Model Checking Expressive Web Service Descriptions

Sudhir Agarwal
Institute of Applied Informatics and Formal Description Methods (AIFB),
University of Karlsruhe (TH), Germany.
agarwal@aifb.uni-karlsruhe.de

Abstract

In order to find suitable web services in a large market of web services, automatic support is needed to filter out web services semantically. Existing matchmaking approaches mainly consider only the types of the input and output parameters, which is not sufficient in practical scenarios. In this paper, we present formalisms for modeling functional and non-functional properties of web services and for specifying user goals. We show how expressive web service descriptions can be checked for satisfiability of the user goal.

1. Formal Descriptions of Web Services

In order to meet richer matchmaking requirements, the web services must be described expressively. We differentiate between functional and non-functional properties of web services. Functional properties describe resources and behaviour whereas non-functional properties describe quality of service attributes.

1.1. Modeling Functional Properties

We use the description logic $\mathcal{SHOIN}(\mathcal{D})$ for modeling resources and resource schemas in an interoperable and maschine reasonable way [1]. We specify concrete resources as description logic individuals. The resources can be further classified into sets that can be hierarchically ordered according to the subset relationship.

We use π-calculus for modeling the dynamic behavior so we first introduce it briefly and refer to [2, 3] for details. π-calculus is a formalism for modeling labeled transition systems. The syntax of an agent can be summarized as follows:

$$P \;::=\; \mathbf{0} \mid \overline{y}x.P \mid y(x).P \mid \tau.P \mid P_1 \parallel P_2 \mid P_1 + P_2 \mid \omega?P{:}\,Q \mid (\texttt{new}\ x)P \mid A(y_1,\ldots,y_n)$$

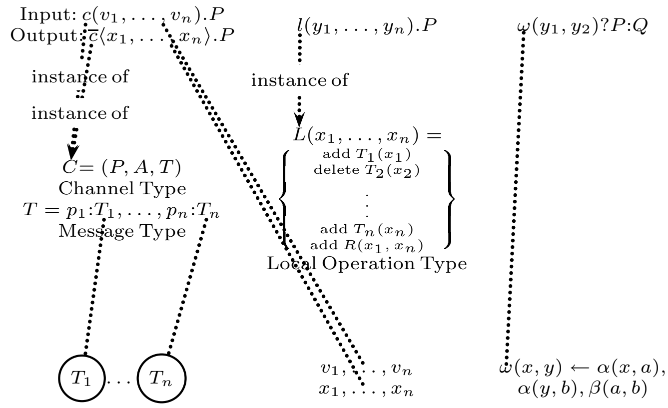

Figure 1. Semantic Description of an Agent

π-calculus is a powerful tool for describing the dynamics of communicating mobile processes. However, π-calculus names, i.e. the objects that are communicated among agent, do not have any structure and any semantics. Figure 1 illustrates our basic idea of resolving this problem by viewing *pi*-calculus names as description logic individuals.

Definition 1 (Communication Channel) *A communication channel c is a tuple (p, a, t), where p is the protocol, a the address and t the type of message that can be transmitted via the channel. A protocol can be something like "http", "phone", "fax", "surface mail" etc.*

Definition 2 (Message Type) *A message type $T = p_1{:}T_1 \ldots, p_n{:}T_n$ is a set of message parts $p_1,\ldots,p_n$ of types $T_1 \ldots, T_n$, which are concepts of a description logic ontology.*

A local operation is a decidable procedure that can add new DL axioms in the knowledge base or remove existing DL axioms from the knowledge base of the agent that executes the local operation. So, we model a local operation as $L(x_1,\ldots,x_n)$ and its effects as a list of changes C, where each element $c \in C$ a parameterized DL A-Box axiom.

Finally, the condition ω in $\omega?Q\colon R$ is a description logic predicate that may be a built-in function or the head of a rule.

1.2. Modeling Non-Functional Properties

The main challenges while describing non-functional properties is their interoperability and their trustworthiness. We use SPKI certificates to model non-functional properties.

Definition 3 (SPKI Certificate) *An SPKI certificate is a tuple* (i, r, p) *digitally signed by* i*, where* i *is the issuing agent,* r *the recipient and* p *is a property. Intuitively, an SPKI certificate* (i, r, p) *means that the agent* i *certifies the agent* r *the property* $p \in \mathcal{P}$*.*

The basic idea to model properties of actors in an interoperable way is to specify the properties that are certified to agents as description logic concepts and to use certificates for issuing properties. Doing this, it becomes possible to use concept subsumption to specify the mapping between different properties. Further, it is possible to certify complex properties with an SPKI certificate while still allowing reasoning over the properties.

2. Model Checking Web Service Descriptions

Now, we present a formalism to specify constraints on web services. We use the temporal logic μ-calculus to build our constraint specfication formalism.

Let Var be an (infinite) set of variables names, typically indicated by $X, Y, Z \ldots$; let $Prop$ be a set of atomic propositions, typically indicated by $P, Q, \ldots$; and let A be a set of action typically indicated by $a, b, \ldots$. The set of formulae (with respect to $(Var, Prop, A)$ is defined as follows:

$$\texttt{true} \mid \texttt{false} \mid P \mid Z \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid [a]\phi \mid \mu Z.\phi$$

In our goal specification language, the set of propositions of an agent correspond to the facts in the knowledge base of the agent. The facts and derived facts in the knowledge base of an agent at some point of time represent the set of propositions that are true at that point of time. We specify propositions of the form $Q_F@Q_A$. The proposition $Q_F@Q_A$ is true if the answer of the query Q_F performed by the agent described by Q_A on his local knowledge base is not empty. For

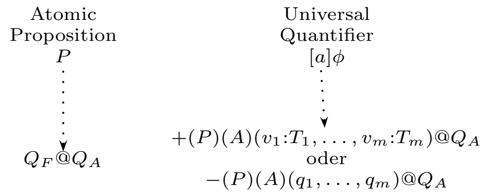

Figure 2. Giving Structure to μ-calculus atoms

input and output actions, we make similar structural extension. We use $+(P)(A)(v_1{:}T_1, \ldots, v_m{:}T_m)@Q_A$ for input actions, which means an an agent satisfying the non-functional properties described in the query Q_A performs an input action that can receive m values of types $T_1, \ldots, T_m$ respectively over a channel of protocoal P at the address A. Similarly, we use $-(P)(A)(q_1, \ldots, q_m)@Q_A$ for an output action, which means an agent satisfying the non-functional properties described in the query Q_A performs an output action that sends m values which are answers of the queries $q_1 \ldots, q_m$ respectively over a channel with protocol P and address A.

3. Conclusion

In this paper, we presented a technique to describe web services semantically by describing involved resources, credentials of participating actors and behavior of the web service including access control policies in a formal and interoperable and unified way. Furthermore, we have presented how expressive user goals can be specified and presented a matchmaking approach based on a decidable, sound and complete model checking algorithm. Furthermore, there exists a prototypical implementation[1,2] of formalisms and algorithms.

Acknowledgements This work was funded by the BMBF project Internetoekonomie and the EU projects NeOn and Xmedia.

References

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory Implemenation and Applications.* Cambridge University Press, 2003.

[2] R. Milner, J. Parrow, and D. Walker. A Calculus of Mobile Processes, Part I+II. *Journal of Information and Computation*, pages 1–87, September 1992.

[3] D. Sangiorgi and D. Walker. *PI-Calculus: A Theory of Mobile Processes.* Cambridge University Press, New York, NY, USA, 2001.

1 https://km.aifb.uni-karlsruhe.de/projects/kaonws/
2 http://sourceforge.net/projects/kasws/