

# Roadmap towards Lifecycle Support for Highly Distributed Web-based Systems

Frederic Majer, Patrick Freudenstein, Martin Nussbaumer  
Karlsruhe Institute of Technology - University of Karlsruhe (TH)  
Department of Telematics, IT Management and Web Engineering Research Group,  
Engesserstr. 4, 76128 Karlsruhe, Germany  
{frederic.majer, patrick.freudenstein, martin.nussbaumer}@kit.edu

## Abstract

*To cope with the increased complexity of today's Web-based systems, approaches facilitating operations, maintenance and evolution of heterogeneous and distributed systems landscapes are vital. In particular, due to the complex interdependencies between their distributed system parts, assuring that services are delivered at a high level of quality is challenging. To this end, we present the integrated information map (i2map) as an aspect-oriented approach. It considers these interdependencies as first-class citizens and provides sophisticated services addressing the lifecycle support of distributed systems. We exemplarily demonstrate the approach's potential in the context of ontology-based evaluation of service level agreements in a highly interrelated system landscape.*

## 1. Introduction

Through the advancement in technology and the spreading of the internet, the World Wide Web has evolved from a decentralized information medium to a platform for a wide range of different application types [8]. These Web-based systems support collaboration scenarios, deliver complex services and are often characterized by the integration of functionality and content from different organizations and software systems. This increases interdependence and interaction between people, companies and especially software systems in disparate domains and locations. While the development cycles of applications can be shortened by reusing existing services, the emerging service-orientation leads to a higher complexity of the resulting system landscapes.

Existing Web Engineering approaches (e.g. OOHDML [10], UWE [5] or WebML [2]) introduced

adequate models, methodologies and tools for the development of Web applications. However, questions regarding operations, maintenance or aspects of evolution of the underlying systems haven't been addressed sufficiently.

In section 2, we describe the problem domain based on an example scenario and introduce common challenges regarding operations, maintenance and evolution of the focused systems. Section 3 gives an overview of the i2map approach, coating the entire system landscape and comprising dedicated models, methodologies and tools to cope with different aspects of the lifecycle support. Furthermore, we demonstrate the realization of first concepts and conclude with an outline of further work in section 4.

## 2. An Example Scenario

Figure 1 shows a representative excerpt of a distributed Web-based system with typical components. It includes a collaboration server for providing appointment and contact data or exchanging mail messages. A Web service wrapper supports the integration of functionality for different Web applications to overcome native and incompatible protocols. Thereby, data about appointments, contacts etc. can be used in the Web Applications. Due to the increased distribution level of participating applications and services, cache mechanisms (*Refresh Cache*) are implemented to reduce communication latency.

Although the scenario may look simple, operating and maintaining the system is complex. In this scenario, a malfunction of one of the scheduled tasks responsible for refreshing the caches would result in the problem that the content of the Web applications originating from the collaboration server was outdated. Furthermore, the distribution of responsibilities

regarding the operation of the various system parts over several roles and persons, partly in different organizational units, increases complexity and results in error-proneness [6]. In this context, any modification to a system part is critical as the effects of a change cannot be foreseen due to a lack of knowledge about all interdependencies in the system. For example, any changes to the interface or the relocation of the Web service wrapper in the scenario lead to a malfunction of the Web applications.

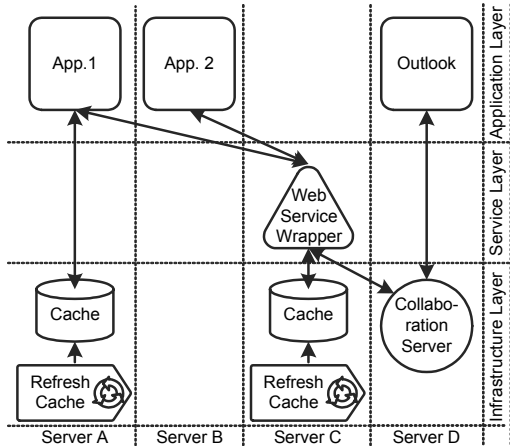


Figure 1. Example scenario

Without an adequate support system, the problem analysis for the above mentioned problems includes the identification and examination of all relevant components and dependencies. Obviously an overview as depicted in Figure 1 can help to obtain an understanding of the problem domain. Although there exist already solutions for the automated gathering and processing of system information (e.g. Microsoft System Center 2007 [12]), these approaches neglect the complexity of today's service-oriented systems. Services provide a plain endpoint related with quality assertions and policies to the service consumer. In this context, the service consumption by an end-user or another system component is based on a contract, partly implicit, defining various aspects of the service delivery. Thereby, several measurable assertions regarding dimensions like security, reliability and performance are defined and agreed upon.

From the perspective of a service consumer, the complex interdependencies of a service (e.g. a Web application) with other components in the system landscape (e.g. Web services, servers, databases etc.) are not visible. However, when negotiating service level agreements (SLAs), their influence is crucial. The validity of a SLA for a particular service strongly depends on the capabilities of all related components

or services, thus leading to a cascade of highly interrelated services and SLAs.

To be able to define consistent and reasonable service level agreements (SLAs) to guarantee a smooth operation of the overall system, knowledge about and sophisticated inference mechanisms on the internal interdependencies are essential. Furthermore, the identification and extraction of the relevant system section (*slice*) in dependence of a given context or problem is challenging. In turn, a generic problem solving approach to gather the relevant information for such a slice can be used to generate adequate views of the system for various stakeholders depending on their role, domain knowledge and the task for which support is needed.

### 3. The Integrated Information Map

To address the above-mentioned challenge, we introduce the *integrated information map* (i2map) as an aspect-oriented system for the lifecycle support of highly distributed Web-based systems. As depicted in Figure 2 the approach supplements an existing Web-based system landscape and provides services for various stakeholders.

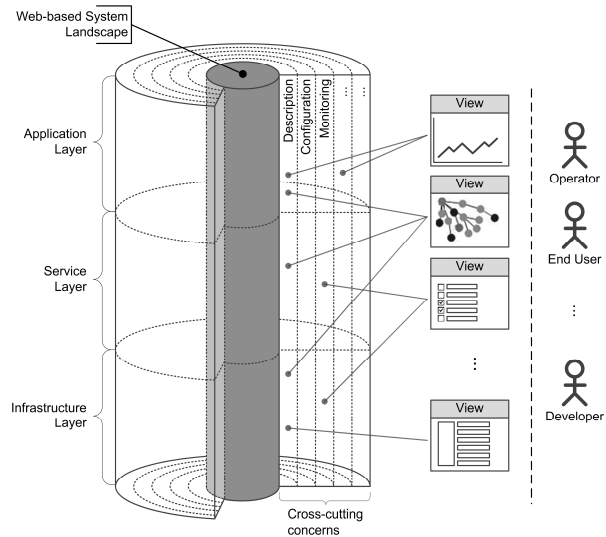


Figure 2. Overview of the integrated information map (i2map)

A major design principle of the i2map is the gradual definition and isolated addressing of several cross-cutting concerns pervading the entire system landscape. This aspect-orientation allows agility and the recentralization of problem solving strategies rather than scattering parts of such a solution all over the distributed system [7]. Due to the accumulation of i2map building blocks, each focusing on a specific

aspect or sub-problem, the integrated information map provides a toolbox with dedicated services for the lifecycle support of distributed systems.

### 3.1 Cross-cutting Concerns

The concern *description* constitutes an essential part of the i2map. It comprises aspects like the automatic gathering, consistency analysis and further processing of descriptive information to provide comprehensive information about all system parts and services, as well as the relations among them. Thus, it forms the foundation for the other concerns. Besides descriptive data about a service's behavior and interface, information about related service level agreements or policies is of interest [1].

The *configuration* concern refers to mechanisms and tools to configure system parts according to defined requirements (i.e. service level agreements). This includes a component's configuration at initial deployment as well as reconfiguration at runtime due to correction or optimization. The latter facilitates the autonomous adaptation of systems in response to user intervention or based on rules triggered by predefined events, e.g. dynamic service substitution in an orchestration to improve the performance or resolve a fault at runtime.

The concern *monitoring* addresses the identification and processing of events occurring in the system. This includes surveillance and testing of services regarding various metrics and related thresholds. The latter comprise both quality factors with respect to availability and performance (reaction time, throughput etc.) and other measurable objectives (e.g. guidelines from the field of Web accessibility). Furthermore, the monitoring of the system involves predictions about possible faults based on logs and the actual performance of system elements.

### 3.2 Realization – The i2map Browser

As a first step in the course of the realization of the i2map approach, we focused on the concern description as the most integral concern for the lifecycle support of Web-based systems. As the manual registration and maintenance of descriptive information is time-consuming and error-prone, an automated solution is essential. Thereby, one of our major goals is to avoid modifications to the underlying systems.

The description concern uses the Common Information Model (CIM) [3], an extensible standard defining how managed objects in an IT environment

are described and managed. The Web-Based Enterprise Management (WBEM) [4] as a set of system management technologies developed to unify the management of distributed computing environments enables the automated gathering of descriptive system information. The retrieved data is stored in dedicated Semantic Web frameworks, thus allowing for inferring further knowledge and sophisticated enquiries about a system.

An essential contribution of our work lies in the identification of relevant CIM classes and dedicated extension for the description of today's Web-based systems. Our current implementation retrieves system information by using the Windows Management Instrumentation (WMI) as an implementation of the WBEM standard for Microsoft platforms. This information becomes stored in the Jena Semantic Web Framework [13]. Now, by using an extended version of an existing CIM ontology [9], further knowledge about the system (e.g. different types of interdependencies) can be inferred and queried via well-defined SPARQL queries.

Figure 3 shows our current prototype of the i2map Browser, visualizing specific aspects of the information gathered about the system. On the one hand, it provides dedicated views of different aspects of the system landscape. Thereby, predefined SPARQL query templates are adapted according to various *dimensions* and the desired scope and granularity (*slice size*). On the other hand, the application supports various stakeholders by the fulfillment of their *tasks*. For example, regarding the introduced problem of defining or examining existing service level agreements in distributed systems, the i2map Browser offers valuable support.

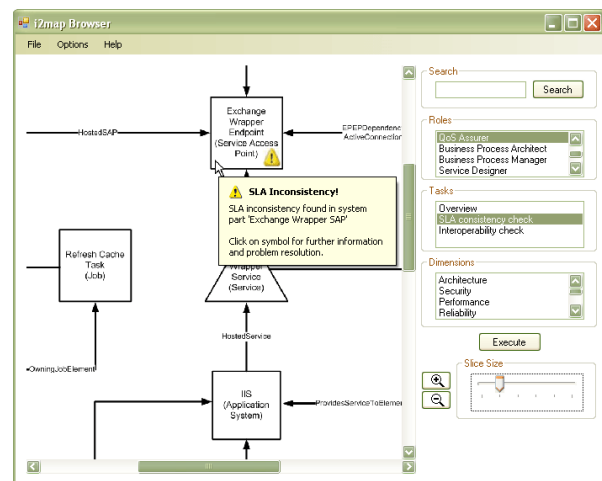


Figure 3. View of the example scenario in the i2map browser

Based on the selected role *QoS Assurer* and task *SLA consistency check*, the i2map Browser evaluates the defined SLAs of all services with strong consideration of their dependencies. Thereby, contradictory SLAs throughout the whole system are detected and highlighted as shown in the figure. Therefore, it takes advantage of the ontology-based inference and query mechanisms provided by the underlying i2map infrastructure.

#### 4. Future Work

Although the results are already promising, further research regarding the concern description and other concerns is needed to provide effective strategies for the lifecycle support of distributed Web-based systems.

In the future, we will concentrate on the formal and comprehensive definition of concepts like roles, domains of interest and lifecycle-related tasks in the context of the Web Engineering domain in form of an ontology. Combined with existing ontologies from other engineering disciplines, this knowledge can be used for the adequate parameterization of the i2map Browser and other tools providing problem solving strategies.

Regarding the verification of defined SLAs, our further work will focus on the analysis of more complex scenarios based on sophisticated metrics. Beyond that, in the context of the monitoring concern, the automated monitoring of the conformance of all system parts to defined SLAs represents the next step. In this context, our work will focus on the automated decomposition of SLAs into measurable values and the development of surveillance and testing strategies based on the existing descriptive information about the system. As already stated above, a major design goal lies in avoiding modifications to the observed system landscape itself and in the application of common standards and existing methodologies and tools where appropriate.

Furthermore, the inclusion of existing data sources providing descriptive information about the distributed system is of interest. For example, this includes the analysis of design artifacts from the development phase. Thereby, the current consolidation efforts of the MDWEnet initiative [11] that strives for achieving interoperability between common Web Engineering methodologies form a promising foundation. Assuming the existence of transformation rules to convert design models of particular methodologies to

design artifacts according to the defined Common Metamodel of the initiative, another transformation to our concepts would allow the usage of the information as descriptive input for our approach. Thus, the i2map framework could immediately serve for the lifecycle support for Web-based systems developed with any of these methodologies.

#### 5. References

- [1] Ackermann, J., Brinkop, F., Conrad, S. et al.: Standardized Specification of Business Components. <http://www.wi2.info/downloads/gi-files/MEMO/Memorandum-english-final-included.pdf>. 2002.
- [2] Ceri, S., Fraternali, P., and Bongio, A., Web Modeling Language (WebML): A Modeling Language for Designing Web Sites. In 9th International World Wide Web Conference, Amsterdam, Netherlands, 2000, pp. 137-157.
- [3] Dmtf, CIM Concepts White Paper - 2003): <http://www.dmtf.org/standards/documents/CIM/DSP0110.pdf> (26.05.2005).
- [4] Dmtf Homepage, Web-Based Enterprise Management (WBEM): <http://www.dmtf.org/standards/wbem/>
- [5] Hennicker, R., Koch, N., A UML-based Methodology for Hypermedia Design. In Third International Conference on the Unified Modeling Language, York, UK, 2000.
- [6] Kajko-Mattsson, M., Lewis, G.A., and Smith, D.B. Roles for Maintenance and Evolution of SOA-Based Systems. In 11th European Conference on Software Maintenance and Reengineering, Amsterdam, The Netherlands, 2007.
- [7] Kiczales, G., Lamping, J., Mendhekar, A. et al., Aspect-Oriented Programming. In Proceedings of the 11th European Conference on Object Oriented Programming (ECOOP'97), Jyväskylä, Finland, 1997, pp. 220-242.
- [8] Phifer, G., et al., Hype Cycle for Web Technologies. Gartner Research, Stanford, CT, 2006.
- [9] Quirolgico, S., Assis, P., Westerinen, A. et al., Toward a Formal Common Information Model Ontology. In Web Information Systems - WISE 2004 Workshop, Australia, 2004.
- [10] Schwabe, D., Rossi, G., and Barbosa, S., Systematic Hypermedia Design with OOHDM. In ACM International Conference on Hypertext. Washington, USA, 1996.
- [11] Vallecillo, A., et al. MDWEnet: A Practical Approach to Achieving Interoperability of Model-Driven Web Engineering Methods. In Third International Workshop on Model-Driven Web Engineering (MDWE'07), Como, Italy, 2007.
- [12] Turner, M.: Microsoft System Center takes on enterprise IT management market leaders.
- [13] Wilkinson, K., Sayers, C., Kuno, H. et al., Efficient RDF Storage and Retrieval in Jena2. In First International Workshop on Semantic Web and Databases, Berlin, Germany, 2003.