A SERVICE-ORIENTED APPROACH TO EFFICIENT DECISION SUPPORT

Goran Mihelcic[1], Bo Hu[2] and Stefan Pickl[1]
[1]*Institute of Theoretical Computer Science, Mathematics, and Operations Research, University of the Bundeswehr Munich, Werner-Heisenberg-Weg 39, Neubiberg, Germany*
[2]*Faculty of Business Administration, University of the Bundeswehr Munich, Neubiberg, Germany*

Keywords: SOA, Decision support, Participatory modeling, Content & collaboration systems.

Abstract: Historically grown IT-landscapes in modern business, often referred to as 'legacy' systems or 'accidental architectures', can have a strong impact on the efficiency of decision support processes. Caused by a high level of separation of essential software applications, crucial information needed may often be difficult and time consuming to retrieve. The authors show one approach of how distributed business-critical software applications can be integrated using the service oriented architecture paradigm. The implementation of webservices can increase the flexibility and integrity of decision support processes. The authors demonstrate their approach in the context of a participatory system dynamics modelling application.

1 INTRODUCTION

Modern businesses and organizations are dependent and relying on effective and efficient decision making of their employees on all strategic levels. In modern economies, about 40% of the employed staff is predominantly engaged to decision making (Johnson, 2005).

An important role within this context is taken by modern IT-systems that support, control, and streamline often business-critical processes. Based on data and information drawn from these systems (also referred to Business Transaction Systems) and supported by his human-judgement and experience, the decision maker aspires to find the best possible solution to a given problem. Quite often, with the partly rapid growing of companies, the rising demands for further IT-support in newly emerging (process) business areas is met by just adding new applications to the existing IT-landscape. Along with this 'strategy' comes the shortcoming of taking the mutual ability of integration among these software components into account.

The arising problems of e.g. redundant data management and inconsistent information in such "ad hoc-ritectures" or "accidental architectures" (see, e.g., Roth, 1999; Booch, 2006) negatively influence the quality of decision making and therefore the performance of the company. Ultimately, this may result in loosing competitive advantages "even though" significant investments into the IT-infrastructure have been made. As in these cases the reorganization respectively replacement of (parts of) the operating IT-systems often result in direct loss of productiveness, a 'smoother' approach aiming at an increase of the level of integration has to be determined.

Service oriented architectures (SOA) and the introduction of webservices (WS) have proven to be useful in such environments and may lead to a streamlining of the given IT-landscape. In the following, a short review of related works is given and based on a prototype (Hu, 2011) we demonstrate in this paper that SOA, in particularly based on WS, can be used to integrate decision-support functions broadly into enterprise business transaction systems.

2 EXTENDING DECISION SUPPORT TO BUSINESS TRANSACTION SYSTEMS

When it comes to the management of large

Mihelcic G., Hu B. and Pickl S..
A SERVICE-ORIENTED APPROACH TO EFFICIENT DECISION SUPPORT.
DOI: 10.5220/0003759802150218
In *Proceedings of the 1st International Conference on Operations Research and Enterprise Systems* (ICORES-2012), pages 215-218
ISBN: 978-989-8425-97-3

amounts of data, data warehouses are one approach to deal with mass data. In order to transform the given data into valuable business information, a class of information systems called "Business Intelligence Systems" is often applied.

Business Intelligence (BI) systems (in a wider sense) comprise all applications that directly or indirectly provide decision support. These systems include analysis and presentation, as well as, data preparation and storage functionality (Kemper, 2006).

Management Support Systems (MSS) can be described as the early predecessors to BI systems. They focused – similarly to BI systems – on "the use of computers and related information technologies to support managers" (Scott, 1983).

Nowadays, competitive businesses have to support decision making not exclusively on higher strategic management levels but furthermore have to support important decision making processes at all strategic levels, and therefore consistently down to the operational levels. Accordingly, an appropriate IT support has to be provided that may e.g. be implemented by a business transaction system.

BI systems build the core for business transaction systems (see e.g. June et al., 2008) that constitute the base application for effective decision making of business personnel. As a front end layer, business transaction systems provide the decision maker with important data and information about the company. In order to establish a web based interface to the user, these front end layers are often implemented as or integrated into content management and collaboration systems that may be accessed using a web browser.

When it comes to the question of closing the gap between the core BI system (or integrating any kind of application from the back end) and the business transaction front end, one approach that can be applied is the service oriented one.

A service oriented (software/system) architecture is an abstract description of how services can be provided, searched for, and applied within a network infrastructure. One of its central features is its platform independency which enables a (business) process oriented view on central (service oriented) software components (Melzer, 2010).

Services can be described as (small) applications or software components that fulfill a certain (sub-)task and that may be accessed via a network. In order to be locatable, services have to provide a public description of their interfaces. Following the design aspect of "information hiding" the implementation, respectively the code of the service, is invisible to the user (ibid.).

One way of implementing and technologically representing the concept of a service is the development of a respective webservice. A common definition of webservices is provided by the W3C (World Wide Web Consortium) (W3C, 2004):

"A Web service is a software system identified by a URI [...], whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols."

An important precondition to the introduction of webservices is the necessity of having thorough knowledge of the underlying business processes. They represent the basic layer for the development of a webservice structure. A business process can use a variety of different webservices from its starting to its end point. A stringent business process management approach will support the understanding of the vital corporate procedures and will furthermore enable a consistent implementation of business processes into corporate IT systems.

The ability of connecting different software systems combined with its platform independency makes (web-)services a crucial factor when a (re-) integration of historically grown "accidental architectures" is intended.

The following case study shows a simple example of how webservices can be used in the context of a heterogeneous software landscape and how it provides flexible support to a decision support system.

3 CASE STUDY: INTEGRATION OF SYSTEM DYNAMICS MODELING INTO A CONTENT AND COLLABORATION SYSTEM

Participatory System Dynamics (SD) modeling can be seen as an important option to support decision-making processes (Andersen, 1997).

Figure 1 shows a use case diagram of a web-based participatory modeling platform which aims to support intensive collaboration between the modelers and the decision-makers. When a decision-maker views the model, comprehensive explanations about the model structure are displayed step-by-step in synchronization with the model which is displayed graphically using any recent web browser (Hu, 2011).

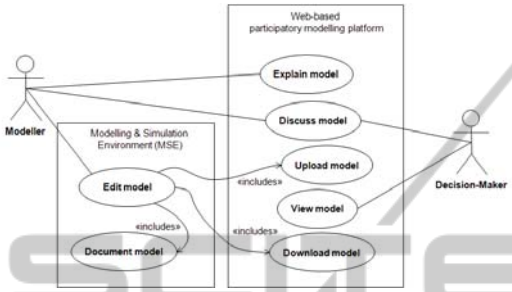

Figure 1: Use cases diagram for a web-based participatory modeling platform.

Compared to a standard content and collaboration system the only key function to be implemented was a server-side viewer for System Dynamics models. As shown in Figure 3 the modules for the viewer "Mdl2Xml" and "Xml2Img" are implemented using Microsoft's .NET technology and connected natively to an Xml-centric content and collaboration system which is also based on .NET technology and provides basis functions like file repository and role based access control as well as tools supporting collaboration. If another content and collaboration system or another server is preferred SOA, especially WS is the right choice of architecture pattern.

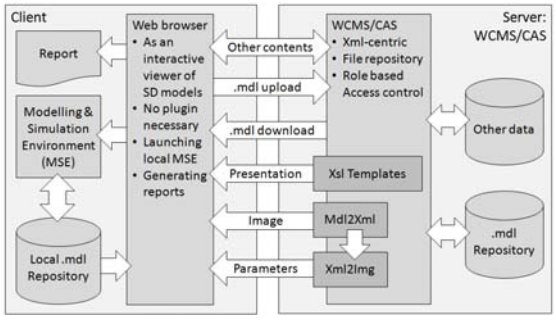

Figure 2: Native connection of modules supporting participatory modeling to a content and collaboration system natively.

The server-side viewer for System Dynamics models is implemented as a webservice and can be integrated into any content and collaboration system which is capable to provide an interface calling webservices, as depicted in Figure 3. Also in the given case the basic functions and collaboration tools are provided by the content and collaboration system into which the server-side model viewer is integrated. The webservice itself has not to manage the persistent data directly.

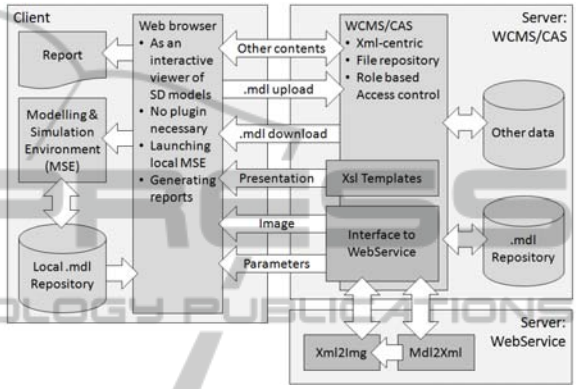

Figure 3: Integrating modules supporting participatory modeling into a content and collaboration system using webservices.

The webservice shown in the figure above is a loosely coupled part of the integrated application and can easily be transferred and integrated into different application contexts. If furthermore, during the productive use of the given application, the need for a different type of visualization occurs, the webservice can comfortably be adapted to the changes needed. Furthermore, different webservices from internal or external providers can be flexibly interchanged.

4 SUMMARY AND OUTLOOK

In this paper the authors described one way of dealing with grown IT-landscapes or so-called "accidental architectures" by choosing the service oriented architecture approach. In particular the application of webservices has been explored and implemented within a case study. This approach has proven to be promising in the developed model case.

Besides the major advantages of being platform independent and enabling connectivity

between different software systems, the concept of webservices also brings along a variety of disadvantages. Besides security issues (e.g. authentication) and performance problems caused by the overhead of XML-message handling, the high effort of integrating legacy systems ("accidental architectures") has to be carefully examined in its relation to the achievable benefit. Especially when the coding and providing of the service is outsourced, further sensitive issues arise as e.g. the dependency on the provider as well as his failure rate and trust issues etc. (Finger, 2009)

Being aware that real-world IT-landscapes in modern businesses are far more complex, the authors still believe that the paradigm of SOA can be one method to increase the level of integration among historically grown and disconnected software applications. Further research of the authors aims at the verification of this hypothesis in the context of larger IT-landscapes.

REFERENCES

Andersen, D. F., Richardson, G. P. and Vennix, J. A. M., 1997. Group model building: adding more science to the craft. *System Dynamics Review, 13 (187–201)*

Booch, G., 2006. The Accidental Architecture. *IEEE Software, Volume 23 Issue 3 (9 - 11),* ISSN 0740-7459, IEEE Computer Society Press, Los Alamitos, CA, USA, May 2006

Finger, P., Zeppenfeld, K., 2009. SOA und WebServices. *Informatik im Fokus, O. Günther, W. Karl, R. Lienhart, K. Zeppenfeld (Hrsg.),* 1. Auflage, Springer, Dordrecht

Hu, B., Leopold, A., 2011. Web-based Participatory System Dynamics Modelling – Concept and Prototype Development. *The 2011 International Conference of the System Dynamics Society, Proceedings,* Washington DC, July 24 – 28

Johnson, B. C., Manyika, J. M., Yee, L. A., 2005. The Next Revolution in Interactions. *The McKinsey Quarterly, Number 4 (20-33)*

June, J. P., Dan, O., 2008. New Perspectives Computer Concepts. USA: Thomson Course Technology, 10th Edition

Kemper, H.-G., Mehanna, W., Unger, C., 2006. Business Intelligence – Grundlagen und praktische Anwendungen. Eine Einführung in die IT-basierte Managementunterstützung. 2. Auflage, Vieweg & Sohn Verlag, Wiesbaden

Melzer, I., 2010. Service-orientierte Architekturen mit Web Services. 4. Aufl., ISBN 978-3-8274-2549-2, Spektrum Akademischer Verlag

Roth, C., 1999. Web "Ad Hoc-ritecture" Pain. *Reining In the Chaos - 3rd 1999 Trend Teleconference,* MATA Group

Scott, M. M., Rockart, M. J., 1983. Implications of changes in information technology for corporate strategy, in: *Interfaces 14: Jan-Feb,* 1984, MIT, (84-95)

World Wide Web Consortium, 2004. Web Service Architecture Requirements, http://www.w3.org/TR/wsa-reqs/ (28.08.2011)