A Component-Oriented Tool-Approach to Enterprise Architecture Management

Jan Stefan Addicks[1], H.-Jürgen Appelrath[2], Philipp Gringel[1], Matthias Postina[1], Ulrike Steffens[1], Christian Zillmann[1]

[1]OFFIS-Institut für Informatik
Escherweg 2, 26121 Oldenburg – Germany

[2]Praktische Informatik, Abteilung Informationssysteme,
Carl von Ossietzky Universität Oldenburg
Ammerländer Heerstraße 114-118
26129 Oldenburg – Germany

1 Motivation

The discipline of Enterprise Architecture Management (EAM), while becoming firmly established with global enterprises, often remains unnoticed by medium-sized enterprises. In global enterprises the role of enterprise architects has been established for years, and they are equipped by EAM tools with underlying EA frameworks such as the Zachman Framework (Zachman 1987, p. 276-292) or TOGAF (The Open Group 2009, p. 7). In smaller companies, however, the role of an enterprise architect may not even be defined or is poorly supported in terms of resources, budget, and influence. Peyret (2007) reports that, in the year 2006, only 26% of the small and medium-sized companies made use EAM tools.

Yet, the need for a proper alignment of business and IT is more than ever a key to success - especially for medium-sized enterprises competing on a global market. EAM is a means to achieve this alignment, providing a holistic view of an enterprise's different architecture domains such as business, application, data and infrastructure (The Open Group 2009, P. 10). With respect to the complexity of these architecture domains and their relations, management tools play an important role in order to keep track of the architecture development and the business IT Alignment (Technische Universität München 2008, P. 23-37).

Tool vendors are most often focusing on large enterprises, providing almost universal support for the entire architecture. This leads to complex and expensive tools requiring huge effort in customizing and professional training courses for the designated staff – a barrier hard to overcome by medium-sized enterprises.

This paper demonstrates an approach to provide entry points to EAM for (medium-sized) enterprises. After a quick glance on related work in the field of EAM in section 2, we identify tool requirements of medium-sized enterprises (section 3) and show how the OSGi implementation Equinox Server-Side could be used as a flexible platform for building a component-oriented toolset to meet these special requirements (section 4). As a proof of concept, section 5 describes a prototypical implementation relying on an extensible meta model in conjunction with an initial set of components in order to form a flexible EAM tool for medium-sized enterprises. Before this contribution is summarized in section 7 along with a brief outlook on future work, section 6 presents a practical usage scenario of our tool. In that scenario, we explicitly refer to both the requirements formulated in section 3 as well as to general requirements raised in a tool survey accomplished in 2008 by TU München (Technische Universität München 2008, p. 5).

2 Related Work

Offering independent EAM parts which can be assembled and gradually refined into enterprise-specific solutions is one important step in order to transfer the benefits of EAM tool-support to medium-sized enterprises. The necessity of adaptable solutions has already been recognized by EAM research and resulted in a number of conceptual approaches to pre-structure the introduction of EAM.

Winter and Fischer (2006, p. 1-4) present essential EA layers and dependencies between them. The elements of these layers can be aggregated and hence enable different levels of abstraction depending on the specific situation within the enterprise. The business architecture layer, for instance, could document business processes, which can be refined into sub processes, which again can be refined into business activities. From an abstract point of view it is sufficient to only consider the topmost aggregation level of a layer and interconnect the layer's elements on this aggregation level to the other architectural layers. Enterprise specific concerns, however, might call for further more detailed elaboration of certain layers. This flexible conceptual perspective conforms to the technical approach presented in this paper. The structures presented in (Winter and Fischer 2006, p. 1-4) can be regarded as one possible starting point for a set of EAM modules providing different ranges of functionality.

Another rather conceptual approach to a stepwise adoption of EAM is presented in Buckl et. al. (2008b, p. 17) who propose a pattern-based approach to construct EAM enterprise models. This is another step towards enterprise-specific EAM, since holistic models do not seem to be adequate due to reasons of complexity and inappropriate model structures and hinder the introduction of EAM in medium-sized enterprises. The EAM pattern approach was refined and consolidated in a pattern catalog, which is intended to support organizations in the EAM procurement phase by offering informal models, methodologies, and viewpoints with regard to certain concerns. For our approach, such concerns may, again, be the starting point for implementing modules to meet the respective enterprises' requirements. Modules may even completely be based upon some of specified patterns.

Componentization of tools has not yet been considered in EAM research. However, there are a number of commercial tools which claim to be extensible by some means or other. One example is ARIS IT Architect, which is integrated into a product suite and thus can be extended by other off-the-shelf ARIS components. Yet, these components themselves are static and ARIS offers no interfaces for adding own EAM software components. Other professional tools are extensible on code level. Typically, this extensibility is restricted to certain functionalities. A popular example is the possibility to add individual graphical visualizations or tabular representations to an existing EAM tool based on the static enterprise model offered by this tool. Telelogic's System Architect can serve as an example for this kind of extensibility. Nevertheless, there are no tools available that provide well-defined logical interfaces for adding new EAM functionality. A third category of extensible EAM tools are those which rely on a highly comprehensive enterprise model and allow for the realization of organization-specific EAM procedures by customizing this enterprise model (Buckl et. al. 2008, p. 21). However, this customization is time consuming, expensive and typically calls for direct vendor support. Against the background of this state-of-the-art of EAM tools, a modular approach like described in this paper seems to be a promising step towards real tool extensibility and hence towards adaptability of EAM tools with regard to enterprise-specific requirements.

3 Tool Requirements of Medium-Sized Enterprises

In consulting both large and medium-sized enterprises on a daily basis, we experience a strong demand for solutions to solve enterprise-wide architecture concerns. Depending on the size and sector of the enterprise, such concerns vary along multiple dimensions:

1. The **Complexity** of the enterprise model in terms of modelled architectures (business, data and application, infrastructure) and the amount of needed viewpoints and complexity of addressed concerns.

2. The **Budget** for enterprise architecture development comprising the size of the architecture team and available resources (time, tools, management support) for EA. This aspect is important since the introduction of an EA management tool represents a major investment (Technische Universität München 2008, p. 1).

3. The **Skills** in the domain of Enterprise Architecture including qualification of engaged staff, usage of frameworks and techniques and usage of EAM tools.

The maturity of these dimensions and of the discipline of EAM in general normally depends on the size of the enterprise. Medium-sized enterprises lag behind global enterprises along these domains. Especially the alignment of EAM and the system of enterprise-wide management with periodic re-consideration of strategic components of the business such as mission, vision, strategies, and action plans, is often not pursued systematically by medium-sized enterprises (Bernus et. al. 2003, p. 345).

On the other hand, strategic re-consideration on the business-side and the flexible alignment of the enterprise architecture is achieved with less strain by smaller organizations, as the necessary communication on architectural aspects is often less formal and hence faster. Nevertheless, architectural concerns are omnipresent and the related models and corresponding data is too complex to be dealt with manually and from scratch – even in some small enterprises.

Vendors of EA tool suites promise to address architecture concerns on a global scale and to support EA management adequately. Such tools are heavily used among large enterprises and support EA architects to a varying degree in all three dimensions. Unfortunately, budget limitations and limited skills hinder medium-sized enterprises to adopt tool-supported enterprise architecture. License fees are one reason for this. The pure complexity of most tools and the therefore expensive and time consuming training is even more relevant, especially when the role of the enterprise architect is performed in part-time – a typical scenario for medium-sized enterprises where long-term architecture development often stays behind operational demands.

Regarding the dimensions complexity, budget, and skills in medium-sized enterprises, we deem the following requirements relevant for adequate tool support:

3a) **Minimal Costs** related to the initial purchase of the tool or yearly license fees and maintenance costs for deploying and running the tool.

3b) **Ease of training** for the different stakeholders. This includes initial training as well as provision of adequate views. Most stakeholders demand only limited functionality - so the tool should be flexible on the relevant scope and training level (Kaisler et. al. 2005, p. 3).

3c) **Adequate complexity** – A tailored tool with a slim enterprise model addressing only the relevant aspects is needed for medium-sized enterprises where not all architecture layers (business, data, application and infrastructure) have become relevant yet. The focus should be settable to key areas of interest, where fast returns are most expected.

3d) Adaptability – Most likely the enterprise architecture initiative of a medium-sized enterprise will begin small and grow over time. At least in our experience such initiatives often start bottom up when a certain amount of complexity in one or more architecture layers can no longer be handled manually or in Excel files any more. Hence, the envisioned tool needs to be able to grow in the same manner - in conclusion a flexible, adaptable and modular approach is essential.

When it comes to the acceptance of EAM in medium-sized enterprises, a modular approach for EAM tool support which enables the tool to grow according to the progress of an enterprise's EA initiative and which relies on a flexible meta model seems to be promising. To this end, we present our OSGi-based EAM tool prototype referencing the requirements listed above where applicable.

4 Technology

Starting from the tool requirements for medium-sized enterprises listed in section 3, we evaluated suitable technologies. Requirement 3a) – providing a solution at minimal costs – combined with requirement 3d) – the need for an adaptable solution – led us to the decision to develop a web-based open source enterprise solution in Java. Hereby we conform to Peyret (2007) whereupon enterprises have called for web-based user interfaces for EAM tools already in 2006. Since Java and especially the Java web server infrastructure does not provide adequate modularity support to allow for the composition of modules (Kaegi and Deugo 2008, p. 688-689) with dependency management or hot deployment, the OSGi Service Platform[1] was selected.

The OSGi Service Platform is not an applicable implementation in the strict sense, but a standard created by companies like IBM, Nokia or Oracle. Yet, some implementations of this standard exist, most of them being open source. The best known open source implementation besides commercial ones like Knopflerfish pro[2] or mBedded Server Professional Edition[3] is probably the Equinox implementation of the Eclipse Foundation (Wütherich et. al 2008, p. 1 - 10) used for instance by the Eclipse IDE (Gruber et. al. 2005, p. 292). As we mentioned in requirement 3a), easy deployment of the EAM tool at minimal cost is an important issue for medium-sized enterprises. Hence, web-based solutions come into play where a simple browser is sufficient on the client side reducing maintenance effort and cost. Thus, the OSGi implementation has to support the development of server based applications, a requirement not fulfilled by Equinox itself (only desktop application development is supported) but by Equinox Server-Side which is

[1] http://www.osgi.org

[2] http://www.makewave.com

[3] http://www.prosyst.com

designed for modularising web applications (Wütherich et. al 2008, p. 1 - 10) and is also developed by the Eclipse Foundation.

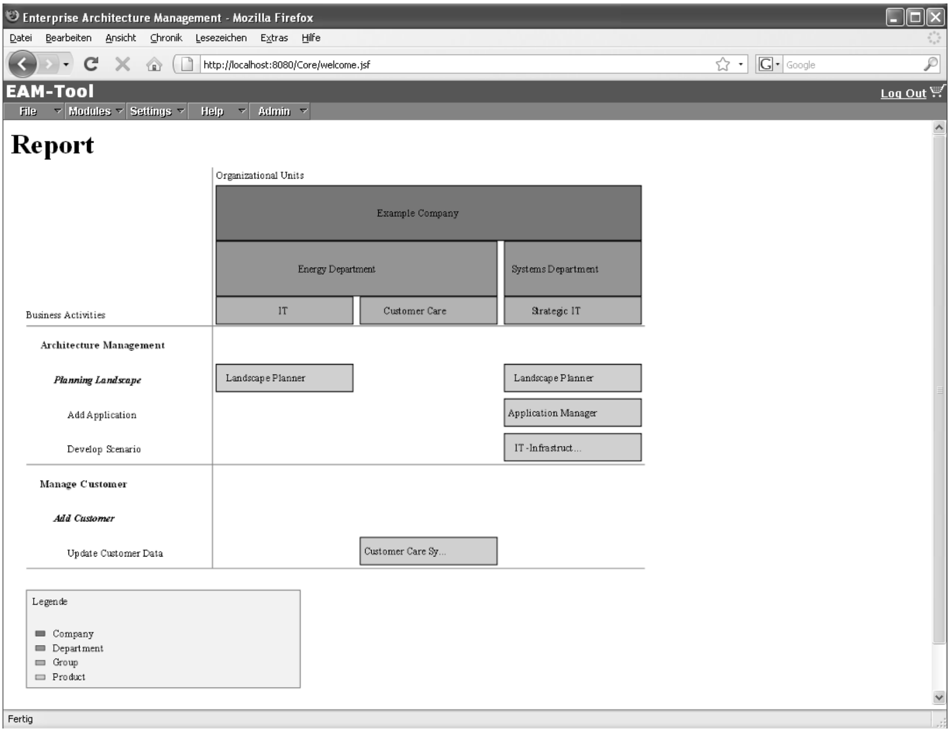

Figure 1: Screenshot of the Prototype – Visualization-Bundle

The OSGi alliance calls modules "bundles" and we adopt this term throughout the remainder of this paper. One benefit of OSGi is the reduction of complexity by splitting functionality into bundles, by making bundles reusable, and by supporting versioning and deploying new functionality at runtime. OSGi was first designed with embedded systems in mind but is used for many other application areas today, for instance desktop or mobile applications. A modular EAM tool implemented on top of the OSGi Service Platform enables companies to focus on the functionality (in terms of bundles) they really need and thus to manage their enterprise architecture in their own enterprise-specific manner. An example for an EAM bundle would be a visualization-bundle, which produces diagrams like the one shown in Figure 1, meeting specific enterprise concerns.

In a nutshell, the use of OSGi is aimed to provide our prototype with the flexibility and customization facility for different kinds of enterprises. Our prototype is intended to show how a cost-effective modular EAM tool can be designed. Therefore, besides the open standard (OSGi) we only make use of open source software. This is mainly Equinox Server-Side, Hibernate for persisting enterprise models and data and JavaServer Faces (JSF) for creating user interfaces.

5 Realization

After having shortly introduced the technology used, this section focuses on the component-oriented architecture of the implemented prototype. Figure 2 contains a general overview. The upper middle shows the core bundle, bundling mandatory functionality needed by all installations regardless of the actual overall bundle configuration. It provides a meta model as basis for creating enterprise models as one of the core functions as well as the support of user management with corresponding operations like managing rights, roles and groups. The definition of data access rights on attribute level and the creation of views (a user should only see data, pages, menu entries etc. according to her permissions) is also part of the core user management.

The core bundle accesses the database through the OR-hibernate layer and offers a public interface for other bundles to retrieve information from the database, partial information on the enterprise model as well as EA instance data for example. Next to Java interfaces, bundles can also offer XML-based declarative services (Tavares and Valente 2008, p. 4) – similar to web services – which can be used by other bundles. Possible bundles' interdependencies defined in the manifest of each bundle are managed by the management agent of Equinox Server-Side.

An example for a service offered by the core- undle is the menu service. The core bundle implements a menu which enables users to navigate to different areas in the prototype. Other bundles can use the menu service to register menu entries with the core, where positions and labels of the entries are submitted during the initialization process of the bundle added.

As stated, the core bundle offers only limited EAM functionality. Depending on the specific circumstances in the particular enterprise, more sophisticated management and analysis functions can be adopted by extension bundles.

To extend our prototype and to demonstrate the potential of our component-oriented approach, we implemented some extension bundles. One extension implements an extended-data-input bundle which enhances the limited functionality of dealing with enterprise models and instances offered by the core bundle. Another extension bundles a visual reporting facility, configurable to visualize parts of the enterprise architecture defined by the enterprise model.

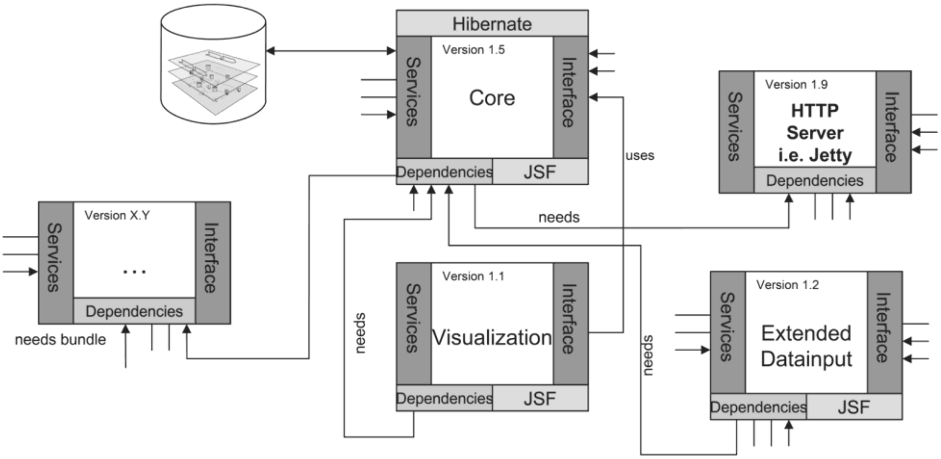

Figure 2: Architectural Overview of the Prototype – Bundle Concept

The modular, component-oriented approach reaches down to the deployment level of our prototype: The application server (Jetty) itself is provided as a bundle and can easily be replaced. Further technical bundles that come with Equinox are essential to run the environment but are irrelevant from an EA perspective.

The adequate complexity requirement (3c) insists on flexible structures to tailor the enterprise model. For the sake of simplicity and to focus on relevant aspects of the specific enterprise's architecture the corresponding enterprise model needs to be adaptable. We distinguish between three abstraction layers for models (see Figure 3) in our prototype. The most abstract model – our meta model – is MOF compliant. The enterprise model is based on the meta model and is used to describe the enterprise architecture model, like for instance the structure needed to describe the application landscape. The last model layer consists of instances of the enterprise model and captures the current state of a company's enterprise architecture.

The enterprise model can be created either through input methods directly offered by the core bundle or be modelled with a UML tool like MagicDraw and imported afterwards. The meta model is standard MOF and cannot be changed. Based on the enterprise model, instances can be created by dynamic input forms provided by the core bundle. Besides import functionality, the core bundle also offers an export function for instance data and enterprise models.

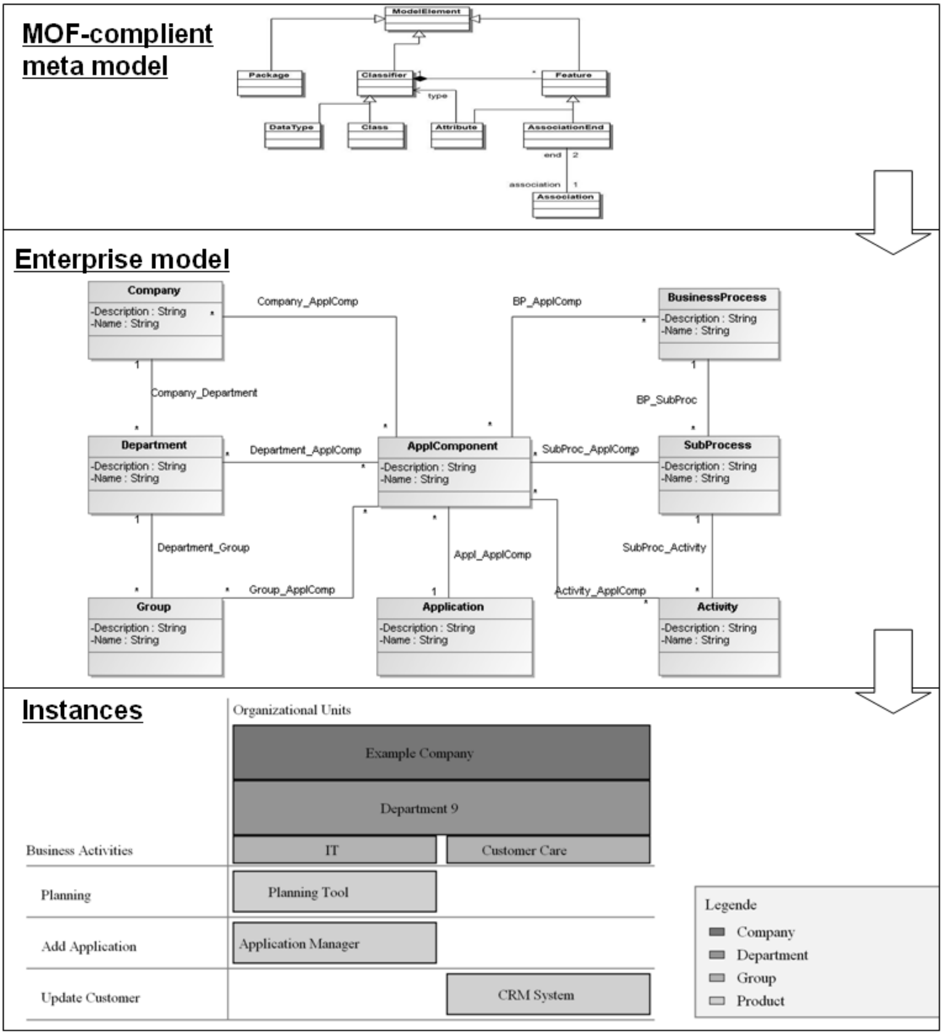

Figure 3: Models of our Prototype

6 Practical Usage Scenario

To demonstrate the relevance of our component-oriented approach, we relate the prototyped bundle functionality (see Chapter 5) to seven out of ten *Scenarios for Analyzing specific Functionality* published by TUM/sebis in the Enterprise Architecture Management Tool Survey 2008 (Technische Universität München 2008, p. 5), namely 6a) *Adapting the Information Model*, 6b) *Supporting lightweight Access*, 6c) *Supporting multiple Users and collaborative Work*, 6d) *Importing, Editing, and Validating Model*

Data, 6e) *Editing Model Data using an External Editor*, 6f) *Creating Visualizations of the Application Landscape*, and 6g) *Annotating Visualizations with Certain Aspects*.

For our prototype, we focus on the basic functionality covered by the seven scenarios listed above and disregard the *Scenarios for Analyzing EA Management Support* (Technische Universität München 2008, p. 54-77) since they address complex EA-management tasks which we considered too detailed for an initial case. Nevertheless the specific functionality "[…] can be seen [to be] contributing to any of these tasks" (Technische Universität München 2008, p. 40).

We want to open the scenario by describing the initial situation at an exemplary company. This situation can be summarized as follows: The company is a medium-sized enterprise starting an initiative for better business-IT alignment, since during the past two years IT appeared to be a bottleneck when it came to the rollout of innovative and new service products.

The IT department of the company is, spoken in EA terms, responsible for the architecture domains infrastructure and application, whereas the operating departments are in charge of the business and data architecture. Both responsible parties are using typical office tools for listing and displaying information related to their architecture domains. Unfortunately, this information is spread widely amongst the different stakeholders.

The head of the IT department is given the task of consolidating the available information and of providing a structured information base for all stakeholders involved. Besides a general overview of the application landscape, the management has a substantial interest on which applications are actually used by which organizational units supporting which business processes.

Starting the EA initiative at the company and being tool-supported by an initial setup of our prototype, stakeholders begin to model relevant aspects of their enterprise according to our meta model using a UML modelling tool. Hereby, full flexibility for creating a common model including business and IT concerns is guaranteed and the evolved questions raised in 6a) concerning new classes, relationships, and attributes as well as requirements 3c) and 3d) are addressed. The enterprise model of the company is loaded into the repository via an import bundle of our prototype. The core bundle already offers user management functionality and authenticated stakeholders can use the dynamic forms generated by the core to enter data according to the imported enterprise model in their web browser. This feature realizes a lightweight access as requested in 6b) and collaborated work as stipulated in 6c). Since the user will only see GUI elements related to her working tasks and relevant to her role requirements, requirement 3b) is addressed and training efforts can be reduced to a role-centric minimum.

As stated above, architecture-related information is available but widely spread over different stakeholders. We implemented an Excel import/export bundle enabling users to import existing data (6d) and also edit data outside the tool (6e). Such an approach fosters data collection and minimizes cost (3a). To meet the requirements of the company's management department, the head of the IT department decides to plug in a visualization bundle capable at displaying certain aspects of the application landscape (6f, 6g), including a matrix map showing which applications are actually used by which organizational units in support of which business processes (Figure 1).

7 Conclusion and Future Work

In this paper, we introduced a modular approach to EAM tool development. Modularity of EAM tools allows for both, software extension as well as software updates in order to meet specific enterprise requirements. This is especially true and beneficial for medium-sized enterprises. These enterprises are often unable to cope with all-embracing tools which come along with complex static enterprise models requiring enormous effort either in data acquisition or in preliminary customization. Technically, the modular architecture is based on the OSGi service platform, a plug-in framework known from the Eclipse context.

We described how an initial enterprise architecture project in a medium-sized enterprise could be supported by our approach. The prototyped bundle functionality was successfully related to the majority of scenarios described by TUM/sebis in the Enterprise Architecture Management Tool Survey 2008 (Technische Universität München 2008, p. 5).

Future versions of our software are intended to be published as open source software raising the hope for a broader dissemination of both, the approach as well as the related software artefacts. For further evaluation, a detailed comparison with other tools regarding relevant technical issues and conceptual EA aspects is targeted. Parts of this evaluation could go along with the next TUM tool survey.

In the long run, a modular approach to EAM tool development may lead to new market situations and business models, since vendors may provide specific and sophisticated functionality in form of plug-ins instead of shouldering the effort of developing broad EAM tool suites. Possibly, even establishing of an open source software market for EAM is conceivable. Both aspects could lead to reduced EA introduction costs for organizations, in particular for medium-sized enterprises, which again might improve the business-IT alignment within these enterprises.

Comprehensive frameworks like TOGAF raise typical EA views, partial models, patterns, typical concerns etc. which can be used as conceptual foundation leading the implementation of single EA modules. These can then be arbitrarily assembled and reassembled in ever-improving individual EAM solutions.

References

Buckl S, Dierl T, Matthes F, Ramacher R and Schweda CM (2008a) Current and Future Tool Support for EA Management. In: Steffens U, Addicks JS, Streekmann N (Editors). Workshop MDD, SOA und IT-Management 2008, Berlin. GITO-Verlag.

Buckl S, Ernst AM, Lankes J and Matthes F (2008b) Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008). München.

Bernus P, Nemes L, Schmidt G (2003). Handbook on enterprise architecture. Springer, Heidelberg.

Gruber O, Hargrave BJ, McAffer J, Rapicault P, Watson T (2005). The Eclipse 3.0 platform: adopting OSGi technology. In IBM Systems Journal, 44(2):289 - 299.

Kaisler SH, Armour F, Valivullah M (2005). US Senate. Enterprise architecting: critical problems. In System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference.

Kaegi SR and Deugo D (2008). Modular Java web applications. In Proceedings of the 2008 ACM symposium on Applied computing ACM Press New York, USA.

The Open Group (2009). TOGAF Version 9. Van Haren Publishing.

Peyret H (2007). The Forrester Wave™: Enterprise Architecture Tools, Q2 2007. Boston.

Technische Universität München 2005, Chair for Informatics 19 (sebis). Enterprise Architecture Management Tool Survey 2005. München.

Technische Universität München 2008, Chair for Informatics 19 (sebis). Enterprise Architecture Management Tool Survey 2008. München.

Tavares A and Valente MT (2008). A gentle introduction to OSGi. In SIGSOFT Softw. Eng. Notes, 33. ACM Press New York, USA.

Winter R and Fischer R (2006). Essential Layers, Artifacts, and Dependencies of Enterprise Architecture. 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW'06).

Wütherich G, Hartmann N, Kolb B, Lübken M (2008). Die OSGi Service Platform – Eine Einführung mit Eclipse Equinox. dpunkt.verlag.

Zachman JA (1987). A Framework for Information Systems Architecture. IBM Systems Journal, 26(3):276–292.