Evolution and Maintenance of Web Service Applications

Mira Kajko-Mattsson
SERVIAM & Software Maintenance Laboratory
Department of Computer and Systems Sciences
Stockholm University/Royal Institute of Technology
Electrum 230, SE-164 40 KISTA
mira@dsv.su.se

Abstract
In this paper, we suggest a panel debate on evolution and maintenance of web services. The panel will discuss the evolution and maintenance of web service applications.

1. Introduction
Web services provide an important underpinning to new business opportunities. They are applications satisfying the needs of a wide variety of customers and businesses. Examples of their use may be found in B2B and B2C applications, e-learning, and the like.

Web services are highly vulnerable and subject to constant change. Hence, they offer a novel challenge to software engineering. This challenge has not yet been investigated. From the evolution and maintenance perspective, there are many things that must be examined. These include the issues of evolution and maintenance processes, products and roles involved within the processes, and the organisational changes required for adopting to the web service application mode.

In this paper, we suggest a panel debate in order to identify this challenge. The invited panelists come from both the industry and academia. They are (1) Dr. Shawn Bohner from Virginia Tech, USA, (2) Dr. Ned Chapin, editor of Journal of Software Maintenance and Evolution, USA, (3) Mr. John Fisher, a practitioner from SmithBucklin Corporation, USA, (4) Mr. Harry Sneed, a consultant from Case Consult GmbH, Germany, (5) Dr. Paolo Tonella, a researcher from Instituto Trentino di Cultura, Italy, and (6) Prof. Cornelia Boldyreff from University of Lincoln, UK.

Each panelist will try to answer to some of the following questions:
- Does evolution and maintenance of web service applications differ from the evolution and maintenance of traditional software systems? The answer should be given from the product-, process-, role- or organisational perspective.
- What are the problems encountered during evolution and maintenance of web services that are not encountered in traditional evolution and maintenance?
- Can we remedy these problems and if yes, how?

Below, each panelist and the panel leader provide their opinion on some of the above-mentioned questions.

2. Mira Kajko-Mattsson
Evolution and maintenance of simple web service applications should not lead to major changes in products, processes and roles. This mainly concerns static web services offered by one organisation. Regarding the high-level complex and dynamic inter-enterprise web services, their management is often not in the control of a single organisation. Hence, some changes are required. First, one must create a new role of a service owner responsible for evolving and maintaining high-level web services. This role may be managed by a separate organisation totally dedicated to manage high-level web services. Second, the maintenance process has to be slightly adapted to manage investigation of problems and impact analysis which have to be made across several collaborating applications belonging to different organisations. Third, there is a great need for well-defined service level agreements (SLAs) and an elaborated SLA process. This is due to the fact that a failure of a web service may substantially affect the productivity of other organisations. Fourth, the current web service technology offers basically a syntactic solution, and lacks a semantic support. In the long run, this may impose severe product maintainability problems.

3. Shawn Bohner
At first glance, maintaining web service applications (WSAs) appears to be largely the same as that for traditional software. Hence, software change principles should apply during corrective, perfective, adaptive, and preventative change actions. However, today’s WSAs are
largely middleware and components -- introducing extra barriers to making changes to software.

WSAs introduce the notion of "maintaining the interoperability" of these layered software capabilities. Also, "web" connotes the application will have its operational environment on the network -- imposing yet another level of complexity when navigating changes. Further, WSAs tend to mix applications and content management into a common intermediate form using eXtensible Markup Language (XML) and all of its supporting capabilities.

The service orientation of web services also presupposes an approach to engaging the service from the customer’s perspective. Ultimately, the customer should be able to have competing software capabilities at his or her disposal. However, this freedom comes at a price as making corrections leads to unexpected circumstances as responsibilities for remedy are, at best, unclear.

4. Cornelia Boldyreff

The extra complexity introduced when systems are even partially composed of Web Services has not yet been appreciated fully. While utilisation of web services gives application developers new flexibility, for example, dynamic reconfiguration of systems, it also requires a rethinking some of our software engineering processes. In the case of dynamically reconfigured systems, current testing practices are no longer valid and require reconsideration.

In general managing the evolution of such systems requires consideration of a number of separate evolution paths; some of which may be beyond the control of the application owner and subject to trusting the guarantees of the service providers.

5. Ned Chapin

The move to Web Services often starts with EDI-using customers and suppliers. The move also often emphasizes enhancing existing ways of transacting business with them by adding the Web. To improve interoperability, the typical approaches have included using wrappers and inserting middleware. Approaches have sometimes relied on vendor-provided packaged software, sometimes on contractor written software, sometimes on in-house produced software, and usually have involved some combination of software from all three sources.

Results have been mixed, thus far. Organizational personality, interoperability and cost matters have been more troublesome than anticipated. Change management, security, and staff training have been common hurdles. But some positive lessons have been learned.

6. Harry Sneed

Seemingly every advance in software technology brings about more demands on the user of that technology. Web-Services are no exception. Incorporating them into a user application environment is no easy task. They have to be configured and tested. Bridges and interfaces have to be built to the existing user systems. The work to be done does not require so much effort, but it does require a great deal of knowledge about the environment, knowledge, which the average users do not have. The total cost of ownership goes up.

The answer is to set up software service centers which have the necessary infrastructure to service several user companies. Even if the users develop their web-systems alone, it is still better for them to outsource the maintenance and evolution of those systems. The economy of scale will make it possible for the service organizations to keep a staff of highly trained specialists, to purchase or develop the necessary tools and to set up the proper processes. With the same staff they will be able to serve many users. In this way, IT will become an external service, which users can rent.

7. John Fisher

Web services have evolved from two distinct sources. The first is the web itself and web based applications, which tend to be short lived, although sometimes very complex. The second is object-oriented software development, which follows a more traditional development life cycle. Web services are in effect large scale business objects, which provide distinct functions and rely on integration with other applications to provide solutions. The major issue with either development effort lies in effective gathering of business requirements.

8. Paolo Tonella

The technology used to realize Web-based systems involves multiple programming languages, both on the server side and on the client side, and dynamic code generation. Reflective facilities are exploited on the client side, to support programmatic, dynamic manipulation of user interface and interaction modes. Proper solutions are adopted to realize session management, to track concurrent users, and to record the session state. Moreover, content and computation must co-exist.

These features give rise to specific problems, demanding for novel approaches and methodologies, during the evolution and maintenance of Web applications. In fact, they complicate the enforcement and preservation of quality attributes such as the modularity, reliability, scalability, efficiency, availability, security, accessibility, portability, maintainability and testability of the Web software.