FiXme Note:
DRAFT
version 1.0

Antonio Arfè

BPEL to Java fixme

Bachelor Thesis Project Oct 2009 - fixmeApr 2010

To be evaluated on fixme 99^{th} 2010

Department of Computer Science
Federico II University
Via Claudio,
80100 Naples
ITALY

Faculty of Engineering
Federico II University

Department of Computer Science

TITLE:

BPEL to Java fixme

PROJECT PERIOD:

Bachelor Thesis,
fixme Oct 1^{st} 2009 -
fixme 99^{th} 2010

GROUP MEMBERS:

Antonio Arfè

SUPERVISORS:

fixme Vittorini, Marrone

NUMBER OF COPIES: 7

REPORT PAGES: 7

APPENDIX PAGES: 5

TOTAL PAGES: 8

SYNOPSIS:

fixme.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed lorem ante, porttitor vitae fringilla et, fermentum fermentum nisl. Nunc fringilla, dui viverra iaculis volutpat, augue massa ornare augue, ac sodales erat lectus non risus. Cras rutrum elit sit amet ligula interdum fermentum. Curabitur tincidunt, dui ut hendrerit mattis, massa turpis pretium lectus, eu sagittis turpis nibh id sem. Vestibulum consectetur sapien vel nisi vehicula eget dignissim diam malesuada. Cras consequat, tellus ut malesuada commodo, lacus libero adipiscing ligula, auctor laoreet ligula enim eu turpis. Aenean eget dui eget mauris facilisis faucibus. Curabitur quis metus est, a vulputate orci. Integer mollis metus ac purus condimentum
fixme.

Contents

1 Introduction

Today, the web is quickly expanding and new functionalities and applications, appearing day by day, are heavily increasing its potentialities and ease of use. Actually, one of the reason why this growth is taking place is the possibility to connect or let communicate different existent applications, or better, *services*.

We should make it clear that we intend the word service as the one used by the UDDI Oasis consortium [2], where services are self-contained, modular, business applications that have Internet-oriented and standard-based interfaces. The given definition of services tightly relates to some of the well known web standards used to implement Web Services (WSs) [4].

1.1 Overview of the thesis

2 MDE: Model Driven Engineering

Model-driven engineering (MDE) is a software development methodology which focuses on creating models rather than computing and algorithmic concepts usually addressed in the classical programming approaches. This discipline attempts an abstraction of the benefits and the features provided by the Software Engineering [3], usually creating a domain specific framework for implementing new systems, based on well known and tested concepts, obtained through a careful and detailled analysis of the domain and its actors. The best known MDE initiative is the Object Management Group (OMG) initiative Model-Driven Architecture (MDA), which is a registered trademark of OMG [8].

2.1 Services and composition, SOC and SOA

During the years preceding the large spread of the internet, companies used to create their own software systems, in order to obtain highly customized and specific services, rarely focused on the accessibility by external partners. Today, with the necessity of exchanging information among different companies and businesses, and the push made by the large growth of the Internet, the focus has moved to the integration and the coordination of the existing softwares, namely, the integration of these systems over larger networks. Defining a *service* as a distributed application that exports a view of its functionalities [4], what is needed is the possibility to compose different services together. For example, it might be useful to integrate a service (already available on a net), providing maps of a city, with a web-based service listings thelephone numbers of a given city zone, resulting in a new service showing telephone numbers on the map.

FiXme Note: I might add a picture to better show the example here

The *Service Oriented Computing* (SOC) is the paradigm that attempts to wrap and adapt exisisting applications into new services [4, 5], keeping security and ease of data sharing. The architectural infrastructure to the SOC is called *Service Oriented Architecture* (SOA). The idea behind the SOA is to describe, publish and make available web services (generally speaking, a combination of services) to requestors from multiple business domains.

it is split up into provider, broker and requestor [1]

- Who is going to deal with the logic and flow of the composition?
- Two main approaches: Orchestration and Choreography, where is the difference [6]
- The concept of Workflow management [7] (picture) and why BPEL is good at it.

2.2 BPEL, its drawbacks and the use of WSs in a lightweight scenario

- BPEL, the de-facto standard concerning web services composition.
- Overview and brief review
- What do we use of it, the whole language or just a part?
- Problems: It works over Distributed Systems, engine powerful but heavy, DS not always available.

- That's our question
- Sometimes we might need services composition working in less powerful environments than distributed systems. Examples?
- But we don't want to create a new application from scratch neither...(change hardware?).

2.3 Other works and proposed approach

- Other similar works: don't know any yet.
- Our attempt is to use the M2T techniques to obtain from a BPEL WSs composition a JavaRMI equivalent application
- Where the limits of our solution are (BPEL subset? Lack of heavy test case? Doesn't work at all?...)
- Where future work might be leaded

3 The methodological approach

- Step by step
- Phase 1
- Phase 2
- ...

4 Tools and Techniques

4.1 The Model Driven Architecture (MDA)

- What it is
- Why we use it, why is good (because it is generic, Architecture independent, Reuse)
- How we use it: From BPEL processes to Java RMI processes

4.2 Java RMI

- Overview, features. - Why? it can run everywhere, lightweight scenarios

4.3 Jet

4.4 EMF and Open Architecture Ware (Eclipse)

4.5 other...

5 Detailed Architecture

6 Implementation

7 Discussion

8 Conclusion

References

[1] P. P. Barbara Pernici, *Un'introduzione ragionata al mondo dei web service*, Mondo Digitale - Upgrade, (2004).

[2] O. Consortium, *Uddi xml.org.* [Online; accessed April-2010].

[3] G. D. Lorenzo, *Un Approccio all'Ingegneria dei Modelli per Sistemi di Elaborazione Critici e Complessi*, PhD thesis, Università Federico II, Napoli, Italy.

[4] ——, *Methodologies, architectures and tools for automated service composition in SOA*, PhD thesis, Università Federico II, Napoli, Italy, November 2008.

[5] M. P. Papazoglou, *Service-oriented computing: Concepts, characteristics and directions*, in WISE, 2003, pp. 3–12.

[6] C. Peltz, *Web services orchestration and choreography*, IEEE Computer, 36 (2003), pp. 46–52.

[7] W. M. P. van der Aalst, *The application of petri nets to workflow management*, Journal of Circuits, Systems, and Computers, 8 (1998), pp. 21–66.

[8] Wikipedia, *Model driven engineering.* [Online; accessed April-2010].

List of Tables

List of Figures