TCOZ Approach to Semantic Web Services Design

Jin Song Dong, Yuan Fang Li
School of Computing
National University of Singapore
{dongjs, liyf}@comp.nus.edu.sg

Hai Wang
Medical Informatics Group
Department of Computer Science
University of Manchester
hwang@cs.man.ac.uk

ABSTRACT

Complex Semantic Web (SW) services may have intricate data state, autonomous process behavior and concurrent interactions. The design of such SW service systems requires precise and powerful modelling techniques to capture not only the ontology domain properties but also the services' process behavior and functionalities. In this paper we apply an integrated formal modeling language, Timed Communicating Object Z (TCOZ), to design SW services. Furthermore, the paper presents the development of the systematic translation rules and tools which can automatically extract the SW ontology and services semantic markup from the formal TCOZ design model.

Categories and Subject Descriptors

D.2.4 [**Software Engineering**]: Software/Program Verification—*Formal methods*; I.2.4 [**Knowledge Representation Formalisms and Methods**]: Representation languages

Keywords

Semantic Web, Formal Methods, TCOZ, DAML+OIL, DAML-S

1. INTRODUCTION

Semantic Web is highly distributed, and different parties may have different understandings for the same concept. One important concept in SW services is ontology. Ontology is the basis for constructing common understand through explicitly defined relations. RDFS and DAML+OIL languages can be used to define the ontology. Another important concept in SW services is the semantic markup of services. Semantic markup of the content and capability of Web services – what a service does, how to use it, what its effect will be – will enable easy automation of a variety of reasoning tasks, currently performed manually by human beings, or through arduous hand-coding that enables subsequent automation. DAML-S [1] is such a semantic markup language for Web services.

SW services may have intricate data state, complex process behavior and concurrent interactions. The design of such SW service systems requires precise and powerful modelling techniques to capture not only the ontology domain properties but also the services' process behavior and functionalities. It is desired to have a powerful formal notation to precisely design the Web system.

Timed Communicating Object Z (TCOZ) [4] is a Formal Specification language which builds on the strengths of Object-Z in modelling complex data and state with strength of Timed CSP in modelling real-time concurrency. We believe that TCOZ as a high level

design technique can contribute to the semantic-web-based system development in many ways. In support of this claim, we conduct a SW service case study, i.e., the online talk discovery system, and apply TCOZ to the design stage to demonstrate how TCOZ can be used as high level design language to specify SW services. The following characteristics of many Web services make TCOZ a good candidate to design such a system.

1. A complex Web service system often has both intricate data state and process control aspects. An integrated formal modelling language, like TCOZ, has the strength to model such systems.

2. A Web service agent often provides several kinds of different services concurrently. TCOZ has the multithreaded capabilities to capture that.

3. A complex Web service system is often composed from sub-services. The sub-services may be provided by other agents, which have their own thread of control. It can be modelled by the active objects feature in TCOZ.

4. A Web service includes highly distributed components with various synchronous and asynchronous communications. It can be specified with various TCOZ communication interfaces – channels, sensors and actuators.

5. A Web service like an online hospital or online bank may have critical timing requirements. TCOZ can capture the real-time requirement well.

Furthermore, the paper presents the development of the systematic translation rules and tools to automatically extract the Web ontology and semantic markup for the SW services from the formal TCOZ design model.

2. THE TALK DISCOVERY SYSTEM

In this section, an online talk discovery system is used as an example to demonstrate how TCOZ notation can be applied to the Semantic Web services development. The talk discovery system is a Web portal which can provide not only the talk's information corresponding to the user's profile in terms of his interest and location constraints, but also can further filter the IT related talks based on information about the user's personal schedule, etc. The completed functionality of the ITTALKS system can be found at http://www.ittalks.org/jsp/Controller.jsp.

2.1 Formal design of the talk discovery system

The system involves four different intelligent agents which communicate interactively. They are the user's Calendar agent, MapQuest agent, user's personal agent and the talk discovery agent. Due to the space limitation, we only present partial design model for the Calendar agent.

The Calendar agent maintains a schedule for each eligible user and supplies some related services.
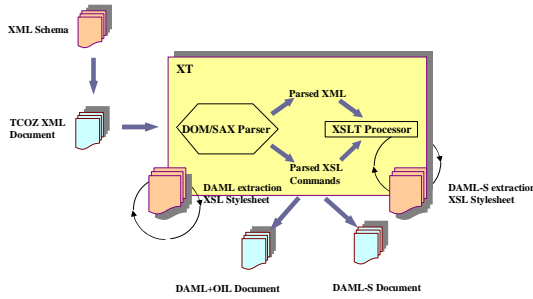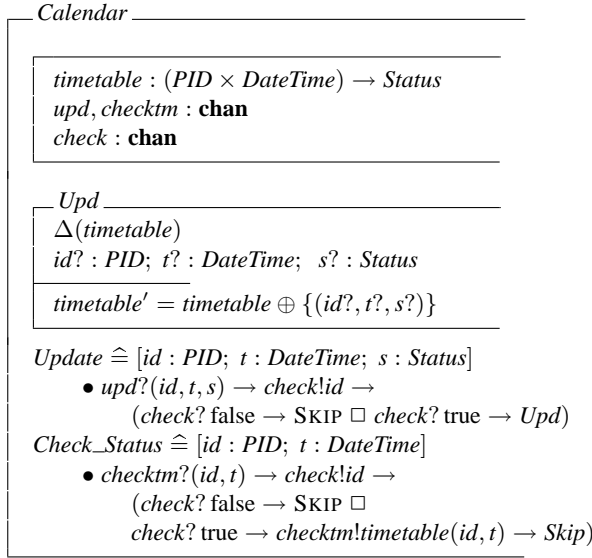
Figure 1: TCOZ DAML+OIL/DAML-S projection

```
┌─ Calendar ─────────────────────────────────────────
│
│   ┌─────────────────────────────────────────────────
│   │ timetable : (PID × DateTime) → Status
│   │ upd, checktm : chan
│   │ check : chan
│   ├─────────────────────────────────────────────────
│   │   ┌─ Upd ────────────────────────────────────────
│   │   │ Δ(timetable)
│   │   │ id? : PID;  t? : DateTime;  s? : Status
│   │   ├──────────────────────────────────────────────
│   │   │ timetable′ = timetable ⊕ {(id?, t?, s?)}
│   │
│   │ Update ≙ [id : PID;  t : DateTime;  s : Status]
│   │     • upd?(id, t, s) → check!id →
│   │         (check? false → SKIP □ check? true → Upd)
│   │ Check_Status ≙ [id : PID;  t : DateTime]
│   │     • checktm?(id, t) → check!id →
│   │         (check? false → SKIP □
│   │         check? true → checktm!timetable(id, t) → Skip)
│
```

Update is used to update the timetable. The operation *Check_Status* is used to check whether a person is available or not for a particular time slot.

3. EXTRACTING WEB ONTOLOGY FROM THE TCOZ MODEL

Designing a clear and consistent ontology is not a trivial job. It is useful to have some tool support in designing the ontology. In this section, we will demonstrate the development of an XSL program to automatically extract the ontology related domain properties from the static aspects of TCOZ formal models (encoded in ZML format [5]). The ontology for the system can be resolved readily from the static parts of TCOZ design documents. In the next section, we will demonstrate tools to automatically extract the semantic markup for services from dynamic aspects of TCOZ formal models. The main process and techniques for the translation are depicted by Figure 1.

A set of translation rules translating from TCOZ model (in ZML) to DAML+OIL ontology is developed. The following are some of the translation rules.

- The given types in the TCOZ model are directly translated into DAML+OIL classes.

- The functions and relations in TCOZ are translated to DAML+OIL ontology property with different domain, range and cardinality constrains.

- The Z state schemas are translated into a DAML+OIL classes. Their attributes are translated into DAML+OIL properties with the schema name as domain DAML+OIL class and the Z type declaration as range DAML+OIL class.

Other translation rules are omitted as the aim of this paper is to demonstrate the approach rather than providing the complete XSL program design.

4. EXTRACTING DAML-S ONTOLOGY FROM THE TCOZ MODEL

In this section, we will demonstrate the development of another XSL program to automatically extract DAML-S information from TCOZ formal models. The semantic markup for the system can be resolved from the TCOZ design documents also. The following are some of the transaction rules we developed.

- Each operation in TCOZ is modelled as a process (AtomicProcess or CompositeProcess) in DAML-S.

- In the case that an operation invokes no other operations, the operation is translated as an AtomicProcess. A precondition appearing in a TCOZ operation schema definition is modelled as *precondition* in the respective service process. A postcondition appearing in a TCOZ operation schema definition is modelled as *effect* in the respective service process.

- An input appearing in a TCOZ operation schema definition is modelled as *input* in the respective service process. An output appearing in a TCOZ operation schema definition is modelled as *output* in the respective service process.

- In the case that an operation calls other operations, the operation is translated as a composite process.

- Each TCOZ process primitive is translated into the proper DAML-S composite process.

Other translation rules for process primitive are omitted due to the limited space.

5. CONCLUSION

In this paper, we demonstrate that TCOZ can be used a high level design language for modeling the SW services ontology and functionalities. Another major contribution of this paper is that it develops systematic transformation rules and tools which can automatically project TCOZ models to DAML+OIL ontology and DAML-S semantic markup.

From a completely different direction, we also recently investigated how RDF and DAML+OIL can be used to build a Semantic Web environment for supporting, extending and integrating various formal specification languages [3].

In summary, there is a clear synergy between Semantic Web and Formal Methods. The investigation between these two paradigms will lead great benefits for both areas. We hope this paper has showed one strong link between the two. The full technical details can be found in [2].

6. REFERENCES

[1] M. Burstein, J. Hobbs, O. Lassila, D. Martin, S. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara, and H. Zeng. Daml service. http://www.daml.org/services/daml-s/2001/05/.

[2] J. D. Dong, Y. F. Li, and H. Wang. A formal approach to semantic web services design. Technical Report TRB2/04, National University of Sinagpore, Feb. 2004.

[3] J. S. Dong, J. Sun, and H. Wang. Semantic Web for Extending and Linking Formalisms. In L.-H. Eriksson and P. A. Lindsay, editors, *Proceedings of Formal Methods Europe: FME'02*, pages 587–606, Copenhagen, Denmark, July 2002. LNCS, Springer-Verlag.

[4] B. Mahony and J. S. Dong. Timed Communicating Object Z. *IEEE Transactions on Software Engineering*, 26(2):150–177, Feb. 2000.

[5] J. Sun, J. S. Dong, J. Liu, and H. Wang. Object-Z Web Environment and Projections to UML. In *WWW-10: 10th International World Wide Web Conference, refereed papers track*, pages 725–734. ACM Press, May 2001.