

A Method for Formal Verification of Service Interoperability

Stanislav Pokraev¹, Dick Quartel², Maarten W. A. Steen¹ and Manfred Reichert²

¹ Telematica Instituut, The Netherlands

² University of Twente, The Netherlands

Stanislav.Pokraev@telin.nl, D.A.C.Quartel@ewi.utwente.nl

Maarten.Steen@telin.nl, M.U.Reichert@ewi.utwente.nl

Abstract

Service interoperability is a major obstacle in realizing the SOA vision. Interoperability is the capability of multiple, autonomous and heterogeneous systems to use each other's services effectively. It is about the meaningful sharing of functionality and information that leads to the achievement of a common goal. In this paper we identify requirements for semantic and pragmatic interoperability. We further propose a method for assessing whether a composite system meets these requirements.

1. Introduction

The lack of interoperability forms a major stumbling block in the service integration process. To address this issue a lot of efforts are currently being invested in standardizing service description languages and protocols for service interactions such as WSDL, BPEL and WSCDL. Unfortunately, these efforts only address what we call *syntactic interoperability*. In this paper we identify the requirements for *semantic* and *pragmatic interoperability* and propose a method for verifying whether a composite system is semantically and pragmatically interoperable.

2. Service modeling concepts

First, we present our conceptual framework for service modeling. This framework defines concepts as well as a notation to model interactions between systems from a *communication*, *behavioral* and *information* perspective. The presented concepts are generic in that they can be applied in different application domains and at successive abstraction levels. This helps to limit the number of required concepts. The core concept in our framework, the *interaction* concept, supports a *constraint-oriented style* of service specification. This facilitates the addressing of interoperability requirements by modeling the participation of all interacting entities as separate constraints. This way, interoperability can be verified by reasoning about satisfiability of the logical conjunction of these constraints. The conceptual framework is based on earlier work[10][11].

The *communication perspective* is concerned with modeling the interacting systems and their interconnection structure. For that purpose we introduce two basic concepts: (i) an *entity* models the existence of some system, while abstracting from its properties; (ii) an *interaction point* models the existence of some mechanism that enables interaction between two or more systems, while abstracting from the properties of the mechanism. In general, the interaction mechanism is identified by its location (e.g., the combination of an IP address and port number can be used to identify a TCP/UDP socket).

The *behavioral perspective* is concerned with modeling the behavioral properties of a system, i.e., the activities to be performed by the system as well as the relations among them. For that purpose we introduce four basic concepts: (i) an *action* represents a unit of activity that either occurs (completes) or does not occur (complete) during the execution of a system. Furthermore, an action only represents the activity result (effect) that is established upon completion, and abstracts from the way this result is achieved; (ii) an *interaction* represents a common activity of two or more entities. An interaction can be considered as a refinement of an action, defining the contribution of each entity involved in the interaction. Therefore, an interaction inherits the properties of an action. In addition, an interaction either occurs for all entities that are involved, or does not occur for any of them. In case an interaction occurs, the same result is established for all involved entities; (iii) an *interaction contribution* represents the participation (or responsibility) of an entity that is involved in an interaction. An interaction can only occur if each involved entity can participate. An entity can participate if the causality condition of its interaction contribution is satisfied (see below). In addition, an interaction contribution may define constraints on the possible results that can be established in the interaction. This means that an interaction represents a negotiation among the involved entities, only defining the potential results of the interaction, while abstracting from how they are established; (iv) for an action or interaction contribution, say *a*, a *causality relation* defines the condition that must be satisfied to enable the occurrence of *a*. Three basic conditions are distinguished: *enabling condition b*, which defines that *a* depends on the occurrence of *b*, i.e., *b* must

have occurred before a can occur; **disabling condition** $\neg b$, which defines that a depends on the non-occurrence of b , i.e., b must not have occurred before nor simultaneously with a to allow for the occurrence of a ; **start condition** \surd , which defines that a is allowed to occur from the beginning of the behavior, independent of any other actions or interaction contributions.

Basic conditions can be combined to represent more complex causality conditions. For this we provide the **AND** and **OR** operators, which define that a conjunction and disjunction of conditions must be satisfied, respectively.

The **information perspective** is concerned with modeling the subject domain[13] of the system. For this purpose we provide five basic concepts: (i) an **individual** represents an entity or phenomenon in the subject domain of the system, e.g., the person “John”, the hospital “Saint Joseph” or the city “London”; (ii) a **class** represents an abstract type of entities or phenomena in the subject domain of the system, e.g., “Patient”, “Hospital” or “City”; (iii) a **property** represent possible relations that can exist between entities or phenomena in the system’s subject domain, e.g., “admitted to”, “is a” or “is located in”; (iv) a **result constraint** models a condition on the result of an action or interaction contribution that must be satisfied after the occurrence of the action or interaction contribution; (v) a **causality constraint** models a condition on the results established in causal predecessors (i.e., actions or interaction contributions) that must be satisfied to enable the occurrence of an action or interaction contribution.

Figure 1 shows how information concepts are related to interactions.

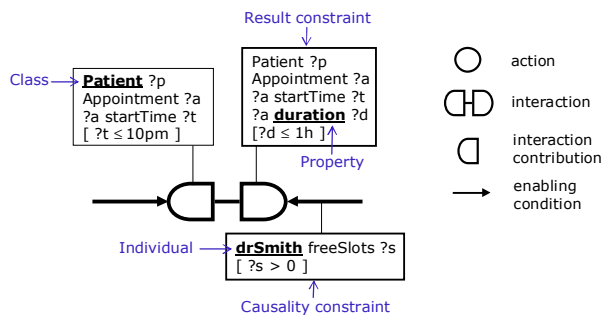


Figure 1. Relating information concepts to an interaction

Putting together the three modeling perspectives yields an **integrated service model** (see Figure 2). A **service** is a set of related interactions between the system and its environment. Our definition of service does not include a sense of direction. It is an interaction that models a common activity of two or more entities in which some results (values) can be established, but abstracts from who

takes the initiative or the direction in which values flow. However, often it is useful to talk about the service that is **offered** by a system without having to specify the constraints of the environment. Likewise, it is also often useful to talk about the service that is **requested** by an entity without making assumptions about the constraints of the service provider. These are two complementary views on a service, which can be obtained by only specifying one entity’s contributions and constraints.

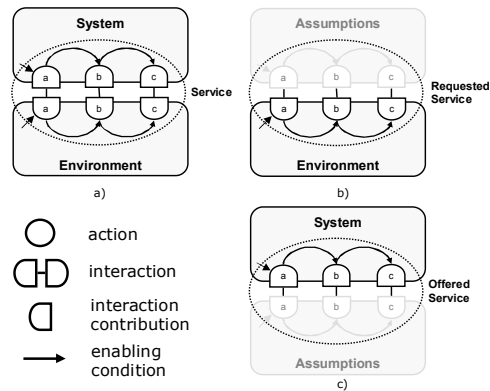


Figure 2. Service model

3. Requirements for interoperability

We distinguish three different levels of interoperability, namely **syntactic**, **semantic** and **pragmatic**.

Syntactic interoperability is concerned with ensuring that data from the exchanged messages are in compatible formats. The message sender encodes data in a message using syntactic rules, specified in some grammar. The message receiver decodes the received message using syntactic rules defined in the same or some other grammar. Syntactic interoperability problems arise when the sender’s encoding rules are incompatible with the receiver’s decoding rules, which leads to (construction of) mismatching message parse trees.

Web Services standards address syntactic interoperability by providing XML-based standards such as SOAP, WSDL and BPEL4WS. Since these technologies address syntactic interoperability sufficiently we focus on semantic and pragmatic interoperability.

Semantic interoperability is concerned with ensuring that the exchanged information has the same meaning for both message sender and receiver. The data in the messages have meaning only when interpreted in terms of the respective subject domain models. However, the message sender does not always know the subject domain model of the message receiver. Depending on its knowledge, the message sender makes assumptions about the subject domain model of the receiver and uses them to construct a message and to communicate it. Semantic

interoperability problems arise when the message sender and receiver have a different *conceptualization* or use a different *representation* of the entity types, properties and values from their subject domains. To address the identified semantic conflicts we define the following requirement:

Requirement 1: A necessary condition for the semantic interoperability of two systems is the existence of a translation function that maps the individuals, classes and properties in the subject domain model of the first system to the respective individuals, classes and properties in the subject domain model of the second system.

Pragmatic interoperability is concerned with ensuring that message sender and receiver share the same expectation about the effect of the exchanged messages.

When a system receives a messages it changes its state, sends a message back to the environment, or both[13]. In most cases, messages sent to the system change or request the system state, and messages sent from the system change or request the state of the environment. That is, the messages are always sent with some *intention* for achieving some desired effect. In most cases the effect is realized not only by a single message but by a number of messages send in some order. Pragmatic interoperability problems arise when the intended effect differs from the actual effect.

Requirement 2: A necessary condition for pragmatic interoperability of a single interaction is that at least one result that satisfies the constraints of all contributing systems can be established.

As said earlier, a service is a set of related interactions between the system and its environment.

Requirement 3: A necessary condition for pragmatic interoperability of a service is that Requirement 2 is met for all of its interactions and they can occur in a causal order, allowed by all participating systems.

4. Assessing service interoperability

To address Requirement 1 we need a method to establish mappings between values, concepts and relations from subject domains of the systems being integrated. This method requires understanding of the meaning of values, concepts and relations from the respective subject domains and cannot be fully automated. However, tools exist that use sophisticated heuristic algorithms to discover possible mappings and provide mechanisms for specifying these mappings. Besides mapping there are two other relevant approaches: alignment and merging of the subject domain models. Alignment is the process of making the subject domain models consistent and coherent with one another while keeping them separate. Merging is the process of creating a single subject domain model that includes the information from all source subject domain models.

In our approach we use OWL-DL[6] as a representation system for our information concepts.

To address Requirement 2 we define an OWL class as the intersection of the admissible results of all participating interaction contributions, and check if this class is satisfiable.

To address Requirement 3 we translate the integrated service model to a corresponding *Coloured Petri Net* (CPN)[7][8] and construct the occurrence graph of that net. We use the constructed graph to check for the existence of a marking in which the results defined by the participating systems can be established. Next, we check if the order of the results establishment meets the causality constraints of the participating systems.

An action in our language maps to a transition in terms of Petri nets. A transition can be executed when all incoming places contain at least one token. On execution it consumes a token from all incoming places and produces a token in all outgoing places. Similar to actions, enabled transitions may execute in parallel.

As said earlier, the occurrence or the result of an action (or interaction) may depend on the result of one or more causal predecessors (actions or interactions). Such dependences can be easily mapped onto guards and bindings in terms of CPNs.

5. State-of-the-art

OWL-S coalition has developed the OWL-S[9] ontology for Web Services, aiming at making Web Service descriptions computer-interpretable, to enable automatic service discovery and invocation, i.e., breaking down interoperability barriers through precise service semantics.

IBM together with LSDIS Lab at University Of Georgia has proposed lightweight approach for adding semantics to Web Service descriptions, WSDL-S[1]. It is based in the work done in METEOR-S[12]. WSDL-S provides a mechanism to annotate WSDL service descriptions by providing extension elements such as input, output, precondition and effect.

The Web Service Modeling Ontology (WSMO)[3] has been proposed by the SDK cluster of EU FP6 projects as an alternative for OWL-S. They argue that OWL-S is only a formalization of WSDL and BPEL4WS, and that true service semantics require a much richer ontology. In addition to the WSMO ontology also a Web Service Modeling Language (WSML)[4] and a Web Service Execution Environment (WSMX)[5] have been defined. The objective of these specifications is to allow automatic service discovery, composition, execution and interoperation in the context of Web and Grid.

The Semantic Web Services Framework[2] is a relatively new initiative, which addresses interoperability by proposing a language and ontology for specifying the

semantics of Web services. The language consists of two parts, namely, a first order logic language for describing web services (SWSL-FOL) and a rule-based language with non-monotonic semantics (SWSL-Rules).

6. Summary and conclusions

The main contribution of this work is a method for formally verifying the interoperability of a composite system to achieve a particular goal. What makes our approach different from related work is that our method is based on a new service modeling framework. This framework provides generic concepts that can be applied in different application domains and at successive abstraction levels. The key concept in our framework (*Interaction*) supports a *constraint-oriented style* of service specification. This style allows service requestors and providers to *explicitly* specify their assumptions about the environment of their systems. This, in turn, enables formal verification of the interoperability of the composite system by checking constraint satisfiability.

Our approach combines the precise, but abstract, definition of the behavior of services and their compositions with a formal definition of the information being exchanged between services. Once we have specified services with this formalism, we are able to apply a combination of a formal logic reasoner and (after a translation to CPNs) a formal behavior analysis tool to verify the semantic and the pragmatic interoperability of a given set of services.

7. Acknowledgements

The presented work has been done in the Freeband Communication project A-Muse (<http://a-muse.freeband.nl>). Freeband Communication (<http://www.freeband.nl>) is sponsored by the Dutch government under contract BSIK 03025.

8. References

- [1] Akkiraju, R., Farrell, J., Miller, J., Nagarajan, M., Schmidt, M.-T., Sheth, A., Verma, K. Web Service Semantics - WSDL-S. W3C Member Submission 7 November 2005, Version 1.0, <http://www.w3.org/Submission/2005/SUBM-WSDL-S-20051107/>
- [2] Battle, S., Bernstein, A., Boley, H., Grosz, B., Gruninger, M., Hull, R., Kifer, M., Martin, D., McIlraith, S., McGuinness, D., Su, J., Tabet, S. Semantic Web Services Framework (SWSF) Overview, W3C Member Submission 9 September 2005, <http://www.w3.org/Submission/SWSF/>
- [3] Bruijn, J. de, Bussler, C., Domingue, J., Fensel, D., Hepp, M., Keller, U., Kifer, M., König-Ries, B., Kopecky, J., Lara, R., Lausen, H., Oren, E., Polleres, A., Roman, D., Scicluna, J., Stollberg, M. Web Service Modeling Ontology (WSMO), W3C Member Submission 3 June 2005, <http://www.w3.org/Submission/WSMO/>
- [4] Bruijn, J. de, Fensel, D., Keller, U., Kifer, M., Lausen, H., Krummenacher, R., Polleres, A., Predoiu, L. Web Service Modeling Language (WSML), W3C Member Submission 3 June 2005, <http://www.w3.org/Submission/WSML/>
- [5] Bussler, C., Cimpian, E., Fensel, D., Gomez, J. M., Haller, A., Haselwanter, T., Kerrigan, M., Mocan, A., Moran, M., Oren, E., Sapkota, B., Toma, I., Viskova, J., Vitvar, T., Zaremba, M., Zaremba, M. Web Service Execution Environment (WSMX), W3C Member Submission 3 June 2005, <http://www.w3.org/Submission/WSMX/>
- [6] Dean, M (eds.), Schreiber, G.(eds.), Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., Stein, L. A. OWL Web Ontology Language Reference, W3C Recommendation 10 February 2004, <http://www.w3.org/TR/owl-ref/>
- [7] Jensen, K. Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1, Basic Concepts. Monographs in Theoretical Computer Science, Springer-Verlag, 1992. ISBN: 3-540-60943-1.
- [8] Jensen, K. Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 2, Analysis Methods. Monographs in Theoretical Computer Science, Springer -Verlag, 1994. ISBN: 3 -540-58276-2
- [9] Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., Sycara, K. OWL-S: Semantic Markup for Web Services W3C Member Submission 22 November 2004, <http://www.w3.org/Submission/OWL-S/>
- [10] Quartel, D.A.C., Dijkman R.M., Sinderen van M. J. Methodological support for service-oriented design with ISDL. In: Proceedings of the 2nd International Conference on Service Oriented Computing (ICSOC 2004), New York City, NY, USA, 2004.
- [11] Quartel, D.A.C., Ferreira Pires, L., Sinderen, van M. J. On Architectural Support for Behaviour Refinement in Distributed Systems Design. In: Journal of integrated design and process science online, 06(01) ISSN 1092-0617.
- [12] Verma, K., Gomadam, K., Sheth, A., Miller, J., Wu, Z. The METEOR-S Approach for Configuring and Executing Dynamic Web Processes", Technical Report . Date: 6-24-05.
- [13] Wieringa, R. J. Design Methods for Reactive Systems: Yourdon, Statemate, and the UML. Morgan Kaufmann, 2003. <http://www.mkp.com/dmrs>