conduct a user-study to evaluate the efficiency and quality of models produced using Protos.

Demonstration Video URL

<http://www.youtube.com/watch?v=7UWb-89w6xE&feature=youtu.be>

Note: Set the video quality to 720p.

3. REFERENCES

- [1] Eclipse. Eclipse modeling project, 2012. <http://www.eclipse.org/modeling/emf>.
- [2] Eclipse. Graphical modeling project, 2012. <http://www.eclipse.org/modeling/gmp>.
- [3] IBM. Ibm rational software architect v8, 2012. <http://www.ibm.com/developerworks/downloads/r/architect>.
- [4] NuSMV. A new symbolic model checker, 2012. <http://nusmv.fbk.eu>.
- [5] Pankaj R. Telang and Munindar P. Singh. Specifying and verifying cross-organizational business models: An agent-oriented approach. *IEEE Transactions on Services Computing*, 4, 2011. In press; available online from IEEE.