A Webservices Nets Based Framework For Process-Integration

Claus P. Priese

Datenbanken und Informationssysteme FB15, Institut für Informatik; J.W.G. University Frankfurt/Main Germany; Robert-Meyer-Str. 11-15; D-60486 Frankfurt/Main
cspriese@gmx.de

Abstract. This paper beginns with a brief introduction to webservices, followed by other fundamental concepts of business process-oriented systems and petri nets. Then the central definition of Web Services-Systems (WSS) is established. After this, a short presentation of the prototypical UFWPI-Toolset is given. Before the end a short look on related work is given. The paper finishes with a summary and outlook.

1 Introduction

Service oriented architectures (SOAs) provide actual technical structures of choice to standardized methods for integration of software-components distributed over the worldwide web. Especially webservices[1][2] provide a W3C-standardized technical platform for establishing asynchronous interoperation and coupling of distributed software parts. But neither service oriented architechtures in general nor webservices especially say anything about the sense of coupling or the sequence-order of coupled software-components. In this situation workflow- and business-process-concepts offer the missing semantic metaconcepts of how to and why to interconnect services. Some discrete mathematic-logical systems, know as Petri-Nets[3], can be used to formalize the semantics of such process-models. As done for example in [4] or [5].

2 Basics

A lot of the service- and process-concepts mentioned below is also contained in different process-ontologies(for example [6][7]) of several so-called semantic-web-initiatives. This emphasize the significance of these reusable basic concepts.

The definition of Web Services-Systems (WSS) given at the end of this subsection is created to be a formal *lingua franca* for process-oriented composition of webservices.

2.1 Web services

The *web services architecture* (WSA) is a specialised service oriented architecture for web-usage. According to [8] there exists different views on the service model, the message-, service-, resource- and policy-oriented view. In general the functionality of an offered web service from a provider is accessible for a requester through an interface only. Web services are implemented and instanciated as agents, which have a concrete network addresses (endpoints) in so called bindings (i.e. URIs in the WWW). The communication with a service works by message exchange following protocols.

The semantic of a message exchange is the contract between requester and provider. It contains all details not already contained in the WSD. The Webservices specifications does intentionally say nothing about format or content of possible semantics. A semantic can be formal, informal or even oral. This is the point where process-semantics are of interest.

The details of the message exchange capabilities and formats are captured in a webservice description (WSD) using the *webservice description language* (WSDL) given in W3C-specification [9][10][11]. The WSDL uses by default the *extensible markup language schema-language* (XML-Schema) [12][13] for typing of messages and service descrption.

Main conceptual WSDL-elements are the *typing*-mechanism in surrounding *definitions*, *interfaces* as a collection of service-*operations*, *message-references*, *features* and *bindings* to concrete realizations. If operations fulfill some prerequisites described in [9], then they represent a *signature*.

2.2 Process concepts

Processes can be seen as computable objects with input- and output-parameters. They have types and can be composed from subprocesses. Processes are interconnected by routing outputs of predecessors to inputs of succeeding processes. The input-parameters of complex processes are routed to inputs of its subprocesses and the outputs of subprocesses are routed to outputs of the surrounding complex process. This routing of parameters between processes and inside compex processes is called *dataflow*.

The arrangement and sequence of execution of processes is called *controlflow* and follows the rules of control constructs. These control constructs are known from programming languages: sequential execution, branching/forking and afterwards joining with conditions of and, or and xor, further parallel execution, iteration and unordered execution (alleation).

2.3 Semantic and Petri Nets

Petri Nets, introduced from C.A.Petri in 1962 [14], provide mathematical founded concepts for modelling and analysing dynamic systems. Because of the mathematical-logical basis, a provable formal semantic is added to petri net theory,

which is missing in nearly all other approaches for modelling process-oriented systems.

Using formal graph-theorie in conjunction with petri nets also provides a graphical representation of business-processes where states are represented as nodes and activities as transitions. These nodes and transitions are connected by (directed) edges. Nodes can be marked with typed objects. When first complex typed objects for marking nodes, second predicates are associated with transition nodes and third edges are labeled with conditions/filters the petri nets are called to be of *higher* style.

Webservice nets are the new variant of higher petri nets used as internal business process modelling structure in the UFWPI Toolset. Webservice nets are intentionally set up at the WSDL-layer of Webservices. This is done to stay independent from any specialised process-ontology or explicit process modelling language while keeping the ability of mapping onto such a specific modelling language.

Webservice nets are defined this way:

Definition: A *Web Services-System* WSS is an algebraic structure *WSS* = (*S, T, F,* ? *, I_S, KB, TI, M^0*) composed of:

a) (*S, T, F*) is a Net, where *S* is a set of predicates (relationschemata) with variable instances(relations) and *T* is a Set of transitionschemata.

b) The structure ? = (*D, Fkt, Pr*) consists of a set of individuals *D*, a set *Fkt* of functions definend on *D* and a set of predicates *Pr* also defined on *D* with nonvariable instances.

c) The funktion $I_S: S \to W_{el}$ determines for each place $s \in S$ an elementar message-description as the type of the place.

d) KB is the labelling of the edges, where all edges out of F get a set of tuples of variables with fitting cardinality fort the adjazent predicates

e) TI stands for inscription of transitions; TI determines for all transition out of T an inscription consisting of a predicate-logic expression formed over ψ and all the variables appearing in the labelling of the adjazent edges.

f) M^0 is the initial marking, i.e. the predicates(relations) are filled in with sets of constant tuples of fitting caridnality for the predicates.

In a WSDL-description the places are given through types of messages as specified in the "types"-element of a WSDL-definition. The concrete marking of places are message-documents which are variable and can change in conjunction with execution of webservice-implementations. Transitions are the WSDL-operations and the inscriptions of transitions are given by the signatures of the operations. The labeling of edges, connecting the places with transitions, is given through the WSDL-features which are related to message-exchange.

Whenever a transition(operation) becomes *enabled* through the availability of all necessary input-messages(determined in a message-exchange-pattern) it is ready to *fire*. If the enabled transition fires, which means it removes the (input-)message-marks

from the before area and emit (output-)message-marks to the after-area, the state of the represented system changes in a semantically secure and controlled manner.

Further the possibility of reducing the complexity of higher petri nets through abstraction, the modelling of concurrency and distributed communication as well as a rich set of known methods for formal analysis and transformation of petri nets [15][16] gives this approach an inevitable level of semantic security and strenght.

3 UFWPI Toolset

The prototype of the Workflow and Process Integration Toolset, created under instruction and with participation of the author at DBIS/Univ. of Frankfurt/M., pursues to realize a software system providing web-based interorganisational integration of services in accordance with process oriented concepts and a strong mathematical-logical basis for the semantics.

3.1 Architecture

The core architectural design of the UFWPI- Toolset- prototype is divided in two major parts: modelling tools and runtime supporting components.

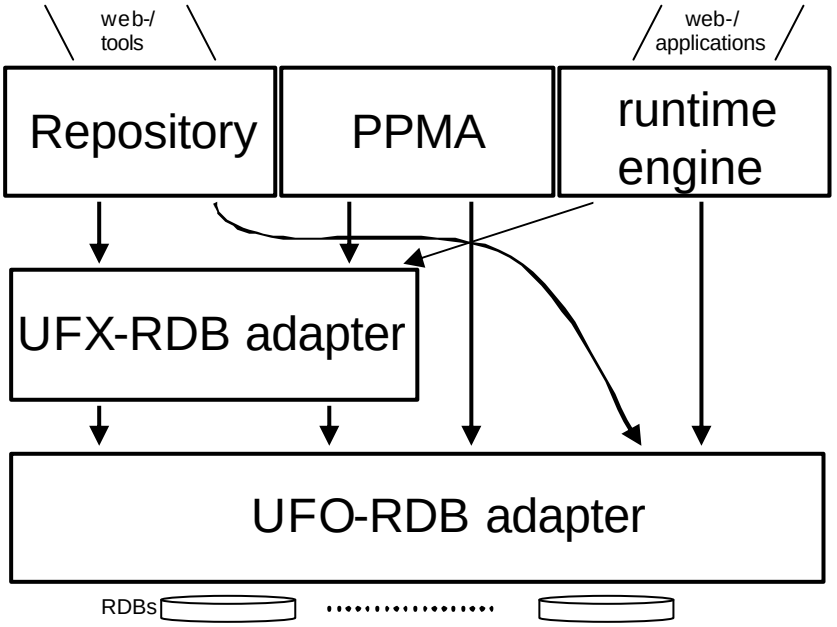

Fig. 1. UFWPI Toolset architectural overview

As shown in figure 1 the basis of the toolset is provided by the object-relational and XML-relational database-adapters UFO-RDB[17] and XFX-RDB. These adapters are

used (denoted by the arrows) to store and retrieve the process-model- and runtime-data of the process repository, the Persistent Processs Modelling Application (PPMA) and the process runtime engine.

3.2 Realization

The protopypical realization enables to read in (i.e. import from external formats), modelling, validating against the formal webservice (petri) net model and creating deployment information or exporting to external formats of webservice-based distributed business-process schemata's.
The prototype of the UF Workflow and Process Integration Toolset is a collection of components realized as (mainly) java-packages. It enables to read and create WSDL-Desriptions, modelling processes by composition with control-constructs and by specifying parameter routing, validating against the formal webservice net model, store process-models in relational database-repositories, creatr runtime deployment information and exporting to external language formats of business-process schemata's. As external representation several alternative vocabularies like OWL-S, BPEL4WS, ebXML's ebBPSS, WPDL/XPDL and other's are possible together with different suitable more or less complex concepts-binding/mapping-mechanisms.

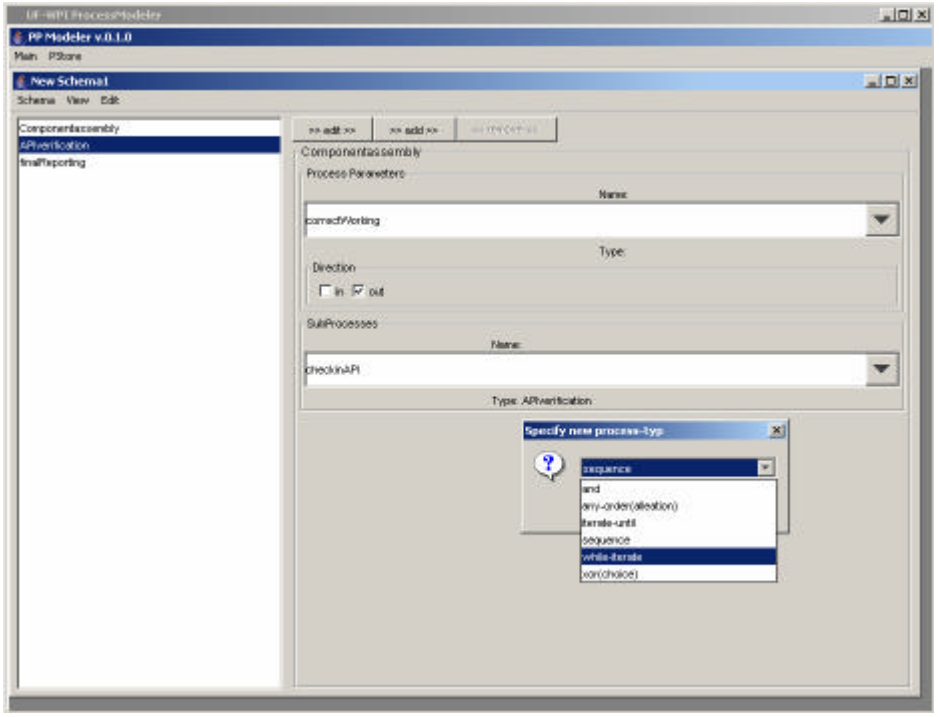

Fig. 2. The persistent process modeling application (PPMA)

4 Conclusion and Outlook

For the challenge of interconnection web-wide distributed services in this paper a process-oriented approach is proposed. To give this appoach a mathematical stable basis a new kind of higher petri nets is introduced: Webservice Nets.

The prototypical UFWPI Toolset implementation uses the database-adapter layers UFX-RDB and UFO-RDB in an appropriate architechture bringing the ideas to work.

Some ideas for additional work in the future could be: to enrich the toolset with goal-planner systems for processes based on logic, heuristics, neural nets or genetic algorithms and further any kind of usefull process-oriented webservices discovery-agents.

1. World Wide Web Consortium: Web Services Architecture Requirements, W3C Working Draft, 11. Februar 2004, abrufbar unter http://www.w3.org/TR/wss-reqs/
2. World Wide Web Consortium: Web Services Architecture (WSA), W3C Working Draft, 11. Februar 2004, abrufbar unter http://www.w3.org/TR/ws-arch
3. Petri, C. A..: Kommunikation mit Automaten; Schriften des Instituts für instrumentelle Mathematik, Bonn, 1962
4. Oberweis, A.: Modellierung und Ausführung von Workflows mit Petri-Netzen, B.G. Teubner-Verlag, Stuttgart und Leipzig, 1996.
5. van der Aalst, W.; van Hee, K.: Workflow Management – Models, Methods and Systems. MIT Press paperback edtion 2004.
6. The OWL-Coalition: Ontology Web Language, available from http://www.daml.org/services/owl-s
7. Web Service Modeling Ontology , available from: http://www.wsmo.org
8. World Wide Web Consortium: Web Services Architecture (WSA), W3C Working Draft, 11. Februar 2004, abrufbar unter http://www.w3.org/TR/ws-arch
9. World Wide Web Consortium: Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, W3C Working Draft, 3. August 2004, abrufbar unter http://www.w3.org/TR/wsdl20
10. World Wide Web Consortium: Web Services Description Language (WSDL) Version 2.0 Part 2: Predefined Extensions, W3C Working Draft, 3. August 2004, abrufbar unter http://www.w3.org/TR/wsdl20-extensions
11. World Wide Web Consortium: Web Services Description Language (WSDL) Version 2.0 Part 3: Bindings, W3C Working Draft, 3. August 2004, abrufbar unter http://www.w3c.org/TR/wsdl20-bindings
12. World Wide Web Consortium: XML Schema Part 1: Structures, W3C Recommendation 2 May 2001, abrufbar unter http://www.w3.org/TR/xmlschema-1/
13. World Wide Web Consortium: XML Schema Part 2: Datatypes, W3C Recommendation 2 May 2001, abrufbar unter http://www.w3.org/TR/xmlschema-2/
14. Petri, C. A..: Kommunikation mit Automaten; Schriften des Instituts für instrumentelle Mathematik, Bonn, 1962
15. Baumgarten, P. : Petri-Netze. 2. Aufl., Spektrum Akademischer Verlag GmbH, Heidelberg Berlin Oxford 1996
16. Priese, L. and Wimmel, H.: Theoretische Informatik Petri-Netze. Springer-Verlag, Berlin Heidelberg, 2003.
17. Priese, Claus P.: Architecture of a reusable and extensible DataBase-Wrapper with rule-set based object-relational schemes. in journal “L'objet” Vol.6 issue N40, Hermes Science Publishing Ltd, Oxford, 2000