

A Lightweight Method for the Modelling of Enterprise Architectures

Henk Koning, Rik Bos, and Sjaak Brinkkemper

Department of Information and Computing Sciences, University of Utrecht,
P.O.Box 80.089, 3508 TB Utrecht, The Netherlands
{h.koning, rik, s.brinkkemper}@cs.uu.nl

Abstract. This paper introduces an easy to learn method to describe enterprise architectures with a limited focus on the relation between enterprise functions and IT-systems. We have designed the Enterprise Architecture Modelling method (EAM), driven by our teaching requirements and based on our ERP modelling experience. EAM consists of these diagram types: the Supply Chain Diagram, showing the business environment; the Enterprise Function Diagrams for the interoperation of enterprise functions; the Scenario Overlay for modelling the main business processes; the System Infrastructure Diagram, depicting the technical infrastructure of IT systems and networks; and the Application Overlay Diagram, showing which applications give support to which enterprise functions. We satisfactorily conducted about 40 case studies based on EAM. To solicit feedback we performed an enquiry among users of EAM. A future step will be testing the use of EAM in managerial decision taking in practice.

Keywords: Enterprise architecture, enterprise functions, applications, modelling method, views, feedback.

1 Lightweight Enterprise Architecture

Several years ago we started at our computer science department of the Utrecht University with a master course in Enterprise Architecture (EA). Since this is a vast subject we were searching for an approach which at the one hand would give the students a good overview of all the developments in EA and on the other hand would enable the students to test their skills in this challenging field. We have sought a limited area within the broad field of EA where the students could develop their skills in communication, in the analysing, condensing and abstract modelling of lots of information, and in mastering abstract concepts in the business and in information technology. We have chosen to concentrate on the bridge between enterprise functions and the IT-support for these functions, and to stay at a high abstraction level. In this we are inspired by our experience in modelling ERP-applications. In that area we have seen that a high level functional breakdown of an organisation is a stable reference to chose ERP-modules. The questions we seek to answer in our EA modelling method for a given enterprise are: what are the main functions the enterprise performs? What are the relations of these functions to each other and to the

outside world? What are, or should be in the future, the information systems that support the enterprise functions? What infrastructure, in terms of computers and network capabilities, is necessary, or will be necessary in the future, to operate these information systems? The emphasis is on understanding the business in a break down of business functions; from thereon the relations are uncovered to information systems (applications) and infrastructure (network and computers). Our method only touches the surface of application layer and infrastructure layer. Also the relations of the business functions to the outside world are described only in general. We limit ourselves to a top down analysis, creating *overview* and *insight*.

The word 'lightweight' in the title of this section should not be confused with 'easy' or 'not to be taken serious'. It can be very difficult to enter an unknown company and derive from all the information that is available a central, homogeneous, balanced model of the enterprise functions. Information systems can have many relations to each other, but not all relations are meaningful at the top, strategic business level. Eppler [3] has studied the communication problems between experts and managers, and has found, amongst others (page 15): "Summarizing these issues, we can conclude that experts struggle with three major issues when transferring their knowledge to managers: First, reducing or synthesizing their insights adequately, second, adapting these trimmed insights to the management context without distorting them, and third, presenting the compressed and adapted findings in a trust-building style and reacting adequately to management questions and feedback." EAM provides means in the synthesizing/compressing and adapting to management context, as will be explained in the sections 2 and 3, and is by no means easy. Also, the ability in EAM to decompose the Enterprise Function Diagrams (EFDs, see section 2) into several layers of detail, means that, if necessary, a lot of complexity can be captured in EAM.

1.1 Related Work

Several methods for developing enterprise architectures are available nowadays, e.g. ArchiMate [6], SEAM [14] and ARIS [10][11]. Each method has its own strengths and usually has its focus on one or more specific points, e.g. integration between different models, alignment between business and IT, business processes, communication etc. We mention some methods here briefly to compare them to our goals. At this moment we have chosen not to use any of these methods (mostly because of the time needed to learn the method) and to create our 'own space'. In the future, when our method has stabilized more, we may again look at this issue.

The ArchiMate project has produced an elaborate language for describing enterprise architectures. The conceptual model consists of 29 entities with a corresponding visual representation consisting of 40 symbols each having a specific meaning. It takes a while to get to learn each of the symbols and, in our experience, after some time of not working with the language one has to relearn the specific meaning of each of the symbols. Because we have only a few basic concepts and stay at a high abstraction level, we practically need no special icons in our method. The authors of ArchiMate have proposed 16 diagram types (viewpoints) as a basic set to work with the language, but many more could be constructed.

ARIS originated as a method for describing business processes in the context of SAP implementation processes, but has developed into a thorough, general purpose EA modelling tool. It is even more complex than ArchiMate. It has 5 basic views, but numerous diagram types to populate the views. On the summary page [10], p. 78, we count 23 ‘main’ diagram types. The business process meta-model contains 300 entities and relations (p. 48). For *students* the time needed to master the basics of ARIS is not proportional to the analysis time needed to study the enterprise architecture of a company.

MEMO, Multi-perspective Enterprise Modelling, was developed by Frank [4]. He proposes a framework of three so called perspectives - strategy, organization and information system - each of which is structured by four aspects: structure, process, resources and goals. MEMO contains three modelling languages: strategy modelling language (MEMO-SML), organization modelling language (MEMO-OrgML) and object oriented modelling language (MEMO-OML), which allow for detailed modelling of the three perspectives. The EAM method we propose in this paper seems to be a subset of MEMO’s strategy goals, organization structure and process, and information system resources. MEMO has an interesting setup, but it falls short of our wishes regarding ‘easy to create’ and ‘easy to understand’.

Braun and Winter [1] describe an Enterprise Architecture meta model which has four layers: strategy, organization, application, software. The models of the first three layers are shown in a slightly simplified manner, and the relationships between these models are elaborated. A successful implementation of these models, using a meta modelling tool, is reported. The three simplified models contain respectively 27, 27 and 22 concepts. So, in our view, they are not easy to learn, they entice into a lot of detailing and underline the need we feel to start anew with a basic method containing few concepts. We like the limited number of layers, with the focus on linking the organization to the IT-support.

The well known framework of Zachman [12][14] has 36 different viewpoints to give aid in categorizing the architectural information, but gives no guidance regarding the modelling of the information. The same goes for an architecture process description like TOGAF (Open group 2002). It gives guidance regarding the activities of the architect, but does not have a modelling method. The 4+1 framework of Kruchten [5] gives an architectural structure and modelling method, but is more geared toward software architecture and not applicable to enterprise architecture.

The team of Wegmann of the École Polytechnique Fédérale de Lausanne has developed an object-oriented enterprise architecture method, called “Systemic Enterprise Architecture Methodology” (SEAM) [14]. As part of SEAM, they have developed the CAD tool “SeamCAD” [7]. SeamCAD enables the modelling of hierarchical systems (spanning from business down to IT) at different levels of detail (e.g. from large business transaction to detailed interactions). It has a philosophical underpinning which is not so easy to understand and doesn’t give guidance as to what modelling concepts should be used or what levels should be distinguished. The notation is UML-like and some training is needed to read the diagrams.

1.2 Outline of the Paper

In the next section we present an overview of the EAM method with the key concepts. In section 3 we present each of the models of EAM and illustrate them with example

diagrams from a case study. In section 4 we outline our efforts to receive usage feedback by a large series of case studies and by a questionnaire. We finish the paper with conclusions and future work.

2 Enterprise Architecture Modelling Method

2.1 Diagramming Tools for Enterprise Architectures

The EAM method consists of the following diagrams:

A *Supply Chain Diagram* (SCD) shows how the enterprise works together with business partners to produce the goods or services for the customers (enterprise context).

An *Enterprise Function Diagram* (EFD) gives a top level breakdown of the main functions of an enterprise. The top diagram covering the complete enterprise is called the *corporate EFD*, and the lower level EFDs are called *function EFD*.

A *Scenario Overlay* (SO) shows how the enterprise functions in an EFD interoperate in a particular situation.

An *Application Overlay Diagram* (AO) shows which applications give support to which enterprise functions.

A *System Infrastructure Diagram* (SID) shows the main network topology, the main computers that function in the network and the main information systems that run on these computers to support the enterprise functions.

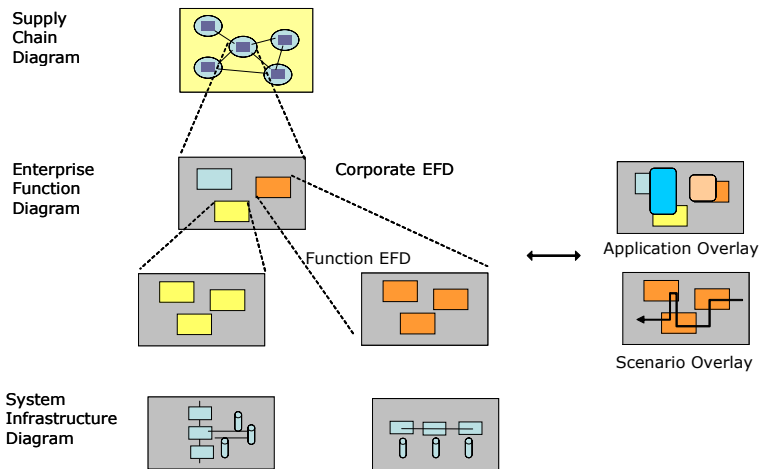


Fig. 1. Overview of EAM models

In Fig 1 the overview structure of these models is shown. We will explain these models each in turn in section 3. With each we give an example diagram taken from a case study performed at the company Center Parcs.

2.2 Concepts of Enterprise Architectures

See Fig 2 for the meta-model of our modelling method. Note that this somewhat simplified model does not contain all constraints as it only shows our key concepts and how they are related.¹

For enterprise we use The Open Group [8] definition “Any collection of organizations that has a common set of goals and/or a single bottom line. In that sense, an enterprise can be a government agency, a whole corporation, a division of a corporation, a single department, or a chain of geographically distant organizations linked together by common ownership”. For the sake of simplicity, when dealing with external parties, we include in this definition any collection of *individuals* (e.g. customers) that have a common set of goals and/or a single bottom line (and for which the enterprise to be modelled develops products and services).

To model the context of an enterprise we use enterprises connected by flows between their enterprise functions. For modelling the enterprise the key concept is the enterprise function. We define an enterprise function as a collection of coherent processes, continuously performed within an enterprise and supporting its mission. To show the interoperation of the enterprise functions we also portray the flow (of information or products & services). Scenarios indicate a sequence of flows. For modelling the information systems and the infrastructure our key concepts are computer, application and network (component). We use ‘computer’ as a general term to indicate all sorts of processing units or executional components. Likewise ‘network’ stands for all sorts of connectivity components. In section 3 we will describe the diagram types and indicate with each on what meta-model concepts they are based.

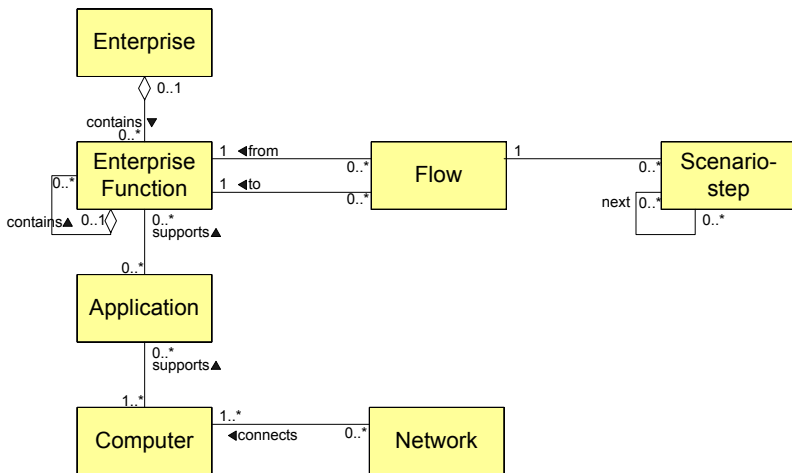


Fig. 2. The meta-model of the Enterprise Architecture Modelling method (EAM)

¹ Additional textual constraints are for example ‘an enterprise function cannot send a flow to itself’ and ‘an enterprise function cannot contain itself’.

3 The EAM Models

In this section we present each of the models of EAM and illustrate them with example diagrams from a case study at Center Parcs Europe. Center Parcs Europe (CPE) is one of Europe’s largest companies in the accommodations rental for short holidays. Its headquarters is in Rotterdam, the Netherlands and it offers about 10,000 bungalows and cottages in 20 parks.

3.1 Supply Chain Diagram (SCD)

The Supply Chain Diagram is a model of the *enterprise context* of the enterprise together with its business partners and the exchange of products and services. Supply Chain Diagrams create a quick overview of the enterprise as a whole and of the main players in its enterprise contexts. The SCD is based on the meta-model concepts ‘enterprise’, ‘enterprise function’ and ‘flow’. In reality the flows connect enterprise functions in the different enterprises, but in the graphical presentation of the SCD these underlying enterprise functions are suppressed, see [2] for this technique. For large companies business units can be treated as separate enterprises. An EA description contains one SCD.

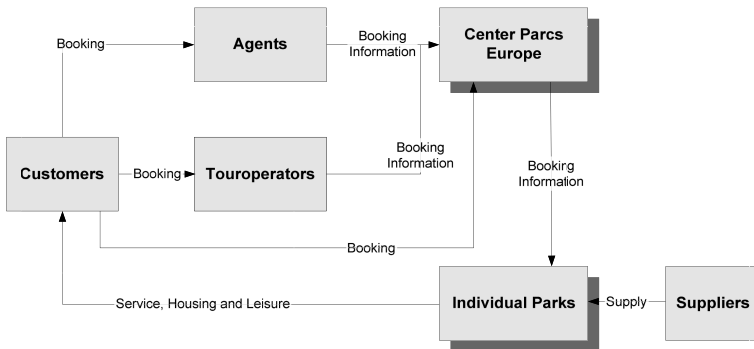


Fig. 3. SCD of Center Parcs

See Fig 3 for an example of an SCD. It shows how CPE and the Individual Parks cooperate with Agents, Tour operators and Suppliers to accommodate Customers. The boxes denote enterprises, the arrows the flow of products and services, or of information. In this example the two main units of CPE are shown, the central Europe headquarters and all the individual parks (in one box), and the main external parties.

3.2 Enterprise Function Diagram (EFD)

An Enterprise Function Diagram is a model from an *enterprise function* perspective. EFDs give a top level breakdown of the main operations of an enterprise with their information flows.

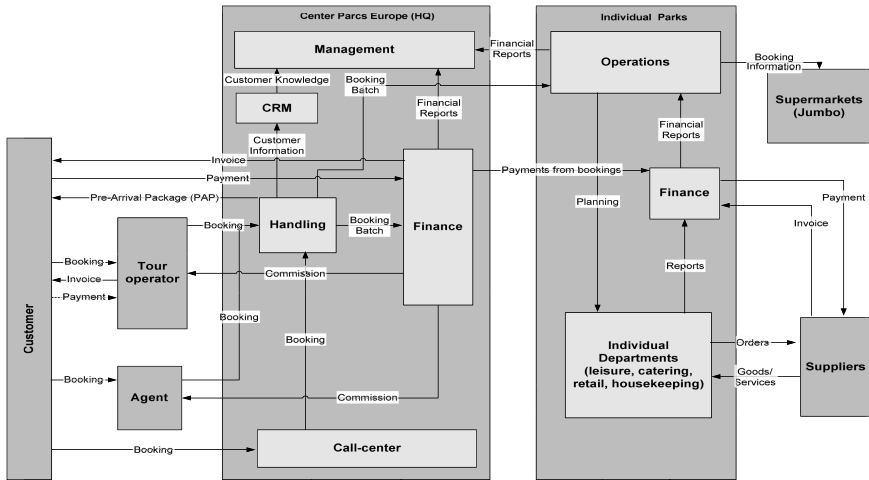


Fig. 4. EFD with the main enterprise functions of Center Parcs

See Fig 4 for an example of an EFD. The lighter grey boxes within ‘Center Parcs Europe (HQ)’ and ‘Individual Parks’ denote enterprise functions. To the left and to the right the external parties are portrayed. Arrows indicate the flow of information (digital or on paper). The EFD is based on the meta-model concepts ‘enterprise’, ‘enterprise function’ and ‘flow’; this is the same as with the SCD but now the focus is on the enterprise functions. This diagram shows these functions of Center Parcs Europe (HQ): *Call center, Handling, Finance, Management, and CRM*. For the sake of brevity these are not described here further. These are the main functions of the Individual parks: *Individual Departments, Finance and Operations*.

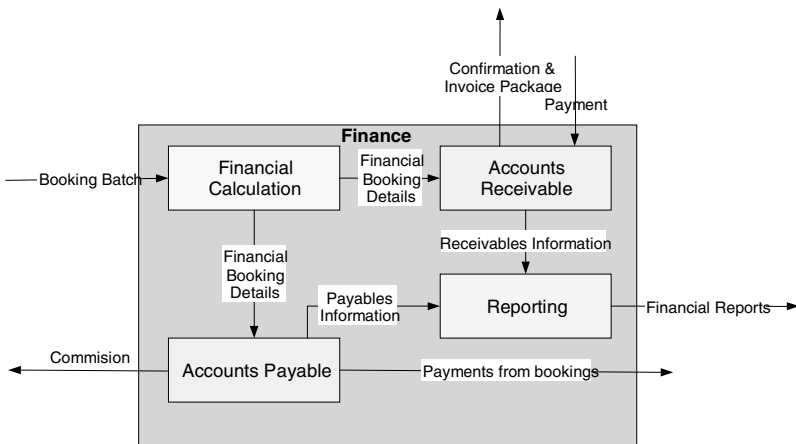


Fig. 5. Decomposition of the Finance function of Center Parcs Europe

Functional EFD. Enterprise functions in an EFD can be decomposed in the same diagram or in a separate EFD. See Fig 5 for an example of a decomposition. A tree structure of EFDs can be set up to analyze the architecture of an enterprise. Case study evidence (see section 4) shows that usually two levels are enough to get sufficient grip on the complexity of an organization. We call the top level EFD a *corporate* EFD (see Fig 4.) and a decomposition a *functional* EFD. The EFD of Finance shows the following sub functions. *Accounts Receivable* , *Accounts Payable*, *Financial Calculation*, and *Reporting*.

3.3 Scenario Overlay (SO)

A scenario is a continuous processing of a request trigger by various enterprise functions, which results in one or more feedback triggers. A Scenario Overlay provides insight in the interoperation of *enterprise functions* and in the completeness of the EFD. A scenario is drawn as an extra diagram level on top of an EFD with a proper explanation. Only essential flows are elaborated into a scenario (highest frequency, large impact). The scenario overlay adds limited, but for our goal sufficient, process information to the EFD. It gives fewer details than process models that have been created with a (dedicated) process modelling language. See Fig 6 for an example. Dark broad lines are drawn that touch EFD functions in the execution of the process that is triggered. The extra information for the scenario, compared to the underlying EFD, is based on the meta-model concept ‘scenario step’. Here we show one scenario overlay for the Finance function. This scenario concerns a booking made via the CPE call center. Scenarios are an optional part of an EA description. Typically an EA description will contain several SOs.

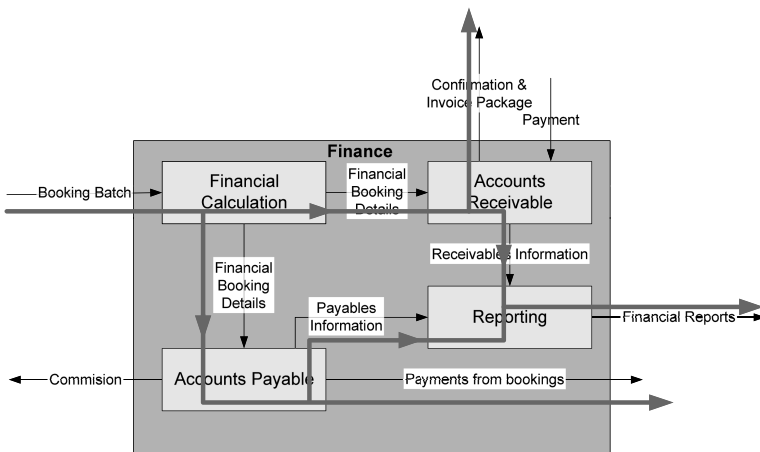


Fig. 6. Example Scenario Overlay, the financial processing of a Booking Batch

3.4 System Infrastructure Diagram (SID)

A SID shows the *information systems* and information technology *infrastructure* of an enterprise, or a well-defined part thereof. A SID shows the main network topology,

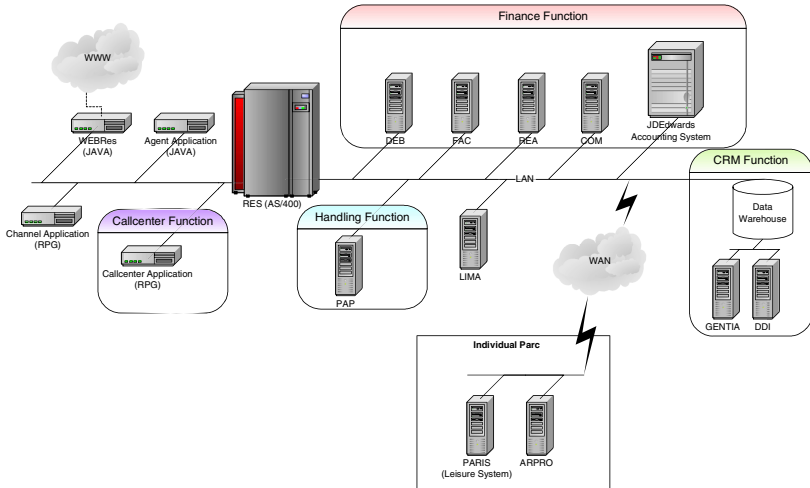


Fig. 7. Example System Infrastructure Diagram, a mixture of platforms at CPE

the main computers that function in the network and the main information systems (applications) that run on these computers to support the enterprise functions. The SID is based on the meta-model concepts ‘application’, ‘computer’ and ‘network’.

The SID is not intended to manage the computers and network in their own right (that would need much more information), but only to show the technical context of the applications. For SIDs there are many popular notational variants. An EA description typically contains one SID, corresponding to the abstraction level of the corporate EFD. Complex parts in this SID can be exposed by creating a SID at a more detailed level. See Fig 7 for an example. The computer icons depict hardware systems, the adjoining acronyms stand for software systems (applications) that run on them. The hardware systems are grouped by the enterprise functions they support. The lines depict network connections. As can be seen in the infrastructure diagram, CPE uses many different systems.

3.5 Application Overlay (AO)

To fill a gap that is felt between the Enterprise Function Diagram and the System Infrastructure Diagram we later introduced a new type of diagram, comparable to the Scenario Overlay, the Application Overlay (AO). The applications are drawn as an extra diagram level on top of an EFD with a proper explanation. The AO deals with information systems (applications) in their own right. Information systems (software components) have a different lifecycle than computers and network (hardware components). The AO is based on the meta-model concepts ‘enterprise function’ and ‘application’. See Fig 8 for an example.

This simple example was not part of the Centre Parcs case study, but was constructed by the authors of this paper as an example. The AO shows clearly which enterprise functions are supported, once or multiple times (possibly a sign of redundancy or fragmentation), and which functions are not supported by IT. If an

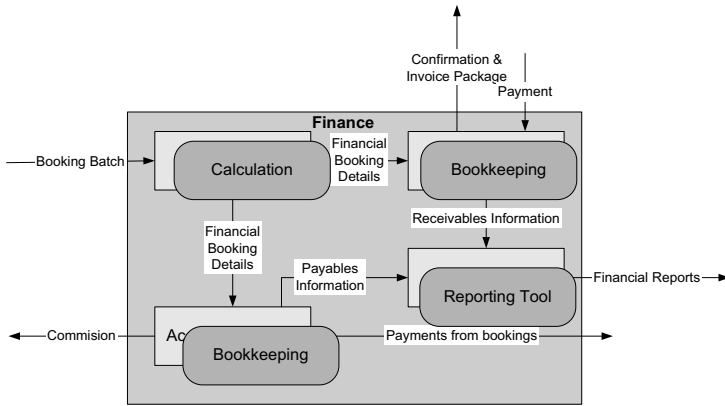


Fig. 8. Example Application Overlay , applications that support the Finance enterprise functions of Center Parcs Europe

application supports enterprise functions that are not adjacent in the underlying EFD diagram, then the application will be drawn more than once (see Bookkeeping in Fig 8).

4 Usage Feedback

4.1 Course Enterprise Architecture

Each fall a master course on Enterprise Architecture (see [13] for a general description) is given as a course in the international MSc program of Business Informatics at Utrecht University. An important assignment in this course is to do a case study in practice, meaning that the students, who already have a bachelor in computer science, have to produce a complete EAM model for a real life company, which includes all diagram types. A representative of the company has to consent to the models, and declare they really represented the given situation at the company. These representatives (contact persons) were mostly having a position in the IT-department, reporting to business management. In the first year we had 40 students in 19 groups working at 15 different companies. In the second year we had 68 students in 23 groups working at 18 different companies. Six companies participated for the second time, making up a total of 27 different organizations. The students are placed in groups of 2 or 3 each and before they visit the companies for the interviews, they get lectures on our EAM technique together with some small exercises. Lectures and exercises take about eight hours. Amongst these case studies we have seen many very good and insightful high level descriptions of enterprises and their IT-support.

EAM was used in practice by former students in several projects in the industry. Here we mention the following since these are published. First, at a municipality to decide on a new e-government portal service to be launched [16]. Second, the integration of enterprise applications at the Royal Netherlands Army was performed by applying EAM. EAM was *extended* with some UML diagrams to design the integrated business processes spanning multiple enterprise applications, see [9].

4.2 Questionnaire

We launched a questionnaire among the students who performed the case studies. See Table 1 for a summary of the results. Six questions related to the ease of creation were asked for each of the diagram types in EAM. The Application Overlay Diagram was not taken into account in this questionnaire, as it was introduced later. The students answered the questions immediately after performing the case study, and with respect to their own work. We have no comparable figures concerning other modelling methods and the students had no normative references for their answers. So we don't take the outcome of the survey as a vindication in an absolute sense, but we subjectively find the average scores satisfying (in line with similar questionnaires regarding other teaching subjects). We mainly look at relative differences in the figures or at extreme values to point us to aspects of EAM that need attention.

The most problems students experience seem to be with the EFD: the readability, the makeability and the correctness. We think this is understandable. The general tendency to add too much information to a diagram and to not stay at a high abstraction level is most felt with the EFD. Here also the intellectual effort needed to construct an abstract balanced model is felt most heavily. We see the same line in the figures pointing to "information lacking on all diagram types". In the presentation of EAM we need to stress more the limited scope e.g. the high abstraction level.

Table 1. Results of the questionnaire among students (n=23)

Question	Supply Chain Diagram	Enterprise Function Diagram	Scenario Overlay	System Infrastructure Diagram
Is the .. diagram easily readable? Rate on a scale from 1 (very bad readable) to 5 (very good readable) how good the .. diagram readable is.	4,3	3,2	3,7	3,8
Has the .. diagram the right level of abstraction? Make a choice: a. less detail preferred / b. just right the way it is / c. more detail preferred.	a 4% b 74% c 22%	a 9% b 91% c 0%	a 9% b 86% c 5%	a 14% b 54% c 32%
Is the correctness of the .. diagram easily established (conformity with the reality within the company)? Rate on a scale from 1 (very difficult) to 5 (very easy) how well the .. diagram can be checked.	3,5	2,7	3,5	3,4
Is there information lacking on the .. diagram? Chose y (yes, information is lacking) or n (no, no information is lacking).	y 35% n 65%	y 35% n 65%	y 23% n 77%	y 24% n 76%
Is there redundant information in the .. diagram? Chose y (yes, there is redundant information) or n (no, there is no redundant information)	y 4% n 96%	y 9% n 91%	y 5% n 95%	y 10% n 90%
How easy is it to produce this kind of diagram on the basis of available information? Rate on a scale from 1 (very difficult) to 5 (very easy) how well the .. diagram can be produced.	3,7	2,7	3,6	3,3

Contacts with some of the contact persons for the students at the participating companies, gave similar indications. Although they gave their consent that the produced EAM models truly reflected the actual situation at their company, they liked to have more information in the diagrams. We did not come round to asking more precisely what more information was needed in their view. The readability of the diagrams was considered good. No training was needed to read the diagrams.

5 Conclusions and Future Work

In this paper we have presented the Enterprise Architecture Modelling (EAM) method consisting of five diagram types for modelling enterprise architectures at a high level of abstraction in a fast and simple way. 40 Different case studies were performed using EAM. We conducted a small scale questionnaire. We conclude that EAM is a good means to express the essential functioning of an enterprise and its IT-support. For authors EAM can be learned and trained in a short course of one day, but they find it sometimes difficult to stay at the high abstraction level in the application of the method. The resulting diagrams can be understood without any specific training.

Further research in practice is needed to assess the necessity and sufficiency of the diagram types for, for instance, managerial decision making. It is required to incorporate non-IT staff in future evaluations of EAM. We want to continue developing EAM and take into account the attention points coming out of questionnaires. For the authors the creation of the EFD needs attention.

Besides these points we want to produce tool support for EAM (an alternative here is possibly to use an existing tool in a customized manner), and, together with a partner from industry, we want to develop the practical application of (an extended?) EAM.

What started as a mere conviction has now been tried out on a modest scale and usage feedback has been received. We would like to see EAM used on a larger scale in the teaching of EA and in practice, which will give us hopefully more feedback on the strengths and weaknesses of EAM.

Acknowledgements

We thank Ronald Bos and Inge van de Weerd for conducting the case study at CPE.

References

1. Braun, C., Winter, R.: A Comprehensive Enterprise Architecture Metamodel and Its Implementation Using a Metamodelling Platform. In: Desel, J., Frank, U. (eds.) *Enterprise Modelling and Information Systems Architectures*, Proc. of the Workshop in Klagenfurt, GI-Edition, Klagenfurt, 24.10.2005, Gesellschaft für Informatik, Bonn, P-75. Lecture Notes (LNI), pp. 64–79 (2005)
2. Buuren, R., van, H., Jonkers, M.-E., Strating, P.: Composition of relations in enterprise architecture models. In: Ehrig, H., Engels, G., Parisi-Presicce, F., Rozenberg, G. (eds.) *ICGT 2004*. LNCS, vol. 3256, pp. 39–53. Springer, Heidelberg (2004)

3. Eppler, M.J.: Knowledge Communication Problems between Experts and Managers. An Analysis of Knowledge Transfer in Decision Processes, ICA Working Paper #1/2004, University of Lugano, Lugano (2004)
4. Frank, U.: Multi-perspective Enterprise Modelling (MEMO) - Conceptual Framework and Modelling Languages. In: Proceedings of the annual Hawaii international conference on system sciences (2002)
5. Kruchten, P.: Architectural blueprints – The ‘4+1’ View Model of Software Architecture. *IEEE Software* 12(6), 42–50 (1995)
6. Lankhorst, M., et al.: *Enterprise Architecture at Work*. Springer, Berlin (2005)
7. Le, L.S., Wegmann, A.: *SeamCAD 1.x: User’s Guide* (2008) (accessed July 5, 2008), <http://infoscience.epfl.ch/getfile.py?mode=best&recid=55774>
8. The Open Group. TOGAF "Enterprise Edition" Version 8.1 (2002) (accessed July 5, 2008), <http://www.opengroup.org/architecture/togaf8-doc/arch/>
9. Roques, J., Vader, H., Bos, R., Brinkkemper, S.: SAIM - A situational method for application integration. UU-CS (Int. Rep. 2007-022). UU WINFI Informatica en Informatiekunde (2007)
10. Scheer, A.W.: *ARIS – Business Process Frameworks*. Springer, Berlin (1998a)
11. Scheer, A.W.: *ARIS – Business Process Modelling*. Springer, Berlin (1998b)
12. Sowa, J.F., Zachman, J.A.: Extending and formalizing the framework for information systems architecture. *IBM Systems Journal*, vol 31(3) (1992)
13. Utrecht University. Course Enterprise Architecture (2008) (accessed July 5, 2008), <http://www.cs.uu.nl/education/vak.php?vak=INFOEIA&jaar=2007>
14. Wegmann, A.: On The Systemic Enterprise Architecture Methodology(SEAM). In: Proceedings ICEIS 2003 (2003)
15. Zachman, J.A.: A framework for information systems architecture. *IBM Systems Journal* 26(3), 276–292 (1987)
16. Zuiderhoek, B., Otter, A., Bos, R., Brinkkemper, S.: Framework for Dutch Municipalities to Ensure Business IT Alignment Using Enterprise Architecture. In: Proceedings of the 6th European Conference on e- Government, pp. 457–466. Academic Conferences International, Reading (2006)