Dunelm

Services Limited

Consultants in Applied Research & Development for
Information & Communications Technology

Services, Service-oriented Architecture & Web Services:

Faster and More Reliable Design using the Unified Modelling Language

2007 © Dunelm Services Limited

Services, Service-oriented Architecture & Web Services

Dr Colin Smythe
Dunelm Services Limited
(colin@dunelm.com)

1

Notes

Aim & Objectives

- Aim - to explain how Internet-based services, realised as Web Services, can be designed and created using the Unified Modelling Language
- Objectives
  - Explain how UML is used to define services
  - Explain how the UML is transformed to a Web Service
  - Present best practices for implementing services as Web Services

2

Notes

Contents

- Service-oriented Architecture (SoA)
- Model Driven Architecture (MDA)
- Web Services
- Unified Modelling Language (UML)
- Profiling UML for service specification
- Workflow for Service design and realization
- What's Next

3

Notes

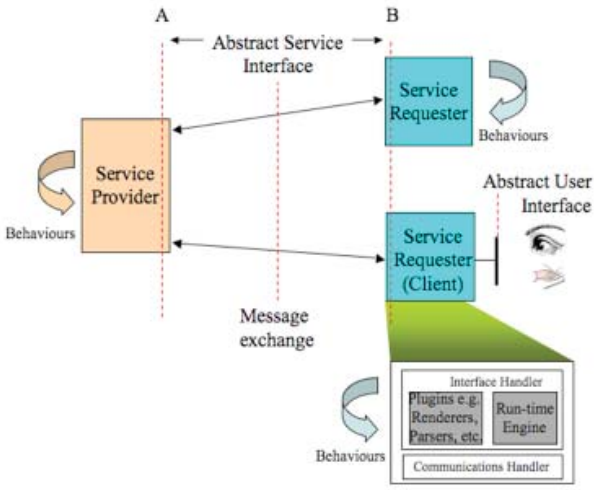
Services

- **Features of a service**
  - **Make it easier to build and maintain complex systems**
  - **Provider(s) & Consumer(s)**
  - **Separation of behaviour from implementation**
    - Functionally cohesive operations
    - Loose coupling
- **Corresponding issues**
  - **Identifying all aspects of a behaviour i.e. error conditions**
  - **Interfaces and protocols**
  - **Combining services**
  - **Language dependent features of the service definition**

4

Notes

Architecture of a Service



5

Notes

Interfaces & Protocols

- **Interfaces**
  - Application-centric
  - Interface contract
  - Realised as language-dependent API(s)
  - Exception handling
  - Driver-oriented approach ?
- **Protocols**
  - Infrastructure-centric
  - Message exchange
  - Realised as Web Service ?
  - Standard distributed systems implementation issues
- **Interoperability options**
  - Interface only, Protocol only, Interface & Protocol combined
  - RPC or object/document/data exchange

6

Notes

Service-oriented Architecture (SoA)

- Building flexible systems from services
  - Not just Internet/Web-based
- Separation of:
  - Domain-specific services
  - Common services
    - Authentication, Authorization, Access Control, etc.
  - Infrastructure from applications
    - Use a common infrastructure across every domain
- Web Services is one way to realise SoA

7

Notes

Using Web Services (Web 2.0)

- Based upon the World Wide Web (WWW) infrastructure (HTTP and URI)
  - The Internet is DNS and IP
- Non-optimal performance
  - Fat pipes and fast processors
  - Verbose data and deep layered protocols
- Minimal additional infrastructure cost
- Constrained by tool support capabilities e.g. .NET Visual Studio

8

Notes

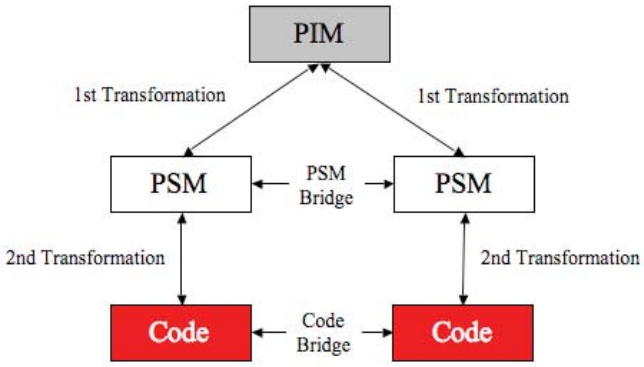
Model Driven Architecture (MDA)

- Specification, design, implementation and testing through model(s)
- Model -> Implementation
  - Correctness maintaining Transformation(s)
  - Automated using transformation rules
- Unified Modelling Language (UML 2.0)
  - Can be profiled for domain-specific modelling
  - Established representational syntax and semantics
  - Wide range of available tools

9

Notes

Models & Transformations



PIM = Platform Independent Model
PSM = Platform Specific Model

10

Notes

Issues for MDA

- Creation of the model
- Verification but not validation
- Language for the PIM and PSM
- Transformation definition language
- Availability of tools and tool interoperability

11

Notes

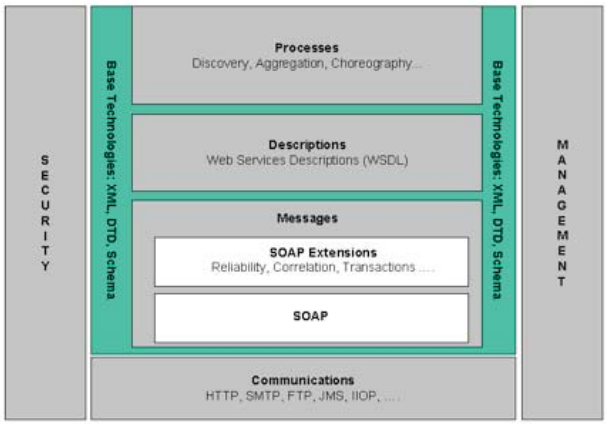
Web Services

- **Objective**
  - Support the creation of services to exchange data over the WWW infrastructure
- **Core Features**
  - XML data
  - SOAP messages
  - HTTP transport
  - Web Services Description Language (WSDL)
- **Extensible Features**
  - Attachments
  - Security
  - Addressing
  - WS-*
    - Reliable messaging
    - Transactions
    - Policy
    - Resource Framework
    - Choreography
    - etc.

12

Notes

The Web Services Stack

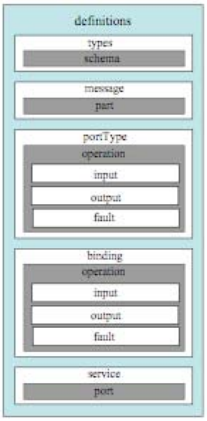


13

Notes

Web Services Description Language

- **WSDL v1.1**
  - Service(s)
  - Port(s) - [Endpoint(s) in WSDL v2.0]
  - Binding(s)
  - Operation(s)
  - PortType(s) - [Interface(s) in WSDL v2.0]
  - Message(s)
  - Types
- **Other features**
  - Extensible - using name-spacing
  - Supports SOAP, HTTP GET, HTTP POST, SMTP

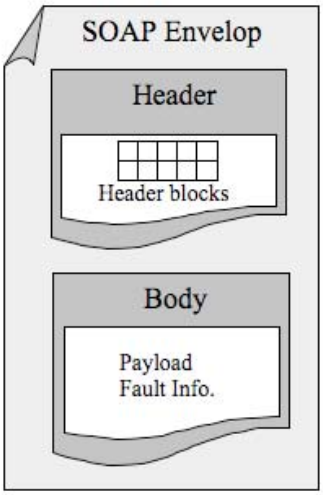


14

Notes

SOAP Messages

- **Header**
  - Extensible - support various WS standards
- **Body**
  - RPC or document-based payloads
  - Fault codes
- **Available versions**
  - SOAP v1.1
  - SOAP v1.2



15

Notes

The WS-I Consortium

- **Web Services Interoperability Consortium**
  - Establish consensus on best practice adoption of Web Service standards from W3C and OASIS
- **WS-I Basic Profile 1.1**
  - Profiling of the W3C Specifications
  - Synchronous communications model

XML schema v1.0
SOAPv1.1*
HTTPv1.1
WSDLv1.1**
UDDIv2.0

16

Notes

UML and Profiling UML

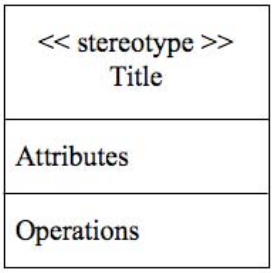
- **Features of UML**
  - Object oriented
  - Diagram-based extensive and flexible modelling language
  - Established representation language (XMI)
  - Tailorable for domain-specific modelling
  - Object Constraint Language (OCL)
- **Types of Diagram**
  - Use-case
  - Class
  - Sequence
  - Communication
  - State
- **Profiling UML**
  - Stereotypes
  - Tags
  - Packages

17

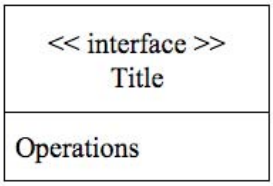
Notes

Classes & Interfaces

- **Classes**
  - Have attributes to define the data
  - Have operations to encapsulate the attributes
  - Have relationships with other classes



- **Interfaces**
  - Contract between providers and consumers of a service
  - Have operations for the service



18

Notes

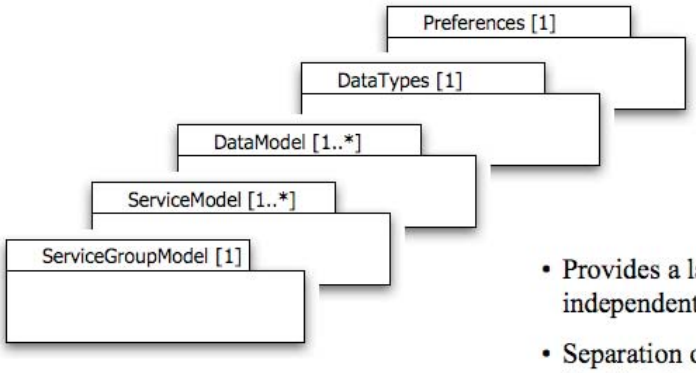
Data-types

- Define the set of data-types available within the PIM/PSM
- Language independent
- Sufficient semantic coverage
- Transformation requirements
  - Define mapping to language-dependent data-types
  - Potential loss of semantic isomorphism

19

Notes

The Specification Packages



- Provides a layout independent context
- Separation of model functional components

20

Notes

Stereotypes

- Tailor the UML meta-description to provide domain-specific modelling features
- Enable computer-based transformation of the model
- Usage
  - Package identification
  - Meta-class definition
- Service and data stereotypes (some)
  - << legend >>
  - << binding >>
  - << import >>
  - << interface >>
  - << container >>
  - << value >>
  - For XML PSM
    - << sequence >>
    - << selection >>

21

Notes

Tools & XMI

- XML Metadata Interchange (XMI)
  - OMG standardised model interchange format
  - Used by tools as the external representation format
  - Different UML tools (and versions) use XMI in different ways
  - Supports graphical/layout information as well as model
- UML Authoring Tools
  - Used to create the UML diagrams
  - Export the representation to XMI file
- XMI Transformation Tools
  - Use XML Stylesheets (XSL) to create an XSLT
  - Purpose-built application

22

Notes

XMI Transformation by XSLT

- **Approach**
  - **Establish transformation rules**
    - Different rules for WSDLv1.1, WSDLv2.0, REST, etc.
    - Consider subsequent file processing implications
  - **Code transformation rules as XSL templates**
  - **Control template sequence using content of the model**
- **Issues**
  - **Fuzzy transformation**
  - **Implicit transformation rules**
    - Imposition of style
    - Imposition of features
  - **XSL processor extensions**
    - Create many output files from single input file
  - **User configuration**
  - **XMI tool-dependence**

23

Notes

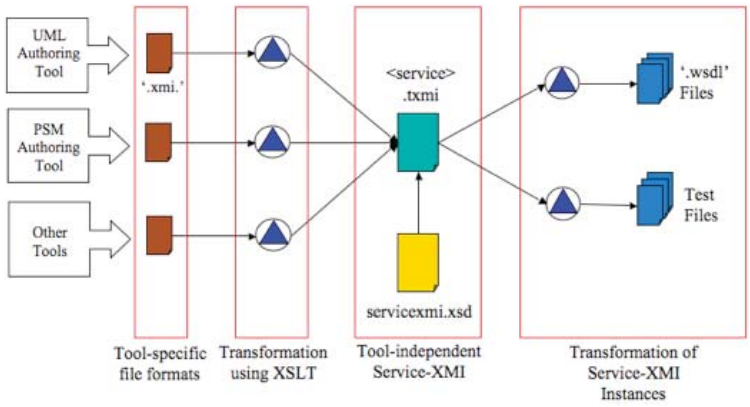
Creating a Web Service using UML

- Establish the UML PIM/PSM profile(s)
- Create the transformation tool(s)
- Create service model and export as XMI file
- Apply transformation tool to XMI file to create WSDL files
- Apply appropriate code generation tools to WSDL files
- Test, amend service model and repeat cycle

24

Notes

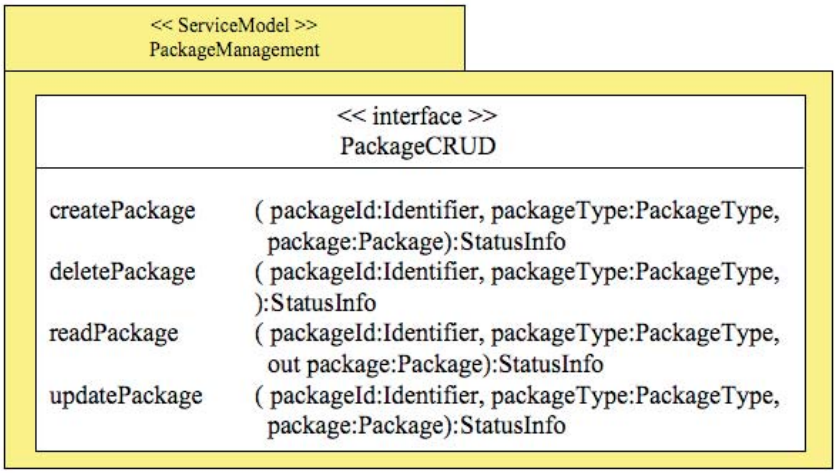
Specification Workflow



25

Notes

Service Model



26

Notes

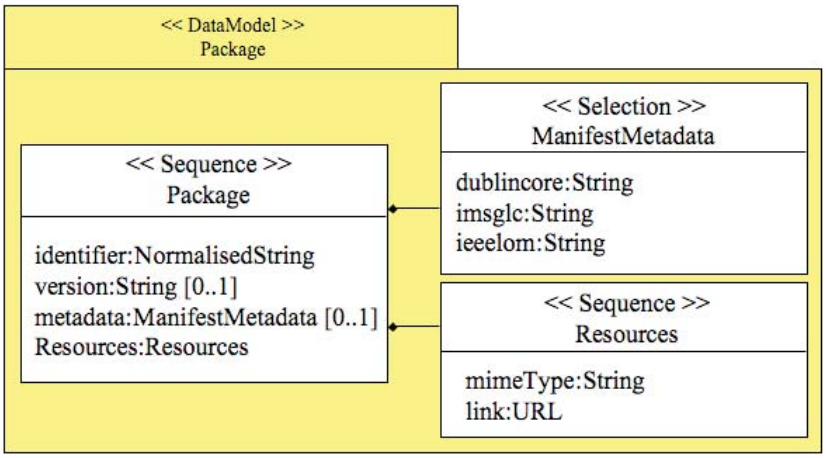
Synchronous & Asynchronous

- **Service**
  - **Synchronous -> blocked**
    - Wait for response
    - Sync Messaging
  - **Asynchronous -> unblocked**
    - Do useful work without waiting for remote data
    - Sync Messaging (x2)
    - Async Messaging
- **Messaging**
  - **Synchronous -> Request/Response**
  - **Asynchronous -> Request/Ack, Response/Ack**
  - **WSDL supports**
    - Request/Response
    - Client-Request
    - Server-Response
  - **Other options inc.**
    - Polled
    - Publish/Subscribe

27

Notes

Data Model

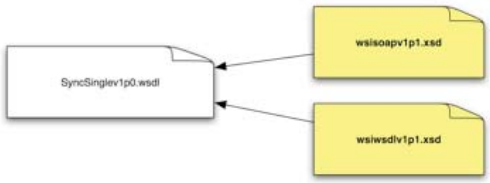


28

Notes

WSDL File(s)

- Options
  - Single WSDL file containing XSD (recommended)
  - Separate single WSDL and XSD files (use xsd:import for linkage)
  - Separate Service Specific and Abstract Service WSDL files (use wsdl:import)
  - Multiple WSDL and multiple XSD files (namespace nightmare)



29

Notes

WSDL to Code

- WSDL feature handling limitations
  - Not all WS-* supported by tools
  - Not all WS-* best practice agreed for WSDL format
- Different tools handle WSDL differently
  - Re-engineer elegant WSDL/XSD structures
- Automated creation of code
  - Code stubs for service provider and consumer
- SOAP message marshalling implications
  - Limitations on XML processing

30

Notes

WSDL Best Practice

- **Policy**
  - Adopt and tailor WS-I recommendations
  - Single service per WSDL file
  - Use request/response messaging
  - Understand name-spacing
  - Consider versioning (many aspects) at the outset
  - Interoperability means be conservative
- **Tool-derived**
  - Do not use XML attributes
  - Avoid restricted simpleTypes
  - Avoid polymorphism
  - Assume global declarations in XSD
  - Single combined WSDL/XSD file
  - Understand name-spacing
  - Use clear naming conventions for all internal WSDL identifiers

31

Notes

SOAP & REST

- **SOAP**
  - Verbose messages
  - Complex message marshalling
  - Flexible and extensive data exchange
  - Supports rich functionality and easy to extend
  - Requires comprehensive implementation
  - Formally standardised and wide adoption
- **Representational State Transfer (REST)**
  - Stateless messaging
    - HTTP_GET
    - HTTP_DELETE
    - HTTP_PUT, HTTP_POST
  - Pass data as HTTP parameters e.g. String
  - Cannot support complex functionality such as security
  - 'Quick and dirty implementation'
  - Set of rules not a standard

32

Notes

More Web Services Standards

WS Standard	Description
MTOM	Message Transmission Optimisation Mechanism for attaching non-XML data to SOAP messages
WS-Addressing	End-point addressing
WS Security	Security framework for authentication and authorisation
WS Resource Framework (WSRF)	Separation of the data resource from the service operations
BPEL	Service composition and choreography using the Business Process Execution Language

33

Notes

Suitable Tools

- UML authoring
  - Poseidon, Enterprise Architect, Rational
- XML/XSL authoring
  - Oxygen, XMLSpy
- WSDL processing
  - .NET Visual Studio
  - Axis (Java)
  - Zend (PHP) and Rails (Ruby)

34

Notes

Interoperability

- **Requirements**
  - Implementation platform independence
  - Backwards and forwards compatibility
  - Separation of service specification and implementation binding
  - Tool independence
  - Full life-cycle support
- **Issues**
  - Great aspiration but reality disappoints
  - Not necessarily to get better over the next 5 years
  - Significant gap between tool capability and technical state-of-the-art
  - Level of real experience is limited even in the big companies

35

Notes

Summary

- Web Services is one way of realising Services across the Web
- Adoption of WSDL/SOAP/XML should be conservative due to tool limitations
- Model driven approaches are increasingly appealing
- UML has wide adoption for system and application design
- MDD using UML for SOA is the way forward

36

Notes

Notes

APPENDIX A – DUNELM SERVICES LIMITED

A1 Dunelm Services Limited

Dunelm, a Sheffield based company, is an Information & Communications Technology (ICT) Applied Research & Development consultancy organization with specialist expertise in data communications, e-learning technologies and service-oriented architecture. Dunelm was incorporated in 1989 and is ISO 9001 registered. For the past five years Dunelm has had an annual turnover in the range of £180k-£220k of which 70% is from international clients. The consultancy skills and services offered are:

- Systems Development – consultancy, software engineering, planning, design, implementation, project and quality assurance management;
- Communications – data protocols, local area networks, spread spectrum systems, internetworking, network modelling, cable networks, network design and evaluation;
- Specification development for on-line learning systems using IMS Global Learning Consortium, ADL (SCORM), Schools Interoperability Framework (SIF) and IEEE LTSC approaches;
- Exploitation of ICT to support regional regeneration: anywhere anytime learning, regional foundations, inward investment.

Key projects completed recently by Dunelm are:

- IMS GLC Specification Development – working for IMS Global Learning Consortium, a global consortium developing specifications for on-line learning. Responsible for the development of the underlying 'IMS General Web Services';
- Rotherham Connected Communities Project – this involved a feasibility study of using the Rotherham Grid for Learning Network to support a broader community-wide digital services infrastructure;
- Digital South Yorkshire (d-SY) Special Interest Group (SIG) Provision – this work entailed the delivery of a number of 'business networking' activities focused on the ICT vendors within the South Yorkshire sub-region;
- Technology Market Development Planning – advising several ICT-based SMEs on how they should exploit their technology globally. Also advising them on how to establish appropriate software development processes and procedures;
- South Yorkshire E-learning Strategy – responsible for developing the e-learning strategy adopted by South Yorkshire. This involved developing the ICT framework to support 'Anywhere, Anytime Learning'.

A2 Colin Smythe Resume

Personal Details

Contact Address: 34 Acorn Hill, Stannington, Sheffield, S6 6AW
Tel/Fax: +(44)-114-2334009
E-mail: colin@dunelm.com

Education: 1982–1985, PhD in Communications (University of Durham)
1976–1979, BSc (Hons) in Applied Physics (University of Durham)

Affiliations: Fellow of British Computer Society (1998)
Member IET (2006)
Member IEEE (1991) and Member ACM (1992)
Chartered Engineer (1998)
Chartered IT Professional (2004)
Associate Member of the Institute of Directors (2006)

Experience in IT: 29 years

Primary Skills

• E-learning specifications and standards	Eight years
• Networks consultancy (Data networking and internetworking)	Thirteen years
• Project management (inc. European projects)	Thirteen years
• Proposal tendering and writing (inc. European projects)	Twenty-six years
• Computer systems implementation (inc. ‘C’)	Twenty-nine years
• IT training (Data networking and software engineering)	Fifteen years

Summary

Colin Smythe has extensive experience in the field of IT, particularly data networking and e-learning. He has successfully managed projects worth millions of pounds and has applied detailed technical knowledge in the fields of data networking, cable modems, spread spectrum communications and internetworking. He has a wide range of experience in corporate and academic management, is a widely used supplier of IT training in the fields of networking, software engineering and e-learning. He has excellent personal and communications skills and is well versed in working with a wide range of companies over long and short periods.

Colin has a degree in Applied Physics and was awarded a PhD for his work on spread spectrum local area networks; both degrees were from the University of Durham. During the past 25 years he has worked in both the academic and the industrial world. He has been a lecturer at the Universities of Durham and Surrey, worked for several years at Logica Ltd., was the founder systems consultancy Hyperion Systems Ltd., e-learning consultancy eLoki and network modelling consultancy Nettonics. He has acted as independent consultant to many private and public organisations, and has also given numerous industrially oriented training courses on communications and software engineering.