Evolving the EAM Pattern Language

ALEXANDER W. SCHNEIDER, Technische Universität München
FLORIAN MATTHES, Technische Universität München

Since 2008, the documentation of Enterprise Architecture Management (EAM) patterns has drawn much interest from the academic as well as industrial community. Nowadays, more and more companies start EAM initiatives and the maturity of some early adopters has reached a high level. Therefore, it is now possible to extend the structure of EAM patterns to reflect and include the growing EAM knowledge in practice. For example, knowledge about typical stakeholders and external influence factors can guide the selection of EAM patterns. In addition, metrics and reports/documents are now commonly used for EAM and can also be included. Furthermore, the problem of collecting data needed to describe an enterprise architecture can also be documented by using the pattern format because practice-proven solutions already exist. Based on questioning 13 different companies new EAM patterns have been identified and documented using the extended structure presented in this article.

Categories and Subject Descriptors: K.6.3 [**Management of Computing and Information Systems**] Software Management

1. MOTIVATION

An enterprise architecture (EA) can be defined as ”the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution“ [IEEE and ISO/IEC 2007]. Respectively, EA management (EAM) includes all activities to document, work up and present information about the EA and to design its evolution. Thereby, EAM establishes a holistic perspective on all elements of an organization, e.g. business processes, IT systems, information, information flows, hardware, and their relationships. Due to the large scope and the given heterogeneity of companies EAM is a field susceptible to practice-driven design research. Although this field has been researched for more than ten years [Langenberg and Wegmann 2004], the variety of challenges and organizational contexts hampered the development of a single and embracing management approach. With no established approach to implement EAM and given the diversity of concerns to be addressed, practicing EAM is far from being a routine activity.

Due to the growing interest in EAM, more and more companies employ enterprise architects. Especially for architects with little experience, EAM patterns form a useful collection of applicable knowledge by providing specific solutions to recurring problems. EAM patterns can also be used to complement larger and more abstract frameworks, e.g. The Open Group Architecture Framework (TOGAF) [Buckl et al. 2009]. Therefore, the pattern-based approach to EAM has been developed to address typical problems of existing EAM approaches lacking appropriate guidance to be used in practice or pursuing an all or nothing approach while neglecting the specific

Author’s address: A. W. Schneider, Technische Universität München, Boltzmannstr. 3, Munich, Germany; email: alexander.schneider@tum.de; Florian Matthes,Technische Universität München, Boltzmannstr. 3, Munich, Germany; email: matthes@in.tum.de

EuroPLoP ’15, July 08 - 12, 2015, Kaufbeuren, Germany

ACM 978-1-4503-3847-9/15/07 . . . $15.00
DOI: http://dx.doi.org/10.1145/2855321.2855367

demands of an enterprise [Buckl et al. 2009]. Since the initial publication of the EAM pattern language in 2007 [Buckl et al. 2007; Ernst 2008; Buckl et al. 2008b], the EAM body-of-knowledge has grown significantly. Unfortunately, a considerable part of this knowledge is spread over several companies and therefore not accessible. This development calls for an evolution of the EAM pattern language in terms of new patterns as well as structural extensions to include the available knowledge and make it accessible to researchers and practitioners.

2. EAM PATTERN OVERVIEW

Following the idea of Christopher Alexander [Alexander et al. 1977], the documentation of patterns seems to be useful also in the Information Systems domain [Winter 2009]. The observation and documentation of EAM patterns has been proposed in 2007 [Buckl et al. 2007]. After a huge data collection and many interviews the first version of the EAM Pattern Catalog has been published in 2008 containing 120 patterns and 43 concerns [Buckl et al. 2008a]. In the following sections we briefly summarize the structure used to document EAM patterns developed by the aforementioned authors and shortly describe a corresponding research method.

2.1 Current structure of the EAM pattern language

The EAM pattern language developed by Buckl et al. [2007] distinguishes between three different types of patterns:

M-Patterns. Methodologies define steps to be taken in order to address given *concerns*. These concerns are addressed by procedures defined by the methodology. Others refer to them as Process Patterns [Moser et al. 2009].

V-Patterns. Viewpoints provide the languages used by methodologies. A viewpoint proposes a way to present data stored according to one or more information model patterns.

I-Patterns. Information models represent underlying models for the data visualized in one or more viewpoints. An information model pattern conveys an information model fragment including the definitions and descriptions of the used information objects.

In addition to these three types of EAM patterns, and in accordance with the IEEE/ISO standard 420101, the EAM pattern language includes a list of typical *concerns*. They can be used as an entry point and help to select appropriate patterns within a given context. Figure 1 depicts the conceptual model underlying the EAM pattern language. Furthermore, typical relationships between patterns of the EAM pattern language (e.g. compounds, complements and sequences [Buschmann et al. 2007]) exist.

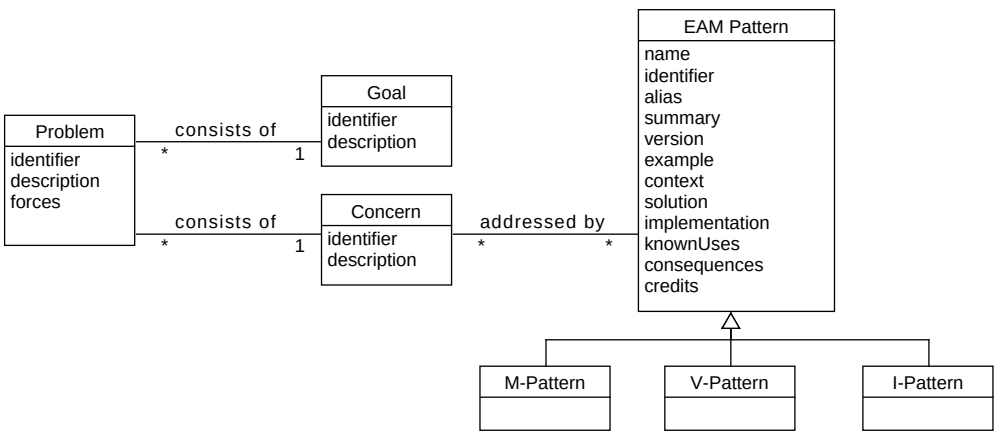

Fig. 1. Conceptual model of the EAM pattern language [Ernst 2008]

2.2 A research method based on the EAM pattern language

The goal of the Pattern-based Design Research (PDR) method is to balance rigor and relevance during design science research endeavors [Buckl et al. 2013]. As visualized in Figure 2, the basis is formed by observations of practice-proven solutions for recurring problems in practice. These observations are then conceptualized by using grounding theories. Within the *pattern-based theory building phase* the observed pattern candidates can evolve to actual patterns which are then integrated in the EAM pattern language. By appropriate problem and context descriptions design theories can be developed. Together, they form an organized collection of reusable practice-proven solutions. Each EAM team using this knowledge base can *select* suitable EAM patterns based on an individual problem and context description. To fit the organizational context the selected patterns have to be *configured*, e.g. concepts need to be renamed to fit the demands of a specific language community. The resulting configured design can then be *established* within the company. The actually *instantiated solution* will be subject to change over time due to changing concerns or improved solutions. The deviations between the initially configured solution and the actually instantiated solution form the basis for additional observations and thereby close the research cycle.

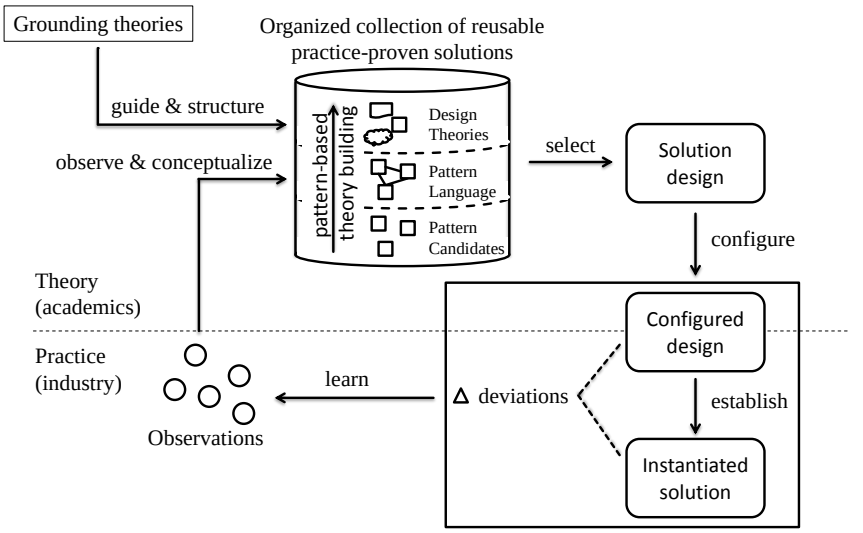

Fig. 2. Pattern-based design research [Buckl et al. 2013]

3. EXTENDING THE STRUCTURE OF THE EAM PATTERN LANGUAGE

Compared to the rich body of EAM literature, the current EAM pattern language is limited with regard to the relevant concepts it covers. However, growing maturity of EAM endeavors within companies and their increasing number allow for respective extensions. Based on the experience of numerous collaborations with industry experts the following concepts qualify for such extension: stakeholders, influence factors and maturity levels, architectural principles, subgroups of V-Patterns, data collection patterns and links to existing EAM frameworks and modeling notations. Figure 3 provides an overview of the resulting structure of the EAM pattern language. Each of the additional concepts will be introduced and explained in the following sections.

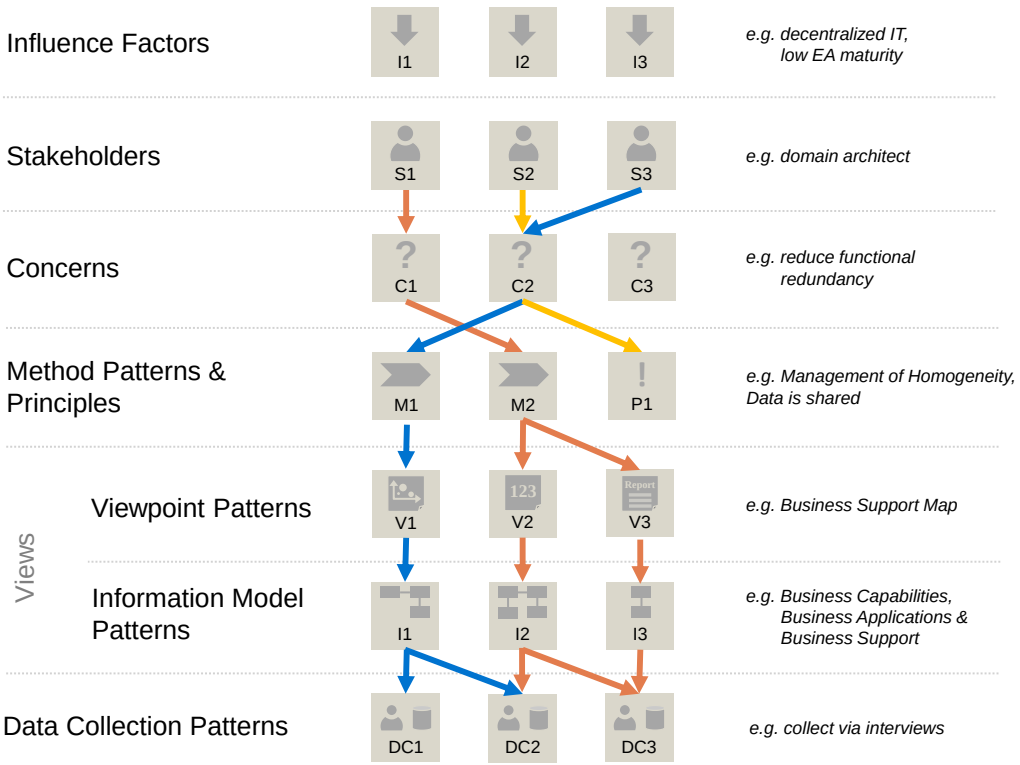

Fig. 3. Conceptual overview of the extended EAM pattern language structure

3.1 Stakeholders

Several observations revealed that concerns and the suitability of V-Patterns are typically linked to a few stakeholders. For example, the level of detail included in visualizations decreases the higher the stakeholder is in a company's hierarchy. Therefore, some V-Patterns might be useful for one but not for other stakeholders. Given the similarity of established roles within companies (stakeholders) and their similar concerns as well as preferences for visualizations of EA information related stakeholders should be mentioned when documenting concerns, M-Patterns or V-Patterns. The challenge consists in defining a generally applicable role model to ensure consistency of stakeholder documentation across companies. After several large-scale surveys among international enterprise architects and including the stakeholder catalog of [The Open Group 2011] the following scheme emerged:

CxO. Members of the board of directors

CIO. The chief information officer, usually the head of the IT department

Chief enterprise architect. The head of the enterprise architecture management team

Enterprise architect. An architect responsible for the enterprise architecture

IT architect. An architect responsible for the IT part of the enterprise architecture

Domain architect. An architect responsible for one business domain

Solution architect. An architect responsible for the development of concrete solution architectures

Business architect. An architect responsible for the business part of the enterprise architecture

Business domain experts. A person responsible for specific business processes

Business executive. A persons responsible for running an organization or a business department

Enterprise security officer. A person responsible, e.g., for corporate risk management or IT security

Program manager. A person responsible for prioritizing, funding, and aligning change activities

Project manager. The person responsible for a project

Software engineer. A programmer developing IT applications

IT operator. A person responsible for the continuing operation of IT applications

Regulatory bodies. Bodies in charge for law enforcement, e.g. supervisory authorities and auditors

Supplier. Other interconnected companies

Although this list might not be complete and not every company defines all of these roles, each company should be able to do a mapping of organization-specific roles to the roles named here.

3.2 Influence factors and maturity levels

Organizations are unique in their way of doing business, the lived culture, used tools, and many other aspects. These different aspects have to be accounted for in the design of an appropriate EAM function and therefore influence the applicability of different EAM patterns. The following aspects can be used to describe the organizational context [Buckl 2011]:

Organizational culture. Open culture, political, culture of negative feedback, culture of positive feedback

Tool support. EAM tools, enterprise 2.0 tools, office tools

Background of the initiative. Bottom-up initiative, top-down initiative, pilot initiative

Organization of the IT department. Centralized IT department, federated IT department, decentralized IT department

In addition to the organizational context the actual maturity level of an EAM approach might impact the applicability of solutions documented by EAM patterns. Therefore, the respective maturity level has to be documented when observing patterns in practice. The following maturity model can be used to assess the EAM maturity level.

—business silos, standardized technology, optimized core, business modularity [Ross 2003].

—no program, informal program, repeatable program, well-defined program, managed program and continuously improving vital program [National Association of State Chief Information Officers 2003].

—Schekkerman [2006] uses six stages and eleven focus areas to assess EAM maturity.

—The GAO maturity framework (Version 2.0) uses seven stages for each of the 15 identified critical success attributes to assess EAM maturity [United States Government Accountability Office 2010].

—van Steenbergen et al. [2010] distinguish between 17 different but interrelated EAM focus areas. For each the maturity can be determined on a four level scale.

3.3 Architectural principles

The steps to be taken to address a given concern described by M-Patterns are not sufficient to capture all steering mechanisms used by today's enterprise architects. A more coarse-grained steering mechanism are architectural principles [Greefhorst and Proper 2011] which form general rules and guidelines for the evolution of the EA. Their scope can vary in terms of architectural levels and business domains. While some principles might help to address a given concern others do not. Therefore, the EAM pattern language has to be extended to include architectural principles in addition to M-Patterns. Typical principles include but are not limited to [Greefhorst and Proper 2011; The Open Group 2011]:

—Information Management is Everybody's Business
—Common Use Applications
—Service Orientation
—Compliance with Law
—Protection of Intellectual Property
—Data is Shared
—Common Vocabulary and Data Definitions
—Technology Independence
—Interoperability
—Business Units are Autonomous
—Customers have a Single Point of Contact
—Data are Exchanged at Real Time

3.4 Documents, reports, metrics and their visualization

Currently, V-Patterns provide qualitative visualizations of the information used to describe an EA. Today, other forms for presenting such information are used including regular documents (e.g. the IT strategy), slide decks (e.g. for communicating with business units) and quantitative representations (metrics) [Matthes et al. 2012] with respective visualizations (bar charts, pie charts, time series) [Roth et al. 2014]. Because all these different types provide a viewpoint on the architectural data the concept of V-Patterns can be extended to cover at least the mentioned sub groups. In line with the extension proposed in Section 3.1 to include typical stakeholders for V-Patterns they should also be documented for documents, reports and metrics.

3.5 Data collection patterns (DC-Patterns)

Many of the already known EAM patterns require some kind of data collection. The data needed is described by I-Patterns but how this data should be obtained was out of the scope of the EAM pattern language. Due to the increasing maturity and increasing degree of automated data integration the respective processes are subject to be observed and documented using the pattern approach. Regarding the classes, attributes and relationships modeled within I-Patterns for each of those elements detailed information is required. Thus, every such element qualifies for being backed by a data collection pattern (DC-Pattern) once. To document DC-Patterns we propose the following structure:

Frequency. How often is the data collected/updated? Real-time, daily, weekly, monthly, quarterly, yearly, event-driven

Responsible person. The person who is responsible for the data collection

Data source. The person/role or IT system having the required data

Type of integration. The way how the data is integrated with other EA data. Manual interface, software supported interface (e.g. Excel), automated interfaces, no integration

3.6 Relationships to other standards

Since the initial publication of the EAM pattern language in 2007 different EAM standards (e.g. TOGAF [The Open Group 2011]) and common modeling notations (e.g. ArchiMate [Iacob et al. 2012]) emerged. Therefore, all patterns should – where appropriate – document existing relationships of their concepts to those standards. In particular, this applies to the glossary of I-Patterns. Therefore, a new section named References should be added to all types of EAM patterns.

4. EXEMPLIFYING THE EXTENDED EAM PATTERN LANGUAGE

During various research projects conducted in close cooperation with industry, we have seen that necessary observations required to use the extended EAM pattern language proposed in Section 3 can indeed be made in practice today. Therefore, we performed a structured survey among 13 companies in Germany to demonstrate its applicability. In the following, we describe the data collection process in detail and present some novel patterns observed in industry at least three times [Coplien 1996] using the extended structure of the EAM pattern language.

4.1 Survey design

Between June and August 2014 we surveyed 13 companies about their current EAM approaches. The companies are all headquartered in Germany, mostly belong to the financial service sector and include also an energy supplier and a governmental agency. Their number of employees ranges between 1,500 and 15,000. All of them established EAM years ago and use a dedicated EAM tool. The survey included open and closed questions about currently used visualizations, modeled information, data collection processes and stakeholders (see also appendix).

The new EAM patterns presented in the following sections exemplarily demonstrate the proposed extension of the EAM pattern language based on the results of this survey. The first pattern is included because it deviates from the typical understanding of V-Patterns and is enriched with stakeholder information. The second pattern is presented because it describes a concept very common today but not yet included in the EAM Pattern Catalog. The third pattern is of type data collection to illustrate this new type of EAM pattern.

4.2 New V-Pattern (Document): APPLICATION PROFILE

Context
Your company uses several business applications.

Problem
You want to provide an overview of business applications which satisfies information demands of most stakeholders.

Solution
The most important elements satisfying the information need of most stakeholders include the application's name, description, list of used infrastructure components, list of information flows to other business applications, the functional domain or business capabilities supported by this application and a list of business processes and organizational units using this particular business application. Use a plain textual information representation.

Application A

This text provides a short overview of Application A. It includes it's main functions and purpose.

Responsibilities
Technically responsible: John Doe

Interfaces
- Application A → Application B
- Application B → Application A
- Application C → Application A

Infrastructure components and platforms
- MySQL database Server 5.0
- Tomcat Webserver 7.0

Organizational units
- Marketing department

Fig. 4. A profile for Application A

Forces

Holistic vs. selected information. Summarizing all available information for a specific business application provides a holistic view whereas the user might be distracted and prefer selected information parts.

Textual vs. visual representation. While lists and textual descriptions might be preferred by some users, others might prefer visual representations.

Consequences
Due to the variety of stakeholders a suitable publishing mechanism needs to be identified accounting for accessibility as well as suitable advertisement to ensure that relevant stakeholders know this source of information.

Stakeholders
All stakeholders mentioned in Section 3.1 except *suppliers*.

See also
The visualized information is based on the following I-Patterns: *Infrastructure Usage with Platforms*, *Interfaces and Information Flows*, *Business Application and Organizational Unit Relationship*.

4.3 New I-Pattern: INFRASTRUCTURE USAGE WITH PLATFORMS

Context
Your company uses business applications which include various infrastructure components.

Problem
You want to model how predefined bundles of infrastructure elements (platforms) are used by business applications but keep the flexibility to assign infrastructure elements also directly to business applications.

Solution
This information model pattern provides concepts for describing reusable bundles of infrastructure components called infrastructure platforms which are used by business applications.

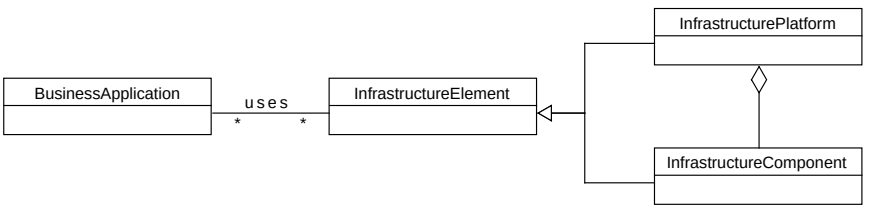

Fig. 5. UML model for I-Pattern *Infrastructure Usage with Platforms*

BusinessApplication. A software system, which is part of an information system within an organization. An information system is therein understood as a socio-technological system composed of a software system (i.e. the business application), an infrastructure, and a social component, namely the employees working with the system. An information system is further described as contributing to the business process support demanded by the organization.

InfrastructurePlatform. Infrastructure components of different types (e.g. database server, web-server and operating system) can be grouped to a reusable bundle for performance as well as standardization reasons.

InfrastructureComponent. Infrastructure components are deployed middle-ware or hardware systems e.g. a database management system.

InfrastructureElement. Infrastructure elements form a generic concept summarizing all infrastructure components as well as platforms made thereof.

Forces

Components vs. Platforms. When infrastructure components are modeled as instances, a huge modeling effort is necessary when assigning them to business applications. If some of those components are frequently used together, then platforms are more convenient. On the other hand, using only predefined platforms limits the flexibility of the business application designer.

Consequences
The observed solution allows to build business applications which are based on infrastructure components, infrastructure platforms or both. Therefore, inconsistent or double modeling can occur limiting comprehensibility. Nevertheless, flexibility is preserved by not forcing business applications to use infrastructure platforms.

Variant
A simplified variant of this I-Pattern is the *Infrastructure Usage* pattern.

4.4 New DC-Pattern: BUSINESSAPPLICATION DATA COLLECTION

Context
Your company uses business applications to support business processes.

Problem
You want to establish a data collection process for your BusinessApplications which minimizes the necessary effort and ensures responsibility of appropriate people.

Solution

Frequency. BusinessApplications are documented at certain events, e.g. go-live or maintenance project closure.

Responsible person. Responsible person is the IT architect.

Data source. Configuration management database. Otherwise knowledge of the IT architect or interviews.

Type of integration. Software-supported integration with a configuration management database.

Attributes to be documented. Name, abbreviation, description, current life-cycle phase, go-live date.

See also
The concept BusinessApplication is used, e.g., by the following I-Patterns: *Business Application and Organizational Unit Relationship*, *Business Application Lifecycles* and *Infrastructure Usage with Platforms*.

5. CONCLUSION AND OUTLOOK

Given the large scope of EAM and the heterogeneity of companies EAM is a field susceptible to practice-driven design research, e.g. via the Pattern-based Design Research method [Buckl et al. 2013]. The EAM pattern language provides the structure for documenting practice-proven solutions to recurring EAM problems [Buckl et al. 2007] [Ernst 2008]. The EAM Pattern Catalog currently forms the most comprehensive collection of EAM patterns [Buckl et al. 2008b]. Based on this foundation, we proposed an extension of the EAM pattern language to account for the rising number and maturity of EAM initiatives in practice as well as the availability of more detailed information. The extended EAM pattern language includes role descriptions for typical stakeholders of Concerns, M-Patterns and V-Patterns, respective influence factors and maturity levels, architectural principles

complementing M-Patterns, subgroups of V-Patterns (documents, metrics, visualizations), data collection patterns describing EA data collection processes and links to other EAM standards and frameworks to allow for an easy integration/comparison. This extended EAM pattern language has been evaluated by observing and documenting new EAM patterns within 13 organizations whereof three are presented in this article.

Nevertheless, the expected value of including stakeholders has not yet been demonstrated due to the small number of investigated companies. By assessing a larger number of companies in the near future we hope to be able to demonstrate this as well. Furthermore, a deeper inspection of influence factors and maturity levels of the surveyed companies would shed light on the applicability of the new sections during pattern documentation. It would also be interesting to see if the EAM pattern collection is able to form the basis for the design of an entire EAM function. Therefore, corresponding tool support, would be helpful. To close the research activity cycle of the PDR method [Buckl et al. 2013], the observation of changing pattern implementations in practice might be used to further develop the EAM pattern knowledge base.

REFERENCES

ALEXANDER, C., ISHIKAWA, S., AND SILVERSTEIN, M. 1977. *A Pattern Language*. Vol. 2. Oxford University Press, New York.

BUCKL, S., ERNST, A. M., LANKES, J., AND MATTHES, F. 2008a. Enterprise Architecture Management Pattern Catalog.

BUCKL, S., ERNST, A. M., LANKES, J., SCHNEIDER, K., AND SCHWEDA, C. M. 2007. A pattern based approach for constructing enterprise architecture management information models. *Wirtschaftinformatik Proceedings 2007*, 65.

BUCKL, S., ERNST, A. M., MATTHES, F., AND LANKES, J. 2008b. Enterprise Architecture Management Pattern Catalog.

BUCKL, S., ERNST, A. M., MATTHES, F., RAMACHER, R., AND SCHWEDA, C. M. 2009. Using enterprise architecture management patterns to complement TOGAF. In *Proceedings of the IEEE Enterprise Distributed Object Computing Conference, 2009*. IEEE.

BUCKL, S. M. 2011. Developing organization-specific enterprise architecture management functions using a method base. Ph.D. thesis, München, Technische Universität München, Diss., 2011.

BUCKL, S., MATTHES, F., SCHNEIDER, A. W., AND SCHWEDA, C. M. 2013. Pattern-based Design Research–An Iterative Research Method Balancing Rigor and Relevance. In *Design Science at the Intersection of Physical and Virtual Design*, J. vom Brocke, R. Hekkala, S. Ram, and M. Rossi, Eds. Springer, Berlin, Heidelberg, 73–87.

BUSCHMANN, F., HENNEY, K., AND SCHMIDT, D. C. 2007. *Pattern-oriented software architecture*. Wiley series in software design patterns. Wiley, Chichester, England and Hoboken, N.J.

COPLIEN, J. O. 1996. *Software Patterns: Management Briefs*. Cambridge University Press, Cambridge, UK.

ERNST, A. M. 2008. Enterprise Architecture Management Patterns. In *Proceedings of the 15th Conference on Pattern Languages of Programs*, J. Yoder and A. Aguiar, Eds. ACM.

GREEFHORST, D. AND PROPER, H. A. 2011. *Architecture Principles*. Springer, Berlin, Heidelberg.

IACOB, M., JONKERS, H., LANKHORST, M., PROPER, E., AND QUARTEL, D. A. C. 2012. ArchiMate 2.0 Specification.

IEEE AND ISO/IEC. 2007. Systems and Software Engineering - Recommended Practice for Architectural Description of Software-Intensive Systems (ISO/IEC 42010 IEEE Std 1471-2000).

LANGENBERG, K. AND WEGMANN, A. 2004. Enterprise architecture: What aspects is current research targeting.

MATTHES, F., MONAHOV, I., SCHNEIDER, A., AND SCHULZ, C. 2012. EAM KPI Catalog v 1.0. *Technische Universität München, Munich, Germany, Tech. Rep*.

MOSER, C., JUNGINGER, S., BRÜCKMANN, M., AND SCHÖNE, K.-M. 2009. Some Process Patterns for Enterprise Architecture Management. In *Software Engineering (Workshops)*. 19–30.

NATIONAL ASSOCIATION OF STATE CHIEF INFORMATION OFFICERS. 2003. NASCIO Enterprise Architecture Maturity Model version 1.3.

ROSS, J. W. 2003. Creating a Strategic IT Architecture Competency: Learning in Stages.

ROTH, S., ZEC, M., AND MATTHES, F. 2014. Enterprise Architecture Visualization Tool Survey 2014.

SCHEKKERMAN, J. 2006. Extended Enterprise Architecture Maturity Model.

THE OPEN GROUP. 2011. TOGAF Version 9.1.

UNITED STATES GOVERNMENT ACCOUNTABILITY OFFICE. 2010. Organizational Transformation: A Framework for Assessing and Improving Enterprise Architecture Management (Version 2.0).

VAN STEENBERGEN, M., SCHIPPER, J., BOS, R., AND BRINKKEMPER, S. 2010. The Dynamic Architecture Maturity Matrix: Instrument Analysis and Refinement. In *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops*, A. Dan, F. Gittler, and F. Toumani, Eds. Lecture Notes in Computer Science Series, vol. 6275. Springer Berlin Heidelberg, 48–61.

WINTER, R. 2009. Patterns in Business and Information Systems Engineering. *Business & Information Systems Engineering 1,* 6, 468–474.

A. QUESTIONNAIRE

A.1 General questions

Name, Organization, Role description, Personal EAM experience level, Organizational EAM experience level, Contact information

A.2 Information model questions

Please describe for each EA concept you model: Name, Name of parent concept (if applicable), Number of modeled instances.

Please describe for each attribute you model for this class: Name, Responsible person/role for data collection, Year of introduction, Data source, Data quality, Data quantity, Frequency of updates.

Please describe for each relationship between two concepts: Name, Responsible person/role for data collection, Year of introduction

A.3 Visualization questions

Please describe for each visualization of your EA: Name, Type, Generation method (manual, semi-automated, automated), Included EA concepts and relationships, Addressed concern, Frequency of generation, Interested stakeholder.

EuroPLoP '15, July 08 - 12, 2015, Kaufbeuren, Germany

ACM 978-1-4503-3847-9/15/07 . . . $15.00
DOI: http://dx.doi.org/10.1145/2855321.2855367