Tool Support for Enterprise Architecture Management - Strengths and Weaknesses

Alexander M. Ernst, Josef Lankes, Christian M. Schweda, André Wittenburg
Technische Universität München
Chair for Informatics 19
E-mail: {ernst,lankes,schweda,wittenburg}@in.tum.de

Abstract

This paper sketches the approach we have taken in an extensive survey to evaluate existing tools providing support for enterprise architecture (EA) management. From there, we outline important strengths and weaknesses of existing EA management tools discovered in the survey and suggest some approaches for improving the tool support available in this field.

The tool evaluation, to which the paper refers, is built on a definition of EA management, which we operationalized via a set of scenarios, which constitute a sufficiently concrete base for making precise statements about the tool support. The paper outlines some scenarios, which are constructed to be representative for EA management, and therefore cover typical tasks. In order to ensure the representativeness of the scenarios, they were validated by ten of our industry partners.

Considering the results of the evaluation, which covered nine EA management tools, we show characteristic strengths and weaknesses common to the tools. From this analysis, which e.g. covered the tools’ metamodeling capabilities, the methodologies they offer to address EA management, and especially their visualization-specific capabilities, we suggest areas for further improvement regarding the software support for EA management. This is meant as a contribution to the field of EA management, which we view as not adequately addressable in practice without tool support due to the inherent complexity of this field.

1 Motivation

Nowadays organizations operate hundreds or even thousands of business applications in order to support their business processes in line with their business strategy. Besides supporting these business processes at different organizational units, the business applications are e.g. further development in projects, have connections to each other via interfaces, or are based on architectural blueprints, just to name a few of their aspects relevant to EA management.

The concepts mentioned above are considered to be part of the enterprise architecture (EA), which encompasses Enterprisewide Technical Architecture (EWTA), Enterprise Business Architecture (EBA), Enterprise Information Architecture (EIA), and Enterprise Application Portfolio (EAP) according to the definition in [10].

Taking into account that these areas target quite widespread functional areas at an abstract level, it becomes obvious that EA management can become a complex task. The considerable information that thereby needs to be stored and managed, the distributed teams potentially involved, and the complex interdependencies of the relevant concepts alluded to above, contribute to the fact that tool support is essential to handle EA management.

As there is, according to [8], no established definition of EA management, we use the following definition here, which was also employed in our Enterprise Architecture Management Tool Survey 2005 [13]:

"EA management is a continuous and iterative process controlling and improving the existing and planned IT support for an organization. The process not only considers the information technology (IT) of the enterprise, but also business processes, business goals, strategies, etc. are considered in order to build a holistic and integrated view on the enterprise.

The goal is a common vision regarding the status quo of business and IT as well as of opportunities and problems arising from these fields, used as a basis for a continually aligned steering of IT and business."

Thus EA management is the discipline of managing the whole enterprise architecture and the artifacts building the enterprise architecture. On the one hand, EA management should be driven by and adjusted to the business strategy. On the other hand, IT should play the role of a supporter or
even an enabler for the business strategies and goals. Therefore, one of the main goals of EA management should be the “Alignment of Business and IT”.

As already described before, EA management is a very extensive and complex task concerning many different elements of the enterprise architecture and also many different roles like Process Modelers, Information Architects, etc. Therefore, EA management should be supported by tools, which support distributed access to consistent data, offer the possibility to structure the information managed, and also aid users in filling out their role in the EA management process.

The interest of our industry partners in information about EA management tools was one of our key motivators for our Enterprise Architecture Management Tool Survey 2005, which we conducted together with ten industry partners, AMB Generali Informatik Service, BMW Group, Deutsche Börse Systems, Deutsche Post, HVB Systems, Kühne + Nagel, Münchenner Rück, Siemens, T-Com, and TUI. In this survey we analyzed the products of nine major vendors in the EA management tool market: Adaptive EAM (Adaptive, Ltd.), ADOit (BOC GmbH), ARIS Toolset (IDS Scheer AG), Corporate Modeler Suite & IT Architecture Accelerator (Casewise, Inc.), MEGA (MEGA International SA), Metis (Troux Technologies, Inc.), planningIT (alfabet AG), process4.biz (process4.biz GmbH), and System Architect (Telelogic AB).

The following section gives a brief overview of literature about EA management. After the discussion of related work, section 3 goes into detail concerning the approach we have taken in our survey. The results are then shown in section 4. Section 5 sketches lessons learned and proposes some improvements to address the weaknesses discovered. The last section summarizes the paper and provides an outlook of how our research project Software Cartography fits into the approaches sketched in section 5.

2 Related work

EA management currently is an important issue in academia as well as in practice. In [8], Langenberg and Wegmann analyzed what aspects of EA management current research is targeting. Thereby they made a conclusion, concerning the number of publications coping with enterprise architecture. Thereby they concluded by the growing number of publications that interest in EA management is continuously growing. We observed the same tendency in regard to the importance of enterprise architecture amongst our industry partners, which may be seen as supportive for the above statement.

According to [8], only some papers concerning EA management could be found before 1996, among them a paper by Sowa and Zachman, who developed a framework for information systems architecture [14]. In recent years the number of frameworks trying to establish an EA management methodology, like e.g. DoDAF [2], TOGAF [15], the Enterprise Architecture Desk Reference [10], etc. has rather increased.

Other articles, like [3] concentrate on developing an exemplary architecture for an EA management tool. Surveys concerning EA management tools are regularly performed by analysts and consultants like Gartner Research, e.g. [6], or Forrester Research, e.g. [12].

3 Scenario Based Operationalization of EAM for Tool Evaluation

In summarizing the state of the art regarding the tool support for EA management, this paper relies on a survey [13], which we conducted after experiencing significant demand for such an evaluation from several industry partners of our research project software cartography, as already described in section 1. While the tool evaluations mentioned at the end of section 2 offer a rather high level view on the market and the respective EA management tools, we wanted to complement such evaluations with an extensive survey more focused on the specific tool functionalities and their application to EA management. This section sketches the approach this survey has taken to evaluate EA management tools.

Basically, our survey took a two tiered approach. On the one hand, we used a list of criteria to evaluate mainly functional aspects of the tools. This list is described in detail in the subsequent section 3.1. As we view that EA management can be adequately supported by tools with largely different sets of functionalities, depending on the specific approach taken, we did not perceive it as sufficient to evaluate solely specific, unconnected functionalities. Therefore, we added, on the other hand, a second step to the evaluation. In this step we created an executable operationalization of our definition and understanding of EA management by developing a set of scenarios, reflecting typical tasks, in a way precise enough to form a basis for a simulation of the aspects we consider EA management to be made up of. The scenarios are introduced in section 3.2.

A cornerstone of our evaluation was the continuous involvement of our industry partners during the design of the list of criteria and the scenarios. As some partners had already chosen a tool, they were able to contribute their insights and experiences to our list of criteria. Furthermore, all of the participating industry partners provided us with exemplary tasks drawn from the field of EA management, similar to the ones they have been encountering in day to day business. This information was used to complement the definition of EA management we sketched above via
scenarios, creating an operationalization of what “supporting EA management” means as a basis for a tool survey.

The approach did not target at establishing a simple ranking of the tools, but at characterizing them according to different aspects, thus showing the strengths and weaknesses of each tool. The survey was performed from January to August 2005 at the Chair for Informatics 19, Technische Universität München.

3.1 List of Criteria

As stated above a major part of the survey was the list of criteria, of which we give an overview here. Basically, the list contained questions about functional, technical, and additional criteria, which we have used for covering different aspects relevant in analyzing EA management tools. As the entire list consisted of approximately 400 questions, it is not included in this article as a whole, but its overall structure is presented below, strongly focusing on functional criteria, which also constituted the major part of the questionnaire.

3.1.1 Functional Criteria

The functional criteria view the tools from a requirements perspective, abstracting from both realization of features as well as their use in the context of specific EA management tasks or processes. These criteria are structured into the following sections: Methodology, Visualization, Integration with related Domains, Integration with other tools, Reporting, Collaboration support, Metadata for Architectural Description, Internationalization/Localization, and Navigation in the Model.

The Methodology section evaluates, to what extent there are aspects of predefined methodologies in a tool and how far it is possible to employ user defined methodologies. Regarding predefined metamodels, which are viewed as parts of a predefined methodology, the size of the predefined metamodel and also its adherence to existing frameworks in the context of EA management like e.g. Sowa and Zachman [14] or DoDAF [2] is taken into consideration. Regarding the creation of user defined metamodels, which is essential to employing user defined methodologies for EA Management, the metamodeling capabilities are evaluated.

The section Visualization is concerned with the tool’s capabilities to display information in a convenient and easily understandable way, going beyond simple tabular and textual reports by e.g. using Boston Square Diagrams (BCG Matrices) or different types of software maps as described in [9]. This section furthermore encloses criteria referring to the tool’s capabilities to reduce the complexity of the visualizations displayed by e.g. filtering. Thereby, also two aspects are taken into consideration: the visualization techniques that are shipped with the tool, and the possibilities to employ user defined visualization techniques.

The section Integration with related Domains evaluates the tool’s capabilities concerning functionalities, that we do not include in the scope of EA management, but can be seen as bordering, such as object-oriented modeling via UML [11], database modeling via E/R diagrams or UML with profiles, or process modeling.

The section Metadata for Architectural Description takes into consideration the tool’s capabilities for describing the views on the enterprise architecture provided, the concerns addressed by these views, and the stakeholders raising these concerns.

Integration with other Tools focuses on integration into portals based on common technical frameworks, like e.g. the Java Portlet API.

Further sections analyze the tools about their capabilities concerning Reporting, Collaboration Support, Internationalization/Localization, and Navigation in the Model.

3.1.2 Technical and Additional Criteria

As an EA management tool has to integrate into the existing IT infrastructure of the enterprise for providing convenient, secure, and flexible access to the data managed therein, the technical criteria focus on aspects of interest in this field such as Hardware/Network/Software Requirements, Software Architecture, Secure Data Transmission, Printing capabilities, Backup/Restore Capabilities, Import/Export Functionalities, and Interfaces for External Data Access. As these are aspects generally relevant to enterprise software and not really specific to the tool support for EA management, they are not further taken into consideration in this paper.

Additional criteria covering information about the tool vendor itself have been included in the list of criteria, but since they were not mandatory to fill out, as some of the respective information might be confidential, they were not relevant to the further evaluation.

3.2 Scenarios

As outlined above, evaluating the tools solely against a list of criteria seemed to us not to be the sufficient for a comprehensive analysis of the tools’ capabilities in supporting EA management. Therefore, we developed, in cooperation with our industry partners, two distinct sets of seven scenarios. The first set is used for analyzing specific functionalities, e.g. visualization techniques and metamodeling capabilities, thereby contrasting the claims made by the tool vendors in filling out the list of criteria with an actual usage of the respective functionality. The second set is used to analyze the tool’s support for tasks and processes typical for EA management. For the survey, each of the scenarios in both sets is described in detail. This begins with the concerns rel-
evant for the EA management task addressed therein, continuing with the questions derived from the concerns and the tasks associated with the execution of the scenario. It concludes with the deliverables showing what the result of the scenario simulation could look like. These deliverables were mainly seen as guidelines, not to be achieved in exactly the same way, leaving open the possibility of alternative solutions as adequate in the different tools.

### 3.2.1 Scenarios for Analyzing Specific Functionalities

The scenarios for analyzing specific functionalities as used in the survey are: Visualization of the Application Landscape, Visualization of Measures, Simplified Access for Readers, Editing Model Data using an External Editor, HTML Export, Metamodel Adaptation, and Large Scale Application Landscape Visualization. The scenario Visualization of the Application Landscape is explained in detail here, as it constitutes the main entry point for the other scenarios. Here, the exemplary data on which the other scenarios are based is entered into the tool. In this paper, the scenario also serves to exemplify the structure used to describe the scenarios in [13]. The other scenarios for analyzing specific functionalities are not further detailed here, focusing on the second set of scenarios, which contains those scenarios more important to showing the strengths and weaknesses of the tool support for EA management currently available (see section 3.2.2).

The scenario Visualization of the Application Landscape starts with initially importing data previously gathered in e.g. a spreadsheet tool into the tool evaluated. The concerns in this scenario are:

"A department store wants to get an overview of its application landscape and its enterprise architecture. This should be accomplished through the creation of three software maps visualizing different aspects of the application landscape: a cluster map (see Figure 1), a process support map (see Figure 2), and an interval map (see [9])."

The three types of maps are introduced in [9] as visualizations for application landscapes. The characteristic feature of a cluster map is that it utilizes the concept of nesting to visualize a relationship between the cluster element and the nested element, as it is shown in Figure 1. In this figure, e.g. the rectangle labeled Online Shop is nested within the rectangle labeled Headquarter. The nesting may e.g. visualize an hosted-at-relationship between an organizational unit and an application system.

The main concept behind the process support map is the alignment of one map symbol in respect to another map symbol. This is visible in Figure 2, which shows an exemplary process support map. In this visualization e.g. the rectangle labeled Inventory Control System is aligned with the chevron labeled Acquisition in respect to the x coordinate, in order to express that the Inventory Control System supports the process acquisition and with the text box Headquarter in respect to the y coordinate to show that it is used at the headquarter.

During the simulation of this scenario the following questions were to be answered:

- Does the tool provide a metamodel capable of covering all aspects of the data about the application landscape?
- Does the tool provide convenient and flexible importing mechanisms and what kind of data formats are supported thereby?
- To what extend does the tool support data manipulation through its user interface and what kind of mechanisms, e.g. graphical modeling, data entry dialogues, for manipulations are provided?
- Does the tool provide mechanisms for creating visualizations similar to those shown in Figures 1 and 2 with shipped functionality and to what extend can the creation be automatized?
- Does the tool support manual adaptations of the visualizations? Is the tool capable of e.g. preserving manual layouting adaptations even if the visualization is re-generated?

![Figure 1. Exemplary cluster map](image)

### 3.2.2 Scenarios for Analyzing EA Management Tasks

The tasks from EA management are analyzed in the second set of scenarios. In order to exemplify the description of a scenario in this set, the scenario Landscape Management is explained in detail.
The scenario Landscape Management simulates tasks relevant to managing the evolution of the application landscape and is thus closely related to project portfolio management, in which concrete procedures and steps for changing the application landscape are managed. The concerns of the scenario are:

"In order to keep information about the future development of the application landscape in the tool there has to be the possibility to create scenarios based on the current application landscape.

The current, planned, and target landscape should be analyzed using three different visualizations. The current landscape represents the landscape as is; the planned landscape shows the landscape as it develops through the changes performed by projects up to a specific date. The target landscape as a long term perspective shows the envisioned architecture of the landscape."

During the simulation of the scenario the following questions have to be answered:

- Does the tool support a visualization of today’s application landscape? Does the visualization comprise similar information as the one displayed in Figure 2?

- Does the tool provide support for visualizing and storing future versions of the application landscape (e.g. a long term perspective)?

- Does the tool provide functionality for highlighting differences between two landscape versions?

- Does the tool support a tracing from the differences discovered to projects potentially causing these differences?

In order to provide an insight, which aspects of EA management are addressed by the scenarios not explained in detail here, a shorthanded overview is provided below: The scenario Management of Business Objects and Business Services simulates tasks concerning the tool’s capabilities to handle business objects, business services, and information flows exchanging business objects between applications. The scenarios Project Portfolio Management and Synchronization Management simulate the management processes concerned with projects. They start with making project proposals, continue with selecting the projects to be actually executed from the portfolio of proposals, and end up in managing project interdependencies especially concerning delays or incidents. The scenario Traceability and Strategy Management addresses issues arising in the context of aligning EA management activities to the organization’s goals and strategies, therefore focusing on capabilities to support e.g. goal decomposition. The scenario Application Architecture Management deals with the introduction and implementation of architectural blueprints for business applications, whereas blueprints are here understood as architectural standards, like e.g. a three-tiered architecture. The scenario Infrastructure Management addresses issues concerning the infrastructure used by business applications, focusing on capabilities to support e.g. life cycle phases for infrastructure components.

3.2.3 Scenario Simulation

In order to simulate the scenarios in an adequately realistic and comparable (among the different tools) way, we created exemplary data around a fictitious department store. Therefore, we developed a simplified metamodel to be able to model and store the data needed for simulating the scenarios.

The scenarios have been simulated with every tool by a team of two people, recording the information gained throughout the simulation process. Especially with scenarios addressing EA management specific tasks the following facts were taken into consideration:

**Achievement of objectives** concerning the creation of deliverables equal or similar to those defined in the scenario description.

**Tool handling** concerning the effort necessary in achieving the deliverables.

**Procedure consistency** concerning the semantically consistent use of the metamodel provided by the tool or potential unavoidable misuse.

---

1These scenarios are not to be mixed up with the scenarios of the tool survey as a whole. Here scenario means a potential future state of the application landscape.
Procedure integration concerning consistent integration of the activities and objectives in respect to other scenarios and other relevant aspects and objectives.

4 Results of the Tool Survey

This section sketches how, based on the data gathered via the survey approach detailed in section 3, the results of the survey were compiled.

Then, strengths and weaknesses generally visible in the tools and the commonalities of their approaches are given, abstracting from the results in respect to specific tools. Therefore we show two kiviat diagrams (see Figure 3 and 4), which show the minimum, median, and maximum values achieved by the tools for each axis. Thereby, each axis represents an aspect of EA management tool support.

The kiviat diagram for visualizing specific tool functionality (see Figure 3) consists of nine axes:

- **Configurability of the Metamodel** reflects the possibility to adapt the supplied metamodel of the tool in order to match the metamodel used for EA management within an enterprise.
- **Coverage of EAM Concepts by Predefined Metamodel** is concerned about the extent of the metamodel shipped with the evaluated tools.
- **Support for Impact Analysis & Calculation** mirrors the ability of the tool to perform calculations or impact analysis based on the repository data.
- **Richness of Predefined Visualization Techniques** focuses on the presentation capabilities of the evaluated tools.
- **Flexibility of Visualization Techniques** on the one hand refers to the tool’s capabilities of defining new visualization templates. In this context, template means kind of visualization, not instance - thus defining a new template creates a new kind of diagram with rules defining the entities to be displayed and manners how they shall be displayed. On the other hand this axis is about the tool’s support for manual adaptations to existing and possibly automatically generated diagrams and its support of persisting such changes.
- **Flexibility of Reporting** reflects the tool’s capabilities of creating tabular reports of the repository data.
- **Collaboration Support** refers to whether concurrent working on the same data is possible or not and which limitations to concurrent work exist.
- **Import/Export & External Data Sources** is about the tool’s capabilities of importing and exporting data.
- **Usability** refers to the general subjective user experience.

The kiviat diagram for visualizing tool support for EA management tasks (see Figure 4) consists of seven axes, which match the scenarios for analyzing EA management tasks already described in section 3.2.2.

For each of the axes a textual description was compiled by the team executing the scenarios on the respective tool, using information from both the list of criteria and the scenario simulation. The axes of the two kiviat were then assigned ordinal valued scores for each of the tools during a group discussion among the evaluation team, based on a consensus finding process.

The following general conclusions can be drawn from our tool evaluation: two axes, which especially came into our focus are Configurability of the Metamodel and Coverage of EAM Concepts by Predefined Metamodel. They rely on the metamodel, which makes up the basis for the repository of the EA management tool, and the possibility to adapt it to own needs.

In analyzing these aspects we discovered that the supported metamodeling concepts vastly differ between the different tools. Some tools are fully Meta Object Facility (MOF) 1.4 compliant, in contrast others did not even support distinguishing between different relationship types. The complexity of the shipped metamodels was another interesting object of investigation. Some tools came with a core metamodel, which only contained a few entities and associations. Other metamodels contained up to 400 entities with corresponding associations. An issue of such extensive metamodels is their complexity, which makes them hard to understand and therefore hard to use in practice, especially if there is only terse documentation available.

The axis Collaboration Support shows that this issue is well understood by all participants, all of them support concurrent use by multiple users. They allow the configuration of different roles and provide functionalities for rights management. The difference between four and five points is based on our appreciation that it is desirable to offer a web based thin client instead of a fat client, because of the simpler maintenance and distribution of the software, especially to a wider audience.

An area, which in our opinion retains potential for im-
Another specific functionality, where we see opportunities for improvement, are the visualizations supported by the tool. We distinguish between two kinds of visualizations, predefined and flexible ones. The (semi-)automatically generated visualizations are usually, if available, at all, rather restricted. Concepts like nesting or alignment to visualize relationships of elements within the repository are not known to all tools. Usually, generated visualizations are based on nodes and edges for showing relationships. In contrast, flexible visualizations have to be created manually and do not have strictly defined semantics, which limits the possibility to make a decision based on these visualizations, as they offer room for interpretation to everyone who uses them.

The last axis of the first kiviat diagram we want to highlight is Import/Export & External Data Sources. Currently, no standardized exchange format for EA management data exists, resulting in high effort to use data of different sources in order to get an aggregated view of the information available within an organization or to migrate from one EA management tool to another.

The second kiviat diagram, shown in Figure 4 visualizes the tool support for EA management tasks (see subsection 3.2). Here, we want to go into details on three of the seven axes. The first two axes are Project Portfolio Management and Synchronization Management. Both require the concept of projects and the concept of time (here meaning e.g. the period in which a certain version is in use), which are not completely supported by all tools, e.g. projects introducing new applications/application versions as deliverables were only known by some tools.

Another important part in EA management is the Management of Business Objects and Business Services. This task is concerned with business objects, representing e.g. a customer which are processed by different business services, e.g. for updating the respective information. Concepts for modeling objects, which can be accessed through services and are transferred over connectors are not completely supported by all tools. As a result, visualizations of information flows are sometimes limited.

Lastly, it can be noted that not all evaluated tools were shipped with a description/manual of a methodology how to address tasks as the ones described in the scenarios using the respective tool functionality. Thus, the evaluation teams often had to develop their own methodologies ad-hoc, complicating the scenario simulation.

The subsequent section presents our lessons learned and suggestions for further improvement in respect to some of the issues described above.

5 Lessons Learned

In this section, we trace back some of the issues described above to approaches commonly encountered in EA management tools, as they have been described in section 4. From there, we sketch alternative approaches, of which we expect that they are suitable for tackling the respective problems.

5.1 Visualization Generation and Visualization Semantics

As described in section 4, two problems arise with the visualizations of application landscape specific information common to EA management tools. One is the fact that automated generation of adequate visualizations is often not possible, and the other is that, more often than not, the semantics of the visualizations are only defined in vague terms.

We see both of these issues as arising from the manual creation of the visualizations that is proposed by many tools and supported by them via capabilities that go into the direction of pure drawing tools, with the key difference being that the shapes have links to objects in a repository, which represent entities of an application landscape or an enterprise architecture and are also called information objects. Although these links between the repository and the visualization serve to establish the semantics of the visualization, many of the concepts used in common visualizations are unknown to the tools. The creators of such visualizations nevertheless often use such concepts, which is basically possible via the drawing capabilities of the tools: they e.g. nest boxes representing business applications into boxes representing organizational units to indicate which organizational unit hosts which application systems, although nesting is basically not known by the tool. Thus, the relationship expressed by the nesting is not automatically captured.
in the repository. This leads to visualizations that, to some extent, exhibit the character of a drawing instead of a model. During one discussion with our project partners about EA management, the problems of such an approach were summarized by the quote: “drawing is no management”.

Therefore, we propose a close link between the information and its visualization (outlined in Figure 5), based on defined metamodels for both of them, as a suitable solution to address the labour-intensive manual visualization creation which often leads to inadequately vague semantics.

Figure 5. Visualization generation via model transformation

A key point of this approach is, as stated above, that the visualization of the information (called symbolic model in Figure 5) is distinguished from the information itself (called semantic model in Figure 5). The semantic model can, depending on the modeling concepts used, e.g. be constituted by tuples in database tables or by a graph of connected objects, if an object-oriented representation is chosen. A similar description of the symbolic model is possible, e.g. via objects that represent what symbols exist on a visualization and in what relation they have to be positioned. Figure 6 exemplifies this by describing a visualization in which several rectangles are nested in a "top-level" rectangle Munich.

Figure 6. Visualization Objects describing a nesting-based visualization, cf. Figure 1

A close link between these two models can be established by generating a symbolic model out of the respective semantic model via a model transformation language as e.g. ATL [1]. Thereby, the model transformation has to be based on the respective metamodel of the semantic model, which we call information model, and of the symbolic model, called the visualization model. These two models are shown in the second layer of Figure 5. The information model is a metamodel of the information stored about the enterprise architecture and the visualization model is a metamodel of the visualizations used to represent this information, defining the concepts that can be used therein. Figure 7 shows an object-oriented visualization model that allows Rectangles with Text to be nested.

Figure 7. Visualization Model of the visualization described in Figure 6

We propose using Meta Object Facility (MOF) 2.0 as a common metamodel for both the information and the visualization model, as the object-oriented concepts of MOF are rather well-known and adequately expressive for the task at hand. Of course, other choices are basically possible as e.g. using ontologies to capture knowledge about an application landscape or enterprise architecture.

Assuming an adequately formal definition of the visualization model², it is possible to create a visualization out of an object graph as shown in Figure 6, which has itself been created out of a semantic model via a model transformation. This automates the process of visualization creation. It further ensures that the information, what graphical concepts represents what concept in the semantic model is explicitly available. This adds to a more strict definition of the visualization semantics³.

5.2 Adequate Information Model support

The tool evaluation described above has shown that most tools come with a predefined metamodel. However, a discussion we held with tool vendors revealed that they view a fully standardized metamodel for EA management as unrealistic. This forms the usage context of the metamodeling capabilities, which are, as described above, frequently offered by EA management tools: before an EA management tool is put into "production” in an organization, among other things, the metamodel has to be adapted to fit the specific needs of the respective organization, often during a

²We achieve this by adding mathematical expressions of what e.g. nesting demands of the positions of the affected map symbols to the respective classes in our object-oriented visualization model.
³Of course, this does not address the problem of adequate semantic definitions for the concepts in the information model themselves.
consulting project.

Therefore, the metamodels offered by EA management tools seem to be in danger of falling into one of two traps: either they are too small to be a major benefit in an EA management project, or they are rather big, contributing to low readability and low understandability of the model.

We view an approach based on the idea of patterns as used e.g. in software design [4] as a way to address the issues outlined above. This approach is based on fragments of an information model, called information model patterns, each of them constructed to address specific concerns. An organization-specific information model is then constructed by choosing the adequate patterns and integrating them into an organization-specific model.

Therefore, we propose describing the information model patterns in adherence to the basic elements of a pattern as described in [4]:

Problem section: This section describes the problems the pattern is designed to address and the usage context in which it can be employed. As the patterns are meant to contribute to an information model that serves as a means of creating architectural descriptions for enterprise architectures, we propose using the concepts concerns and stakeholders, as used by the IEEE Std 1471-2000 [5]. By adhering to this standard, which demands that a view on an architecture always has to serve concerns of stakeholders, it is possible to avoid the creation of architectural descriptions which exist just for their own sake.

Solution section: Here, a proposal of how to address the problems outlined above is given. It is not meant to offer an exact description of the solution, but similar to [4], a template, which describes in an abstract way how the respective issues can be solved. This involves a diagram of an information model fragment (e.g. a class diagram), but also a description that conveys to the reader how the information model fragment is meant to be used and what the semantics of the concepts in the fragment are. This may include e.g. evaluation procedures or visualization techniques, extending the information model pattern in the direction of an underlying model for addressing specific concerns, as described in the section 5.3 below.

Consequence section: This section describes the most important consequences that stem from the use of the respective information model pattern. It can encompass a broad range of consequences, ranging from effects, which using a given pattern has on an information model (e.g. in respect to understandability or modifiability) to the demands a pattern poses on employees using the respective information model (e.g. knowledge in certain analysis techniques). Also consequences in respect to data needs due to certain patterns in an information model should be stated here.

By composing the information model out of patterns as described above, an organization specific model can be built of units that are more manageable than one big monolithic model, parts of which are then trimmed, abandoned, or modified to adapt to customer-specific details.

The patterns, which constitute smaller units covering a limited number of explicitly given concerns, are a more inviting environment than a monolithic metamodel to reason about adequately precise semantic definitions of the used concepts. This facilitates the creation of a documentation for the information model which states precisely enough what is meant by the concepts employed.

5.3 From Metamodels to Methodologies

Taking an engineering-like approach to conducting EA management, characterized by planned procedures, takes more than the information models commonly offered by EA management tools. An engineering discipline is based on methodologies [7], and a methodology is more than a metamodel.

Although an information model (and patterns as mentioned above) clearly can be part of a methodology, an in-depth definition of a methodology shows some interesting aspects of which the patterns mentioned above can be extended, evolving them into a way to provide systematic, methodic support to EA management. [7] defines a methodology/a method within an engineering discipline as a way to conduct a process, which consists of an underlying model, which refers to the classes of objects represented, manipulated, and analyzed by the method, a language, which refers to the concrete means of describing the products of the method, defined steps and ordering of these steps, which refers to activities performed by the user of the method, and guidance for applying the method.

The specific parts of the definition can be interpreted in the context of tool support for EA management as follows:

- The underlying model can be an information model pattern.
- The language can be some formalism via which the information structured by the information model can be expressed, e.g. as described with scenario Visualization of the Application Landscape.
- The extension of information model patterns can go even further with the defined steps and the ordering of the steps, which can be seen as part of the solution section of a pattern. They describe how the concern is addressed using the concepts described above (underlying model and language).
- The guidance for applying the method can make statements about the usage context of the methodology.
which includes the concerns to which the methodology can be applied and the roles that are able to successfully use it (IT staff, business managers, etc.).

A methodology based approach can address a problem that is described in section 4: although the tools are ranked quite well in metamodel and visualization specific criteria, the score for some scenario simulations is lower than one might expect. A possible explanation is that the tool does not always give enough guidance in how to apply the functionality to solve a specific EA management problem.

6 Resume and Outlook

In this article, we have sketched the general strengths and weaknesses of the tool support commonly available to EA management today, based on an extensive survey we conducted in 2005.

While software like inhouse solutions or solutions currently in development have not been considered, the survey represents the software offered by major players in this sector, thus evaluating what is state of the art in this market.

While the tools generally offer adequate support in respect to a significant share of the aspects evaluated, as visible in Figures 3 and 4, there are some facets of the tools where we view that the approaches currently taken lead to a suboptimal support of EA management. For these areas, we sketched alternative approaches of which we believe they are capable of enabling tools to provide better support to EA management.

These approaches are currently object of research in our project Software Cartography, in which we explore and develop methods for the description, evaluation, and design of complex application landscapes.

In respect to automated visualization generation, we are detailing the approach sketched above, thereby especially focusing on the visualization model, which serves as the basis of a prototypical tool we are currently developing.

Regarding the information model pattern and methodology based approach, we are working out a scheme how to best document such patterns/methodologies. From there we are describing first patterns, as we have been encountering them in our project software cartography, e.g. at industry partners. These are meant to form the basis of a pattern catalogue, which can serve as a knowledge base for the construction of organization-specific, methodical approaches to EA management.

References