Knowing When to REST:
Simple Object Access Protocol vs.
Representational State Transfer Web Services

Version 1.0

Revision History

Date	Version	Description	Author
June 10, 2009	1.0	Initial release	Valerie Smothers vsmothers@medbiq.org

Table of Contents

MedBiquitous Consortium XML Public License and Terms of Use

MedBiquitous XML (including schemas, specifications, sample documents, Web services description files, and related items) is provided by the copyright holders under the following license. By obtaining, using, and or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

The Consortium hereby grants a perpetual, non-exclusive, non-transferable, license to copy, use, display, perform, modify, make derivative works of, and develop the MedBiquitous XML for any use and without any fee or royalty, provided that you include the following on ALL copies of the MedBiquitous XML or portions thereof, including modifications, that you make.

1. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the following notice should be used: "Copyright © [date of XML release] MedBiquitous Consortium. All Rights Reserved. http://www.medbiq.org"
2. Notice of any changes or modification to the MedBiquitous XML files.
3. Notice that any user is bound by the terms of this license and reference to the full text of this license in a location viewable to users of the redistributed or derivative work.

In the event that the licensee modifies any part of the MedBiquitous XML, it will not then represent to the public, through any act or omission, that the resulting modification is an official specification of the MedBiquitous Consortium unless and until such modification is officially adopted.

THE CONSORTIUM MAKES NO WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, WITH RESPECT TO ANY COMPUTER CODE, INCLUDING SCHEMAS, SPECIFICATIONS, SAMPLE DOCUMENTS, WEB SERVICES DESCRIPTION FILES, AND RELATED ITEMS. WITHOUT LIMITING THE FOREGOING, THE CONSORTIUM DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY, EXPRESS OR IMPLIED, AGAINST INFRINGEMENT BY THE MEDBIQUITOUS XML OF ANY THIRD PARTY PATENTS, TRADEMARKS, COPYRIGHTS OR OTHER RIGHTS. THE LICENSEE AGREES THAT ALL COMPUTER CODES OR RELATED ITEMS PROVIDED SHALL BE ACCEPTED BY LICENSEE "AS IS". THUS, THE ENTIRE RISK OF NON-PERFORMANCE OF THE MEDBIQUITOUS XML RESTS WITH THE LICENSEE WHO SHALL BEAR ALL COSTS OF ANY SERVICE, REPAIR OR CORRECTION.

IN NO EVENT SHALL THE CONSORTIUM OR ITS MEMBERS BE LIABLE TO THE LICENSEE OR ANY OTHER USER FOR DAMAGES OF ANY NATURE, INCLUDING, WITHOUT LIMITATION, ANY GENERAL, DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, OR SPECIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF ANY USE OF MEDBIQUITOUS XML.

LICENSEE SHALL INDEMNIFY THE CONSORTIUM AND EACH OF ITS MEMBERS FROM ANY LOSS, CLAIM, DAMAGE OR LIABILITY (INCLUDING, WITHOUT LIMITATION, PAYMENT OF ATTORNEYS' FEES AND COURT COSTS) ARISING OUT OF MODIFICATION OR USE OF THE MEDBIQUITOUS XML OR ANY RELATED CONTENT OR MATERIAL BY LICENSEE.

LICENSEE SHALL NOT OBTAIN OR ATTEMPT TO OBTAIN ANY PATENTS, COPYRIGHTS OR OTHER PROPRIETARY RIGHTS WITH RESPECT TO THE MEDBIQUITOUS XML.

THIS LICENSE SHALL TERMINATE AUTOMATICALLY IF LICENSEE VIOLATES ANY OF ITS TERMS AND CONDITIONS.

Knowing When to REST: Simple Object Access Protocol vs. Representational State Transfer Web Services

1. Acknowledgement

The following members of the MedBiquitous Technical Steering Committee have contributed to this document:

- Joel Farrell, IBM, Technical Steering Committee Chair
- Dan Rehak, Learning Technologies Architect
- Andy Rabin, CECity
- Carl Singer, CECity
- Valerie Smothers, MedBiquitous

2. Scope

This document provides general guidelines for those considering the development of Simple Object Access Protocol (SOAP) Web services and Representational State Transfer (REST) Web services. The document does not strictly adhere to the definition of REST presented by Roy Fielding in his dissertation [Fielding]. This document interprets REST as a general approach broadly following the architectural style for distributed resources.

3. Status

This document is a technical guideline developed for the MedBiquitous community. It is currently a draft document; we welcome your comments.

4. Introduction

SOAP is a packaging protocol for the messages that applications send back and forth to execute some functionality described by a Web service [Snell1]. SOAP can be used for calling specific procedures or functions remotely, or it can be used to send electronic documents for specific transactions [Snell1].

REST is an abstraction of the architecture commonly used by websites. A Web resource has a URL; accessing this URL returns a representation of the resource, such as a web page [Ray]. A link within that resource then leads the client to another resource, or a change in state [Ray]. Resources may be dynamic or static.

Many implementing Web services question which approach to use. These guidelines provide compare the two approaches and offer some general guidelines for when to use each approach. These guidelines are designed to complement the MedBiquitous Web Services Design Guidelines [MedBiq] and the MedBiquitous REST Services Design Guidelines (in development).

This document assumes a general conceptual understanding of Web services.

5. Resources vs. Activities

James Snell and others suggest that REST is appropriate for resource-oriented services, while SOAP is more appropriate for activity-oriented services [Snell2]. Resource-oriented services offer a few basic operations that can be applied to a dataset or object. These operations are usually the CRUD operations: Create (PUT), Retrieve (GET), Update (POST), Delete (DELETE) [Snell2]. Activity-oriented services have a variety of operations that vary depending on the activity that must be performed. Transferring funds is an example of an activity-oriented service [Snell2].

The ATOM Publishing Protocol is an example of a "RESTful" Web service, i.e. a Web service designed using REST principles. The protocol provides a standard way to create, edit, and delete resources, such as newsfeeds. It also provides protocols for retrieving sets of resources and discovering resource collections [RFC5023]. Retrieving a collection of news items on e-learning would be an example of RESTful Web service using ATOM.

One example of a SOAP service could be a service for scheduling operations in a hospital [Snell2]. There is an activity, scheduling, that is at the center of the service, and it is likely more complex than retrieving and updating a static document.

6.   When is something a Resource?

At the root of the activity resource distinction is the ability to distinguish a resource from an activity. A resource is information that can be named; it has an identifier (such as a URL), and a representation (such as a web page). The representation may change, but conceptually, it remains the same [Ray].

Resource identifiers must be a Uniform Resource Identifier (URI). Practically, these are usually URLs, which allow users to find the resource (for example, http://www.medbiq.org/rss/medbiquitousnews.xml). Generally, REST services are used to expose these resources [Ray].

Consider the example of digital continuing education certificate. Each certificate has a unique identifier that could be constructed as a URI. There is also a representation of the certificate that may be text or code, such as MedBiquitous Activity Report XML. The certificate can be considered a resource.

Now consider an example related to clinician credentialing. A hospital grants privileges in part based on credentials, such as licensing. A hospital may receive notice that a staff member's license has been revoked. The hospital could develop a web service to suspend privileges at other hospitals in the health system. While the license may be a resource, the suspension of privileges is an activity more complex than retrieving or updating data.

7.   Other Factors to Consider

Other factors are important to consider when weighing the options for Web services design. In many cases, SOAP offers the benefits of a predefined approach for factors surrounding the core transaction. SOAP is complemented by a variety of WS standards with various degrees of implementation support. RESTful solutions do not have equivalent standards defining approaches to addressing these issues. Architectural consistency can also be a factor to consider.

- **Security** [Snell2] – WS-Security and related specifications offer fairly comprehensive and standardized mechanisms for securing SOAP Web services. These mechanisms include digital signatures that offer proof of origin and integrity of data while ensuring confidentiality. REST web services may use security mechanisms afforded by HTTP, such as Secure Sockets Layer (SSL), but high level security services would have to be implemented on a case-by-case basis.

- **Reliable messaging** [Snell2] – SOAP offers Web Services-Reliability, a predefined approaches for reliable messaging. REST web services may use reliability mechanisms afforded by HTTP, but such services would have to be implemented on a case-by-case basis.

- **Other complexities**, including Message routing, lifecycle management, and event notification [Snell2]. Various WS standards allow predefined approaches for these complexities. The same functionality may be accomplished through REST, but these implementations would be on a case-by-case basis.

- **The overall application or environment** – If a task analysis leads you to one style of Web services, the fact that a couple of individual services do not fit cleanly into that design does not mean that they should be implemented using the other paradigm. The determination should be made on the central thrust of the

system; an exception doesn't mean you should deviate from the overall design pattern. Architectural consistency is important. If you are working in an environment where one approach dominates over the other, that may be reason enough to choose one approach over another, provided the approach will fit the needs of the application.

8. References

[Fielding]
Feilding R, 2000. *Architectural Styles and the Design of Network-based Software Architectures.* Doctoral dissertation, University of California, Irvine.

[MedBiq]
MedBiquitous, 2009. MedBiquitous Web Services Design Guidelines ver 2.0. Accessed May 27, 2009:
http://www.medbiq.org/std_specs/techguidelines/webservicesguidelines.pdf

[Ray]
Ray RJ, Kulchenko P, 2002. Programming Web Services with Perl. O'Reilly: Sebastapol, CA.

[RFC5023]
Gregorio J., de hOra B, 2007. RFC 5023: The Atom Publishing Protocol. IETF Trust. Accessed May 27, 2009:
http://www.ietf.org/rfc/rfc5023.txt

[Snell1]
Snell J, Tidwell D, and Kulchenko P, 2002. Programming Web Services with SOAP. O'Reilly: Sebastapol, CA

[Snell2]
Snell J, 2004. Resource-oriented vs. Activity-oriented Web services. Accessed May 26, 2009:
http://www.ibm.com/developerworks/webservices/library/ws-restvsoap/