Softw Syst Model (2016) 15:397–426
DOI 10.1007/s10270-014-0407-3

REGULAR PAPER

A situational method for semi-automated Enterprise Architecture Documentation

Matthias Farwick · Christian M. Schweda · Ruth Breu · Inge Hanschke

Received: 14 August 2013 / Revised: 21 January 2014 / Accepted: 1 March 2014 / Published online: 3 April 2014

Abstract The business capabilities of modern enterprises crucially rely on the enterprises' information systems and underlying IT infrastructure. Hence, optimization of the business-IT alignment is a key objective of Enterprise Architecture Management (EAM). To achieve this objective, EAM creates, maintains and analyzes a model of the current state of the Enterprise Architecture. This model covers different concepts reflecting both the business and the IT perspective and has to be constantly maintained in response to ongoing transformations of the enterprise. In practice, EA models grow large and are difficult to maintain, since many stakeholders from various backgrounds have to contribute architecture-relevant information. EAM literature and two practitioner surveys conducted by the authors indicate that EA model maintenance, in particular the manual documentation activities, poses one of the biggest challenges to EAM in practice. Current research approaches target the automation of the EA documentation based on specific data sources. These approaches, as our systematic literature review showed, do not consider enterprise specificity of the documentation context or the variability of the data sources from organization to organization. The approach presented in this article specifically accounts for these factors and presents a situational method for EA documentation. It builds on four process-supported documentation techniques which can be selected, composed and applied to design an organization-specific documentation process. The techniques build on a meta-model for EA documentation, which is implemented in an EA-repository prototype that supports the configuration and execution of the documentation techniques. We applied our documentation method assembly process at a German insurance company and report the findings from this case study in particular regarding practical applicability and usability of our approach.

Communicated by Prof. Franck Barbier.

M. Farwick (✉) · R. Breu
Institute of Computer Science, University of Innsbruck, Innsbruck, Austria
e-mail: matthias.farwick@uibk.ac.at

R. Breu
e-mail: ruth.breu@uibk.ac.at

C. M. Schweda · I. Hanschke
iteratec GmbH, 82008 Munich, Unterhaching, Germany
e-mail: christian.schweda@iteratec.de

I. Hanschke
e-mail: inge.hanschke@iteratec.de

Keywords Enterprise Architecture · Documentation · Maintenance · Model · Automation · Situational method

1 Introduction

Enterprise Architecture Management (EAM) is a management function that targets inter alia the optimization of business-IT alignment, IT cost savings as well as IT standardization. The common means to achieve these goals is the creation of Enterprise Architecture (EA) models that represent the business side of the enterprise and document its dependencies to the supporting information systems and the underlying IT infrastructure, as summarized by Winter et al. [53]. These models are then used to analyze the current state of the architecture to plan transformations toward future optimized states [3]. EAM frameworks such as The Open Group Architecture Framework (TOGAF) [49] or the Zachman Framework [54] provide high-level guidance for the execution of this management function and information models that hold EA data. In practice, specialized EAM tools are often used to model the architecture and visualize it in a stakeholder-specific way. Still, organizations face numerous challenges when practicing EAM that diminish the return on often large EAM investments. In an EA literature survey, Lucke et al. [40] present a categorization of these EAM challenges. The main identified categories are *Management Issues*, such as lack of management support for the EA endeavor, *Semantic Problems*, that complicate the communication of different domains/departments, *Representation Issues*, that refer to the difficulty of creating adequate views for the stakeholders, as well as the *Complexity Problem* that refers to "the challenge to model and maintain models of heterogeneous, dynamic and complex social systems." In other words, in practice, it is difficult to maintain a model of systems with many components and many connections that are constantly changing. The human factor adds to the complexity by introducing manual and biased abstractions into modeling and interpretations of models. As stated above, Lucke et al. see this as one of the key challenges in EA practice that needs to be tackled by research.

In our research project "Living IT-Landscape Models," we try to tackle the aforementioned complexity problem, i.e., we aim at facilitating the documentation of the current, or often called as-is, architecture. Our overall goal is to increase the cost-benefit ratio of EA documentation which, we argue, will lead to a higher return on investment of EAM since better decisions can be made based on more accurate information.

Let us, for now, further detail on the complexity problem described by Lucke et al. and other authors [11]. As described by Winter et al. [53], EA models typically contain elements describing the business side of the organization and how it is supported by processes, applications as well as the IT infrastructure. Each of these so-called *EA model layers* contains an abstract and often aggregated representation of the actual architecture. These abstractions are often subjective interpretations of reality, as described in [7], such that it is mostly not feasible to automatically generate such models from existing data sources.

Although each architectural layer contains only a fraction of the possible detail level, the size and interconnected nature of the resulting model is usually still very large. Hence, in practice, it is often hard to maintain the quality of such large and complex models that need to be aligned with reality at reasonable costs. In two practitioner surveys with, in sum, 123 participants, we validated the claim that the maintenance of EA models is a relevant and major problem in the EAM practice [19,20]. This is in line with the observations of several other authors [36,52]. Nevertheless, EA literature from research and practice, as well as EA frameworks give little advice on how to maintain an EA repository at adequate quality for the enterprise-specific, organizational, financial and technological context as well as information demands.

Recent work [16,18,19,35] applies automated data collection to improve the actuality of EA models. However, these works focus only on the collection of structured data and its integration into the repository. As outlined by Hauder et al. [32] and validated by our empirical research in [20], there are numerous challenges in automatically importing structured data into an EA model (cf. Sect. 3). We argue that the circumstance that each organization raises unique information demands, according to Aier et al. [4], makes it necessary to adapt a method for EA documentation to each individual organization, i.e., to derive an organization-specific EA documentation method.

Hence, our main research question is: How can EA documentation processes be devised and supported by a tool that optimally supports the information demand and context of a specific organization?:

Such situated approach to EA documentation that can be tailored to the organization appears to be absent in EA literature (cf. Sect. 2.2).

In the paper at hand, we present a method for designing organization-specific EA documentation processes. Therein we lay specific focus on:

1. *Reducing manual modeling* via automation.
2. *Increasing the actuality* of the overall EA model by triggering manual, semi-automated and automated model updates at the *right time*.
3. Optimizing the quality of resulting EA models by automated as well as forced *quality assurance*.

Our approach consists of several configurable documentation techniques, a method assembly process as well as an accompanying meta-model to hold necessary meta-data for the process execution.

We propose several techniques to assemble semi-automated organization-specific EA documentation processes. The high-level techniques are:

1. Task-based reminders for the correct persons that are responsible for parts of the model at specific points in time (*Reminders (T0)*).
2. Automated structured data collection with semi-automated data quality assurance (*Automated Collection (T1)*).
3. External events from information systems such as project portfolio management (PPM) tools that trigger manual tasks at the right time (*External Events (T2)*).
4. EA repository internal model events that trigger manual tasks, such as element expiry (*Internal Model Events (T3)*).

We present these techniques in detail in Sect. 4.

In this work, we applied the design science research paradigm according of Hevner et al. [34]. In this vein, the paper presents three contributing artifacts. First, we propose a situational method for the construction of semi-automated EA documentation processes that aims at reducing manual work. Second, we present a meta-model that when instantiated can flexibly support the organization-specific documentation processes. The third artifact is the prototypical implementation of a tool that supports such situational methods. The requirements for these artifacts were gathered from systematically reviewing related literature, two empirical studies as well as from our own experience as consultants in the EA field. We establish *rigor* by basing our work on the foundational research of situational method engineering (SME) [31], which aims on constructing methods that can be adapted to the organization-specific contexts. We highlight the *relevance* of our work by presenting the results of two empirical studies on EA documentation and a systematic literature review which both showed that EA documentation is underresearched and a pressing challenge in practice. By highlighting our previous work in this area, we emphasize that the work is the result of an iterative *design science search process* (cf. Hevner et al. [34]). The contribution of this paper is both constituted by the described artifacts as well as by providing foundations for research conducted on situational EA documentation. Finally, the artifacts are evaluated in multiple ways. We practically evaluate the artifacts by showing their *implementability* with a prototypical implementation, as well as an additional partial implementation of the presented approach in an industry project. With the latter, we also establish *efficacy* and *utility* of our approach. In addition, we evaluated our work theoretically by comparing its capabilities with related literature and the EA documentation challenges that can be identified therein.

The remainder of this paper is structured as follows. In the next section, we first introduce our preliminary work in this area and also analyze related work on EA documentation via a systematic literature review on EA documentation. In Sect. 3, we present EA documentation challenges that we identified in EA research literature to motivate our research. The main part of this paper (Sect. 4) first briefly introduces the reader to the concept of situational method engineering and then discusses context factors as well as typical EA documentation roles that influence the documentation processes. We then proceed by detailing our documentation techniques that can be assembled to form an organization-specific documentation method. The main section is concluded by the description of this method assembly process. In Sect. 5, we then detail our meta-model that holds the meta-data to support the processes and is used to model organization-specific information models. In the evaluation Sect. 6, we then first describe our EA repository prototype that can execute the assembled documentation processes. We continue the evaluation by presenting a case study which we performed at a German insurer where we applied our method assembly process and implemented according tooling. We conclude the evaluation by discussing how our approach covers the EA documentation challenges identified in Sect. 3. In the final section, we conclude and discuss open issues of our approach.

2 Preliminary and related work

In this section, we present the work we have already done in the field of (semi-automated) EA documentation and analyze related work via a systematic literature review (SLR).

2.1 Preliminary work

In our preliminary work, we have created the foundational concepts for our situational method for EA documentation. In the initial phase, we focused solely on full automation of data collection. To lay the motivational basis for our research, we first conducted a survey on the requirements for automated EA documentation [19]. From these requirements and a literature review, we derived semi-automated data collection processes that ensure the mapping of structured data from external data sources into an EA repository in adequate quality. These processes were presented in [18]. To support the processes with concepts like identity reconciliation to identify duplicate model elements, internal model events, such as expiry, or the distribution of manual tasks to the correct stakeholders we presented a corresponding meta-model in [21] which holds the required meta-data and provides elements to create organization-specific information models.

However, it is clear that organizations always have different sources of EA-relevant structured data available. This is why we conducted another survey on structured data sources for the purpose of automated EA documentation [20]. One of the key findings of the survey was that, although data about the more technical EA layers can be automatically collected, in many cases relevant data will not be available in an automated manner with current approaches, such as the ones presented in [16,35].

This raised the question of how the actuality of EA models can be kept high without the existence of automated updates. Our approach to this is the integration of EA-relevant change events that can be fired from information systems like project portfolio management tools. Events such as an *architecture change project end event* can be used to trigger manual modeling tasks of the respective architecture, at the point in time when the changes are implemented in the enterprise. This approach was initially presented in [22].

The realization that all organization differ in their context for EA documentation lead us to a shift in the research question. This research question, which we presented in the introduction, does not exclude manual data collection and fosters situativness of the approach to documentation. The results of the research on that newly formulated question are presented in this paper.

In the following, we continue by analyzing related work in the area of EA documentation.

2.2 Related work (systematic literature review)

Although the problem of maintaining an up-to-date model of the EA has been identified by researchers and practitioners alike, we noticed that it is not holistically addressed in EA literature. To verify this observation, we executed a systematic literature review with the main research question to identify categories of related work on manual and automated EA documentation. In addition, we laid special attention on whether any of the identified literature includes situativeness in their approach. In the following, we briefly describe the applied literature review method, outline the identified categories and contained papers. Finally, we discuss the gap between the identified literature and the work presented in this paper which constitutes its main contribution. More details on the research method, such as the full list of search terms, can be found in Appendix 8.1.

2.2.1 Review method

The review was executed between the 9th and 25th of July 2013. As a first step, Google Scholar[1] was searched using a set of search terms that included the terms "enterprise architecture" in conjunction with relevant search terms such as "model maintenance" or "documentation." In addition, we manually inspected the proceedings of relevant scientific events such as the Trends in Enterprise Architecture Research (TEAR) workshop. Also, relevant scientific journals like the Journal of Enterprise Architecture (JEA) were inspected. We did not include EA frameworks in our analysis; however, we can state with confidence that the leading frameworks such as Zachman [54] or TOGAF [49] fall short in giving concrete advice on how to collect EA information in practice, as stated by Buckl et al. [14].

2.2.2 Categorization of documentation approaches in research literature

The review resulted in a list of 28 publications targeting EA documentation or giving recommendations on how to conduct it. We categorized them into eight different groups of approaches to EA documentation where one paper can be included in several categories. In the following, we discuss these categories.

Interviews and forms This most basic form of data collection is the interview of related stakeholders including the use of predefined forms that should be filled out by the data providers. Not surprisingly, this form of documentation is suggested exclusively by EA books originating from practice [1,17,29,37,38]. This is the traditional form of data collection for EA modeling, which seems rarely sufficient on their own to keep EA models in adequate quality. Hence, the publications in the following categories try to enhance this basic practice with tooling, automation and according data collection processes.

Wiki collaboration A more sophisticated category of EA documentation approaches tries to leverage Web 2.0 technologies in the form of wikis to include knowledge of more stakeholders into an EA repository [13,25,30]. Here, the authors argue that the inclusion of many stakeholders which do "casual" modeling increases the actuality and availability of architectural information. In this field of research, one can distinguish between the use of semantic wikis such as in [25,30] and hybrid wikis that store structured and unstructured data [13].

Defined data collection processes As the next step toward a more sophisticated approach to EA data collection, some authors propose pre-defined processes that assign data collection roles to stakeholders and include steps for automated data collection [18,21–23,42]. However, no concepts for organization-specific adaption of the processes are discussed by these works.

Generic import concepts Another category of literature is mentioning the idea of import of data. The literature stems both from research and practice [9,18,24,25,29,37,39,47]. Some works give hints on the components of an implementing tool, such as [25,39]. The majority, however, stays vague on how to tackle technical challenges such as data mapping or the avoidance of duplicates.

Tool-, model, semantic integration Yet another category of publications deals with the idea of integrating other existing tools or models [7,38,46,48,51]. Ter Doest [48], Arbab et al. [7] and Lankhorst [38] focus on the integration of the content of diverse modeling tools into the main repository fostering a seamless navigation between the tools. Chen et al. [51] as well as Schmidt et al. [46] make use of semantic technologies, i.e., ontologies, in order to integrate models in different data formats into a main repository.

Automation via specific data sources More detailed descriptions on how to integrate data from existing data sources are provided by some papers which concentrate on integration issues of specific data sources [6,16,35]. Holm et al. [35] as

[1] http://scholar.google.com.

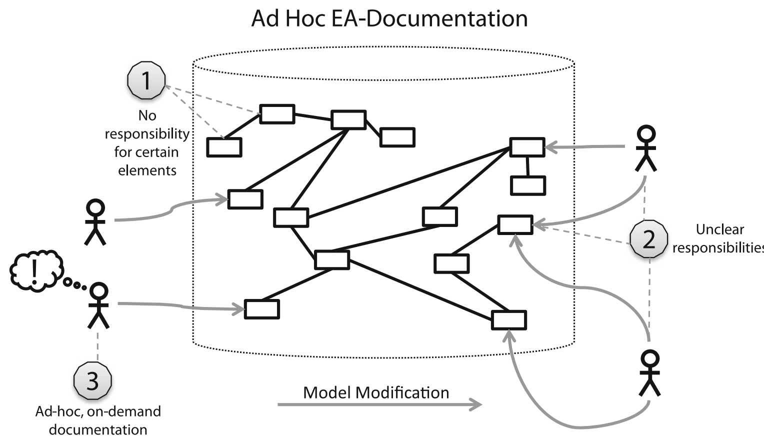

Fig. 1 Ad hoc EA documentation processes

well as Alegria et al. [6] make use of network analysis tools in order to infer information on the IT infrastructure. Buschle et al. [16] on the other hand interpret the configuration of an Enterprise Service Bus (ESB) to include knowledge on communicating information systems in the EA model. These approaches have in common that they are limited to a specific layer of the EA model and do not include methods on how to ensure the quality of the collected data.

Change events and notifications Another class of publications recognizes the importance of receiving events from other information systems that indicate architecture-relevant changes [1,15,22,29,47]. Some simply mention the importance of recognizing external events [1,29,47]. Others, such as the thesis of Buckl consider triggers for manual action in the provided meta-model [15]. Only one paper actually discusses implementation issues of utilizing external events to drive automated EA documentation processes [22].

Conflict resolution and quality assurance Finally, a set of papers considers the problem of resolving conflicts and quality assurance when data are automatically collected [18,21,23,41,42,45]. Here, again only one paper discusses implementation details for conflict resolution mechanisms [21].

To summarize the results of the above categorization, one can say that there exist small islands of research on EA documentation which are driven by few different researchers. Also, no work was identified that combined the topic of EA documentation with situational method engineering.

3 Challenges in (automated) Enterprise Architecture Documentation

As described in the introduction, the documentation of Enterprise Architectures often poses a challenge in practice. In this section, we highlight the specific challenges that make EA documentation a difficult task.

A common problem is that too fine-grained EA data are collected. This increases the effort that needs to be applied in order to keep the quality of the model high. Literature from research [5], as well as practice [28] advices to start with small information models that address a limited set of EA-relevant questions. Once it is clear that the model size can be handled in adequate quality, the information model can be expanded to support deeper analysis. Also, based on the findings presented in [44], the team structure plays a major role for the success of EA documentation. Another major finding from our study was that organizations that have defined EA data collection processes with clearly defined responsibilities have fewer difficulties with keeping their EA models up to date. In [36], Kaisler et al. state that currently "there is minimum tool support to track collection of entities." While this has certainly improved in recent years, EA tools still do not fully satisfy the data collection needs of EA practitioners [44,52]. In many cases, the EA-relevant information is dispersed within the organizational units. It is therefore often difficult to get hold of the right stakeholders to collect the desired data [20].

Figure 1 visualizes the typical ad hoc documentation processes applied in many organizations. Different stakeholders maintain the EA model in an "on-demand" manner or simply when they are accidentally reminded of maintaining the model. The numbers indicate some of the core problems in such unstructured approaches. Number 1 shows the problem of unclear responsibilities for the quality of certain model elements in the repository. In many cases, model elements become orphaned after they have been initially entered to the repository and no person is assigned responsibility for the quality of the element. The number 2 indicates the problem of no assigned responsibility for certain elements. If the responsibilities are not explicated, it can quickly lead to inconsistencies. Finally, number 3 shows that the actual documentation is just triggered in an ad hoc manner and not as the result of clearly defined events, such as the event of a finished architecture change project.

Table 1 Challenges in automated EA documentation according to Hauder et al. [32] and additional challenges discovered in [20] (italic rows)

ID	Challenge description	Covered by rel. work
Data challenges		
DC 1	Overload of productive systems due to large volume of transactions for automated data collection	
DC 2	Selection of the right productive systems as information sources for EA documentation	
DC 3	Detection of changes in the real-world EA and their propagation to the EA model in the repository	[6,16,22,35]
DC 4	Data quality in the productive systems not sufficient for the documentation of EA information	[21]
DC 5	*Often no structured data sources for model elements in the upper layers*	[22]
DC 6	*Detecting removed elements*	
Transformation challenges		
TC 1	Model transformation for the exchange of EA information necessary due to missing interfaces and standards	[16,21,35,51]
TC 2	Ambiguous concepts imported from the productive systems in the organization require a consolidation	[18,21,45]
TC 3	Administration of collected data from the productive systems is required to ensure actuality and consistency	[18,21,45]
TC 4	Duplicate EA elements imported from different productive systems of the organization	[21,45]
TC 5	Abstraction between the EA model and the imported information from productive systems of the organization	
Business and organizational challenges		
BC 1	Security vulnerability through monitoring tools in the infrastructure of the organization	
BC 2	Not enough return on investment due to large initial investment efforts	
BC 3	Involvement of data owners for the maintenance of imported EA information	[18,21,45]
BC 4	*Responsibility for data sources, model elements and quality assurance of automated changes not clear*	[21,45]
Tooling		
T 1	Synchronization of changes in the EA model to the underlying productive systems	
T 2	Collection of information not relevant or too fine grained for decision makers in the EA	
T 3	Analyses have to be decoupled from the meta-model	
T 4	Not enough tool support for automated EA documentation available	

Hauder et al. summarize their view on the challenges in automated EA documentation in [32]. This research is synthesized from a model integration case study, a literature review and a practitioner survey. The challenges collected in their work are categorized into Data, Transformation, Business and Organizational as well as Tooling challenges. *Data Challenges* are concerned with the issues of the integration of EA-relevant data sources, *Transformation Challenges* are concerned with the abstraction gap between source and target data models, *Business and Organizational Challenges* discuss issues around the business value, responsibilities and security issues of EA documentation, and finally *Tooling Challenges* relate to the issues around the tool support for (automated) documentation. We have listed the challenges presented by Hauder et al. in Table 1. In addition to the challenges identified in their work, we have added more challenges from our own experience. These are highlighted with a gray background.

As one can see from this table, the related work identified with the systematic literature review only covers a small percentage of the challenges identified by Hauder et al. In particular, mostly challenges of technical, but not organizational nature have been covered. For example, so far we could not identify any related work that tackles the problem of choosing which automated data source to choose (*DC 2*).

In Sect. 6, we revisit these challenges and discuss how the situated documentation approach presented in this paper addresses them.

In addition to these generic challenges, each organization has its own constraints in terms of resources, available data sources, team structure and information demand (see Sect. 4.2). Hence, each organization needs tailored documentation processes. We present our approach to constructing such situated processes in Sect. 4.

4 A situational method for EA documentation

A missing piece in practice and research is a situational method that enables the assembly of various organization-specific documentation approaches in a structured and coherent way. In this section, we describe a situational method for EA documentation. It combines and consolidates the concepts and artifacts of our previous work with a structured method assembly process and a system for maintenance responsibility that can be tailored according to the context of each organization.

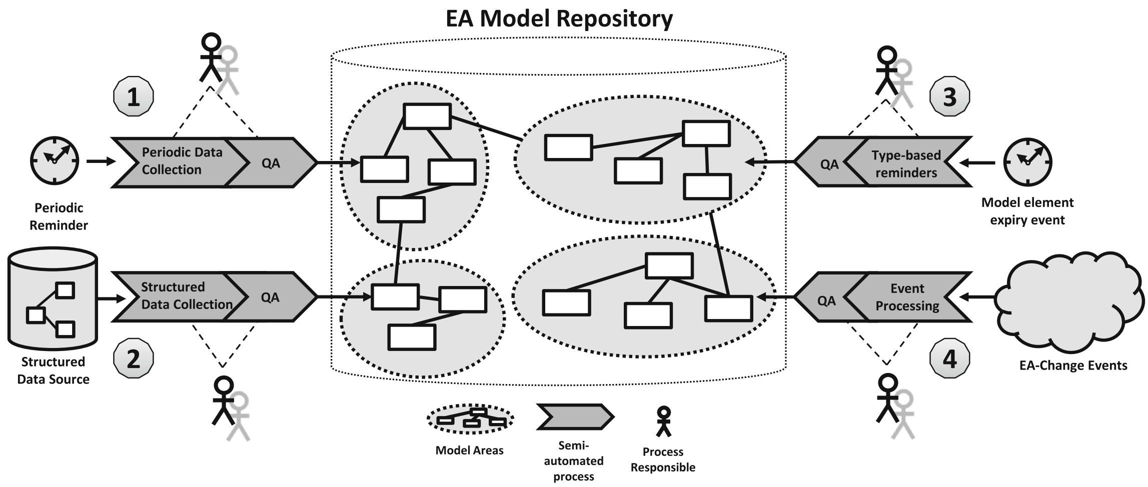

Fig. 2 Enhanced EA data collection process including automation techniques

The main constituents of the configurable data collection method are the processes and tooling concepts for:

- Recurring periodic manual data collection reminders for appropriate stakeholders.
- Semi-automated data collection from external structured data sources.
- An eventing mechanism that receives events from external information sources to initiate manual maintenance processes.
- Internal model events for appropriate stakeholders, such as model element expiry.

These main concepts are called *documentation techniques* in the following.

In Fig. 1, we showed the typical problems of EA documentation processes that are characterized by ad hoc maintenance and underspecified responsibilities. Figure 2 shows the enhanced processes that can be assembled with the approach presented in this section utilizing the documentation techniques that we detail further below. Each data collection or maintenance process is accompanied by an optional quality assurance process, clearly defined processing roles and defined parts of the model that are the target of the maintenance process.[2] The result of each process execution is that, to some degree, statements about the actuality and quality of the model elements can be made. For example, it is possible to analyze when a person checked a model element for its actuality for the last time or when it was last imported from data source. In Fig. 2, the numbers beside the process symbols indicate the type of the technique. The techniques will be further detailed in Sect. 4.4.

In the following, we first briefly introduce the reader to the foundational concepts of situational method engineering. We then list context factors that influence the decisions on when to apply which documentation technique. The roles that are needed to be defined for each fragment type are discussed in the following Sect. 4.3. We then describe the details of each method fragment in Sect. 4.4. As the final part of this section, we introduce the method assembly process that is used to tailor a situational documentation method for a specific organization, taking the context factors and the available method fragments into account.

4.1 Situational method engineering

The theoretical basis for our approach is situational method engineering (SME). Brinkkemper defines it as "the discipline to build project-specific methods, called situational methods, from parts of the existing methods, called method fragments." [8]. Originally, the concepts of SME have been created for assembling information system development methods [31], but have also been applied for other types of projects in research and practice (e.g., change engineering in Health Care [26]). As argued by Harmsen et al., the practice of SME provides a framework for the desired attribute of *controlled flexibility* of methods. The goal of controlled flexibility is to achieve a harmonization between rigid standardized (information system development) methods and the flexibility to adapt to organization-specific needs, by guiding the construction of methods via best practices.

In SME literature, the atomic elements of method construction are called *method fragments*, sometimes also called *method chunks* [33]. Method fragments and method chunks have slightly different definitions but serve the same purpose of method construction from atomic method parts [33]. In this work, we use the term *documentation technique* which results in process fragments and product fragments according to the SME definition. Process fragments, as the name already implies, are process definitions that can be reused

[2] We call them "Model Areas".

to serve a specific purpose. These fragments have a producing relationship with product fragments. Product fragments are artifacts whose instantiation behavior is governed by an information model that describes possible products. In the case of this work, process fragments describe EA data collection and maintenance processes which produce product fragments, i.e., updated or newly created EA model elements. In our approach, the process fragments are relatively organization independent, i.e., generic, but produce highly organization-specific product fragments. For example, an organization chooses the implementing the automation of automated collection of data about network nodes via a network scanner (cf. *Technique 1* in Sect. 4.4.2). The elements that are collected by the process fragment of the technique are mapped to the *Server* information model element in the organization's information model. The latter corresponds to the organization-specific nature of a product fragment. This is why the actual product fragments that are created by a process are actually not listed in the description of each technique. They are selected based on a specific situation, i.e., a specific information model.

4.2 Context factors of documentation

The decision for the selection of the strategies to assemble a situated EA documentation process depends on various organization-specific context factors. In the following, we list and discuss these factors which we identified from research literature.

– *Information demand* The most important context information for assembling the documentation process is the actual information demand of the EA model and its stakeholders. In general, it is important to distinguish between subjective and objective information need. The effort to collect data should hence be only made for data that is objectively needed as stated in [50]. In the context of EA, as pointed out in [28], it is important that in any EA documentation process, the collected EA data is actually needed to make specific decisions and can be collected and maintained with reasonable effort.
– *EA team structure* The documentation of the as-is architecture might be organized in a centralized fashion where a small number of enterprise architects collect and consolidate data. On the other hand, federated approaches, such as proposed in [23], or a mixture of both might be employed. As we found in our survey, this setup positively influences the documentation quality significantly [20].
– *Organizational structure and culture* Similar to the EA team structure, the organizational structure is an important context factor for EA documentation. The availability of information from stakeholders in potentially geographically dispersed organizational units influences the options for communication to drive documentation. Also there might exist an organizational culture that inhibits data sharing such as mentioned in [12].
– *Available structured data sources* Each individual organization is characterized by a set of information systems that potentially contain relevant EA information that can be reused for EA modeling. In [20], we have empirically analyzed the existence of such sources of EA data. Also, we studied the provided information types (i.e., on which layer of the EA model data can be provided) as well as the respective data quality. The type, availability and content of the data sources are an important context factor on which to base method construction of automated data collection.
– *Available event sources* In addition to automated structured data sources, organizations might have information systems that are able to produce relevant EA change events, as described in [20,22]. The availability as well as the type and quality of events is another important context factor for the method assembly.
– *Management commitment and budget* The introduction of new data collection processes and the implementation of automation techniques pose initial and ongoing costs. Hence, the support by management as well as the available budget needs to be taken into account when planning the method assembly.
– *Security concerns* Finally, the specific data sources might contain highly confidential data. This should be taken into consideration when discussing the inclusion of such data sources in any automated documentation process.

Above context factors are important decision criteria for assembling the situational documentation method as explained in Sect. 4.6 as are the documentation roles which we present in the next section.

4.3 Documentation roles

As it is unreasonable to assume full automation of EA documentation, there still need to be humans involved in any data collection and maintenance processes. In the following, we describe the roles that are necessary for the assembly of methods as well as their execution. For each documentation technique, we refer back to the here-described roles that are actually needed to execute a given technique. Individuals can potentially be assigned multiple roles.

1. *Documentation manager:* This role is responsible to oversee the overall documentation and in particular to assign the appropriate roles to the right stakeholders. In addition, this role needs to take action if specific tasks have not been processed for a too long time span.

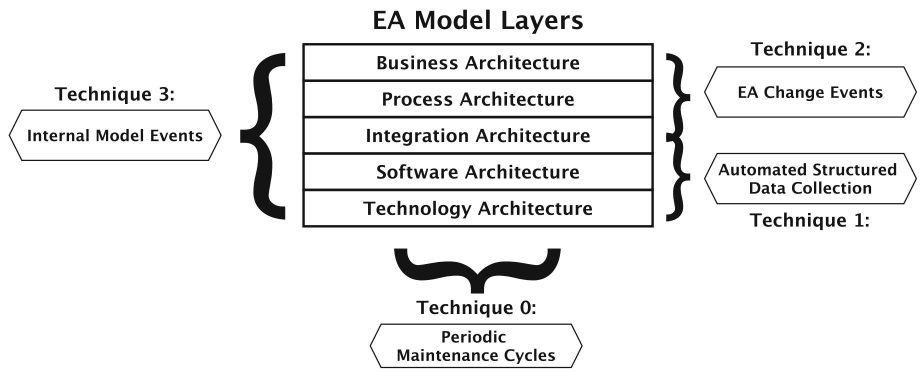

Fig. 3 Overview of EA model maintenance techniques and the EA layers they apply to

2. *Documentation method engineer:* The EA documentation method engineer is the person who coordinates the method assembly and oversees its execution as well as its performance. In particular, the feedback loop from the process execution as input for the method engineer is of major importance to improve the processes and maintain its quality.
3. *Data/event source manager (Technical):* This role is responsible for the technical integration of the EA repository with specific remote EA data and event sources.
4. *Event responsible:* Users who are assigned to handle architecture change events from external event sources, such as project portfolio management tools, have this role. They are in charge of interpreting the events and applying according changes to the EA model or forwarding the task to stakeholders with the appropriate knowledge.
5. *Data owner:* Data owners are the persons who have direct knowledge on specific parts of the EA model. They not necessarily are the persons who enter the data into the EA repository, but provide the authority for parts of the model. They can also be responsible for the data provided by a data source. In particular, this group of users is in charge of resolving conflicts resulting from data that were manually entered or originate from a data source they have been assigned responsibility for.
6. *EA stakeholders:* The EA Stakeholders are the group of people who are actually using the EA data to make decisions, e.g., regarding architectural changes. Typical examples are the CIO, Enterprise Architects and IT managers that need to stay informed about changes.

In the following, we use a RACI matrix[3] to show each role's responsibilities for the setup and execution of each technique. Such a matrix is commonly used to express the responsibility assignments of stakeholders to execute specific tasks. RACI stands for **R**esponsible, **A**ccountable, **C**onsulted and **I**nformed.

4.4 Techniques for model maintenance

Our SME approach acknowledges the fact that the majority of EA model contents are not fully automatically discoverable. Hence, different composable techniques including their processes for collecting those parts of the model are described in the method.

Figure 3 shows an overview of the typical EA layers according to Winter et al. [53] and how our documentation techniques support the maintenance of each layer. As can been seen in the figure, each of the techniques potentially only satisfies parts of the information demand of the EA stakeholders, and others are capable of covering the whole layer stack. The processes of each technique are relatively similar from organization to organization such that they only need little adaption in the method assembly process. For example, in some cases, quality assurance sub-processes need to be adapted. The produced artifacts, however, are dependent on the information model applied in the organization.

Figure 4 gives an overview of the basic components that play a role in our situative approach. It is inspired by the groundwork of Gutzwiller who proposed the basic constituting components of SME in [27]. On the bottom left of the figure, one can see the concept of an *EA tool*. This EA tool supports the execution of documentation *techniques* which can be initiated by different triggers, such as time-based events or others. Techniques are broken down into *documentation activities* which resemble process activities that have a sequence. These can be either automated activities or manual ones that need human intervention. In the latter case, *viewpoints* make it possible for stakeholders with specific *roles* to perform such manual activities. Activities lead to a *changed EA model* that conforms to an organization-specific *EA information model*.

The techniques and their accompanying process fragments can be seen as best-practice patterns. The key guiding principle for their selection was their coverage of the EA layer stack as visualized in Fig. 3. Hence, the goal was to devise a set of techniques that can potentially cover elements of all layers of the stack. They were developed via several iterations from different inputs, which we list in the following:

[3] http://myclass.peelschools.org/sec/12/4268/Resources/RACI_R_Web3_1.pdf.

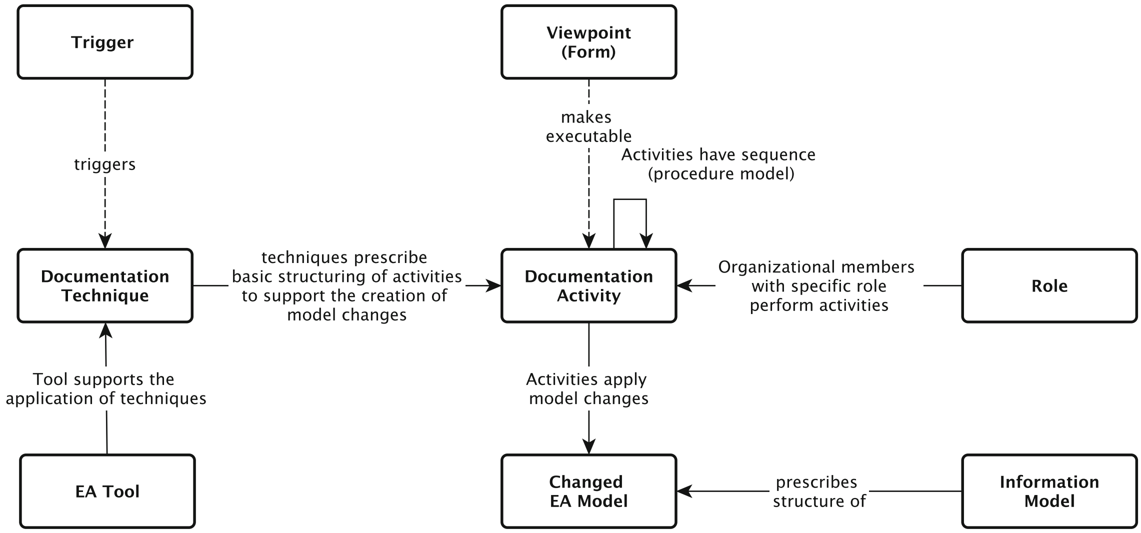

Fig. 4 Components of a situative approach to EA documentation on the basis of the work of Gutzwiller [27]

1. *Related work* We based the development of the techniques on several related works. For example, both Fischer et al. [23], as well as Moser et al. [42] introduce (semi-automated) processes for EA documentation. Other authors work toward semantic tool integration such as Chen et al. [51] or automation from specific data sources, such as Holm et al. [35]. Each of the propose processes, however, only covers a specific aspect of possible documentation. Also integration between these aspects is missing. We gave a thorough discussion of the related works in Sect. 2.2.
2. *Empirical analysis* Further, we executed several empirical analyses that aimed at eliciting requirements and state of the art in EA documentation [19,20,44]. These gave us insights about the data sources and processes that are executed in the organizations to gather data, but also allowed us to identify documentation problems.
3. *Experience as EA consultants* As we are working as consultants in the field of EAM we are exposed to the typical problems in EA documentation. This experience further aided us in developing the techniques.

Note that we do not see this set of techniques/patterns as fixed. The list can and should be refined and extended with additional techniques based on further input from research and practice. This feedback loop of enhancing methods is explicitly mentioned in the SME literature, for example by Brinkkemper [8] who proposes a so-called *method base* to store, retrieve and improve practice proven methods.

Each technique is discussed in detail in the following section including its purpose, process, roles and the context in which it makes sense to apply the technique. Also, we describe the configuration options for each technique to adapt it to the organizations' context. The process fragments described in the following are visualized as BPMN 2.0 diagrams.[4]

4.4.1 T0*: periodic quality assurance and data collection*

The most basic and likely still most important EA model maintenance technique is the manual collection and quality assurance of the EA model. However, this basic technique can still be supported by appropriate tooling. Current tools fail in aiding this simple task. Literature from research [23] and practice [28] suggest the usage of, so-called data delivery contracts from data owners. These contracts require that the data owners contribute the parts of the EA-relevant data they are responsible for, to the EA model on a regular basis. Without tool support, however, the check for adherence to the contracts is a difficult task with much communication overhead.

The basic idea of this technique is to formalize and automate the process of recurring manual data delivery on the basis of a contract. To setup this technique, a data owner is made responsible for a certain part of the model. This definition of a model part is achieved via, so-called, area-of-responsibility (AoR) definitions, which are further explained in Sect. 5. For now, it suffices to explain that AoRs allow for grouping certain parts of an EA model based on types, instance IDs, data origin and other constraints. An example is the AoR of all information systems deployed in a specific data center in Munich. After an AoR has been setup, it can

[4] In BPMN 2.0 cogwheels denote automated tasks, person icons denote human tasks, hand icons denote offline tasks, three parallel lines in a task denote multiple instances, a plus sign in a task denotes a sub-process contained in a task, arrows starting with a diamond denote conditional flows and arrows starting with a short crossing line denote the default flow.

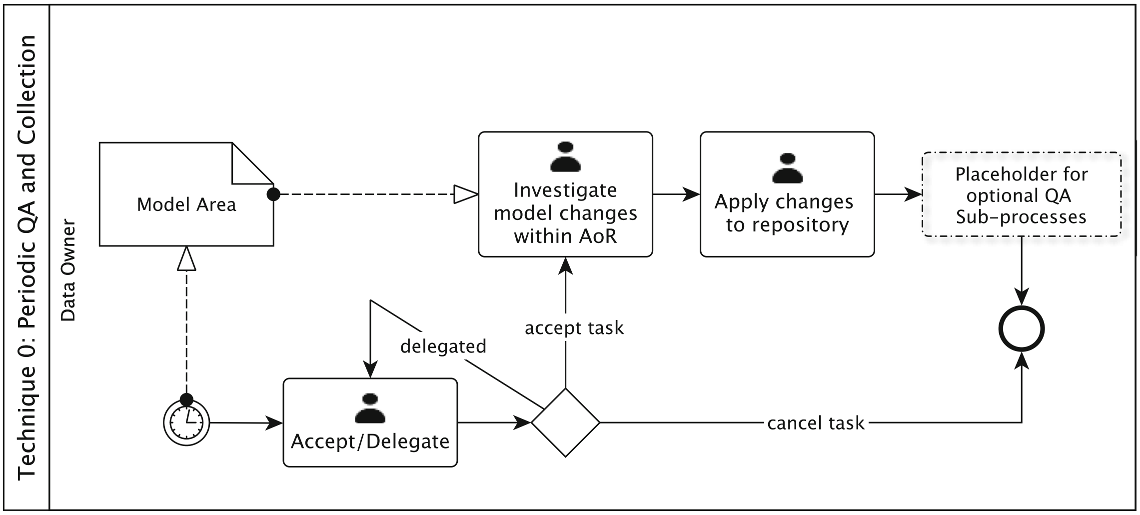

Fig. 5 Process fragment for technique 0 consisting of a setup process and the actual execution path

Table 2 Summary of the responsibilities for documentation techniques T0–T3 and the method assembly process (R: responsible, A: accountable, C: consult, I: inform)

	Documentation mgr.	Method engineer	Data/event source mgr. (tech.)	Event responsible	Data owner	EA stakeholder
T0 setup	R				C	C
T0 periodic QA					R	I
T1 setup	I		R			C
T1 data mapping setup	C		R		R	C
T1 automated data coll.					R	I
T2 setup	R			C	C	C
T2 event handling				R	C	I
T3 setup	R				C	C
T3 internal events	C				R	I
Method assembly	C	R	C	C	C	C

be decided at which frequency data should be delivered by the data owner. A supporting tool can then remind the data owner at selected intervals and list the parts of the model that should be checked for changes. In the following, we list the core attributes of this model maintenance technique.

Purpose This technique can be used to cover the maintenance of a complete organization-specific information model or parts of it. It resembles “traditional” data collection processes in organizations but embeds them into a repeatable and traceable semi-automated process structure. In addition, periodic checks can be applied to achieve regular QA intervals that help to increase the overall model quality.

Situation/context This most basic technique should be applied independently of other context factors since it allows for the general structured quality assurance of the complete EA model. It also incurs only little setup time, i.e., the assignment of responsibilities for model areas.

Process fragment Figure 5 shows the process for this technique. When a maintenance interval has passed, a maintenance event is triggered which also notifies the respective data owner. The tooling collects the model elements that are part of the area-of-responsibility and presents them to the data owner in a task for checking and updating. The data owner can either accept the task or delegate it to another user. In the delegation case, the process starts over for the new user. If accepted, the data owner is responsible for investigating and applying any changes. The investigation can be the simple lookup in existing documentation or even the setup and execution of extensive stakeholder interviews. Finally, the process foresees an optional placeholder for quality assurance sub-processes.

Roles Table 2 shows the involved roles in this documentation technique. The setup of its process is performed by the documentation manager which consults the EA stakeholders to elicit EA information demand and data owners that can

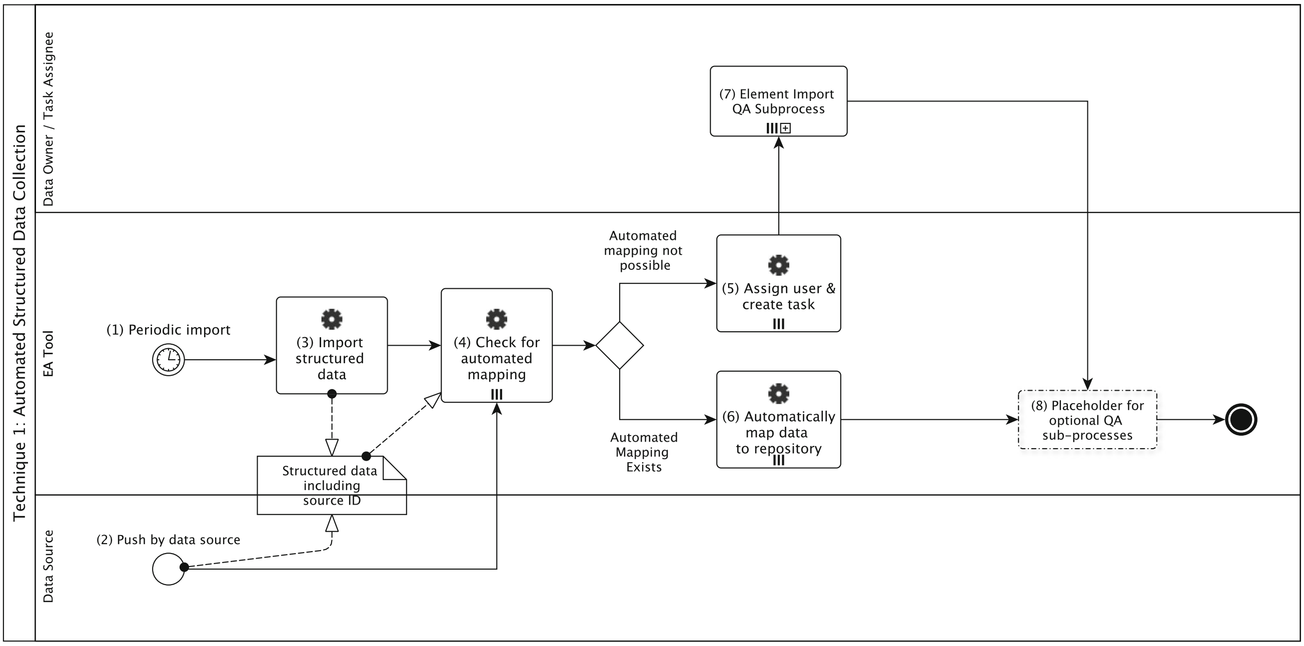

Fig. 6 Process fragment for automatically collecting structured data from external data sources

potentially provide relevant data. During the process execution, only the data owners are involved. If requested, EA stakeholders can be informed about any changes.

Configuration To configure this technique-specific model areas need to be setup, to which roles and users are assigned. For example, one user could be made responsible for the quality of the model elements that represent the physical hardware in a specific data center. In addition, the intervals of process execution need to be set depending on the expected change frequency of specific model areas.

4.4.2 T1*: automated structured data collection*

This technique has the highest potential of reducing manual data collection efforts. However, it might not be applicable in all organizations due to unavailable data sources or data sources providing information in inferior quality. The basic concept is the automated collection of data from data sources that can deliver structured EA-relevant data, such as CMDBs, network scanners or Enterprise Service Buses (ESBs). We present and discuss a list of potential data source in [20]. However, as we showed in our survey [20], the effectiveness of such an approach highly depends on the available data sources and their associated data quality. In our previous work, we elaborated the requirements for such semi-automated data collection [19], sketched according processes [18] and presented a meta-model that supports the processes with context information as discussed in Sect. 2.

Purpose The purpose of this technique is the reuse of external structured data sources into the EA model in order to reduce or even eliminate the manual data collection effort for specific model elements in the repository.

Situation/context This technique should be applied when data sources exist or can be created with justifiable effort, that are able to supply relevant EA data which can be mapped to the organization-specific information model. As will be described in the method assembly process description of Sect. 4.6, a weighing of the effort between data source integration and continuous manual integration has to be conducted in order to justify the initial setup costs.

Process fragment Figure 6 shows the process fragment for semi-automatically integrating structured data into the EA model. The data import can be initiated by the providing data source (push) or by the EA tool as a periodic import (pull). Hence, there exist two different start nodes, a periodic import event on the EA tool side (1) and a simple start node in the data source role lane (2). In both cases, structured data are imported into the EA tool (3), which first checks whether parts of the data can actually be automatically imported into the EA model without manual intervention. For each collected element, this step can be parallelized (4), as indicated by the three vertical parallel lines in the action. If an automated mapping is possible for an element, it can be directly written to the model (6) when no further QA sub-process is included (8). Else, the process identifies an appropriate user to handle the manual processing of the imported model element (5). This manual step (7) is further decomposed into

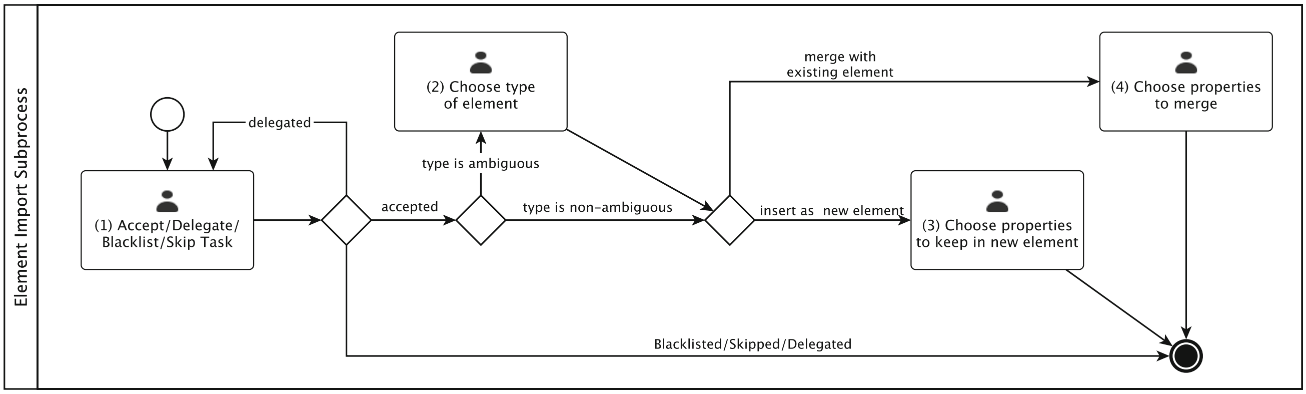

Fig. 7 Manual intervention process for quality assurance of automatically collected structured data

a sub-process which can be seen in Fig. 7. After this manual processing, an optional QA sub-process can be added to further validate the quality of the newly added or changed model elements.

In the manual process of Fig. 7, the assigned user first has the option to accept, delegate blacklist or skip the processing of the collected model element (1). If the user accepts the task, the system first has to decide if the type of the to be created model element is non-ambiguous. If the type is unclear, the user first has to decide which model element type to instantiate (2). After that she can either decide to create a new model element and then choose the properties to keep from the input and manually add additional information (3), or choose an existing model element to merge the data with (4). The selection of appropriate existing elements in step 4 can technically be aided by concepts like *identity reconciliation* that we detail in [21].

Roles Table 2 shows the responsibility matrix for this documentation technique. For setting up an instance of this technique, the *Data Source Manager* is the responsible role. This person has to create the technical infrastructure for the integration.

However, also the data owner, i.e., the person with the domain knowledge on the provided data, is responsible for creating the mapping between incoming data. This is denoted by the responsibilities in the fourth line of Table 2.

For the actual semi-automated data collection process, the respective data owners which are needed to execute the quality assurance are responsible. After changes have been applied, EA stakeholders can potentially be informed.

Configuration To setup this process, the selected data sources need to provide interfaces which can be accessed by the EA repository and read in an appropriate format. A mapping has to be defined that transforms the external data format into the EA repository's format and its organization-specific information model. Also the interval of data integration needs to be set for each data source, and responsibles for technical problems as well as manual intervention have to be set.

4.4.3 T2*: external information system events*

In [20], we showed that structured data sources, as of *technique 1* can only cover parts of the organization-specific EA models, i.e., mostly technical layers can automatically be collected. However, in the same research, we realized that some information systems, while not capable of providing relevant structured EA data, are relevant sources for EA change events. A typical example is the *architecture change project end event* from a project portfolio management tool, as encouraged in [41]. Such events can be utilized to initiate EA model maintenance processes at the right time and with additional context information to process the event as we outlined in [22].

Purpose Utilizing external events to drive documentation processes is useful to catch changes in the EA at the right time that cannot be covered by structured data sources. In addition, they can help in triggering maintenance cycles when changes occur similar to *Reminders (T0)*. The technique further helps to align the documentation processes with parallel processes in the organization.

Situation/context This technique should be applied when there exist event sources in an organization that can be used to produce events on architecture-relevant changes. In [22] we list possible sources and event types.

Process fragment Figure 8 shows the event handling process that can be applied to handle EA change events from external event sources. Once an EA change event (notification) is received by the EA tool, the process first checks whether the specific event instance is blacklisted. This system then assigns an appropriate user to the task (1). The user selection is based on the area-of-responsibility as well as the event's

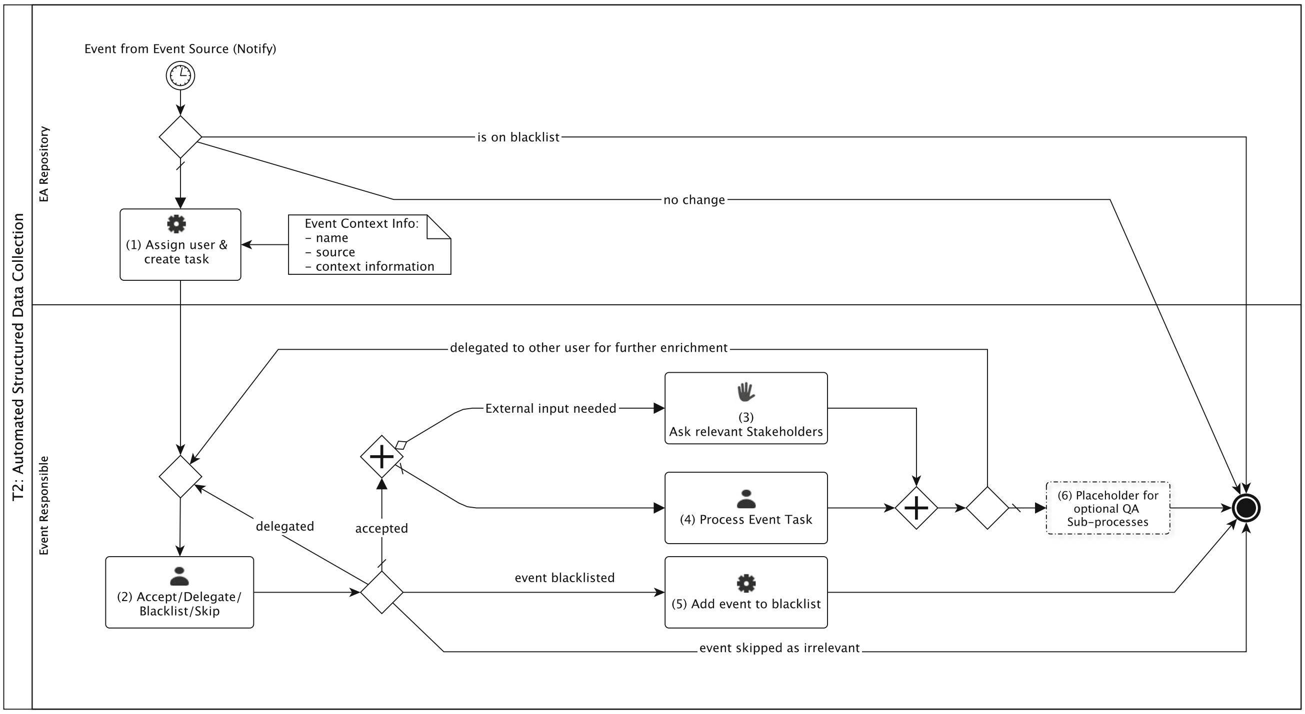

Fig. 8 The event handling process of *technique 2* in BPMN notation

context information. An assigned user can then accept, delegate, blacklist or skip this task (2). When the user accepts the task, she is presented with the event's context information and is then responsible for acquiring the necessary information online or offline (3) in order to apply the changes to the EA model (4). Opposed to blacklisting (5), skipping means that the event type is not permanently blocked, but just skipped for this one event occurrence. Also, after editing the user can delegate the task to another user for further editing. Finally, an optional QA sub-process can be added as well.

Roles In Table 2, we show the required roles to setup and perform this process. For the setup, the *Documentation Manager* is responsible for identifying the *Event Responsibles*, as well as consulting the *Data Owners*, which are the persons in charge of the systems that fire events. Also *EA stakeholders* should be consulted in order to find out which events could be important. For the event handling itself, the assigned *Event Responsibles* are in charge, data owners are consulted in specific cases and EA stakeholders are optionally informed about changes.

Configuration For each event source, the types of events need to be defined and in particular which roles or users are responsible for specific types of events. In addition, the interval at which events are pulled from their sources or when events are pushed to the repository from their sources needs to be set.

4.4.4 T3*: internal model events*

Different EA-relevant artifacts in an organization have differing life spans as discussed by Aier et al. in [2]. In their work, the authors elaborate on the survival time of information systems; however, other artifacts contained in EA models also have limited life spans such as virtual or physical hardware. Being aware of the general expiration of model elements helps in realizing that these lifespans can be utilized by an EA tool to fire model element expiry events. For example, in an exemplary organization, it could be the case that new information system releases are generally rolled out in a four month cycle. This information can be used to adjust the expiry duration of the model element type *Information System*. We initially described this idea of model element expiry in [21]. An actual expiry event should be fired when the duration since a model element has been manually changed or has been explicitly reviewed is longer than the expiry duration that was set for the respective model element type. Besides these expiry events, other event types can be thought of such as semantic consistency checks. An example is the check whether two information systems communicate and one information system is already tagged as retired. In Sect. 5, we describe the necessary meta-model concepts to implement and trigger such checks in the form of constraints.

Purpose The technique is used to remind appropriate stakeholders to check the actuality and general validity of specific model elements after appropriate time spans or when a

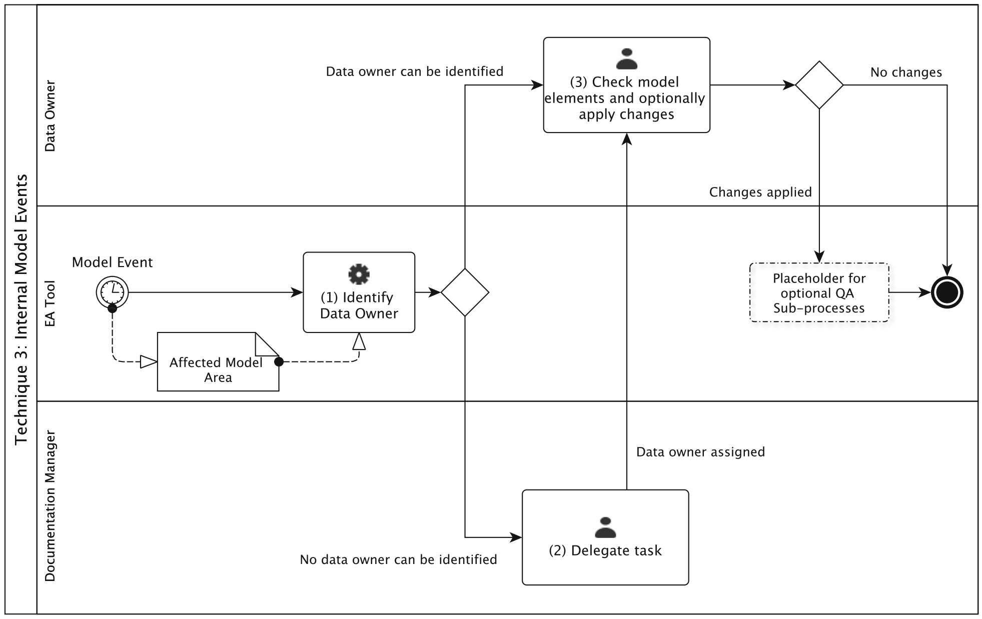

Fig. 9 Process fragment for the integration of internal model events into the documentation process-*Internal Model Events (T3)*

constraint is violated. This helps to increase the overall quality and actuality of the EA model and is particularly useful since it can potentially cover all EA layers.

Situation when to apply This method should be applied when no event or structured data sources exists for a specific type of model element. For example, this is often the case for upper-layer model elements, such as high-level business processes or strategy elements. However, the technique can be applied for elements of all layers.

Process fragment Figure 9 shows the workflow that is executed when a model event is fired from an EA repository. First, the tool has to identify the appropriate data owner that is responsible for the affected model element or model area (1). If none can be calculated, the task of identifying the appropriate user is escalated to the EA documentation manager (2). Then the assigned data owner receives the manual task to check the quality and actuality of the respective model elements (3). If changes are applied they might go through an optional quality assurance processes. If nothing was changed, the process terminates.

Roles To setup the internal model event process, the documentation manager coordinates the discussion with data owners and EA stakeholders. This leads to a list of model element types for which expiry events should be fired and an expiry duration for each of the types. During the actual process execution, the data owners are the key performers. However, if no data owner can be identified for an expiry element, the documentation manager is consulted. In addition, EA stakeholders are potentially informed if changes have been applied.

Configuration To configure this technique, the types of internal events need to be modeled. As explained above, these can be model element expiry events or constraint-based checks that are executed in a specific interval. Hence, for expiry events, the expiry time of each model element type needs to be set, and for the constraint-based events, the constraint needs to be defined. Also the responsibles for the events or corresponding model areas need to be set.

4.5 Quality assurance sub-processes

As can be seen in the process diagrams, each of the aforementioned techniques can be optionally equipped with sub-processes for quality assurance. These are executed when changes are applied to the model, and it is required that a very high model quality is achieved. Moser et al. [42] give an example for a quality assurance release process that can be used to check and enforce organization-specific modeling guidelines before any changes are committed to the EA repository. In the work of Fischer et al. [23], a more complex process for vetoing EA model changes is described. Due to space limitations, we do not further elaborate on these possible QA processes. Please refer to the abovementioned related work.

4.6 Method assembly

In the previous section, we introduced four techniques that can be assembled to make up a situated EA documentation

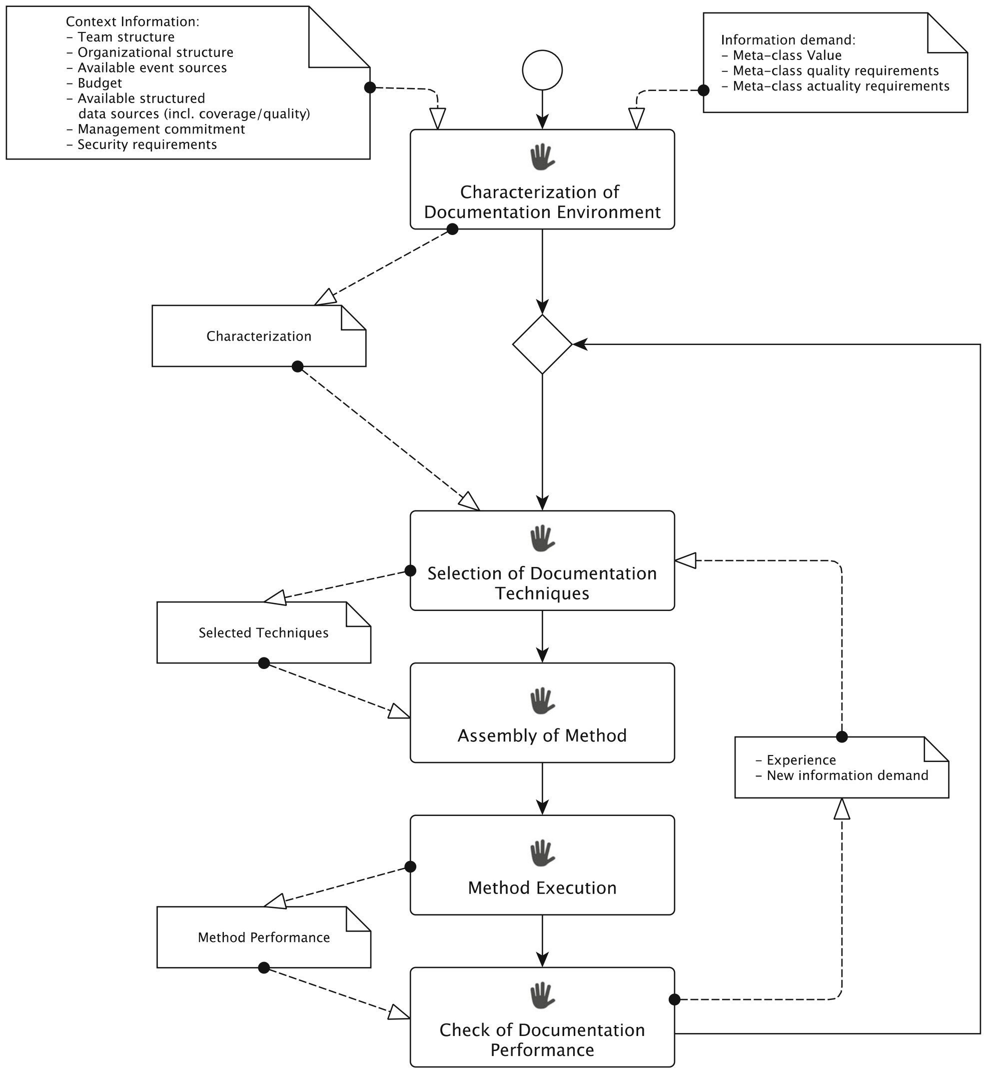

Fig. 10 Organization-specific process for the selection of appropriate EA documentation techniques (on the basis of Fig 1. in [8]). All tasks except the execution are executed by the method engineer

method. However, it is important that the use of each technique is applied to satisfy a specific information demand and is taking respective context factors into account. A careful construction of the documentation method in an organized fashion allows for "controlled flexibility" of the used method according to Harmsen [31].

Figure 10 shows the general process for the selection and assembly of techniques to form a coherent documentation method.[5] In the first step, the documentation context of the organization needs to be assessed. This context consists of the organizational structure, such as EA team structure, information providers as well as available data and event sources. In addition, the information demand for each type of model element needs to be assessed as well as the corresponding requirements with respect to the quality and actuality of instances. Based on this context data, the documentation techniques can be selected and configured to the documentation needs of the organization as described in the following section. Consecutively, process fragments are adapted to these needs, roles are assigned and data as well as event sources are setup or configured. While the documentation process is running, its performance, i.e., the quality of the EA model in relation to the effort of maintaining this quality, is continuously checked. This information is then used to redesign the documentation method if needed.

4.6.1 Method ranking and selection

The aim of the method assembly process is to produce a set of documents that can be used to facilitate the decisions on

[5] To increase the readability, we have omitted the role lanes in this figure. All tasks are performed by the method engineer except the method execution.

Step 1 – List Information Model Quality Requirements

Model Element	Actuality Req.	Completeness Req.	Importance
Information System	2 Weeks	Only information systems needed of main branch in Munich	4/5
...			

Step 2 – List and Analyze Available Data Sources

Data Source	Structured Data	Data Quality	Completeness	Constraints	Responsible
Network Scanner Data Center Munich	• Information Systems • Hardware	• Information Systems: MEDIUM • Hardware: HIGH	• Information Systems: 60% • Hardware: 80%	Security clearance?	Karl ...
...					

Step 3 – List and Analyze Available Event Sources

Event Source	Events on	Event Quality	Constraints	Responsible
Global PPM Tool	Information Systems, ...	Medium	Possible Firewall Issues	Peter ...
License Mgmt. Tool	Information Systems, ...	High	No constraints	Joe ...
...				

Step 4 & 5 – Merge Lists with Techniques and Rank

Technique	Provides	Source	Quality	Cost of Impl.	Ranking
T1 – Automated from structured Data Source	Information System	Network Scanner	60% of information systems in Munich	Detailed costs here	1
...					

Fig. 11 Production steps of documents/spreadsheets for the method assembly

which data and event sources to integrate and which other documentation techniques to apply. In the following, we describe this sequence of analysis steps that are applied to create the necessary documents. Figure 11 exemplarily shows documents that need be created and the steps to be executed to support the selection and ranking. As pointed out in the evaluation section (cf. Sect. 6), we successfully applied this ranking method in practice.

1. *Preconditions* First the preconditions need to be met in order to evaluate whether the situated approach for EA documentation is applicable. The preconditions are in particular the existence of an EA information model that only covers the actual information demand of EA stakeholders and the availability of an appropriate EA tool that can guide the documentation processes. Also the tool needs to be able to define appropriate data collection roles.
2. *Step 1: Listing of information model quality requirements* In this step, the EA responsibles, and in particular, the EA stakeholders assess the quality requirements for each model element type, i.e., potential product fragments, in the EA information model. This, for example, results in a spreadsheet that can be seen in Step 1 of Fig. 11. It includes the requirements for actuality (i.e., after what time does the validity of an instance expire, see Sect. 4.4.4), as well as the requirements for completeness of documentation of all elements of this type.
3. *Steps 2 and 3: Listing of data and event sources* Next, lists of data providers (data owners) and event- as well as structured data sources need to be created. This list should contain:
   – Type of the provided data from data owner or data/event source.
   – The quality of the provided data with respect to actuality, correctness and completeness.
   – Granularity of provided data with respect to the abstraction gap of the data to the abstraction level of corresponding model element types of the organization-specific EA information model.
   – Constraints regarding security and privacy issues on the supplied data.
   – Information about possible runtime performance degradation of data sources.
4. *Step 4: Merge in documentation techniques* In the next step, the items of the previously compiled lists of information suppliers and sources are integrated with possible techniques to gather information from them. For example, *T0* for manual collection from data owners, *T1* for structured data sources etc.. For each EA model element type, the document contains the information which data owners can potentially provide the data, which automated source can potentially supply structured data on instances of the type (product fragments) and which event sources potentially provide change events for this type. Also the cost for the implementation is approximated in this list.
5. *Step 5: Final ranking and decision* The previously merged list then needs to be ranked. The general approach should be to choose automated sources for model element

types of high importance, low implementation cost and low security risk where the completeness is additionally high. In cases where the level of abstraction of a data source is very low, it might be advisable not to use the specific data source, since the effort of manually raising the level of abstraction might outweigh the benefits of automation. Of course in each method instantiation, several different instantiations of each technique can be applied. For example, if several automated data sources are included, several specifically configured process of *T1* need to be deployed.
We refrain here from giving a concrete ranking formula since we consider the ranking decisions as specific to an organization and a creative process.

4.6.2 Final method assembly

Once it is decided which techniques to apply and which data and event sources should be used, the overall method needs to be assembled in the form of executable processes. This entails the deployment of the processes of each applied technique in a tool such as the process-based EA-repository prototype we describe in the evaluation section below (cf. Sect. 6). For each technique, the executing roles need to be assigned according to the role descriptions presented above. In addition, the processes need to be adapted according to the configuration options that we listed with each technique. For example, for the automated integration of structured data according to *Automated Collection (T1)* one needs to set up the mapping of the data structure from the source to the repository format and semantics as well as the interval in which updates are checked.

Note that the process fragments are not weaved together to form one large process, but are deployed separately. However, they can trigger each other's execution, e.g., via model change events (cf. Change Triggers in Sect. 5).

4.6.3 Method assembly roles

Table 2 shows the RACI matrix for the documentation method assembly process in the last line. The method engineer is responsible for coordinating and collecting the context information as well as assessing the information demands of each stakeholder. This is achieved by consulting all other documentation stakeholder roles in the organization. The method engineer, in essence, brings together all EAM interest groups and negotiates the importance of each information demand in relation to the effort of collecting the data in an automated fashion. Finally, the method engineer also oversees the final configuration of the techniques once they have been selected.

In this section, we have introduced four configurable techniques for semi-automated EA documentation. Since these techniques need an information model to be applied to, as well as extensive meta-data to be executed, we present a supporting meta-model in the following section.

5 Supporting meta-model

As noted by Buckl et al. [14], the practice of EAM consists of the dichotomy between the method of EAM execution as well as the language to describe the EA. In the previous section, we introduced an approach to create the documentation part of the EAM function to elicit the as-is EA model. In order to provide the basis for tool support that can create and maintain the language part of the EAM function, we developed a meta-model that we initially presented in [21]. It allows to define organization-specific EA information models, responsibilities, event and data sources, model element identity conditions, blacklists as well as model element expiry intervals. The possibility to access meta-data constitutes the context information that drives the manual, semi-automated or automated data collection process in practice. Some of the questions an instance of the meta-model (i.e., the organization-specific information model) is able to answer are:

1. **What** is the information demand of the organization?
2. **When** does the EA model have to be updated from a data source?
3. **Where** does integrated data originate from (manual entry or automated update)?
4. **Which** part of the EA model should be updated?
5. **How** is the quality of the resulting EA model governed, e.g., how are duplicate data entries avoided?
6. **Who** is responsible for a specific manual tasks?

Figure 12 shows the meta-model. It conforms to the object-oriented meta-modeling conventions of the MOF [43] and can be divided into five parts. One part that provides the means for information model design (middle-left), a part for process management and responsibility assignment (top), two parts for external and internal events (center and middle-right) and a part for model element identification (bottom).

In practice, i.e., with an implementing EA tool, different parts of the meta-model would be instantiated with different means and at different times. For example, the information model could be modeled with a graphical notation provided by the tool that follows a similar syntax as UML class diagrams. This would typically be done at the initialization phase of the EA project or when the information model needs to be adapted. The assignment of roles to tasks, on the other hand, can be more efficiently done in a form-based manner in a tool and is also a purely runtime activity. The same applies, for example, for the definition of events or identity conditions in a tool.

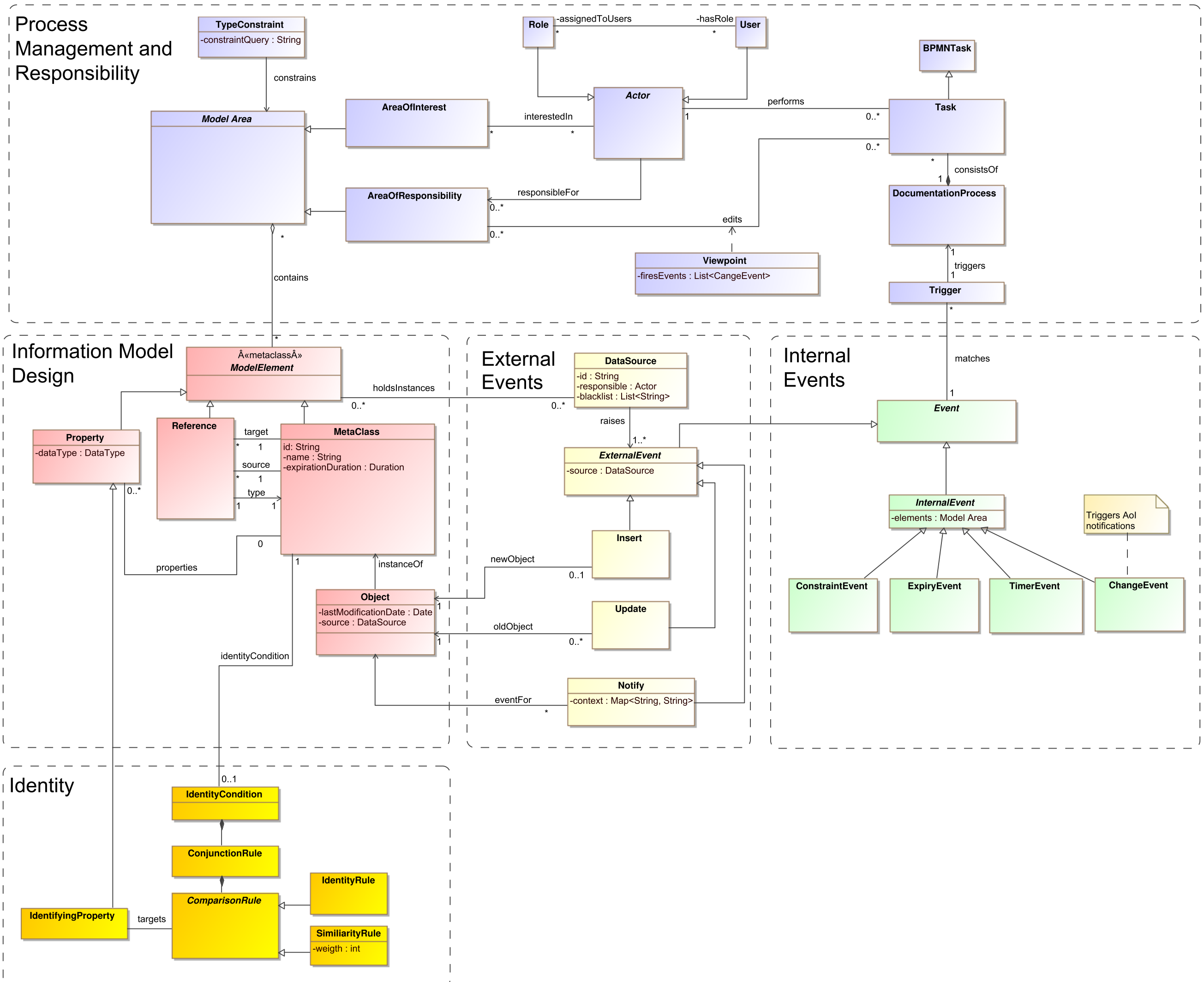

Fig. 12 The meta-model to create organization-specific information models and to hold meta-data that drives the data collection processes

In the following, we describe the different parts of the meta-model in detail.

5.1 Information model design mechanism (meta-modeling)

The information model design mechanism of the meta-model resembles the mechanism of the MOF specification (middle-left). The mechanism allows for the design of organization-specific information models[6] that can be realized by instances corresponding to the information demand of an organization.

The main concept from which elements of this part inherit is *Meta-Element*. The inheriting elements are *Class*, *Reference* and *Property*. As is typical for meta-modeling mechanisms, a class can have zero or more properties and references to other classes. Additionally, each instance of a class, i.e., an information model element type, has a name and unique id. Instances of the types are realized by the *Object* model elements via the *instanceOf* relationship. An implementing tool would, for example, enable the instantiation of an information model via a tree-based editor or as mentioned above in a graphical manner.

5.2 Process management and responsibility

The top part of the meta-model is responsible for modeling the instantiation of the documentation techniques presented in Sect. 4. On the right side, one can see the concept of a *DocumentationProcess* that consists of *Tasks*. A task covers the processing of an *AreaOfResponsiblity*. Such an area is a specialization of the abstract concept *ModelArea* which is used to group model elements. The processing of the model elements contained in an *AreaOfResponsiblity* in the realm

[6] Information models are organization-specific meta-models according to the terminology used in [10].

of a *Task* is done via a *Viewpoint* realization. Such a viewpoint is a specialized view in an EA tool to handle model changes. This could, for example, be a view that is specialized for processing external change events. *Actors* are responsible for executing the tasks that are attached to a specific *AreaOfResponsiblity*. These actors can be defined via a direct *User* assignment or *Role* assignments. Actors are also connected to the subclass of *ModelArea* called *AreaOfInterest*. These specialized model element collections are used to inform registered actors upon changes on the contained model elements.

An implementing tool would likely allow the modeling of the processes and according role assignment via the selection of pre-built processes with specific configuration points such as optional quality gates. The pre-defined viewpoints could then just be selected for manual tasks.

5.3 External events

External events (center) are fired from *DataSources*. These have a unique id, a responsible actor as well as a list of external ids that serves as a blacklist for unwanted events. A data source holds instances of specific model elements. Such instances of the same type can be delivered from different data sources. A data source can raise external events of three different types, *Insert*, *Update* and *Notify*.

Inserts are events that indicate a newly detected model element for which it is unclear whether it maps to any existing model element in the repository. Methods for identity reconciliation and information fusion can help to identify and merge *Inserts* into corresponding existing elements. We further discuss these methods in Sect. 5.5.

Updates on the other hand are external events where it is known which model elements they concern. *Updates* and *Inserts* are the main meta-model elements to support *Automated Collection (T1)*.

Notify events are triggered from information systems that cannot deliver structured data on specific model elements, but can notify adequate stakeholders when changes happen. The typical example here is the "architecture project end event" that is fired from a project portfolio management tool when a project has reached its deadline. These events can contain context information in the form of key value pairs that can be used by the corresponding viewpoint to display some context information about the event.

All three external event types are related to *Objects* which have a last modification date property. This property can be used for calculating the expiry date of model elements in combination with the expiry duration of its corresponding class as explained in the next section.

In an implementing tool, the events would be typically configured when setting up the configuration for the data sources. For example, one would set which data sources potentially provide which types of model elements and who is responsible for processing such events.

5.4 Internal events

Internal events (middle-right) are raised by the implementing EA tool itself and contain a set of affected model elements as a *ModelArea*. *ConstraintViolationEvents* are used to trigger manual quality assurance processes when certain constraints of the overall EA model are violated. For example, an information system still has a property stating that it is running in production mode; however, its planned retirement date has already passed. Such an event can then request a responsible actor to check this constraint violation and apply necessary changes to the model. As already discussed, expiry events remind actors to check the quality of specific model elements if the element was not changed or explicitly checked for a certain amount of time (*Internal Model Events (T3)*). *Timer Events*, on the other hand, are events that are fired in regular intervals in order to realize *Reminders (T0)* by triggering regular maintenance processes. *ChangeEvents* are fired, as the name implies, when changes to model elements occur. These events can then be used by an implementing tool to notify interested actors upon changes to specific model elements.

The configuration of such internal events in an implementing tool would typically be done in a configuration interface where the timers and the responsible stakeholders are set. In the current version of the tool complex constraint, events are not supported; however, technically, there seems to be only little limitation in using a query language for expressing such constraints that are executed on a regular basis.

5.5 Model element identity management

In order to avoid duplicate model elements originating from different data sources as well as manual entry, the meta-model provides mechanisms for identity reconciliation and data fusion (bottom). The first mechanism is responsible for detecting duplicates, and the second is responsible for merging duplicate elements. These mechanisms are important to reduce the chances of duplicate data that could specifically be introduced when *Automated Collection (T1)* is applied. The elements on the left side of Fig. 12 represent the meta-model elements that support these mechanisms.

An *IdentityCondition* can be attached to classes in order to set the condition that approximates whether two model elements actually refer to the same physical thing. An *IdentityCondition* consists of several *ConjunctionRules* that in turn consist of *ComparisonRules*. A comparison rule always targets an *IdentifyingProperty*, which is a subclass of a normal property. It simply indicates that this property can be used to apply comparison rules to it. A comparison rule can either be an *IdentityRule* or a *SimilarityRule*. The former is an overriding rule that, if it evaluates to true, indicates the identity of two model elements. Such a rule can be used to identify two model elements as the same physical entity when their IDs and their names are equal.

A *SimilarityRule* is used when no identity rule holds. It can be used to compare the properties of model elements and their string similarity. These can then calculate a similarity score as the basis for a human decision on merging two model elements. This form of merging is also often called information fusion. We explain these concepts in more detail in [21].

An implementing tool would allow the modeling of this part of the meta-model via configuration forms for each information model element type.

So far we have introduced our situational method for EA documentation and a supporting meta-model. In practice, this meta-model will not be instantiated with one single tool. Rather, several tools or views within one tool are used to compose a "virtual" instance that is a composition of process models, the created information model, as well as trigger and identity management configurations. These separate parts reference each other in an implementation. For example, a trigger configuration in an implementing tool, just knows the ID of the process to initiate when the trigger is fired. Hence, there is a loose coupling between the different parts of the overall model.

We continue with evaluating our approach both practically by introducing our prototypical tool implementation and its application in practice, as well theoretically by comparing our approach to the challenges identified in Sect. 3.

6 Evaluation and related work

In this section, we evaluate the presented techniques as well as the method assembly process from multiple perspectives. First we compare the approach with the documentation challenges identified in Sect. 3. Thereby we establish *relevance* and *utility* of our work. We then demonstrate the viability of the proposed concepts by presenting our prototypical EA repository implementation that enables the flexible deployment and execution of the configured EA documentation processes. This shows the *implementability* of our work. We then present a case study we executed at a German insurance provider where we applied our method for the construction of an organization-specific documentation processes. This shows the *purposefulness* of our approach in a practical example.

6.1 Prototypical tool implementation

To show the full implementability of the methods created with our situational approach, we developed a web-based EA-repository prototype. It is based on a workflow engine that is used to deploy organization-specific data collection processes with manual as well as automated tasks. The semi-automated processes consist of manual tasks via task forms and automated data collection as well as quality assurance services. Process definitions can be added or updated at runtime to add new data collection functionality or change existing process parts according to the method adaption step that is part of the assembly process (cf. Sect. 4.6).

To show its capabilities, we first outline the most important features of the tool and then briefly discuss implementation details.[7]

- *Customizable information model* As mentioned in Sect. 4.2, each organization has its specific information demand that is reflected by the underlying organization-specific information model. The tool can work on arbitrary information models enriched with relevant information for the data collection process as described in Sect. 5.
- *Customizable data collection processes* A core feature is the ability to adapt the processes that schedule the time when data are collected, determine who does the manual processing and state what data sources are tapped. This is realized by processes that can be deployed on a BPMN engine.
- *Plugin architecture for collection of structured data* In order to integrate structured data sources such as a network scanner, the tool provides a plugin and mapping mechanism. It allows adding new data sources via separately implemented connectors. With the data mapping mechanism, one can define mappings from the original data source's data format to the repository's internal data representation. Figure 13 shows a screenshot of the data mapper. The two columns in the center can be used to specify the data mapping from the source data format on the left, to the repository internal information model on the right via drag-and-drop. In the top right, one can see the notification area, where users are notified on assigned and unassigned tasks, as well as currently running data collection processes.
- *Event interface* Further the tool provides a REST interface that can receive arbitrary events with context information to trigger architecture change event processes according to *Internal Model Events (T3)*.
- *Triggered process execution* Different triggers can start process instances at desired points in time. These can be simple timers and repeated queries of quality attributes such as element expiry. They can also be based on external events or change triggers. Currently, the check-

[7] Note that the implementation focus of the prototype is laid on data collection mechanisms and does not include EA visualization capabilities.

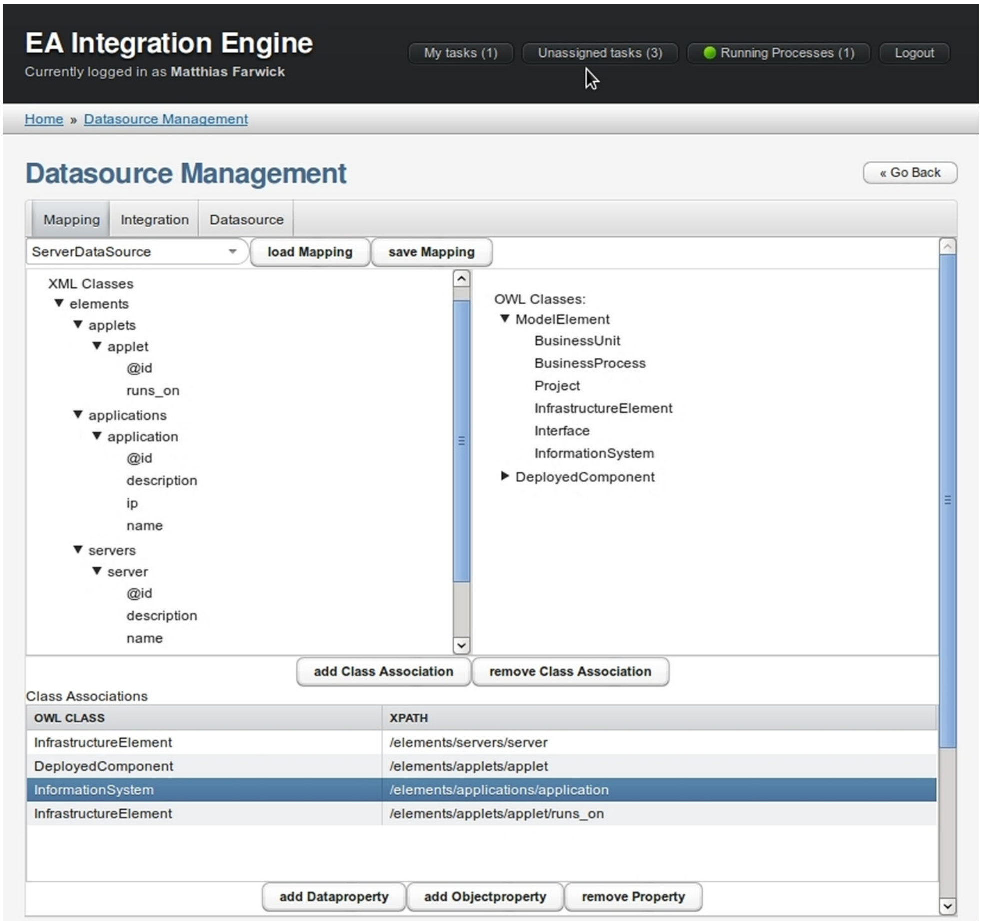

Fig. 13 Screenshot of the data import mapping of the tool, with the source data format on the *left* and the organization-specific information model format on the *right*

ing of complex constraints for internal events is not supported.

- *Blacklisting of events and model elements* Data sources and event sources may repeatedly produce model elements or notification events that are not relevant for the EA. The tool provides a mechanism for blacklisting of such elements. The blacklisting behavior can be edited for each data/event source. This feature is important to reduce the manual effort of filtering unwanted data.
- *Model quality assurance Mechanisms* Several quality assurance mechanisms are implemented to assure the overall quality of the EA model. For example, the *identity reconciliation* mechanism [21] is used to identify duplicates of model elements coming from automated data sources. If automated identification is not possible, the tool calculates similarity scores to assist manual inspection. A *data fusion* mechanism is implemented to combine new external data with existing model elements in the repository.
- *User management and task assignment* The tool allows to manage users and their roles in a fine-grained manner. This is an important feature in order to assign manual data processing tasks to the appropriate stakeholders who have the best knowledge to fulfill a specific task based on their areas of responsibility.

The EA repository is realized as a Java-based Rich Internet Application (RIA). As a modern web-application, its user interface is characterized by a high usability and instant feedback. For example, if a new architecture change notification reaches the system, the appropriate user is selected and he or she will receive an immediately visible notification message if logged in.

The application is implemented using Java and the Spring framework[8] to realize its business logic. Its presentation layer is built with the web-framework Vaadin[9] which allows developing web-based applications completely in Java in a similar style to the development of desktop applications. Any J2EE servlet container which supports the J2EE specification 2.5

8 http://www.springsource.org.

9 http://www.vaadin.org.

and higher can be used to host the application. The information modeling mechanisms are realized with the semantic-web technology RDF/OWL,[10] its persistence layers is implemented with the RDF repository Jena.[11] For the deployment of processes and task management we used the BPMN engine Activiti.[12]

With the implementation of the prototype, we showed the general *implementability* of our approach with current tools. In the next section, we present a case study where experience from the prototypical implementation was used for implementation and our method assembly process was successfully executed.

6.2 Case study

In this section, we outline the application of our situational approach to EA documentation at a German insurance provider to underline its practical relevance. It was executed as part of the master thesis project of a student with our supervision where we also frequently attended the meetings and guided the work. The goal was to elicit automation potential for keeping the EA repository of the organization up to date and to implement according tools and processes.

Context The EA endeavor of the company has top-level management support, where even the automation project was directly supported by the CIO. The documentation is mainly executed by one chief enterprise architect; however, it is targeted to federate the documentation more to the information providers in the course of the project. By attending architecture meetings of various departments and projects, the architect acts as an integrator with overarching architecture knowledge. An EA tool was already in place that supports the import of data via a programmatic interface. The tool, however, lacks the appropriate means to assure the quality of imported data before it is added to the repository as well as means to coordinate the right stakeholders for a given task.

Course of action In initial meetings with the chief enterprise architect, we analyzed the actual EA information demand of the company. This was based on the analysis of the applied EA information model and a discussion on which parts of the model change frequently. The analysis of the information model was relatively simple because the company made use of the EA tool *iteraplan*[13] which consists of a small and static information model. Also, we analyzed for which types of elements actuality is most important. As a result, a description of the quality requirements for *information system* model elements and standardized *technical components* was created as outlined in Sect. 4.6.

Based on this document, appropriate stakeholders in different organizational units were interviewed that have knowledge on potential data sources that can provide change events or structured data. The interviews resulted in a list of potential data sources. Initially, it consisted of a CMDB, a network scanner, regularly updated Excel sheets, a project portfolio management tool, an enterprise service bus (ESB) and server configuration files. All were accounted for in a spreadsheet according to the method assembly process we described in Sect. 4.6. These spreadsheets included their respective provided data, expected data quality and options for automation. Subsequently, a comparison of the information demand and the available data sources lead to the selection of the CMDB as a source for *technical components* and *information systems*, the ESB also as a source for *information systems* and the server configuration as another source for *information system elements*.[14] These were the selected product fragments for the applied techniques.

Since the organization already used an existing EA tool, it would have been difficult to implement *Reminders (T0)* and *Internal Model Events (T3)* in the tool. Hence, it was decided to implement a process-based data accumulation and preprocessing tool on the basis of the prototype presented in the previous section. The tool provides a tree-based editor for modeling information model elements and allows assigning responsibility to model element types. Also for each data source, time triggers can be specified when the respective process fragment should be executed.

Several specifically configured instances of the process fragments of *Automated Collection (T1)* were deployed as BPMN processes for the different data sources. The final processes resemble the process descriptions that can be seen in the Figs. 6 and 7 with the following adaptions:

1. Periodic imports are triggered by time triggers that can be configured for each data source in the web-interface (Step (1) in Fig. 6).
2. In the QA placeholder step (8), manual inspection by the EA manager was added. Here, this role can also decide to push the collected data to other information systems that might be interested in the newly collected data.
3. In the last step, data are pushed to the existing EA tool *iteraplan*.

Figure 14 shows the simplified architecture and data flow of the implemented EA data accumulation tool. Several data mapper implementations pull data from the above described data sources and transform them into the organization-

[10] http://www.w3.org/RDF.

[11] http://jena.apache.org.

[12] http://activiti.org.

[13] http://www.iteraplan.de.

[14] The server configuration file contains the names and routing information of the main web-based information systems of the organization.

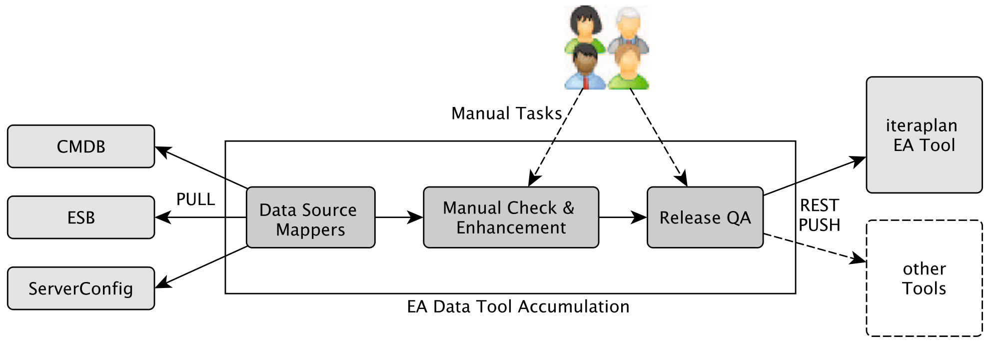

Fig. 14 EA data accumulation tool implemented in case study

specific information model format. In the following manual steps, data owners can enhance and blacklist the imported data. In a final step, the chief enterprise architect has to check the data before it is further pushed to the previously existing EA tool iteraplan.

In the current tool, the integration of events according to *External Events (T2)* from the PPM tool was not implemented in this first phase of the project.

As the manual processing and enriched data is only done by a small group of architects including the chief enterprise architect, no additional quality assurance sub-processes, such as a veto mechanism, were implemented. However, a special emphasis was laid on the black listing mechanism, since the CMDB includes many elements that are not of interest at the EA abstraction level. Another important aspect of the tool was the ability to let users decide what type of model element to instantiate from a given input element. This is important because it is often difficult to automatically detect whether an incoming element is actually a technical component or an information system. At the time of writing, the tool is introduced in the organization.

Discussion Our experience from this case showed that our approach covers the important questions to be asked, when automation of EA documentation is desired. Our method assembly process can lead the way to elicit the actual information demand, available data sources, which data sources should be tapped and finally which techniques should be applied. It also shows that by using a data accumulator and preprocessor, existing EA tools can be enhanced with sophisticated automation techniques. However, since the processes are not running for a long enough time period, we cannot yet make any claims about the actual performance of the implemented data collection processes.

6.3 Evaluation against EA challenges

In Table 1 of Sect. 3, we listed a set of challenges for automation from literature [32] and from our own experience. In the following, we discuss if and how our approach addresses these challenges in order to evaluate the appropriateness of our solution for the context of EA documentation and its automation. We marked these challenges with the attributes *ADDRESSED*, *PARTLY ADDRESSED*, *NOT ADDRESSED* and *NOT RELEVANT* according to how well we deem our approach to tackle the specific challenge.

6.3.1 Data challenges

DC 1 *(ADDRESSED)* is concerned with the overload of the data providing information systems (data sources) due to automation. We explicitly consider this problem in the data source selection process in Sect. 4.6. However, we also argue that performance degradation of data sources due to data export will only be a problem in very rare cases.

DC 2 *(ADDRESSED)* considers the problem of selecting the right data sources for automation. With our assembly method and data source selection process discussed in Sect. 4.6, we have addressed this issue. We also successfully applied our data source selection approach in a real-world case, as explained in Sect. 6.2.

DC 3 *(ADDRESSED)* acknowledges the problem of detecting changes in the EA and propagating these to the EA model. To cope with this challenge, our approach uses multiple automated and semi-automated means. These are the polling of structured data sources, the production of events from event sources, as well as regular reminders for manual check and reminders based on model element expiry.

DC 4 *(ADDRESSED)* deals with the bad data quality at the providing data sources. We approach this problem from two angles. First with our data source selection method, we avoid integrating sources that do not provide data in appropriate quality and second we cope with sources providing some data of bad quality with the blacklisting that helps to avoid noise and manual quality assurance steps.

DC 5 *(ADDRESSED)* identifies the challenge of nonexistent automated structured data sources for model elements in the upper EA layers. We cope with this challenge by not only including structured data sources as a documentation technique but also event sources. In addition, the different triggers for manual collection, such as expiry events and timer-based events, can help to keep these layers up to date by regularly triggering manual checks.

DC 6 *(PARTLY ADDRESSED)* mentions that detecting removed elements in the EA is a challenge. We do not directly cover this challenge, however, with triggered manual documentation processes such changes in the EA can be covered to some extent. Also, external events can be utilized to make EA modelers aware of removed architectural elements.

6.3.2 Transformation challenges

TC 1 *(ADDRESSED)* states the challenge of the necessary transformation between source and target data models. In our prototypical implementation, we created an intuitive user interface for creating data mapping configurations between data coming from external data sources to the internal data format.

TC 2 *(ADDRESSED)* mentions the challenge of ambiguous concepts imported from a source where it is not clear to which type the data will be mapped in the EA model. We cope with this challenge by including an activity that lets a user manually select the appropriate type during the import process of *Automated Collection (T1)*.

TC 3 *(ADDRESSED)* requires a meta-model that includes context information to support data collection processes. We have introduced such a meta-model in Sect. 5.

TC 4 *(ADDRESSED)* identifies the challenge of duplicated model elements in the repository due to automated data collection. We tackled this challenge by including an identity reconciliation mechanism, both in our meta-model, as well as in the implemented prototype.

TC 5 *(PARTLY ADDRESSED)* recognizes the problem of a too fine-grained level of abstraction of data that can be provided by structured data sources. In Sect. 4.6, we mentioned that in cases of a very low level of abstraction, it might not be advisable to make use of the source. The reason is that the effort of manually raising the granularity level of the imported data might outweigh the benefits of automation. The automated raise of the abstraction remains an open research question as we discuss in Sect. 7.

6.3.3 Business challenges

BC 1 *(PARTLY ADDRESSED)* identifies the problem of security vulnerabilities when importing data from sources such as network scanners. We argue in the method assembly section that security concerns should be taken in to account when choosing EA data sources. Hence, we approach this challenge by explicitly recognizing the problem in the method assembly process.

BC 2 *(ADDRESSED)* is concerned with the problem of little return on investment for automation. We approach this in the method assembly process by explicitly including the cost of automation and comparing it to the model coverage and change frequency of that automation. This way costs and benefits are explicitly taken into consideration before an investment is made.

BC 3 *(ADDRESSED)* identifies the challenge of including data owners in the documentation processes since they have the best knowledge on the data they are responsible for. We tackle this challenge via the area-of-responsibility assignment explained in Sect. 5. Via this feature, tasks can be assigned to the stakeholders that are responsible for specific types of model elements or parts of the model, in order to apply changes or check the quality of a model part.

BC 4 *(ADDRESSED)* identifies the need for assigning responsibility for the handling of data sources and accompanying quality assurance processes. As with *BC3*, we handle this by allowing modeling these relationships as explained in Sect. 5.

6.3.4 Tooling challenges

T 1 *(NOT RELEVANT)* discusses the challenge of pushing changes in the EA tool to the attached data sources. We argue that, in most cases, this should not be a goal, since the providing data sources should be the master data providers for their respective information. Hence, we do not approach this challenge listed by Hauder et al. [32].

T 2 *(ADDRESSED)* mentions the problem when information provided by data sources is too fine grained or simply not relevant. Again, as with *DC2* and *TC5*, we tackle this challenge by introducing a data source selection process that explicitly recognizes this problem and helps to identify relevant sources.

T 3 *(NOT ADDRESSED)* discusses the challenge that EA model analysis mechanisms have to be able to cope with a changing EA information model. Since EA information model changes are currently an open research area, we did not consider this challenge of automatically adopting analysis rules, such as event triggers, according to these changes.

T 4 *(ADDRESSED)* asserts the general challenge of missing tool support for automating EA documentation. With the meta-model discussed in Sect. 5 and the EA repository prototype presented in the preceding section we showed how automation can be approached holistically.

As can be seen from the discussion of the challenges for automated EA documentation listed in Table 1, the majority of the challenges were tackled. In particular, our approach handles the selection procedure of appropriate data sources and thereby covers many of the described challenges. We did not cover the tool challenge *T3* which is concerned with changing EA information models. Also, the data challenge *DC 6* is only partly covered since automated update of removed model elements still can only be partly achieved via events and task-based reminders. Finally, we consider *T1* as not directly relevant for EA documentation because we argue that the data sources should still be the master data sources for their respective information.

We conclude that the paper at hand constitutes one of few approaches that aim to consolidate different efforts to enhance EA documentation via processes and automation. However, it also goes beyond existing work by including a situational approach and corresponding documentation process assembly methods. The required functionality for the complete approach can be implemented with current technology, as we showed with the prototype. This usability and performance of the prototype, however, has not been evaluated under realistic conditions. Nevertheless, experiences drawn from the prototype were used to implement the EA tool that was implemented for the case study which proofed its utility in practice. The presented method assembly process was also successfully applied in practice to elicit the context and requirements for EA documentation in an organization. Its applicability in aiding the development of an organization-specific documentation process was shown.

However, there are several threats to the validity of the work presented in this paper. First, it has to be stated that we did not yet fully evaluate all parts of our approach in the case study in practice. For example, events have not been utilized in the tool that was implemented. In addition, we executed the method assembly process only in one organization where it showed to be useful. More practical evaluations need to be executed to underline the utility of our work. Also the actual difference our method assembly process made is hard to measure, because there was no prior method in place to compare it to.

Nevertheless, the systematic literature review showed, to the best of our knowledge, our approach covers more challenges than current related work.

7 Conclusion and future work

EA documentation is a labor-intensive, yet crucial process for EA management. Current practical implementations of this process are usually ad hoc and fail to account for already established documentation processes and responsibilities for related management activities. Our literature review further showed that academic approaches to systematically design organization-specific documentation processes are scarce or limited to automated EA documentation (see Hauder et al. [32]).

In the introduction of this paper, we posed the research question: How can EA documentation processes be devised and supported by a tool that optimally supports the information demand and context of a specific organization?. We have presented composable techniques, a tool and a method assembly process to answer this question.

The approach presented here does not solely focus on automation, but also accounts for manual activities and interventions where necessary. The four different model maintenance techniques that constitute the core of our approach are designed to be applied to architecture elements on all different layers of the EA, ranging from business to IT infrastructure. For each of these techniques, the relevant application context is stated and provides a basis for selecting the most appropriate technique for a specific organizational setup. A comprehensive meta-model allows to precisely describe the selected model maintenance techniques and provides the basis for a prototypical implementation of our method. The presented case study showed the practical applicability of the method to create a viable, useful and tool-supported documentation process at an insurance company.

Our work impacts practice from two different angles. First, EA practitioners might use the presented techniques, and in particular, the assembly method in order to analyze and enhance their documentation strategies. Second, tool vendors might recognize the importance of process support in order to enhance their offerings with the presented techniques.

Further research needs to be executed both in evaluating the techniques in practice and in enhancing or creating new techniques. Another interesting research area is the problem of the abstraction mismatch between data sources and the detail level needed for EA. Further case studies are necessary to understand this mismatch in more detail. This might lay the basis for more sophisticated techniques that address this abstraction problem. In particular, the open issue of detecting removed elements needs to be tackled.

Our future work specifically targets practical evaluation of the situational method and tooling. Additional cases will show, which architecture elements can be drawn from which data sources using which techniques. Thereby, we aim to iteratively enhance the presented techniques and discover new documentation techniques where appropriate.

8 Appendix

8.1 Systematic literature review details

As a first step of the literature review, the main two research questions were fixed. These were the following:

1. What is the related work in the context of (automated) Enterprise Architecture Documentation in EA research literature?
2. Is there literature that combines EA documentation research and situational method engineering?

We then crafted a protocol of the review execution that included the method of data collection, a timeline as well as quality criteria for inclusion of literature. The categories for of the literature were established during the review process. In the following, we describe the review process in more detail and then list the data sources and keywords that were used for the review.

8.1.1 Review process

In the first step, we used the scientific literature search engine Google Scholar to search for terms which we list below. We went through the first ten pages of results, leading to a manual search of the top 100 search results on Google Scholar per search term. Relevant papers were first collected in a preliminary list. Also in this step, we manually searched through the last ten years of publications of the journals and proceedings of scientific events listed below. From the first list of identified papers, we performed a backward search of the included references as well as a forward search of papers citing a given article via Google Scholar. A second list was created in this step for which this procedure was repeated until no new relevant literature was identified.

In addition to the search of scientific literature, we included several books from practitioners in the review. The general inclusion criteria were the description of practices or recommendations for the manual or automated maintenance of EA models and in particular any reference to situativeness in the documentation approach. Since the amount of identified literature was comparatively little and we wanted to get a general overview of the state of research in EA documentation, we did not specify any exclusion criteria regarding the quality of evaluation of the identified literature. Publications by the same author on the same topic were not considered twice. Hence, we always included the newest publication in such cases.

8.1.2 Data sources

As already mentioned above, the main database that was searched was Google Scholar as it integrated several other main search engines including the ACM Portal[15] and the IEEE XPlore Digital Libary.[16]

Journals

- Software and Systems Modeling (SOSYM)[17]
- Journal of Enterrise Architecture (JEA)[18]
- Enterprise Modeling and Information Systems Architectures (MoBIS)[19]
- Management Information Systems Quartely (MISQ)[20]

Scientific events

- Hawaii International Conference On System Sciences (HICSS)[21]
- Enterprise Distributed Objects Conference and Workshops (EDOC)[22]
- Trends in Enterprise Architecture Research Workshop (TEAR)[23]
- International Conference on Advanced Information Systems Engineering (CaiSE)[24]
- Americas Conference on Information Systems (AMCIS)[25]
- European Conference on Information Systems (ECIS)[26]

8.1.3 Search terms

In the following, we list the search terms that were used to search Google Scholar. The term "enterprise architecture" was combined with the following list of keywords. The first ten pages of results were examined.

{"*enterprise architecture*"} × {
"*documentation modeling*",
"*maintenance*",
"*model maintenance*",
"*update*",
"*up − to − date*",
"*documentation process*",
"*modeling process*, "
"*architectural description*",
"*data collection*",
"*model repository*",
"*model repository update*",
"*information retrieval*",

[15] http://portal.acm.org.

[16] http://ieeexplore.ieee.org.

[17] http://www.sosym.org.

[18] http://www.globalaea.org/?page=JEAOverview.

[19] http://www.wi-inf.uni-duisburg-essen.de/MobisPortal/index.php?lang=en.

[20] http://www.misq.org/.

[21] http://www.hicss.hawaii.edu.

[22] http://www.edoc2014.org.

[23] The proceedings of TEAR were published with several different publishers of which some can be found here: http://www.informatik.uni-trier.de/~ley/db/conf/tear/index.html.

[24] http://www.informatik.uni-trier.de/~ley/db/conf/caise/index.html.

[25] http://aisel.aisnet.org/amcis/.

[26] http://www.ecis.org/.

"*model quality*",
"*model quality assurance*",
"*actuality*",
"*update*",
"*semantic integration*",
"*data source*",
"*information source*",
"*data collection*",
"*automated data collection*",
"*information collection*",
"*automatic*",
"*automated*",
"*semi − automated*",
"*challenge*",
"*master data management*",
"*situational method*"
}

References

1. Ahlemann, F., Stettiner, E., Messerschmidt, M., Legner, C.: Strategic Enterprise Architecture Mangement. Springer, Berlin (2012)
2. Aier, S., Buckl, S., Franke, U., Gleichauf, B., Johnson, P., Närman, P., Schweda, C., Ullberg, J.: A survival analysis of application life spans based on enterprise architecture models. In: 3rd International Workshop on Enterprise Modelling and Information Systems Architectures, pp. 141–154. Ulm, Germany (2009)
3. Aier, S., Gleichauf, B.: Application of enterprise models for engineering enterprise transformation. Enterp. Model. Inf. Syst. Arch. **5**, 56–72 (2010)
4. Aier, S., Gleichauf, B.: Understanding Enterprise Architecture Management design—an empirical analysis. In: Proceedings of the 10th International Conference on Wirtschaftsinformatik WI 2011 (Zürich) (2011)
5. Aier, S., Kurpjuweit, S., Saat, J., Winter, R.: Enterprise architecture design as an engineering discipline. AIS Trans. Enterp. Syst. **1**(1), 36–43 (2009)
6. Alegria, A., Vasconcelos, A.: IT architecture automatic verification. In: 2010 Fourth International Conference on Research Challenges in Information Science (RCIS) (2010)
7. Arbab, F., de Boer, F., Bonsangue, M., Lankhorst, M., Proper, H., van der Torre, L.: Integrating architectural models: symbolic, semantic and subjective models in enterprise architecture. Enterp. Model. Inf. Syst. Archit. **2**(1), 40–57 (2007)
8. Brinkkemper, S.: Method engineering: engineering of information methods and tools. Inf. Softw. Technol. **38**(4), 275–280 (1996)
9. Brückmann, Tobias, Gruhn, Volker, Pfeiffer, M.: Towards real-time monitoring and controlling of enterprise architectures using business software control centers. Lect. Notes Comput. Sci. Softw. Archit. **6903**, 287–294 (2011)
10. Buckl, S., Ernst, A., Lankes, J.: Generating visualizations of enterprise architectures using model transformations. Enterp. Model. Inf. Syst. Archit. **2**(2), 03–13 (2007)
11. Buckl, S., Ernst, A., Matthes, F., Schweda, C.: How to make your Enterprise Architecture Management endeavor fail! In: Pattern Languages of Programs 2009 (PLoP 2009), Chicago (2009)
12. Buckl, S., Ernst, A.M., Lankes, J., Matthes, F., Schweda, C.M.: State of the Art in Enterprise Architecture Management. Technical Report (2009)
13. Buckl, S., Matthes, F., Neubert, C.: A lightweight approach to enterprise architecture modeling and documentation. Inf. Syst. Evol. **72**, 136–149 (2011)
14. Buckl, S., Schweda, C.M.: On the State-of-the-Art in Enterprise Architecture Management Literature. Tech. rep, Technische Unversität München (2011)
15. Buckl, S.: Developing Organization-Specific Enterprise Architecture Management Functions Using a Method Base. Ph.D. thesis, Technical University Munich (2011)
16. Buschle, M., Ekstedt, M., Grunow, S., Hauder, M., Matthes, F., Roth, S.: Automating Enterprise Architecture Documentation using an enterprise service bus. In: Americas Conference on Insormations Systems (AMCIS) (2012)
17. Engels, G., Hess, A., Humm, B., Juwig, O., Lohmann, M., Richter, J.P.: Quasar Enterprise: Anwendungslandschaften Serviceorientiert Gestalten. Dpunkt Verlag, Heidelberg (2008)
18. Farwick, M., Agreiter, B., Breu, R., Ryll, S., Voges, K., Hanschke, I.: Automation Processes for Enterprise Architecture Management. In: 2011 IEEE 15th International Enterprise Distributed Object Computing Conference Workshops, pp. 340–349. IEEE, Helsinki, Finland (2011)
19. Farwick, M., Agreiter, B., Ryll, S., Voges, K., Hanschke, I., Breu, R.: Requirements for automated enterprise architecture model maintenance. In: 13th International Conference on Enterprise Information Systems (ICEIS), Beijing (2011)
20. Farwick, M., Hauder, M., Roth, S., Matthes, F., Breu, R.: Enterprise Architecture Documentation: empirical analysis of information sources for automation. In: Proceedings of 46th Hawaii International Conference on Systems Science (HICCS-46). IEEE Computer Society, Wailea, Maui, USA (2013)
21. Farwick, M., Pasquazzo, W., Breu, R., Schweda, C.M., Voges, K., Hanschke, I.: A meta-model for automated enterprise architecture model maintenance. In: 2012 IEEE 16th International Enterprise Distributed Object Computing Conference, pp. 1–10. IEEE (2012)
22. Farwick, M., Schweda, C., Breu, R.: On enterprise architecture change events. In: Trends in Enterprise Architecture Research (TEAR), pp. 129–145. Barcelona, Spain (2012)
23. Fischer, R., Aier, S., Winter, R.: A federated approach to enterprise architecture model maintenance. Enterp. Model. Inf. Syst. Archit. **2**(2), 14–22 (2007)
24. Frank, U., Heise, D., Kattenstroth, H.: Use of a domain specific modeling language for realizing versatile dashboards. In: Proceedings of the 9th OOPSLA Workshop on Domain-Specific Modeling (DSM) (2009)
25. Fuchs-Kittowski, F., Faust, D.: The semantic architecture tool (SemAT) for collaborative enterprise architecture development. In: Groupware: Design, Implementation, and Use, pp. 151–163. Springer, Berlin (2008)
26. Gericke, A., Winter, R.: Situational Change Engineering in Healthcare. In: M. Stormer, H., Meier, A., Schumacher (Ed.) European Conference on eHealth 2006, Lecture Notes in Informatics, pp. 227–238. Gesellschaft für Informatik, Bonn (2006)
27. Gutzwiller, M.: Das CC-RIM Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen. Physika, Heidelberg (1994)
28. Hanschke, I.: Strategic IT Management: A Toolkit for Enterprise Architecture Management. Springer, Berlin (2009)
29. Hanschke, I.: Strategisches Management der IT-Landschaft: Ein praktischer Leitfaden für das Enterprise Architecture Management. Carl Hanser Verlag GmbH & Co. (2010)
30. Happel, H., Seedorf, S.: Documenting service-oriented architectures with ontobrowse semantic wiki. In: Heinzl, E.J., Armin, Appelrath, Hans, J., Sinz (Eds.) Proceedings of the PRIMIUM Sub-

conference at the Multikonferenz Wirtschaftsinformatik (MKWI) (2008)
31. Harmsen, A., Brinkkemper, J., Oei, J.: Situational method engineering for information system project approaches. In: Proceedings of the IFIP WG 8.1 Working Conference, Maastricht, Netherlands, pp. 169–194 (1994)
32. Hauder, M., Matthes, F., Roth, S.: Challenges for automated Enterprise Architecture Documentation. In: Trends in Enterprise Architecture Research (TEAR), pp. 21–39. Springer, Berlin, Barcelona, Spain (2012)
33. Henderson-Sellers, B., Gonzalez-Perez, C., Ralyte, J.: Comparison of method chunks and method fragments for situational method engineering. In: 19th Australian Conference on Software Engineering, 2008 (ASWEC 2008), pp. 479–488 (2008)
34. Hevner, A., March, S.T., Park, J., Ram, S.: Design science in information system research. MIS Q. **28**(1), 75–106 (2004)
35. Holm, H., Buschle, M., Lagerström, R., Ekstedt, M.: Automatic data collection for enterprise architecture models. Softw. Syst. Model. 1–17 (2012). doi:10.1007/s10270-012-0252-1
36. Kaisler, S., Armour, F., Valivullah, M.: Enterprise architecting: critical problems. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences 00(C), pp. 224b–224b (2005)
37. Keller, W.: IT-Unternehmensarchitektur: von der Geschäftsstrategie zur optimalen IT-Unterstützung. Dpunkt-Verlag, Heidelberg (2012)
38. Lankhorst, M.: Enterprise Architecture at Work, vol. 36. Springer, Berlin (2013)
39. Laube, A., Gagnerot, J.M.v.d.W.G., Blösch, M., Verbeek, E., van der Aalst, W., Plate, H.: Concept and Architecture for Automated Model Creation, Population, Maintenance and Audit. Tech. Rep. 257109, FP7 Project PoSecCo (2012)
40. Lucke, C., Krell, S., Lechner, U.: Critical issues in enterprise architecting a literature review critical issues in enterprise architecting a literature review. In: AMCIS 2010 Proceedings (2010)
41. McClure, D.: Integrating Enterprise Architecture and Portfolio Management Processes in Government Gartner. Tech. Rep. June, Gartner Inc. (2006)
42. Moser, C., Junginger, S., Brückmann, M., Schöne, K.: Some process patterns for Enterprise Architecture Management. In: Proceedings of the Workshop on Patterns in Enterprise Architecture Management (PEAM2009), Bonn, pp. 19–30 (2009)
43. omg: Meta Object Facility (MOF) Core Specification Version 2.0 (2006)
44. Roth, S., Hauder, M., Farwick, M., Matthes, F., Breu, R.: Enterprise Architecture Documentation: current practices und future directions. In: International Conference on Wirtschaftsinformatik, pp. 911–925. AIS, Leipzig, Germany (2013)
45. Roth, S., Hauder, M., Michel, F., Münch, D., Matthes, F.: Facilitating conflict resolution of models for automated Enterprise Architecture Documentation. In: AMCIS 2013 Proceedings (2013)
46. Schmidt, A., Osl, P., Back, A., Brenner, W., Österle, H.: A Method for Establishing Transparency on Integration Objects. Tech. rep., University of St. Gallen (2008)
47. Sousa, P., Gabriel, R., Tadao, G.: Enterprise Transformation: The serasa experian case. In: Proceedings of the Third Working Conference, PRET 2011, Luxembourg-Kirchberg, Luxembourg, September 6, 2011. pp. 134–145 (2011)
48. ter Doest, H., Lankhorst, M.: Tool Support for Enterprise Architecture-A Vision. Tech. rep, Telematica Instituut, Enschede (2004)
49. The Open Group: TOGAF "Enterprise Edition" Version 9.1 (2009). http://www.togaf.org
50. Wigand, R., Picot, A., Reichwald, R.: Information, Organization, and Management: Expanding Markets and Corporate Boundaries. Wiley, New York (1999)
51. Chen, W., Hess, C., Langermeier, M., von Stuelpnagel, J., Diefenthaler, P.: Semantic Enterprise Architecture Management. In: 15th International Conference on Enterprise Information Systems (ICEIS) (2013)
52. Winter, K., Buckl, S., Matthes, F., Schweda, C.: Investigating the state-of-the-art in Enterprise Architecture Management methods in literature and practice. In: 2010 Proceedings MCIS (2010)
53. Winter,R., Fischer, R.: Essential layers, artifacts, and dependencies of enterprise architecture. In: Enterprise Distributed Object Computing Conference Workshops, 2006. EDOCW '06. 10th IEEE International, Hong Kong, China (2006)
54. Zachman, J.A.: A framework for information systems architecture. IBM Syst. J. **26**(3), 276–292 (1987)

Matthias Farwick has received his MSc in Computer Science at the University of Innsbruck, Austria, in 2010. Currently, he pursues his PhD studies at the University of Innsbruck in the area of Enterprise Architecture (EA) documentation. In the course of his research, he participated in several R&D projects in the USA, Canada, Germany and China. He also works as a consultant in the field of IT-Landscape management for QE LaB Business Services and is the core architect of the IT documentation tool Txture.

Christian M. Schweda currently is a private researcher in the field of Enterprise Architecture (EA) Management. Formerly, he worked at Technische Universität München (TUM—Munich University of Technology), where he received a PhD for his work modeling EAs. During his work at TUM, he was active in the project teams evaluating different EA Management tools (2005 and 2008). In 2011, Christian joined the iteratec GmbH as head of EA research. In this occupation, he developed and furthered methods and techniques for tool support EA management, both in practice as well as in close cooperation with academic researchers.

Ruth Breu is full professor at the University of Innsbruck, head of the Institute of Computer Science and the research group Quality Engineering (QE) and head of the competence center QE LaB. Her research interests include model engineering, security engineering, enterprise architecture management and requirements engineering. Contact her at ruth.breu@uibk.ac.at, and follow the QE team at qe-informatik.uibk.ac.at, qe-lab.at and @QEresearch.

Inge Hanschke is Managing Director of iteratec GmbH. Her remit includes the iteratec EAM method and the open-source EAM tool iteraplan (http://www.iteraplan.de). In the 24 years since gaining her degree in information technology, she has worked as an IT manager at user-side enterprises, at an ERP product company and at IT service providers. She has successfully aligned both IT and service portfolios to the requirements of business and has professionalized IT management. “Simple and effective” is her motto. This is also reflected in numerous publications and in the book series of the same name published by Hanser Verlag (in German). Titles already published (in German) in this series are “EAM - Simple and Effective” in November 2011 and “Strategic Process Management - Simple and Effective” in February 2012. Her next book, “Business Analysis - Simple and Effective”, appeared in September 2012, and the third edition of her book “Strategic IT Management” in 2013.