

Stanisław Ambroszkiewicz

enTish:

An Approach to Service Description and
Composition

INSTYTUT PODSTAW INFORMATYKI
POLSKIEJ AKADEMII NAUK

Warszawa 2003

Redaktor Naukowy
Instytut Podstaw Informatyki PAN
Prof. dr hab. *Antoni Mazurkiewicz*

Prace do druku opiniował:
Prof. dr hab. inż. Krzysztof Cetnarowicz

Adres autora:
Stanisław Ambroszkiewicz
Instytut Podstaw Informatyki PAN
al. Ordona 21
01-237 Warszawa
oraz
Instytut Informatyki Akademii Podlaskiej
al. Sienkiewicza 51
08-110 Siedlce

© Copyright by Instytut Podstaw Informatyki PAN, 2003

ISBN 83-910948-7-1

The work is dedicated to the memory of Professor Jerzy Łoś

Contents

Preface	iii
1 Developing open distributed systems	1
1.1. Introduction to Open Distributed Systems	2
1.2. Component-Oriented Programming	3
1.3. Software agents and multi-agent systems	4
1.4. What are distributed systems for?	4
1.5. The RPC Model	6
1.6. CORBA	7
1.7. Tuple Space	8
1.8. Service Oriented Computing	9
1.9. Web service composition	11
1.10. Open Grid Service Architecture	14
1.11. Chapter summary	17
2 A New Approach to Developing Open Distributed Systems	19
2.1. Client Server model revised	20
2.2. Agents and implementation of the middleware	25
2.3. Running examples	26
2.4. How to implement the examples?	29
2.5. Chapter summary	31
3 enTish: A New Approach to Service Description and Composition	33
3.1. Beyond the RPC-style and simple document passing	34
3.2. Namespaces for Description Language	35
3.3. XLM-syntax of Description Language: upperEntish	36
3.4. State of agent and service	44
3.5. properEntish as Description Language	47
3.6. Entish and the meaning	49
3.7. Entish and the Semantic Web - a comparison	52
3.8. Message format	55
3.9. Composition protocol entish 1.0	56
3.10. Chapter summary: What is enTish?	60
4 Implementation of the First Running Example	61
4.1. Service operation and state	62

4.2.	Publication of the operation type of FirstClass	63
4.3.	Task	64
4.4.	Agent	65
4.5.	Workflow construction	66
4.6.	Workflow execution	69
4.7.	Transactional semantics	71
5	Implementation of the Second Running Example	73
5.1.	Service operation and state	74
5.2.	Publication of the operation type of AnyBook and BigBank . . .	76
5.3.	Task	77
5.4.	Agent	79
5.5.	Workflow construction	80
5.6.	Workflow execution	87
6	Service Composition Protocol entish 1.0	93
6.1.	Introduction	94
6.2.	Service state and operation type	96
6.3.	Publication of the service operation type to an infoService	97
6.4.	Task and agent state	97
6.5.	Intentions	100
6.6.	Commitments	101
6.7.	Control of workflow formation	105
6.8.	Workflow execution and control	106
6.9.	Task formula is an arbitrary term	108
6.10.	Transactional Semantics	111
7	Abstract Architecture and Prototype Implementation	113
7.1.	enTish abstract architecture	114
7.2.	Conclusion and summary of the work	117
A	XML Sources of the enTish Technology	123
A.1.	XML-schema message.xsd	125
A.2.	XML-schema info.xsd	126
A.3.	XML-schema state.xsd	127
A.4.	XML-schema formula.xsd	129
A.5.	XML-schema definitions.xsd	131
A.6.	XML-schema properEntish.xml	134

Preface

Motto: *Entish - say nothing that isn't worth saying.*

Introduction and motivation of the work

Sometimes there is a tendency to oppose technology to theory and vice versa. As a matter of fact they are complementary. Technology needs theory to justify its solutions and provide sound and solid foundations, and sometimes to explain reasons for its failure. On the other hand theory needs technology to verify its ideas, theorems, and claims; if it does not, then the following citation¹ is in the right place.

As [theory] travels far from its empirical source, or still more, if it is a second and third generation only indirectly inspired by ideas coming from "reality," it is beset with very grave dangers. It becomes more and more purely anesthetizing, more and more purely l'art pour l'art. In other words, at a great distance from its empirical source, or after much "abstract" inbreeding, a [theory] subject is in danger of degeneration. At the inception the style is usually classical; when it shows signs of becoming baroque, then the danger signal is up.

We start with one of the greatest technology challenges: *How to integrate heterogeneous software applications in an open and distributed environment.* It will turn out that the crucial aspect of this challenge is, in fact, one of the most fundamental problems in the theory: *How to construct a generic open language describing data processing with precise machine understandable semantics.*

So that our primary goal is extremely ambitious, it is automatic integration of heterogeneous applications in open and distributed environment. The key requirement for a technology that could realize this goal is to make it possible to automatically discover and use an application in order to perform some given task, and automatically compose multiple applications in order to perform more complex tasks.

One may say that such technologies already exist, e.g., RPC, CORBA, and Web services to mention only the most important ones. However, none of them realizes the requirement. Although these technologies are well established as CORBA, or backed by big vendors like Web services, the dream of integrating business processes in an automatic way is still far from reality. What is the reason for that? Perhaps it is our lack of understanding of the nature of distributed systems.

¹J. von Neuman. The Mathematician. In *In the Works of the Mind*. Chicago, IL: University of Chicago, (1943)

It seems that this very nature is interoperability at several levels. It is not only interoperability at wire and transport levels; actually this was already realized as a working technology (e.g., Internet) quite successfully. It is also interoperability at semantic level, that is, heterogeneous applications must "understand" each other. For this very purpose a language with precise semantics that can be processed (i.e., understood) by applications is needed. This very language is the place where technology and theory meet each other.

The conclusion may be as follows. A great fundamental research is to be done both by the technology as well as by the theory. The research should be done in cooperation, that is, solution proposed by theory should be verified by technology, whereas system proposed by technology should be grounded in theory.

The work presented in the book may be seen as a small contribution to this research. A new approach to distributed systems consisting of heterogeneous components is presented. On the basis of this theoretical approach, an experimental technology for service composition is proposed. The technology is based on the new service description language with machine processable semantics. The work may be summarized in the following way.

There are two general approaches to integration of heterogeneous applications. The first one corresponds to business-to-business point of view, whereas the second one to the client's point of view. The first one is based on the assumption that applications are composed, orchestrated, or choreographed in order to create sophisticated business processes, whereas the second one assumes that applications are composed (typically on the fly) in order to realize clients' requests.

In our work the client's point of view is taken and a new experimental technology for service description and composition in open and distributed environment is proposed. The technology consists of description language called Entish, and composition protocol called entish 1.0. They are based on software agent paradigm.

The description language is the contents language of the messages that are exchanged between heterogeneous applications according to the composition protocol. The language Entish is fully declarative and has clear machine processable semantics.

A task (expressed in Entish) describes the desired static situation to be realized by the composition protocol entish 1.0. Note that a transactional semantics (similar to 3PC) is realized in the protocol.

The syntax of the description language as well as the message format are expressed in XML Schema. The language and the protocol are mere specifications. To prove that the technology does work, the prototype implementation was realized. A demo of the prototype is available as applets via web interfaces starting with <http://www.ipipan.waw.pl/mas/>.

The work was preceded by several papers (see (2; 3; 4; 5; 6; 7)) published in the last four years. Related work was done by SOAP + WSDL + BPEL4WS + (WS-Coordination) + (WS-Transactions), WSCI, BPML, WS-CAF, Semantic Web, DAML-S, Grid services, and Self-Serv. The references to these technologies as well as a short overview is presented in Chapter 1. The reader is cordially

invited to the next chapters for details.

The work is structured in seven chapters and an Appendix where the XML sources of the enTish technology are presented.

In the first chapter, an introduction to distributed systems is presented, as well as a short overview of the most important technologies such as RPC, CORBA, software agents and multi-agent systems, Web services, and Grid services.

Chapter 2 is devoted to presentation of a new theoretical approach to distributed systems; the classical model of Client-Server is revised, and two simple running examples are introduced.

In Chapter 3, the enTish technology is presented. The description language Entish is presented formally whereas the composition protocol is introduced in an intuitive way. Two sections of this chapter are devoted to the semantics of Entish, and to the comparison of Entish to the Semantic Web.

Chapter 4 and Chapter 5 are devoted to formal and detailed presentation of implementation of the two running examples introduced in Chapter 2. These two chapters are essential to understand properly the composition protocol as well as the whole technology that is proposed.

In Chapter 6, the complete specification of the service composition protocol entish 1.0 is presented.

Chapter 7 summarizes the work by presenting an abstract architecture that may be used to implement enTish technology, as well as some details of already realized implementations.

Acknowledgments

Many people have contributed to this work by taking part in long and sometimes exiting and sometimes boring discussions with the author. The author is grateful to Leszek Rozwadowski, Tomasz Nowak, Dariusz Mikulowski, Dariusz Pawluczuk, Michal Calka, Piotr Izdebski and many others.

Special thanks go to Prof. Andrzej Barczak for his encouragement and support from Institute of Informatics of the University of Podlasie.

The author is grateful to the members of the second enTish implementation team (i.e., Dariusz Pawluczuk, Michal Calka, Piotr Izdebski, and Dariusz Mikulowski) for their implementation work and assistance.

The work was done within the framework of KBN project No. 7 T11C 040 20, as a part of the author's research in the Institute of Computer Science of the Polish Academy of Sciences, and in the Institute of Informatics of the University of Podlasie.

The author is also grateful to Steven Willmott for the cooperation within the framework of Service Description and Composition Working Group of Agentcities.NET project.

Special thanks are due to Prof. Andrzej Salwicki for motivations to writing this book.

Categories and Subject Descriptors:

C.2.4 [Computer-Communication Networks]: Distributed systems.

Keywords: Service-Oriented Architecture, service description and composition, middleware, Web services, software agents.