Introduction

Service-Oriented Architecture

Jeremy Gibbons

Contents

1 Aims

- understanding of principles
- engagement with phenomenology
- exploration of emerging standards
- some investigation of practical toolkits
- situation in wider context

2 Prerequisites

At the start of the course you should:

- fully understand principles of object orientation
- be fluent in programming in Java
- have a reading knowledge of UML and XML

3 Contents

- components
- web services
- REST
- qualities
- composition
- services vs objects
- software architecture
- semantic web
- engineering

4 Connections

- objects (OOR, OOD, OOP, DPA, SPL)
- formal techniques (SEM, SDE)
- concurrency (CDS, ACT, MOB)
- security (SPR, SRO, DES, NES)

5 Resources

- Singh & Huhns, *Service-Oriented Computing* (Wiley, 2005)
- Josuttis, *SOA in Practice* (O'Reilly, 2007)
- Erl, *SOA* (Prentice-Hall, 2005)
- Szyperski, *Component Software* (Addison-Wesley, 2e 2002)
- Richardson & Ruby, *RESTful Web Services* (O'Reilly, 2007)
- Antoniou & van Harmelen, *Semantic Web Primer* (MIT, 2004)
- any number of W3C, OASIS, OMG standards

6 Rules of engagement

- please feel free to ask questions as we go along
- the timing estimates are flexible
- mobile phones turned off (or at least silent), please

7 Introductions

- DPhil 1991
- SEP since 1999
- programming languages
- generics, patterns, fp, objects...

- and you —
- name? role? objective?

Index

Service-Oriented Architecture

Monday	Tuesday	Wednesday	Thursday	Friday
Introduction Components	REST	Composition	Architecture	Engineering
coffee	coffee	coffee	coffee	coffee
Components	REST	Composition	Architecture	Conclusion
lunch	lunch	lunch	lunch	lunch
Web Services	Qualities	Objects	Semantic Web	
tea	tea	tea	tea	
Web Services	Qualities	Objects	Semantic Web	