Service-Oriented Programming

Course Module 3

Version 1.0
July 29, 2013

Xumin Liu, Rajendra Raj
Computer Science
RIT
[xl, rkr]@cs.rit.edu

Tom Reichlmayr
Software Engineering
RIT
tjr@se.rit.edu

Chunmei Liu
Systems & Computer Science
Howard University
chunmei@scs.howard.edu

Alex Pantaleev
Computer Science
SUNY at Oswego
alex@cs.oswego.edu

This material is based upon work supported by the National Science Foundation under Awards DUE-1140567, DUE-1141112, and DUE-1141200. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

1. Overview

This course module is the third of three modules being developed to teach Service-Oriented Programming (SOP) to undergraduate students in computing. Course modules are well-defined, self-contained units of instruction that may be incorporated, either as-is or with minor modifications, within one or more courses offered in diverse disciplines. This modular approach also allows for easy vertical integration of SOP into the undergraduate curriculum in introductory, mid-level and senior courses in disciplines such as Computer Science and Software Engineering.

This course module focuses on the architecture, design, and implementation of enterprise-critical software systems, the so-called "Programming in the Large." It is targeted at upper division students, and assumes some SOP knowledge, as can be gained by taking either one of Course Module 1 or Course Module 2; if not, an appropriate subset of that material should be incorporated either as a review or complete introduction. This module can be used in a variety of senior courses such as Concurrent & Distributed Software Systems; Enterprise Software Systems; Software System Requirements & Architecture; Upper Level Software Engineering Course; or Senior Capstone Projects.

Prerequisite knowledge. This course module is typically meant for students who have had extensive exposure to programming in object-oriented languages such as Java, C#, Objective-C or C++, and have taken software engineering and software design courses.

Module Learning Outcomes. Students completing this course module will be able to:

- Describe the fundamental differences in design approaches between centralized and distributed applications
- Analyze an existing application with respect to non-functional requirements (performance, scalability, reliability, failure models).
- Design a distributed application that uses public or private web services
- Explain how software reuse and quality are improved by the use of web services' composition

2. Rationale

The global service-oriented middleware market will grow to at least $8.2 billion by 2016, according to a WinterGreen Report[1]. It is crucial that undergraduate Computer Science (CS) and Software Engineering (SE) students learn about these developments in web services and service-oriented computing as these concepts and technologies will dominate software development for the next few years. The students need repeated exposure to the same concepts at different degrees of depth, and this upper-level course module builds on earlier knowledge of SOP.

[1] WinterGreen Research, Inc. SOA Application Middleware – Markets Reach $8.2 Billion by 2016. In http://www.wintergreenresearch.com/reports/SOA%202010 %20press%20release.pdf, 2010.

3. Recommended use

The authors have typically targeted this module to a course from one of Concurrent & Distributed Software Systems; Enterprise Software Systems; Software System Requirements & Architecture; Upper Level Software Engineering Course; and Senior Capstone Projects. As usual, the module may also be used in any course where the students have the prerequisite knowledge of object-oriented programming, software engineering and software design.

4. Lecture Slides

The lecture slides for this module are also included in this folder. The instructor may use them as-is or incorporate them into his or her materials.

5. Sample questions

The sample questions for this module, also included in this folder, are meant for low stakes quizzes. The instructor may use them as-is or incorporate them into his or her tests.

6. Labs/programming assignments

A sample lab assignment is provided in this folder to permit students gain hands-on experience in making use of existing web services in their programming.

7. FAQ

This section will be populated after the authors have gained further experience with the use of these modules in actual courses.

8. Readings

These readings are primarily for faculty teaching these courses; a subset of the material may also be handed out to students as appropriate.

1. Web Service: Principles and Technology by Michael Papazoglou, Prentice Hall, 2007, ISBN: 0321155556.
2. Web Services by Gustavo Alonso, Fabio Casati, Harumi Kuno, Vijay Machiraju, Springer 2004, ISBN: 3540440089.
3. Engineering Distributed Objects, Wolfgang Emmerich, John Wiley & Sons Ltd 2000, ISBN: 0471986577
4. SOA in Practice: The Art of Distributed System Design, Nicolai M. Josuttis, O'Reilly 2007, ISBN: 0596529554
5. Distributed Systems: Concepts and Design Fifth Edition, George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair, Addison Wesley 2012, ISBN: 0132143011

6. Service Oriented Architecture: Entities, Services, and Resources, Dominic Duggan, John Wiley & Sons, Inc. 2012
7. Patterns of Enterprise Application Architecture, Martin Fowler, Addison Wesley, 2003

9. Links

These links are primarily for faculty teaching these courses; a subset of the material may also be handed out to students as appropriate.

1. http://www.w3.org/TR/ws-arch/
2. http://www.w3.org/TR/wsdl
3. http://www.w3.org/TR/soap/
4. http://www.w3.org/TR/wscl10/
5. http://msdn.microsoft.com/en-us/library/ms954638.aspx

10. Module Evaluation

This section will include assessment tools to measure student learning and module effectiveness. In addition to direct student assessment using quizzes, assignments and tests, module evaluation requires student self-assessment and instructor self-assessment of the materials. The latter assessment materials will be provided to instructors who are early adopters of this course module.