

A UML Profile For Service Oriented Architectures

Mr. Rafik Amir
0020105410035

American Univerity in Cairo
raamir@aucegypt.edu

Dr. Amir Zeid
0020101118516

American Univerity in Cairo
azeid@aucegypt.edu

ABSTRACT

Service Oriented Computing is the new paradigm for Distributed computing and e-business processing that is changing the way software applications are designed, architected, delivered and consumed. Services are autonomous platform-independent computational elements that can be described, published, discovered, orchestrated and programmed using standard protocols for the purpose of building agile networks of collaborating business applications distributed within and across organizational boundaries. Engineering and modeling service-oriented architectures need extensions to existing modeling techniques and methodologies. In this paper, we propose a UML profile for service-oriented architectures.

Categories and Subject Descriptors

I.6.5 [Model Development]: UML Profiles.

General Terms

Design

Keywords

UML Profiles, Service Oriented Architectures, Software engineering

1. INTRODUCTION

Service-oriented architectures can be considered the natural evolution of component based development. In recent years, web-services have become a field of interest for both academia and industry. For an industry to gain momentum and evolve as a scientific discipline, software engineering best practices have to be applied. In this poster we illustrate our efforts in developing a UML profile for service-oriented architectures.

2. Motivation

The building of a solution with a Service Oriented Architecture will issue the problem of the standard modeling of this solution; so normally any architect will tend to model his solution as components and map the SOA components into object oriented components and he must take into consideration all the constant design issues of the SOA. So in order to reduce this whole work and to standardize it, we present a solution for the modeling of any type of SOA with the standard UML modeling notation base on the architectural models defined by the W3C web services group [2].

Copyright is held by the author/owner(s).
OOPSLA'04, Oct. 24–28, 2004, Vancouver, British Columbia, Canada.
ACM 1-58113-833-4/04/0010.

3. Background

3.1 UML

OMG has published the UML for the Object Oriented Analysis and Design since 1997 offering unified modeling notations for analysts and designers. Since then most of the software have adopted this standard. OMG has published also with the UML an Extension mechanism for the notations such that any organization can easily evolve in an agile and responsive manner that would allow the definition of new notations that will serve a new domain of interest. This extension mechanism is called the UML Profiles [1].

3.2 Service Oriented Architectures

Service Oriented Architecture (SOA) is an architecture composed of loosely coupled discrete functions. Each function can be considered an independent service that can be used in many scenarios. The SOA is not a new architecture but it has been the talk of the hour because of the emergence of the web services technology.

4. SOA Profiles

Our SOA UML profiles consist of five profiles:

- The resource profile which is intended for modeling any resource by identifying a main key of the resource (identifier) (i.e. URI or QName).
- The service profile that extends the resource profile describes how to model a service with a Controller that controls service tasks and how to package all these elements in a service component. It also describes how to compose and model the discovery service which is kind of a service component also.
- The message profile describes how each message exchanged by the service should be modeled as component identifying the contents of the message and its addresses such as HTTP or SMTP address. It also models the Message transport component that represents the network layer protocol component as HTTP or SMTP.
- The service policy profile describes the modeling of the policies that should be applied to the service usage either as an obligation of who to do the task or as a permission requested by other party to use the service.
- The agent profile shows how to introduce the agents dealing with the service into your model, since they are of a great importance when the services come into action.

4.1 Resource Profile

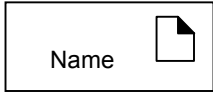
Stereotype	Base Class	Notation
Resource	Component	<<resource>>
Identifier	Class	<<identifier>>
Resource Interface	Interface	<<resourceInterface>>
Resource Description	Artifact	

Table 1. Resource Profile

4.2 Service Profile

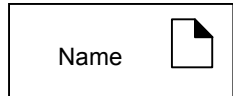
Stereotype	Base Class	Notation
Service	Component	<<service>>
Service Task	Component	<<serviceTask>>
Service Manager	Class	<<serviceManager>>
Service Interface	Interface	<<serviceInterface>>
Service Description	resourceDescription	
Service Task Interface	Component	<<serviceTaskInterface>>
Discovery Service	Service	<<discoveryService>>
Discovery Service Interface	serviceInterface	<<discoveryServiceInterface>>
Service Template	Class	<<serviceTemplate>>
Publish	Operation	Publish
Search	Operation	Search

Table 2. Service Profile

4.3 Message Profile

Stereotype	Base Class	Notation
Message Transport	Component	<<messageTransport>>
Message Interface	Interface	<<messageInterface>>
Message Transport Interface	Interface	<<messageTransportInterface>>
Message Controller	Class	<<messageController>>

Controller		
Message	Component	<<message>>
Address	Class	<<address>>
Envelope	Class	<<envelope>>
Body	Class	<<body>>
Header	Class	<<header>>

Table 3. Message Profile

4.4 Service Policy Profile

Stereotype	Base Class	Notation
Audit Guard	Class	<<auditGuard>>
Permission Guard	Class	<<permissionGuard>>
Policy	Component	<<policy>>
Permission	Policy	<<permission>>
Obligation	Policy	<<obligation>>
Policy Interface	Interface	<<policyInterface>>

Table 4. Service Policy Profile

4.5 Agent Profile

Stereotype	Base Class	Notation
Agent	Component	<<agent>>
Agent Information	Class	<<agentInformation>>
Agent Manager	Class	<<agentManager>>
Requester	Agent	<<requester>>
Provider	Agent	<<provider>>
Requester Manager	agentManager	<<requesterManager>>
Provider Manager	agentManager	<<providerManager>>

Table 5. Agent Profile

5. Conclusion

The UML Profiles presented in this paper is based on the newly presented web services architecture models. We have mapped these architectures into UML profiles presenting the notations and the stereotypes. According to these profiles we will be able to model any SOA from the interacting agents to the services and the messages that are being transported between them.

6. REFERENCES

- [1] Object Management Group UML 2.0 SuperStructure Final Adopted Specification
<http://www.omg.org/cgi-bin/doc?ptc/2003-08-02>
- [2] Web Service Architecture
<http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>