

Workflow-supported Creation and Administration of XML-based Sustainability Reports

Ralf Isenmann¹, Tobias Jordan², Jorge Marx Gómez³

Abstract

Sustainability reporting is an approach striving for sustainable development which meets environmental, economical and social aspects. It is supported by indicators for sustainable managing as well as by the integration of external costs. In this paper we present the conception of a software prototype for workflow-supported creation and administration of XML-based sustainability reports.

1. General

Today, companies are moving from an outdated stage, of simply providing brochures on print media which are usually prepared as "one size fits all" documents, towards an advanced internet-based online approach (Brosowski et. al. 2004). Such an approach enables companies to provide detailed and substantial information, in order to automate the overall reporting workflow and to create reporting tools on different media and in various presentation styles (Isenmann et. al. 2005). The creation and administration of sustainability reports in an enterprise is characterized by a huge number of involved persons of different organizational units. The efficient collaboration of these persons is only possible with the use of a software tool, which is capable to map the work routines (workflows) and furthermore allowing an output oriented structuring of the creation and administration processes. Besides the mapping of the workflows, the software tool should be able to support the simultaneous work of several persons at the same part of a report. These tasks require different resources and specific expertise. Therefore a number of people is involved, e.g. an editorial team, the environmental department, special task forces, accountants, lawyers, public relations specialists, auditors and external consultants (GRI 2002). Within the scope of this project a draft of a software tool for the creation and administration of XML-based sustainability reports has been developed and implemented as a prototype.

2. Conceptual model

The software tool should allow the modelling of workflows in an own, pragmatic modelling language. The report components and the responsible persons are assigned to a single activity of the workflow (Jablonski 1997). Beside the manual processing of report components an automated processing is planned. The contents of an automated processing should be obtained through XML interfaces of different data suppliers, as for example ERP systems or operational industrial environmental information systems. The introduction and application of the software tool offers the following advantages:

¹ Universität Bremen, Institut für Projektmanagement und Innovation, Wilhelm-Herbst-Strasse 12, D-28359 Bremen, E-Mail: ralf.isenmann@innovation.uni-bremen.de, Internet: <http://www.innovation.uni-bremen.de>.

² Otto-von-Guericke-Universität Magdeburg, Institut für Technische und Betriebliche Informationssysteme, Universitätsplatz 2, D-39016 Magdeburg, E-Mail: tobi@wi2000.de.

³ Universität Oldenburg, Abteilung für Wirtschaftsinformatik, Ammerländer Heerstr. 114-118, D-26129 Oldenburg, E-Mail: marx-gomez@wi-ol.de; Internet: <http://wi-ol.de/>.

- Through the XML pattern-oriented sequence control of the relevant activities it is ensured that certain report components must be finished, before the preprocessing of a next report component can begin. For example, the statement of the general manager about certain key data assumes that such numbers are available in the report.
- The quality of the report components can be controlled and significantly improved through editorial staff or specific controlling activities. Furthermore, it is guaranteed that all report components do exist before the whole report is published (Müller 2005).
- Input errors or incorrect content structures can be avoided by the use of input masks, which provide input fields according to the content types agreed in the XML pattern.

Figure 1 illustrates the software composition and the interfaces to external systems in an UML deployment diagram. The component *ui* belongs to the *presentation layer* of the system. It provides the HTML interface for the clients (web browsers). The *ui* component accesses the *business logic layer*. It consists of the *workflowEngine*-, the *model* and the *di* component. The *workflowEngine* includes together with the *model* the application business logic. The *di* component provides the XML interface to external data suppliers. The *data management layer* is represented by an external *database management system* which contains data of both the *model* and the *reportBase* for the administration of the sustainability reports or report components (Jordan 2005).

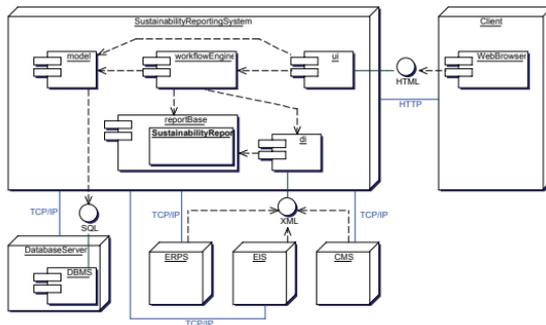


Fig. 1: System Design (UML Deployment Diagram)

3. Prototypical implementation

The implementation of the software tool focuses on the application of Java technology and XML. The presentation layer was developed using the open source framework Jakarta Struts. It allows the separation

of presentation, application logic and data management (model-view-controller pattern) under the use of Java Servlets or Java Server Pages (JSP). The JSP pages contain the information about the presentation, while the navigation logic is encapsulated in a central Controller-Servlet, which is provided by Struts and thus does not need to be implemented. With the request of a JSP page all necessary data will be taken from so called forms. In this case the forms serve merely as a wrapper for the access to Java Enterprise Beans (Backschat et. al. 2002), contained in the business logic layer.

The business logic layer builds up on the Java 2 Enterprise Edition (J2EE). It allows the development of scalable, robust and platform independent applications for a heterogeneous environment. The central access to the business logic layer occurs through a businessDelegate. A businessDelegate encases the access to the business logic layer and provides a well defined interface between presentation and business logic layer. The actual business logic is included in Enterprise Beans. Enterprise Beans are part of the architecture specification of Enterprise JavaBeans that aims at the realization of component based, transaction oriented, and semi-layer applications in Java (Schmietendorf et. al. 2002). Session Beans and Entity Beans are also used in the software tool. Session Beans represent the Facade pattern. A facade hides the complexity of a subsystem from the clients. Entity Beans are called by the Session Beans. Entity Beans represent persistent data objects which are commonly retrieved out of a database. In this case Container-Managed-Persistence Entity Beans (CMP Entity Beans) are used. At CMP Entity Beans the EJB container is responsible for the persistence of the beans. The developer has to declare the desired persistence fields and the container itself controls all database accesses.

4. Conclusions and Outlook

This paper outlines the conceptual foundations for a complete implementation of the software tool. The developed prototype covers only a limited amount of use-cases. Therefore it is not yet applicable in a real industrial environment. Within the next step the entire implementation of the software tool is suggested. As far as the interfaces to external IT systems are concerned we propose to support 3 typical external information systems:

- mySAP as ERP-System,
- Umberto as IEMIS and
- RedDot as CMS (Ehlers 2003).

The integration steps are necessary in order to support the whole reporting workflow and to improve the underlying core processes, from material and process flows to eco input-output-inventory. The resulting software product should be tested very carefully in a pilot project afterwards.

Bibliography

- Backschat, M., Gardon, O. (2002): Enterprise JavaBeans: Grundlagen, Konzepte, Praxis. Heidelberg, Berlin: Spektrum Akademischer Verlag.
- Brosowski, J., Isenmann, R., Beisel, M., Marx Gómez, J. (2004): XML Schema for sustainability reports meeting the needs of the GRI guidelines. In Proceedings of the 18th International Conference Informatics for Environmental Protection, S. 184-193.
- Ehlers, L. (2003): Content Management Anwendungen: Spezifikation von Internetanwendungen auf Basis von Content Management Systemen. Berlin: Logos Verlag.
- Global Reporting Initiative (GRI) (2002): Sustainability reporting guidelines. Boston: GRI.

- Isenmann, R., Beisel, M., Brosowski, J., Marx Gómez, J. (2005): Software-Werkzeug für Single Source Cross Media Multiple Requirement Sustainability Reporting. In: Betriebliche Umweltinformationssysteme – Best Practice und neue Konzepte. Shaker, 2005.
- Jordan, T. (2005): Konzeption und prototypische Implementierung eines Softwaretools für die workflow-gestützte Erstellung und Verwaltung XML-basierter Nachhaltigkeitsberichte. Otto-von-Guericke-Universität Magdeburg.
- Jablonski, S., Böhm, M. und Schulze, W. (1997): Workflow-Management: Entwicklung von Anwendungen und Systemen; Facetten einer neuen Technologie. Heidelberg: dpunkt Verlag.
- Lenz, C. (2003): Empfängerorientierte Unternehmenskommunikation, Dissertation.
- Müller, J. (2005): Workflow-based Integration: Grundlagen, Technologien, Management. Berlin, Heidelberg, New York: Springer Verlag.
- Schmietendorf, A., Dimitrov, E. und Dumke, R. (2002): Enterprise JavaBeans. Bonn: mitp-Verlag.