Inf Syst Front
DOI 10.1007/s10796-015-9559-0

CHOOSE: Towards a metamodel for enterprise architecture in small and medium-sized enterprises

Maxime Bernaert[1,2] · **Geert Poels**[1] · **Monique Snoeck**[3] · **Manu De Backer**[1,4]

Abstract Enterprise architecture (EA) is a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and IT infrastructure. Recent research indicates the need for EA in small and medium-sized enterprises (SMEs), important drivers of the economy, as they struggle with problems related to a lack of structure and overview of their business. However, existing EA frameworks are perceived as too complex and, to date, none of the EA approaches are sufficiently adapted to the SME context. Therefore, this paper presents the CHOOSE metamodel for EA in SMEs that was developed and evaluated through action research in an SME and further refined and validated through case study research in five other SMEs. This metamodel is based on the essential dimensions of EA frameworks and is kept simple so that it may be applied in an SME context. The final CHOOSE metamodel includes only four essential concepts (i.e. goal, actor, operation, object), one for each most frequently used EA focus. As an example, an extract is included from the specific model that was created for the SME used in our action research. Finally, the CHOOSE metamodel is evaluated according to the dimensions essential in EA and the requirements for EA in an SME context.

Keywords Enterprise architecture · Small and medium-sized enterprises · CHOOSE · Metamodel

✉ Maxime Bernaert
Maxime.Bernaert@UGent.be; Maxime.Bernaert@inno.com

Geert Poels
Geert.Poels@UGent.be

Monique Snoeck
Monique.Snoeck@KULeuven.be

Manu De Backer
Manu.DeBacker@UGent.be

1 Department of Business Informatics and Operations Management, Ghent University, Tweekerkenstraat 2, 9000 Ghent, Belgium

2 Master of ICT Enterprise Architecture, inno.com Institute, Heiststeenweg 131, 2580 Beerzel, Belgium

3 Department of Decision Sciences and Information Management, K.U. Leuven, Naamsestraat 69, 3000 Leuven, Belgium

4 Department of Management Information Systems, University of Antwerp, Prinsstraat 13, 2000 Antwerp, Belgium

1 Introduction

According to IEEE Computer Society (2000), architecture is "the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution". Architecture could thus be defined as "structure with a vision", providing an integrated view of the system designed or studied. At the level of an entire organization, it is commonly referred to as enterprise architecture (EA). This refers to a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and IT infrastructure (Lankhorst 2013). Rather than specific solutions for specific problems, EA is assumed to capture the essence of the business, IT, and its evolution, as this essence is much more stable. In this respect, EA considers an enterprise as a system in which competencies, capabilities, knowledge, and assets are purposefully combined to achieve stakeholder goals. The tangible outcome of this line of reasoning is a blueprint or holistic overview of the enterprise in the form of an integrated collection of models. Hence, architecture can help maintain the essence of the business, while still allowing for optimal flexibility and adaptability (Jonkers et al. 2006).

Published online: 26 May 2015

EA approaches are often experienced as complex, over-engineered, and difficult to implement. Because of the technical detail required for full-scale implementation, EA models tend to become very large, making them more difficult to understand and less effective to reflect on or design enterprises and their supporting systems (Balabko and Wegmann 2006). Due to their resource poverty, SMEs experience even more difficulties than larger enterprises in employing EA experts or hiring external consultants (Kroon et al. 2012). Yet, as some studies have confirmed, they may encounter several problems if they fail to implement EA (Bidan et al. 2012; Bhagwat and Sharma 2007).

Bernaert et al. (2013b) did an extensive problem analysis of EA and SMEs and proposed the concept of EA as a good solution to be used for SMEs to solve problems related to a lack of structure and overview. However, EA is still unknown and hardly used in SMEs. A recent exploratory field study by Bernaert et al. (2013b) examined 27 SMEs and observed that nearly all of them were missing a clear overview of their business organization and none of them actually were using EA (Bernaert et al. 2013b). The authors concluded that there is a pressing need to develop an EA approach specifically adapted to the SME context, consisting of a metamodel, a method, and software tool support.

The goal of the current research is to develop such an EA approach for SMEs, called CHOOSE. As some research has already focused on how to bring EA to SMEs in general (Bernaert et al. 2013b; Bidan et al. 2012; Wißotzki and Sonnenberger 2012; Aarabi et al. 2011; Bernaert and Poels 2011; Jacobs et al. 2011), the value of the current research lies in the fact that, to our knowledge, CHOOSE is the first effort to actually develop an EA approach specifically adapted to the SME context. The present paper will elaborate on the design of the CHOOSE metamodel. The development of the other CHOOSE artefacts is on-going research, consisting of a method to guide the development of CHOOSE models through the instantiation of the metamodel and a suite of software tools to support this instantiation process.

The development of the metamodel was guided by the requirements for EA in an SME context proposed by Bernaert et al. (2013b) and involves a constant trade-off between comprehensiveness and simplicity. Intended for EA, the metamodel needs to provide a holistic overview and thus incorporate the essential dimensions of existing EA approaches. At the same time, though, the metamodel is also intended for SMEs, so it is kept as simple as possible, without being too simple. In order to find the right balance, a set of EA frameworks used in business and academia was analyzed to capture the essential dimensions of EA approaches.

After the essential dimensions of EA approaches had been defined, a suitable starting point for designing the CHOOSE metamodel needed to be found. From different investigated metamodels, the metamodel of the KAOS requirements engineering methodology (Van Lamsweerde 2009) was found to be the most suitable as it is rather elaborate and provided a good match with the essential dimensions that had been determined.

Next, during multiple rounds of action research (Järvinen 2007) in one specific SME that complied with the characteristics of SMEs as proposed by Bernaert et al. (2013b), the KAOS metamodel was adapted and transformed into the CHOOSE metamodel. Some of the changes to the developing metamodel were, however, triggered by parallel case study research in five other SMEs, which was initiated to design the CHOOSE method. Any changes that the action research participants considered useful were also incorporated into the final CHOOSE metamodel.

This final metamodel comprises four viewpoints: (1) a goal viewpoint for the motivational part (i.e. why), (2) an actor viewpoint for the active performers (i.e. who), (3) an operation viewpoint for the behavioural part (i.e. how), and (4) an object viewpoint for the description of the concepts and relationships (i.e. what). In this way, the core part of the CHOOSE metamodel only consists of the bare minimum of concepts (only one main concept per viewpoint) in order to maintain the balance between both comprehensiveness for EA and simplicity for SMEs. Since in the original KAOS metamodel all the viewpoints are tightly integrated, in the resulting CHOOSE metamodel also a high traceability within and between the four viewpoints was achieved.

The CHOOSE metamodel is written in UML (Unified Modeling Language). Its elements are defined using SBVR (Semantics of Business Vocabulary and Rules) and intra- and inter-view constraints are specified as OCL (Object Constraint Language) constraints. These SBVR definitions are based on definitions of well-known modelling languages and thus contribute to the unambiguous definition of the metamodel concepts. The OCL constraints, in their turn, help ensure the completeness and consistency of the models that instantiate the metamodel.

The instantiation of the proposed metamodel is demonstrated by means of the EA model that was developed during the action research programme in the SME. This also provided the basis for the evaluation of the metamodel, a process that was guided by the EA essential dimensions and the requirements for EA in SMEs.

Section 2 of this paper elaborates on the research problem, the intended contribution of this research, and the requirements for EA in an SME context. In Section 3, the solution approach, the scope of the research presented in this paper, and the research methodology are presented. The results are shown from Section 4 onwards: the definition of essential EA dimensions based on an analysis of EA frameworks (Section 4); the choice of KAOS as a starting point for the metamodel design (Section 5); the adaption of the initial metamodel and the development of the CHOOSE metamodel during the action research and case studies (Section 6); the formal definition of the resulting CHOOSE metamodel (Section 7); and, finally, its evaluation (Section 8). The final section, Section 9, presents conclusions and outlines the current and future research required to complete the development of CHOOSE.

2 Problem description and solution requirements

This section describes the research problem and the requirements for its solution, based on a review of related and previous research.

2.1 Problem description

A good EA gives a static overview of the enterprise and offers a means for supporting change. A good architectural practice helps a company innovate and change, by providing both stability and flexibility (Jonkers et al. 2006). Jonkers et al. (2006) further mention that it is important to realize that most stakeholders of a system are probably not interested in its architecture, but only in the impact of this architecture on their concerns. In addition, although they often have radically different backgrounds, an architect should be able to explain the architecture to all of the stakeholders just as clearly. This highlights one of the most important roles of EA: it serves as an instrument in the communication among diverse groups and interests and produces a common ground for discussion and decision-making.

EA has become one of the top priorities of IT executives and is considered an important instrument for aligning the required changes in corporate strategy and business processes with an increasingly complex IT landscape (Luftman and Ben-Zvi 2011). Some of the most recognized benefits of EA are that IT can be used more efficiently and flexibly, business and IT can be better aligned (Radeke 2011; Tamm et al. 2011; Daneva and van Eck 2007; Lindström et al. 2006), and a better fit between business operations and strategy can be achieved (Hoogervorst 2004; Veasey 2001). Braun and Winter (2005) underscore that in order for business-IT and strategy to be aligned, EA must be adaptable and constantly held up-to-date.

SMEs constitute over 90 % of operating businesses in many countries, in the U.S. even 99.7 % (Small Business Administration 2011) and in Europe 99.8 % (European Commission 2011). There is therefore a great need for more rigorous research that is relevant for this important sector of the economy (Devos 2011).

Right now, existing EA frameworks are primarily used in large enterprises (Gartner 2012). Wißotzki and Sonnenberger (2012), among others, recognize the importance of EA and EA management (EAM) in particular, but also notice that EAM is still mostly unexplored and rarely used, especially in the context of SMEs (see also (Bernaert et al. 2013b; Devos 2011)). Yet, such specific research is crucial, as research findings based on large businesses cannot be generalized to small businesses due to the inherent differences between SMEs and large businesses (Aarabi et al. 2011).

Lybaert (1998) discovered that SME owners or managers with a greater strategic awareness use more information and that SMEs that use more information are generally more successful. Hannon and Atherton (1998) further revealed that for SMEs success is correlated with higher levels of strategic awareness and better planning of owners-managers. In addition, there is evidence to believe that companies that make strategic rather than just financial business plans perform significantly better financially than those that do not (O'Regan and Ghobadian 2004; Smith 1998). Jacobs et al. (2011) argue that from the perspective of change and complexity, EA could assist SME management during the growth of a small enterprise. For example, according to Aarabi et al. (2011), ERP (Enterprise Resource Planning) systems cannot be successfully implemented and utilized in SMEs if EA is disregarded. In fact, it is EA's integration of strategic goals, business processes, and technology planning methods that provides the standards, roadmap, and context for ERP implementation (Zach 2012). As Bidan et al. (2012) conclude, process standardization in SMEs is more important than the deployment of technology (e.g., ERP systems) to improve organizational performance. In short, SMEs need to get a structured view of their company, even before they start implementing an ERP solution.

Hence, while EA might offer SMEs a solution to typical problems related to a lack of overview, strategic awareness, IT planning, and business-IT alignment, EA approaches that cater for the specificities of small businesses are still missing. This lack of research on an EA approach that can readily be used for SMEs is exactly the problem that is addressed in the present research.

2.2 Requirements for EA for SMEs

To guide the development and evaluation of an EA approach for SMEs, requirements for an appropriate solution are needed. These requirements were specified in previous research (Bernaert et al. 2013b) and will be summarized here. First, the requirements for EA in general are presented, followed by those for the adoption and successful use of IT in SMEs. To end, the combination of these two sets of requirements into a single set for EA in an SME context, as per (Bernaert et al. 2013b), is also described.

2.2.1 Requirements for EA

The essential requirements for EA (Bernaert et al. 2013b; Lankhorst 2013; Zachman 1987) are the following:

1. Control: EA should be usable as an instrument in controlling the complexity of the enterprise and its processes and systems.

2. Holistic Overview: EA should provide a holistic overview of the enterprise and be able to capture its essence: the stable elements that do not vary across specific solutions found for the problems currently at hand.
3. Objectives: EA should facilitate the translation from corporate strategy into daily operations.
4. Suitability: EA should be suitable for its target audience. It needs to be understood by all those involved, even if they come from different domains.
5. Enterprise-wide: EA should enable optimization of the company as a whole instead of doing local optimization within individual domains.

The fourth requirement refers to the target audience. In our case, the target audience is SMEs and, more specifically, their owners or managers. Therefore, requirement 4 is refined using the requirements for the adoption and successful use of IT in SMEs. This topic has been dealt with extensively in several studies, listed by Bernaert et al. (2013b). The authors argue that since Moody (2003) showed that IT adoption models are also useful for evaluating the adoption of IT-related methods (e.g., information systems design methods), and that EA, with its origins in IT research (Zachman 1987), can be seen as such a method, IT adoption models for SMEs can provide useful insight into the determining factors for successfully using EA in SMEs.

2.2.2 Requirements for the adoption and successful use of IT in SMEs

The requirements for the adoption and successful use of IT in SMEs (Bernaert et al. 2013b) are as follows:

4.1 The approach should enable SMEs to work in a time-efficient manner on strategic issues.
4.2 A person with limited IT skills should be able to apply it.
4.3 It should be possible to apply the approach with little assistance of external experts.
4.4 The approach should enable making descriptions of the processes in the company.
4.5 The CEO must be involved.
4.6 The expected revenues of the approach must exceed the expected costs and risks.

By combining these requirements with the EA requirements of the previous section, Bernaert et al. (2013b) obtained a set of requirements for the adoption and successful use of EA in SMEs.

According to requirement 4 and thus 4.1–4.6, the EA model should be understandable and adaptable by non-EA experts in SMEs. The previously mentioned role of EA as a communication instrument can only be established by tailoring an EA approach to the specificities of SMEs. Bernaert et al. (2013b) therefore argue for a different EA approach for SMEs, based on simplicity. We are fully aware that focusing on simplicity rather than on completeness is not common in an academic context. However, also Balabko and Wegmann (2006) emphasized that current EA approaches are often experienced as complex, over-engineered, and difficult to implement.

3 Solution approach and research methodology

In this section, we will present CHOOSE as the solution to the problem described in the previous section. We will limit the scope of the research presented in this paper to the primary artefact of CHOOSE (i.e. its metamodel) and we will describe the research methodology that was followed to develop and evaluate this metamodel.

3.1 CHOOSE: Balancing comprehensiveness and simplicity

Our solution consists of developing a new EA approach guided by the requirements for EA in an SME context (cf. Section 2.2). The approach was called CHOOSE, so that these requirements would always be kept in mind. CHOOSE is an acronym for "maintain Control, by means of a Holistic Overview, that is based on Objectives and kept Simple, of your Enterprise".

It is clear that the development of the CHOOSE metamodel will involve an on-going assessment of comprehensiveness and simplicity (see the methodological pragmatism (Rescher 1977)), because it should include the necessary information to get a holistic overview of the enterprise, while still being as simple as possible. As Albert Einstein once said, "A scientific theory should be as simple as possible, but no simpler".

The meaning of simplicity and complexity of a metamodel can be found in related work by Erickson and Siau (2007), in which a simplified core of the UML metamodel is proposed, based on key constructs. They argue that any increase of this core comes at the expense of increased complexity. Their work is mainly based on the work of Rossi and Brinkkemper (1996), who argued that "the relative complexity of methods and techniques based on metamodels is significant because it can be expected to affect the learnability and ease of use of a method". In other words, the number of metamodel objects, relationships, and properties to be learned adds to the complexity.

There is of course a trade-off between a metamodel's learnability and its expressive power. When organizations select metamodels, they should be aware that more powerful metamodels may be harder to learn, yet may also be more effective for experienced users. As previously mentioned, though, related research on EA in SMEs shows that SMEs hardly use EA, even hardly know about its existence, and can therefore be seen as novice users.

3.2 Research process and scope

This work extends the earlier research by Bernaert et al. (2013b). Their research investigates why EA has not yet been adopted by SMEs, despite its possible benefits. In this respect, Bernaert et al. (2013b) also present a research process (Fig. 1) for developing an EA approach adapted to the SME context.

The dark grey lines in Fig. 1 express the work that has been done by (Bernaert et al. 2013b). In step 1, both the literature on EA and IT use in SMEs were analyzed and relevant characteristics were examined. From these characteristics, requirements were extracted for EA in an SME context, which have already been summarized in this paper in Section 2.2.

The black lines in Fig. 1 highlight the part of the research process that is reported in this paper. The light grey lines in Fig. 1 refer to the (on-going) research required for developing the CHOOSE method and supporting software tools, which lies beyond the scope of this paper.

Step 2 was desk research based on a literature study and analysis, which involved choosing a suitable starting point to design the CHOOSE metamodel. While constantly keeping in mind the balance between comprehensiveness and simplicity, we analyzed a large number of existing EA frameworks in order to extract the essential dimensions of EA frameworks. In the end, an initial metamodel (i.e. the KAOS metamodel) was selected that matched these dimensions.

Step 3 was field research conducted primarily by means of action research in an SME and complemented with case study research in five other SMEs. Through the action research programme, the metamodel was gradually further developed, with the initial metamodel as a starting point. The outcome of the action research was also used to evaluate the research results with respect to the EA essentials and the requirements for EA in an SME context (step 6).

After the start of the action research, five case studies involving the use of CHOOSE were initiated in SMEs with different characteristics (e.g., size, sector). These case studies were primarily used to develop the CHOOSE method (step 4). As the development of this method required us to implement CHOOSE, the initial version of the metamodel that was available at that time in the action research programme was also tested in these other SMEs. Hence changes to the initial metamodel were also tested in other SMEs. Conversely, the experiences in the case study companies were used as additional input to the action research. Therefore, when necessary, these other case studies are briefly referred to in Section 6, where the development of the CHOOSE metamodel is described.

3.3 Action research

The main research methodology employed in step 3 of Fig. 1 was action research (Susman and Evered 1978). Action research employs the researcher as an active participant rather than a passive observer. It is a cyclical process of actively participating in an enterprise change situation while at the same time doing research. The basic steps are *planning* (i.e. problem identification), *acting* (i.e. changing and learning processes), and *evaluating* (i.e. measuring results) (French and Bell 1973). According to Järvinen (2007), action research is an instance of the design science methodology (Hevner et al. 2004) that is suitable when little theoretical background or experience is available, which is the case for the implementation of EA in SMEs.

Baskerville and Myers (2004) provide three guidelines for good action research, which we applied as follows:

1. Demonstrate a contribution or potential contribution to practice (i.e. the action): EA for SMEs could provide SMEs with solutions to problems related to a lack of structure and overview (Bernaert et al. 2013b).
2. Demonstrate a clear contribution to research (i.e. the theory): This research develops the CHOOSE metamodel for EA in an SME context, an artefact that can be further refined and tested in other research.
3. Identify the criteria by which to judge the research and demonstrate how these criteria are met: The criteria for our research were presented as requirements for EA in an SME context in Section 2.2 and are part of the evaluation in Section 8.

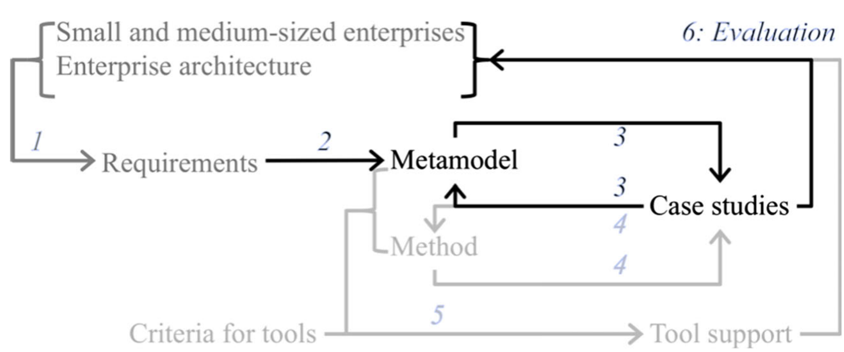

Fig. 1 Research process for developing an EA approach for SMEs (from (Bernaert et al. 2013b))

The action research was performed in multiple rounds in an SME that sells car tyres and performs small maintenance jobs on cars (i.e. case study 1 in Fig. 2). It has six permanent employees and works with temporary employees during the busy winter season. This SME was chosen because it complied with the common characteristics of SMEs (Bernaert et al. 2013b): management has little time to look at strategic matters, no EA experts are employed, no funds to hire external consultants are available, the extent of employees' responsibility for certain tasks is often discussed, the CEO is the central figure, and the CEO takes the decision of whether or not to adopt a new approach.

In the first action research cycle, of which the results were published in (Bernaert and Poels 2011), the KAOS metamodel was used in its original form as a feasibility test (see Fig. 2) to see if it could be used to model the EA of an SME. It turned out that KAOS in fact did have the ability to document and analyze the EA of an SME, although it was originally developed for modelling software-intensive systems within their organizational or physical environment (Van Lamsweerde 2009). Nevertheless, the test also showed that the metamodel needed to be adjusted in order to change its scope from a system on the software level (KAOS) to a system on the enterprise level (CHOOSE). This called for more action research cycles.

Four further cycles of action research were performed (see action research cycles in Fig. 2). In each round, the CEO of the SME was involved in completing the SME's EA model according to the CHOOSE metamodel version available at that moment. To ensure more objectivity in evaluating the results, in each round two researchers were involved to obtain investigator triangulation (Denzin 2006). Each round was voice recorded to obtain raw data and both researchers made additional notes. The voice recordings, notes, and models were stored in a case study database. As most of the data involved strategic issues, a limitation of this research is that the case study database contains confidential data and cannot be made public.

To analyze the data obtained in each action research cycle, the process presented in (Susman and Evered 1978) was followed:

- Diagnosing: The model, voice recordings and the notes of both researchers were analyzed, on the basis of which a list was established of encountered problems that called for adaptations to the metamodel.
- Action planning: For each problem, a set of possible adaptations to the metamodel was considered by the researchers, favouring adaptations that were likely to be more generally accepted by CEOs of SMEs.
- Action taking: The SME's EA model was changed according to the proposed adaptations to the metamodel.
- Evaluating: The model changes were evaluated to see if the problems were solved and if new problems would surface.
- Specifying learning: Positively evaluated adaptations were included in the next version of the CHOOSE metamodel.

As expected, after each round fewer changes had to be made and after three of the four additional rounds the metamodel had become stable. In the meantime, some other adaptations triggered by the case study research in the other five SMEs (i.e. case studies 2–6 in Fig. 2) were tested and evaluated in the SME used in the action research. If these adaptations were positively evaluated, they too became part of the final version of the CHOOSE metamodel.

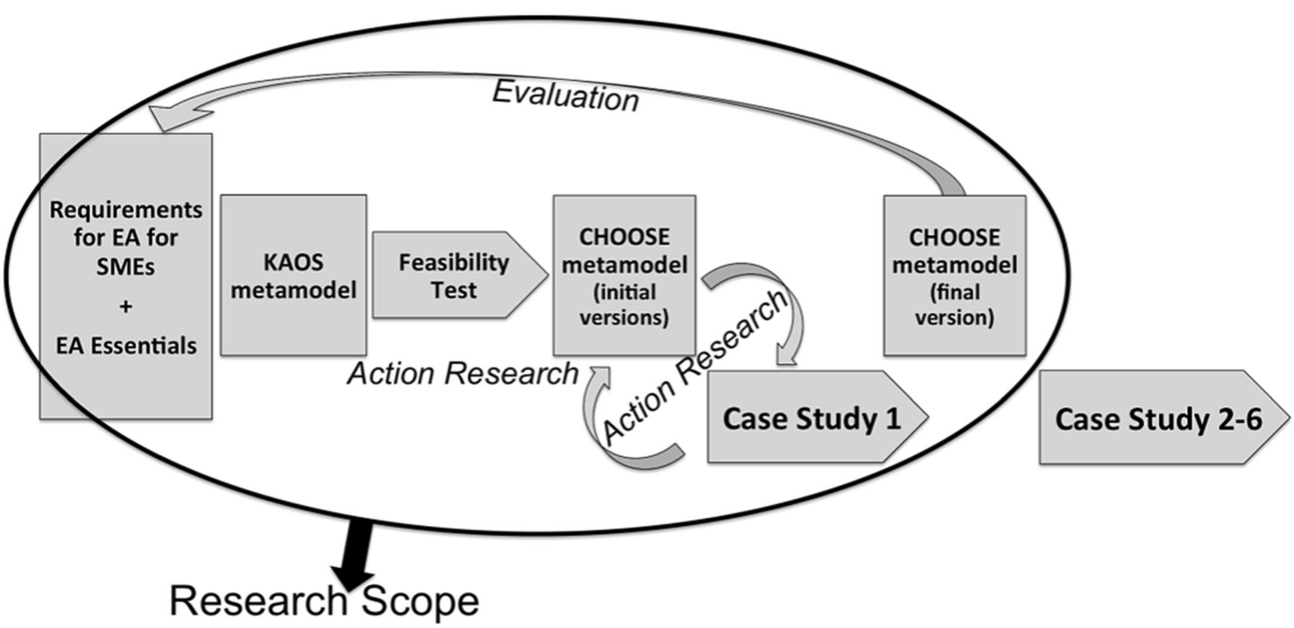

Fig. 2 Research process for developing the CHOOSE metamodel

This final metamodel provided input for the development of prototype software tools (step 5 in Fig. 1) (Bernaert et al. 2013a; Dumeez et al. 2013; Ingelbeen et al. 2013). One such tool was installed in the SME in the last round and enabled it to manage its EA model after the end of the action research programme. As such, this tool can supply longer-term feedback on the CHOOSE metamodel.

4 Essential dimensions of enterprise architecture

In this section, existing EA frameworks are first reviewed, so that the most important ones may be pinpointed. Next, the identified frameworks are analyzed in order to determine the essential EA dimensions. These dimensions are then used in the next section to help select a suitable starting point to design the CHOOSE metamodel.

4.1 Enterprise architecture frameworks

Since the publication of the Zachman framework in 1987 (Zachman 1987), a multitude of EA frameworks have been proposed. In order to identify the essential elements of an EA metamodel in its most simple form, balancing comprehensiveness and simplicity (see Section 3.1) and meeting the EA requirements for SMEs (see Section 2.2), this section aims to identify the most common elements in the most important EA frameworks proposed so far. These essential dimensions of EA define the degree of freedom that can be exerted in adapting the CHOOSE metamodel during the action research cycles, as they set clear and minimal boundaries for the key elements that the metamodel should include.

To identify the most important frameworks, we studied several reviews and historical overviews of EA frameworks, such as the one provided by Georgiadis (2015) (Fig. 3). The

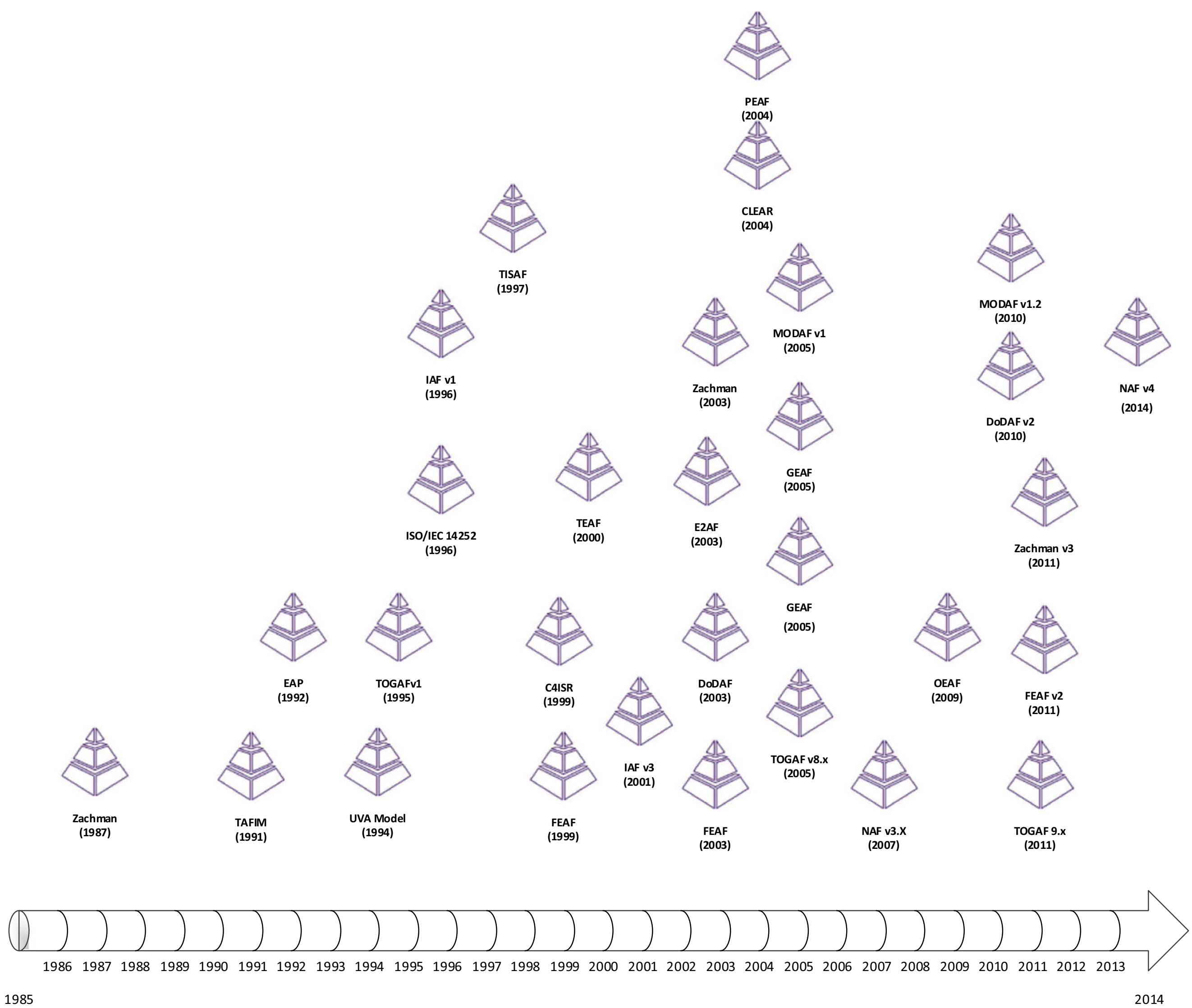

Fig. 3 Historical overview of EA frameworks (updated by Georgiadis (2015) from (Schekkerman 2006))

overview by Schekkerman (2006) is less recent, but interesting for its explanation of the influences EA frameworks have had on each other. Based on these influence relationships, Zachman (Zachman International 2011), TOGAF (The Open Group 2009), DoDAF (DoD 2010), and E2AF (IFEAD 2006) appear to be important EA frameworks. Zachman gave rise to another EA framework, TEAF, which was created for the US Department of the Treasury. Yet, since it is subsumed in the Federal Enterprise Architecture (FEA), just like FEAF, it is better to include FEA instead of TEAF.

Zachman, TOGAF, DoDAF, FEAF, and TEAF are all analyzed in the study of Urbaczewski and Mrdalj (2006). Sessions (2007), on the other hand, compares the first two, Zachman and TOGAF, with FEA and Gartner's GEAM. Yet another study by Leist and Zellner (2006) juxtaposes Zachman, TOGAF, DoDAF, FEAF, TEAF, ARIS, and MDA (model-driven architecture). The last one, MDA, is more a general systems development approach, so it will not be included in our further analysis here.

In short, the most widely discussed EA frameworks that should also be included in the present analysis are Zachman (Zachman International 2011), TOGAF (The Open Group 2009), DoDAF (DoD 2010), E2AF (IFEAD 2006), FEA (The White House OMB 2012, 2013), GEAM (Gartner) (Bittler and Kreizmann 2005; James et al. 2005), and ARIS (Scheer 2000) (see Table 1).

This selection of relevant EA frameworks is confirmed by the survey of IFEAD (2005) and, more recently, by the survey of Gartner (2012) on the use of EA frameworks in companies (Fig. 4). However, a lot of companies also use a homemade EA framework or hire a consulting firm (e.g., IBM, Deloitte) to help them craft a best-of-breed framework. ArchiMate (Lankhorst 2013) was also included in Table 1, because it was recently adopted as a standard by The Open Group (2012) to be used in combination with TOGAF. Capgemini's IAF (van't Wout et al. 2010) was also added because it was built based on experience in more than 3000 EA projects and it evolves faster than any standard ever can. As such, it lies at the basis of large parts of TOGAF 9's content framework. The Business Motivation Model (BMM) (OMG 2010) is also relevant for our study because of its emphasis on the motivational dimension. Yet it does not give a holistic EA overview and is not actually an EA framework, so it has been placed between brackets in Table 1. At the same time, though, BMM is often included in business architecture analyses (Glissman and Sanz 2009), so it should definitely be taken into account in our analysis. Finally, Sogeti's DYA (Wagter et al. 2005) offers a holistic view and should therefore also be included in Table 1.

To make sure that recently developed EA frameworks were not ignored, we also included several EA frameworks developed in academia, namely CARP (derived from DoDAF) (Business Transformation Agency 2009), Enterprise Modelling (Bubenko 1993) and its successors Enterprise Knowledge Development (EKD) (Stirna and Persson 2007) and 4EM (Sandkuhl et al. 2014), REA extended with goal modelling (Andersson et al. 2009) (Fig. 4), SEAM (Wegmann et al. 2007), and LEAP (Clark et al. 2011).

4.2 Essential dimensions of EA

The essential dimensions of EA were determined in three consecutive steps.

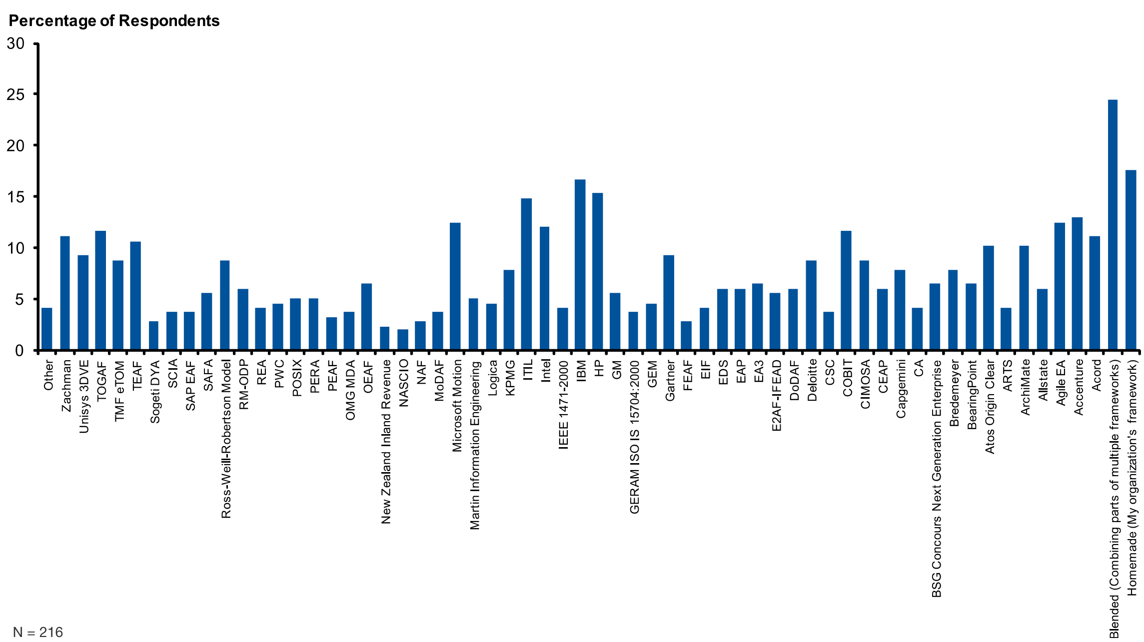

Fig. 4 EA frameworks currently being used (from (Gartner 2012))

Table 1 Analysis of EA frameworks

	What	How	Where	Who	When	Why	Business IS IT	Strategy–Operations
Zachman	What	How	Where	Who	When	Why	B/IS/IT	+
TOGAF	Data entity, …	Process, …	Infrastructure extension	Organization unit, actor, role, …	Event, …	Motivation extension	B/IS/IT	+
ArchiMate	Information	Behaviour	Network, …	Structure	Event, …	Motivation	B/IS/IT	+
DoDAF	Resource	Activity	Location	Performer	–	Capability	Blend	+
CARP	Resource	Activity	–	Performer	–	Capability	B	+
IAF	Object	Activity	Interaction	Actor, role, …	Event	Why, goal, …	B/IS/IT	+
E2AF	Business objects, resources, …	Business activities, …	With who?	Organization structure, actors, …	When?	Why?	B/IS/IT	+
FEA: FEAF	Objects, …	Business process, …	Business locations, …	–	–	–	IS/IT	–
FEA: TEAF	Information, …	Business process, …	Information exchange, …	Organization chart, …	Event, …	Mission, vision, …	B/IS/IT	+
GEAM	–	–	–	–	–	Requirements vision	B/IS/IT	+
ARIS	Input, output, …	Function	–	Organizational unit, …	Event	Goal	Blend	+–
(BMM)	-	(Business process)	–	(Organization unit)	–	End	B	+
DYA	Product, data, …	Process	Network	Organization, …	–	Business objectives	B/IS/IT	+
Enterprise modelling/EKD/4EM	Concepts model	Business process model	–	Actors and resources model	–	Goals model	B/IS	+
REA	Resource	Event	–	Agent	–	Goals	B	–
SEAM	–	–	–	–	–	Strategies	B/IS	+
LEAP	Object	Operation	–	Object	Condition	OCL constraint	B/IS/IT	–

Firstly, according to Schekkerman's (2006) and Georgiadis' (2015) overview of EA influence history, Zachman (1987) seems to be at the very origin of many EA frameworks. The collection of EA frameworks identified in the previous section will therefore be analyzed by means of the six focuses (columns) of the Zachman framework (*what*, *how*, *where*, *who*, *when*, *why*). These focuses make it possible to classify architectural descriptions according to content or subject focus (e.g., objects or data for *what*, processes for *how*, networks or locations for *where*, etc.), so that architecture models according to a particular focus represent a single aspect of the enterprise, abstracting from relationships with the other aspects.

Secondly, Winter and Fischer (2007) identified five essential architectural layers in EA frameworks (i.e. business, process, integration, software, infrastructure). These architectural layers allow a further classification of (parts of) architectural descriptions, so that architecture models are expressed using concepts that represent the enterprise elements that are relevant to a certain perspective, in a way that is comprehensible for the stakeholders in that perspective. The process architecture layer can then be further merged with the business architecture layer and, in its turn, the integration architecture layer can be combined with the software architecture layer. This results in three essential EA layers: business, software, and infrastructure. These layers were used to analyze the EA frameworks in Table 1 (i.e. *business (B)*, *software (IS)*, *infrastructure (IT)*, or a *blend* of the three).

Thirdly, during the analysis of the selected EA frameworks, no additional focuses or architectural layers were identified. However, what also became apparent during the analysis was that most EA frameworks make it possible to translate strategy into operations and often stress the importance of a long and thorough analysis of the strategy space, free from all implementation constraints. Lankhorst (2013), for instance, refers to the strategic alignment model of Henderson and Venkatraman (1993), according to which EA can help in executing the business or IT strategy and enable the alignment between (business or IT) strategy and (organizational or IT) infrastructure and processes. In fact, many EA frameworks provide guidance for the translation from corporate strategy into daily operations. For example, Zachman (2011) defines six views (rows) from "scope" all the way to the "full enterprise", adding more implementation constraints towards the "full enterprise" view. Another example is IAF (van't Wout et al. 2010), which is primarily built upon the principle of analyzing the strategy space for as long as possible without taking into account the constraints of operations beforehand, by using contextual, conceptual, logical, and physical abstraction levels that are closely related to the different views of Zachman.

Since this aspect can be found in many EA studies, we too have decided to incorporate this, so the last column of our analysis shows whether the EA frameworks provide a means to analyze the (business or IT) strategy space while still disregarding the constraints of (organizational or IT) operations (i.e. *strategy–operations*).

Table 1 gives an overview of the analyzed EA frameworks. For each Zachman focus, one or more concepts that represent enterprise elements according to that focus are provided as examples, if they are defined in the metamodel of the EA framework. In the *strategy–operations* column, a minus/plus-minus/plus indicates that a translation from (business or IT) strategy into (organizational or IT) operations is not/limited/clearly supported.

Most of the frameworks use (at least) four focuses from Zachman's framework: *what*, *how*, *who*, *why*. The *where*-focus is usually only implicitly present in, for instance, relationships between elements and in networks. Often, an explicit metamodel concept for expressing enterprise elements according to this focus is missing. The *when*-focus, if used, is mostly related to conditions or events that trigger processes. In this respect, it is closely related to and often included in the *how*-focus (e.g., event-driven process chains). Yet, Winter and Fischer (2007) argue that in EA, "business processes should not be decomposed further than to the subprocess level. Detailed process descriptions including specifications of activities and work steps are out of EA scope and should be maintained by using specialized business process modelling tools". This holistic overview function of EA is confirmed by other authors, such as (Lankhorst 2013; Jonkers et al. 2006).

The importance of these four focuses is confirmed by a large number of application cases performed with EKD. Stirna and Persson (2007) point out that, while EKD specifies six sub-models, it focuses predominantly on the goals model, business process model, concepts model, and actors and resources model. According to these authors, these sub-models correspond to the *why*, *how*, *what* and *who* questions, which are the four essential Zachman focuses that we identified. EKD sub-models thus represent a single aspect of the enterprise using concepts related to a particular focus.

Most of the time, all three layers (i.e. *business (B)*, *software (IS)*, *infrastructure (IT)*), or a *blend* of them are used. Most EA frameworks also emphasize the importance of analyzing the strategy space without worrying about the constraints of operations beforehand (*strategy–operations*).

Hence, these three things are defined as the essential EA dimensions to be supported by the CHOOSE metamodel: (1) the presence of the four focuses (*why*, *who*, *how*, *what*), (2) at least a *blend* of three architectural layers (*business*, *IS*, *IT*), and (3) analyzing the strategy space without considering any future constraints of operations (*strategy–operations*). This means that the CHOOSE metamodel needs to define concepts for each of the four essential focuses, that the metamodel concepts may represent elements related to business, IS and IT, and that CHOOSE models can be constructed for representing and analyzing enterprise strategy without being constrained by the current operations, so that strategy (needs) and operations (means) are not mixed.

5 Initial metamodel

We will first explain why the metamodel of the KAOS approach was chosen as a starting point for designing the CHOOSE metamodel. Next, the KAOS metamodel itself will briefly be presented. A more detailed description is provided in Appendix 1.

5.1 KAOS as a starting point

In addition to the EA approaches listed in Table 1, we also investigated goal-oriented requirements engineering (GORE) approaches. The main reason for choosing KAOS as a starting point is that from the investigated EA and GORE approaches, only KAOS (Van Lamsweerde 2009) and EKD (Stirna and Persson 2007) are explicitly built around the four essential EA focuses. Furthermore, KAOS was preferred to EKD as its metamodel is formally defined, which helps provide precise definitions for the concepts in the CHOOSE metamodel. The KAOS metamodel also explicitly distinguishes between concepts related to strategy and concepts related to operations. On the other hand, it should be pointed out that KAOS is not an original EA approach, but rather a requirements engineering approach intended to model systems. Therefore, its selection as the initial metamodel for CHOOSE was not trivial and had to be based on well-reasoned considerations, as explained below.

Engelsman et al. (2011) wrote an interesting paper on the use of GORE in EA in order to deal with the problem that current EA frameworks offer little support for modelling the underlying motivation of EAs in terms of stakeholder concerns and the high-level goals addressing these concerns. Their work lay at the basis of the ArchiMate 2.0 standard for EA modelling that extended ArchiMate 1.0 with a motivational extension (The Open Group 2012). The need for (a simple version of) goal refinement in EA approaches was confirmed after tests in case studies (Engelsman and Wieringa 2012). Therefore, GORE approaches were also considered as candidates for the selection of the initial metamodel, apart from the EA approaches listed in Table 1.

Well-known GORE techniques are i* (Yu 1993) and KAOS (Dardenne et al. 1991; van Lamsweerde et al. 1991). KAOS is a requirements engineering approach for software-intensive systems within an organizational or physical environment (Van Lamsweerde 2009). It is important to stress that KAOS is primarily intended to model organizational or physical systems based on goals and requirements, rather than used to model software. However, since enterprises are regarded as systems within EA (Jonkers et al. 2006), they can also be modelled as systems with KAOS. Compared to i*, which is more focused on the early requirements engineering phase and the modelling of dependencies between actors (Engelsman et al. 2011), KAOS has an important advantage since it makes it possible to make a broader overview of a system within its environment.

The ultimate choice for KAOS was, however, based on its great fit with the essential EA dimensions that we identified after analyzing important EA frameworks (see Section 4). First of all, its metamodel is based on four viewpoints that provide a one-to-one mapping with the four essential EA focuses. Second, KAOS models systems that can be composed of business (or real-world), software, data and technology components, so a blend of the three architectural layers can be used. Third, since KAOS is a GORE approach, it provides a means to analyze the strategy space without anticipating any constraints of operations. In GORE, abstract higher-level goals are gradually refined to more concrete lower-level goals, which are used to specify requirements for systems (Anton 1996; Anton et al. 1994; Dardenne et al. 1993). These goals, which are part of the why-focus, are then linked to operations, which are part of the how-focus, in order to maintain traceability (Mostow 1985). Engelsman et al. (2011) state that a company is a good example of a system and goals can be a good basis for modelling the motivational dimension of a company. Other research concludes that business goals form an integral part of enterprise models (Boman et al. 1997; Loucopoulos and Kavakli 1995).

A final motivation for choosing KAOS is that its metamodel is well elaborated after more than 20 years of research, and is hence a good starting point to reuse existing knowledge.

5.2 KAOS metamodel

The KAOS metamodel consists of four main viewpoints that define different sub-models (Fig. 5): goal, agent, operation, and object. These viewpoints are mapped onto the four essential EA focuses of *why*, *who*, *how* and *what*:

- Goal viewpoint (why-focus), where goals are refined and justified until a goal hierarchy has been put together for tackling a particular problem.
- Agent viewpoint (who-focus), in which agents are assigned to the goals they are responsible for.
- Operation viewpoint (how-focus), which defines various behaviours that the agents need to fulfil their requirements.
- Object viewpoint (what-focus), which is used to define and document the objects (i.e. entities, agents, and associations).

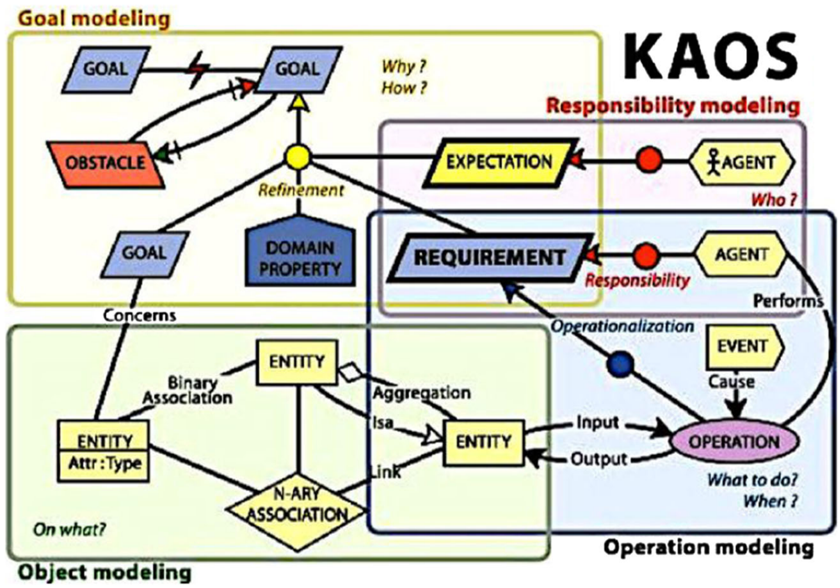

Fig. 5 A simplified overview of the KAOS metamodel (from (Respect-IT 2007))

There is an additional viewpoint (not shown in Fig. 5), which completes the static representation of system functionalities by capturing the required system dynamics. This behaviour viewpoint defines sub-models that can be represented using UML sequence diagrams and state diagrams. The concepts used in these sub-models are most closely related to the *when* Zachman focus, but this is not among the essential focuses of EA frameworks that we identified. Hence, it is clear that the behaviour viewpoint is not essential for EA modelling and can therefore be left out of the initial metamodel.

In the remainder of the paper, concepts from the goal, agent, operation, and object viewpoints will further be coloured in yellow, red, purple, and green, respectively. Definitions can be found in Appendix 1 and will be provided in the remainder of the paper when relevant to the discussion regarding the changes made during the action research and case studies (see next section).

6 From KAOS to CHOOSE

The most important change to transform the KAOS metamodel into the CHOOSE metamodel entailed deleting the elements that were not further used after the feasibility test of the full KAOS metamodel in the SME (Bernaert and Poels 2011) and were not asked for in the following rounds. As Moody (2003) mentioned, adoption is related to both effectiveness (i.e. benefits) and efficiency (i.e. costs). In order to develop CHOOSE, we first focused on efficiency and started with the essential part of an EA approach. During the action research in the SME, we then found out which parts had to be added for which the increase in effectiveness (i.e. increase in benefits) was larger than the decrease in efficiency (i.e. increase in costs).

Firstly, it is important to note that only two meta-attributes are mandatory for any meta-concept of all viewpoints in KAOS: *Name* and *Def*. These meta-attributes are also the attributes of the four central CHOOSE concepts; all other KAOS meta-attributes were omitted. *Def* was changed into a less formal *Description* attribute. This attribute has to be comprehensive and precise, yet also needs to provide a clear, though informally stated, description in natural language.

Secondly, other parts were omitted, changed, or added in each viewpoint and will be discussed for each viewpoint in the next sections. As mentioned before, after the action research was set up in the SME, five more case studies in SMEs with different characteristics were performed. Therefore, sometimes a particular change was triggered by a problem experienced with the use of CHOOSE in a case study company. If a similar problem was noticed in the action research SME, the solution chosen for the case study company was also evaluated in the action research SME, and after a positive evaluation this solution was then also incorporated into the metamodel.

In Appendix 1, a visual overview is given of the transformation of the KAOS metamodel into the CHOOSE metamodel (Figs. 11, 12, and 13). It is important to note that the following discussion is based on the complete KAOS metamodel (Fig. 11 in Appendix 1) and not on the simplified overview presented in the previous section (Fig. 5). In the following sections, changes to the different viewpoints will be discussed.

6.1 Goal viewpoint

The *Goal* concept was retained together with the concept of *Refinement*. The attribute of *Refinement* became *Id*, in order to enable the SME to distinguish between alternative *Refinements*. The distinction between *BehaviouralGoals* and

SoftGoals was omitted, because the SME was not interested in qualitative (Mylopoulos et al. 1992) nor quantitative (Letier and van Lamsweerde 2004) analyses, because of time constraints.

The same holds for the *Obstacle* analysis. This part was left out because it was never used as such. In the SME, *Obstacles* were implicitly addressed by naming the *Goals* according to the problem they aim to resolve (e.g., "decrease out of stock situations") instead of using *Obstacles* (e.g., "out of stock situation") and then *Resolving* them by means of a *Goal*. However, the SME wanted to model conflicting *Goals* of different stakeholders of the company in order to resolve these conflicts. The *Conflict* relationship between *Goals* was thus retained and explicitly represented by a relationship in the CHOOSE metamodel.

DomDescript was never used as it corresponds in KAOS to physical laws that cannot be broken. This aspect relates to formulating business rules in the context of an enterprise as part of an EA model. For example, an SME can express that a specific bank account (*Object*) can only be *Controlled* by maximum three *Human Actors*. As no business rules have been expressed so far in the SME, this concept of business rules is not yet explicitly represented in the CHOOSE metamodel. Nevertheless, the tool support we are developing for CHOOSE (Bernaert et al. 2013a; Dumeez et al. 2013; Ingelbeen et al. 2013) does make it possible to check such rules, by means of queries of the model level.

Finally, there is no longer an explicit distinction between *Expectations* and *Requirements* in CHOOSE. Since *Actors* can be different types, these types of *Goals* can simply be queried from the CHOOSE model to determine which *Goals* are from a specific type of *Actor*.

6.2 Agent viewpoint

The *Agent* concept was renamed into *Actor*, so that it would be more consistent with the terminology used in most EA frameworks (Table 1). The distinction between *SoftwareToBeAgent* and *EnvironmentAgent* was turned into a distinction between *Human Actor*, *Role*, *Software Actor*, and *Device* (hardware and equipment), which were only implicitly present in the KAOS metamodel via the optional *Category* attribute of an *Agent*. This change was not initiated by the SME, but rather by the need to adapt KAOS so that it would support the EA essential dimensions better (see Section 4.2) and be able to model a blend of the three architectural layers (business, IS, IT). The SME used the distinction between these types of *Actors* sometimes but not all the time, in order to speed up the modelling task. As a consequence, the specialization became optional in CHOOSE.

The SME experienced problems linking *Operations* to functions, for which a solution had to be found. Sometimes, functions appeared to switch between *Human Actors*, depending on the availability of the actors themselves, as well as their available time. The use of *Roles* and *Human Actors* that *Perform Roles* is briefly mentioned in KAOS, but not explicitly present in its metamodel. Still, as this is widely supported by EA frameworks (see Table 1), it was explicitly added to the CHOOSE metamodel so that this issue could be addressed.

The reflexive *Supervision* relationship was added between *Human Actors* because the SME immediately became aware of the need to make organizational charts. A many-to-many *Supervision* relationship was chosen to also enable matrix organizational structures in which a *supervisee* can have more than one *supervisor*.

Another reflexive relationship between *Actors*, *Aggregation*, was initially not included in the metamodel. However, one of the SMEs in which we performed case study research (to design the CHOOSE method, see Section 3) had 37 employees and the metamodel did not allow us to group *Actors* into departments or other categories, because such units are neither *Human Actors* nor *Roles*. In order to be able to group *Actors* according to different levels of granularity (e.g., business unit or department), which is also common in EA frameworks (Table 1), the reflexive *Aggregation* relationship was thus added again. However, the SME did not express the need to make a further specialization of *Actor* in department or business unit. Therefore, the specialization of *Actor* in its subtypes is not covering (incomplete), since an *Actor* can be something other than a *Human Actor*, a *Role*, a *Software Actor*, or a *Device*. The problem of not being able to group *Actors* was initially not brought to the attention in the action research SME, because this is an SME with only six employees. When the SME discovered in the second additional action research cycle that it could model its organizational chart more precisely, it fully supported this change in the metamodel.

At first, only one type of relationship was retained between *Goals* and *Actors* (i.e. *Assignment*). However, this soon became insufficient, because the relationship was used to assign *Actors* to *Goals* (as executing *Actor*) for lower-level *Goals*, but was also incorrectly used to express that an *Actor* "wanted" a *Goal* to be fulfilled for higher-level *Goals*. Therefore, the *Wish* relationship between *Actors* and *Goals*, only implicitly present in the KAOS metamodel as an attribute of *Agent*, was made explicit in the CHOOSE metamodel as a relationship. This was usually on a higher *Goal* level than the *Assignment* relationship between both.

The *Assignment* relationship, however, has a different meaning than the relationship in KAOS. In KAOS, an *Assignment* relationship makes it possible to *OR-Assign* different *Agents* to the same *Goal*, while only one *Agent* can be made *Responsible* of that *Goal*. In CHOOSE, *Actors* have an *Assignment* relationship with a *Goal* if they have been instructed to achieve that *Goal* (i.e. they are IOR-assigned).

This enabled the SME to assign multiple *Actors* to the same *Goal*, which was more in line with the SME's business reality. Whether or not the *Actors* were responsible for that *Goal* at a specific time, there were more ad-hoc decisions and there was no need for this to be expressed in the metamodel. The SME also *Assigned* some *Actors* to non-*LeafGoals* so that the model would reflect reality more clearly. Yet, this is not possible in the original KAOS metamodel. This is a subtle, yet important difference between KAOS and CHOOSE: in KAOS, *Goals* have to be *Refined* until they can be under the *Responsibility* of just one *Agent*. These *LeafGoals* can then be *Operationalized* by one or more *Operations*. The *Operations* also have to enable *Performance* by just one *Agent*. In contrast, in CHOOSE, *Goals* at any level can be assigned to *Actors* and can be *Operationalized* by *Operations* that can be *Performed* by more than one *Actor*. This clearly reflects the real-life organizational levels that can exist in a company. Nevertheless, some consistency problems still occurred in the SME due to this adaptation, for which additional OCL constraints (Appendix 2 Table 3: constraints 5–6, 21–22) were added.

At the start, only one relationship was retained between *Actor* and *Object* to express that an *Object* belonged to an *Actor*. However, this relationship could be more correctly modelled with an *Association* between an *Actor* and *Entity* in the object viewpoint if *Actor* was seen as a subtype of *Object*. The relationship between *Actor* and *Object* was therefore omitted and *Actor* was kept as a subtype of *Object* and could be used in the object viewpoint by the SME. Thus, the CHOOSE metamodel did not contain any extra relationship between *Actor* and *Object* anymore. One of the case study SMEs did a lot of administrative work and some discussions arose based on read and write rights of documents. This problem did not occur in the action research SME at first. When the EA model of the action research SME became more complete, it did become an issue as the action research SME also wanted to express the confidentiality of financial data. For example, it had to be decided who could see a particular bank account and who could make payments. In order to solve this problem, some options were considered and most often, a distinction between creating (if the object is newly created), transforming (if it is changed), and using (if it is only used and not modified) was found to be of importance. The SME did not make a distinction between creating and transforming. Therefore, the *Monitoring* and *Control* relationships from KAOS were added again, but instead of linking them to *Associations* and *Attributes,* we provided a direct link from *Actors* to *Objects* (Fig. 11). This was a logical step, because the SME did not specify any additional *Attributes* for *Objects* and because *Associations* are still subtypes of *Objects*. As a result, if an *Association* has to be explicitly *Monitored* or *Controlled*, the *Association* can be objectified, and *Actors* in CHOOSE can thus *Monitor* and/or *Control* an *Object*.

Finally, *Dependencies* between *Actors* can be queried from the CHOOSE model, and were therefore omitted from the metamodel. The assumption behind this is that if *Actors* are *Assigned* to the same *Goal*, or if they have to *Perform* the same *Operation*, they are dependent on each other.

6.3 Operation viewpoint

The operation viewpoint differs significantly between KAOS and CHOOSE. As mentioned before, an *Operation* in KAOS can only have a *Performance* relationship with exactly one *Agent*. However, when more *Operations* were added to the SME's EA model during the action research, there was no clear overview anymore. A solution to this problem was found by examining how ARIS (Scheer 2000) and BPMN (OMG 2011a) structure processes. A reflexive *Includes* relationship was added to enable the SME to make *Operations* part of other(s) in order to make it possible to create a structured *Operation* overview (sometimes called a map or landscape). Some constraints (Appendix 2 Table 3: constraints 9, 11, 22, 25) were adapted or added to maintain consistency.

It has already been pointed out that process modelling should not be included in EA. The SME, for its part, did not feel the need to make any process descriptions either. Still, some SMEs are likely to be confronted with this need for process modelling if standardization becomes more important (Ross et al. 2006). To make sure that they have some kind of EA overview of *Processes* at their disposal, a *Process* overview is included, while detailed process modelling is left out of the CHOOSE metamodel. However, process modelling descriptions can easily be linked to this *Process* overview of CHOOSE (e.g., with attachments in the software tool), which has as an advantage that the choice of process modelling language (e.g., BPMN, EPC, UML activity) can be made based on the SME's preferences, without this affecting the CHOOSE metamodel.

The name *Operation* was retained, since there is a clear distinction in a business context between a *Process* (Weske 2012) and a *Project* (Kerzner 2013), which was confirmed by the action research SME. A *Process* will typically be performed multiple times, while a *Project* is performed only once and has time, budget, and other constraints. In CHOOSE, an *Operation* can therefore either be a *Process* or a *Project*. The SME had some *Projects* that could be quite disruptive for their business and wanted to treat these *Projects* differently than the *Processes* (e.g., some milestones were formulated for these *Projects)*. Therefore, the SME sometimes, but not always, wanted to distinguish between a *Process* and *Project*. That is why the specialization is optional: if not further specified, the SME is not interested in making the difference between a *Process* and *Project*.

The *Performance* relationship helped the SME to make a load analysis of all *Operations* linked to an *Actor*. However, this load analysis needed some corrections, because for example sometimes the *Actor* would only be informed about the *Operation* once in a while, which was less time-consuming than being held responsible for it. For this problem, different solutions exist (e.g., RACI, RASCI, CAIRO). In order to be able to use a RACI (Responsible, Accountable, Consulted, and Informed) labelling of the *Performance* relationships, an association class including the attribute *Type* was added to the *Performance* relationship. This RACI chart is also used, for example, by the IT governance reference framework COBIT to define responsibilities (ISACA 2012). Working with a generic *Type* attribute instead of a specific set of labels makes the modelling effort much more flexible, so that the SME may choose another responsibility assignment matrix. The load analysis during the action research would then be more accurate, based on the different *Types* of *Performance* relationships between *Actors* and *Operations*.

As the SME linked *Operations* with *Goals*—not only *LeafGoals*—at different levels, the *Operationalization* link needed to be adapted. A constraint (Appendix 2 Table 3: constraint 9) was added to maintain consistency. However, it is best to delay the *Operationalization* of a *Goal* as long as possible, to make sure that the constraints of operations are still disregarded during the analysis of the strategy space, which is an essential element in EA frameworks (Table 1). This aspect could be further investigated with regard to the future development of the CHOOSE method, but is beyond the scope of the present paper.

The *Input* and *Output* links between *Operations* and *Objects* were retained for the same reasons as the *Monitor* and *Control* links between *Actors* and *Objects*: to give the SME the possibility to express which *Objects* are the *Input* (i.e. using) of an *Operation* and which ones are *Output* (i.e. creating or transforming). These relationships were also directly linked to *Objects* instead of *Associations* and *Attributes* (Fig. 11). An *Object* that is the *Input* of an *Operation* was often a resource in the SME, while an *Object* that is *Output* was often a product of the SME. However, as there are multiple exceptions, this was not included in the metamodel. For example, for more administrative *Operations* documents were sometimes needed as *Input*, in which case the *Output* would be an invoice, for example.

6.4 Object viewpoint

The object viewpoint was less used than the other three viewpoints. The SME only needed to model *Objects* and the *Associations* between them. Only if more specificity was required, was an *Object* further broken down into *Entity*, *Actor* or *Association*. As a consequence, this specialization could be optional. There was no need to include extra *Attributes*, *DomDescripts* or *DomInits* either, because the *Description* attribute of an *Object* was sufficiently specific and the CHOOSE metamodel is not focused on precise system specification like KAOS. The *Event* concept, referred to in this object viewpoint but part of the behavioural viewpoint in KAOS, was also omitted. This can again be accommodated by process modelling languages and state diagrams.

In the SME of the action research, only *Associations* that *Link* two *Objects* were used. To enhance semantic clarity, *Associations* between more than two *Objects* were disregarded. The SME did not use any specific *Attributes* for *Objects*, nor did it define *ApplicationSpecific Associations*. Instead, the two attributes of an *Association*—*Name* and *Description* because an *Association* is a subtype of an *Object*—were sufficient to clearly describe the different *Associations*.

Aggregation and *Specialization* were first hardly used. However, when the CEO of the SME tried to specify a bill of materials (for example, by asking himself which car parts could be replaced by the SME), the *Aggregation* relationship offered a good solution (Hegge and Wortmann 1991). The same happened when the CEO tried to get a product overview (for example, by asking himself how the SME sorts the warehouse according to tyre type), the *Specialization* relationship was a good solution (Eriksson and Penker 2000). A good method to explain these options can also be recommended. Preferably, this explanation does not use the terms *Aggregation* and *Specialization*, which were unknown to the CEO in this particular case. The choice to specify an *Association* as either an *Aggregation* or *Specialization* is an *Optional*, disjoint (*Or*) choice (OMG 2011c). This means that an *Association* does not have to be further specified in CHOOSE if the SME does not need it, but if it is, it can only be one of the subtypes.

The *Concern* relationship between *Goal* and *Object* was retained, although the SME in fact did not use it frequently. Further research in more SMEs could give more insight into the use of this relationship, for example in order to detect consistency conflicts (Appendix 2 Table 3: constraint 12).

An *Object* can be *Input* and *Output* of the same *Operation*, as its *State* can be changed by an *Operation* (for example, the customer file that was updated in the SME). However, there was no need to explicitly model these *States*. This is in line with the choice to also exclude process modelling from the CHOOSE metamodel, because this can also be achieved by process modelling languages and state diagrams.

7 CHOOSE metamodel

7.1 Complete CHOOSE metamodel

The CHOOSE metamodel was robust after the third action research cycle and no further changes needed to be made during the fourth cycle. According to the Object Management Group (OMG) (2012b, 2013) standards, the metamodel presented in this research is a computation independent model (CIM) at M2-level. Since it is described as a unified modelling language (UML) class diagram (OMG 2011b, c), this model can also serve as a platform independent model (PIM) for software tool support development (Bernaert et al. 2013a; Dumeez et al. 2013; Ingelbeen et al. 2013). The models made with this metamodel, and thus instantiating it, will be at M1-level and will be EA models for the specific SME being modelled.

Figure 6 shows this final CHOOSE metamodel, including all optional parts. *Actor* is represented twice for clarity's sake, but refers to the same concept.

7.1.1 CHOOSE goal viewpoint

Goal is the central concept in the goal viewpoint and has the attributes *Name* and *Description*. A *Goal* can have a *Conflict* relationship with zero or more other *Goals*.

An *OR-Ref* links one higher-level *Goal* with a *Refinement*. If different *Refinements* are linked via *OR-Ref* links to the same higher-level *Goal*, this means the *Goal* is *OR-Refined* several times. If only one *Refinement* is linked via an *OR-Ref* link to a higher-level *Goal*, this means the *Goal* can only be refined in one possible way. Each *Refinement* is then linked via *AND-Ref* links with one or more lower-level *Goals*. This implies each alternative *Refinement* of a higher-level *Goal* is linked via *AND-Ref* links with one or more lower-level *Goals*, which all have to be fulfilled in order to meet the higher-level *Goal* with which this *Refinement* is linked via an *OR-Ref* link. A special case is when a higher-level *Goal* is *OR-Refined* by just one *Refinement* and this *Refinement* is *AND-Refined* by just one lower-level *Goal*. In this case, the higher-level *Goal* is simply refined by the lower-level *Goal*. If a *Refinement* thus only has one upper *Goal* (*OR-Ref*) and one lower *Goal* (*AND-Ref*), it can be seen as a single refinement of a higher-level *Goal* in a lower-level *Goal*. A *Goal* does not have to have a link with a higher *Refinement* (reached through an *AND-Ref* link) if it is one of the highest-level *Goals* in the *Goal* hierarchy, and it does not have to have a lower *Refinement* (reached through an *OR-Ref* link) if it is one of the lowest-level *Goals* in the hierarchy (i.e. a leaf *Goal*).

A *Goal* can have a *Wish* or *Assignment* relationship with zero or more *Actors*, can be *Operationalized* by zero or more *Operations*, and can have a *Concern* relationship with zero or more *Objects*.

7.1.2 CHOOSE actor viewpoint

Actor is the central concept in the actor viewpoint and has the attributes *Name* and *Description*. An *Actor* can be an aggregation of zero or more other *Actors* and can be part of zero or more *Actors* via a *Division* relationship. However, if an *Actor* is a *Human Actor*, it cannot be an aggregation of other *Actors* (Appendix 2 Table 3: constraint 14) and other relevant constraints to limit the *Aggregation* of different *Actor* types are added (Appendix 2 Table 3: constraints 15–19). An *Actor* can, but does not have to be (i.e. it is *Optional*) *Specialized* in either (disjoint *Or*) a *Human Actor*, a *Role*, a *Software Actor*, or a *Device*. A *Human Actor* can be *Supervised* by zero or more *supervisors*, or can *Supervise* zero or more *supervisees*. A *Human Actor* can *Perform* zero or more *Roles*, while a *Role* can be *Performed* by zero or more *Human Actors*.

Actors can have a *Wish* (only unspecialized *Actors* or *Human Actors*, see Appendix 2 Table 3: constraint 4) or *Assignment* relationship with zero or more *Goals*, they can have a *Performance* relationship (some kind of RACI or other *Type*) with zero or more *Operations*, and they can *Monitor* or *Control* zero or more *Objects*.

7.1.3 CHOOSE operation viewpoint

Operation is the central concept in the operation viewpoint and has the attributes *Name* and *Description*. An *Operation* can be *Included* in zero or more other *superOperations* and can *Include* zero or more *subOperations*. An *Operation* can, but does not have to be (i.e. it is *Optional*) *Specialized* in either (disjoint *Or*) a *Process* or a *Project*.

An *Operation* can *Operationalize* zero or more *Goals*, can have a *Performance* relationship (some kind of RACI or other *Type*) with zero or more *Actors*, and can have zero or more *Objects* as *Input* or *Output*.

7.1.4 CHOOSE object viewpoint

Object is the central concept in the object viewpoint and has the attributes *Name* and *Description*. An *Object* can, but does not have to be (i.e. it is *Optional*) *Specialized* in either (disjoint *Or*) an *Entity*, *Actor*, or *Association*. An *Association Links* two *Objects*, while an *Object* can have zero or more *Associations* with one other *Object*. An *Association* inherits the attributes *Name* and *Description*, which are also visualized for clarity's sake, and a *Link* has the optional attributes *Role* and *Multiplicity*. An *Association* can, but does not have to be (i.e. it is *Optional*) *Specialized* in either (disjoint *Or*) an *Aggregation* or a *Specialization*.

An *Object* can have a *Concern* relationship with zero or more *Goals*, can be *Monitored* or *Controlled* by zero or more *Actors*, and can be *Input* or *Output* for zero or more *Operations*.

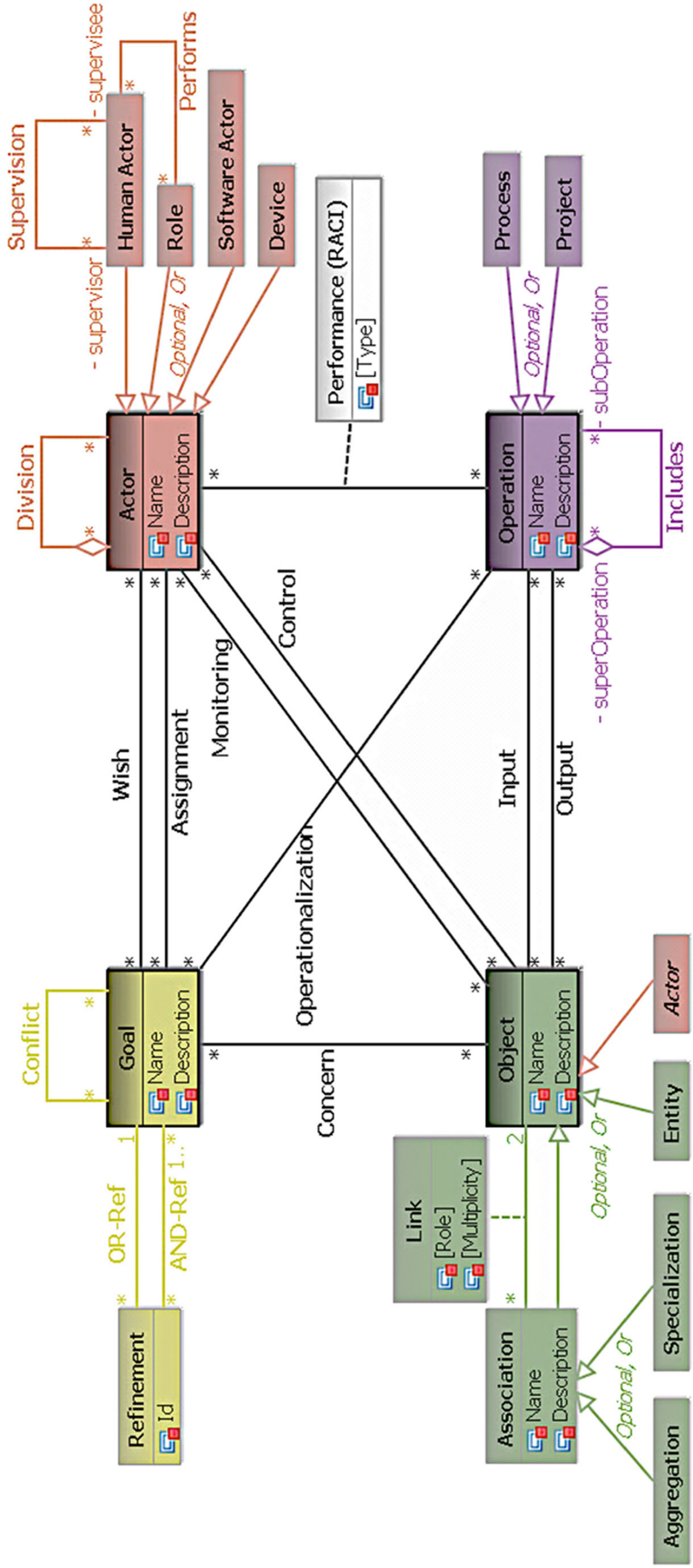

Fig. 6 CHOOSE metamodel

7.2 Core part of the CHOOSE metamodel

Figure 7 shows the core part of the CHOOSE metamodel, which only includes the minimum set of concepts and relationships of the CHOOSE metamodel required to model an SME's EA. The optional specializations and attributes are left out of this core representation of the CHOOSE metamodel, since they were only useful in some scenarios for the action research SME and do not belong to the essential EA dimensions.

These minimal parts of the CHOOSE metamodel can be used by SMEs to quickly and easily create an EA model. If needed, however, an *Actor* could then for example later on be specialized as a *Human Actor* in order to use the *Supervision* relationship of the complete CHOOSE metamodel. For instance, if the SME treats a *Process* differently than a *Project*, it could use the extensions of the complete metamodel and specialize the *Operations* into *Processes* and *Projects*. This core part of the CHOOSE metamodel represents the bare minimum, while still conforming to the EA essentials from Section 4.2.

7.3 CHOOSE definitions using SBVR

In order to decrease misunderstandings, formal definitions are provided to contribute to the unambiguous definition of the CHOOSE concepts. Not all concept definitions in the CHOOSE metamodel could be retained from KAOS, since KAOS is used for system specification, while CHOOSE is used to make EA models of SMEs. Hence, the context in which the concepts are used is different.

When the KAOS definitions had to be adapted, the first choice was to relate the definitions, if possible, to ArchiMate definitions for two reasons. First, ArchiMate has been adopted by The Open Group (2012) as a standard and second, in future research, CHOOSE will be mapped onto ArchiMate to make bidirectional translation possible (Roose et al. 2013). For the organizational chart, definitions are linked to OMG's (2009) organization structure metamodel (OSM), since this metamodel is a widely used standard. The *Project* definition is adapted from the project management body of knowledge (Project Management Institute 2013) and the *Process* definition from OMG's (2011a) BPMN and ArchiMate (The Open Group 2012). Finally, the concepts derived from KAOS that cannot be related to a relevant EA definition are taken from the original KAOS definition (Van Lamsweerde 2009).

In Table 2, the definitions of the entities and relationships of CHOOSE are explained by means of SBVR (OMG 2008). Only the business vocabulary part of SBVR is used, as the rules are expressed in OCL (Section 7.4).

It is important to note that *Aggregation* and *Specialization* of *Objects* cannot directly be used for further model-driven development of systems, since additional information needs to be added by IS experts, like for example whether the *Specialization* between *Objects* is total or not. CHOOSE is not intended to be directly used for implementation (e.g., to build an enterprise database for the SME), but rather a means to provide

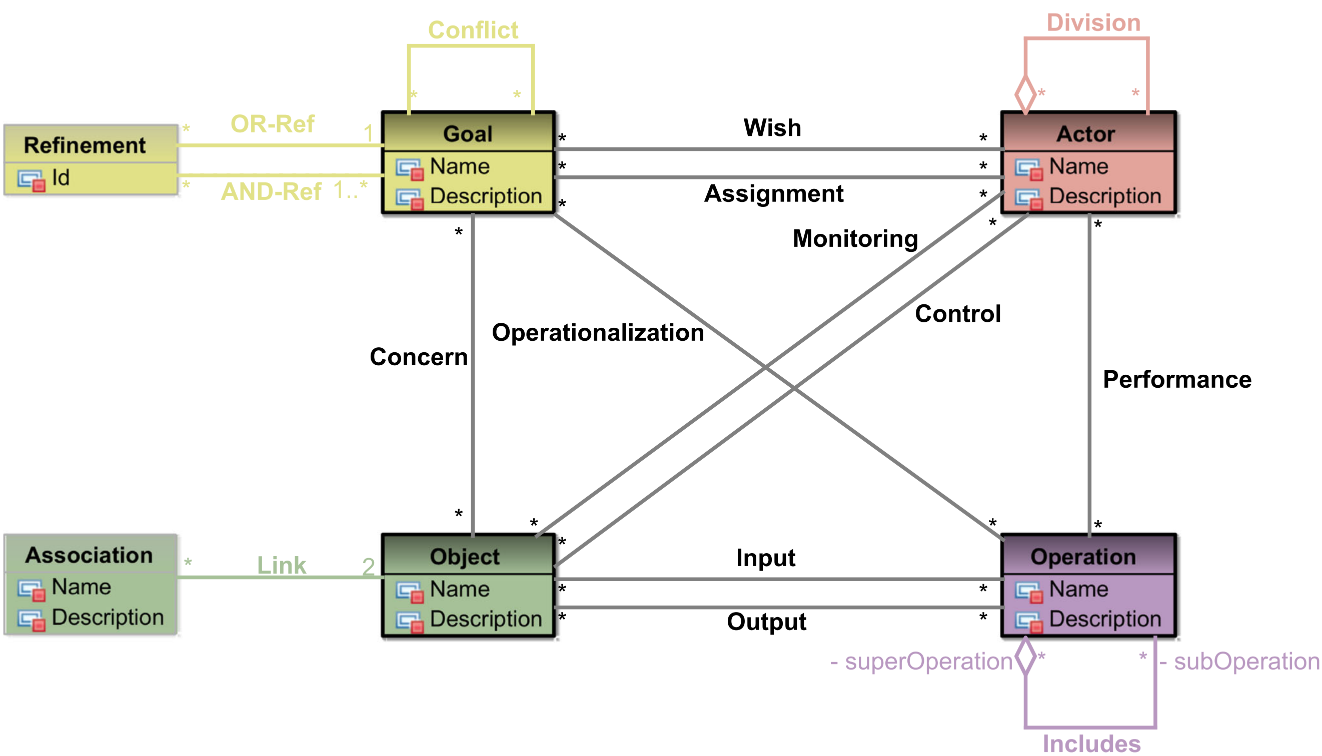

Fig. 7 Core part of the CHOOSE metamodel

Table 2 CHOOSE entities and relationships defined with SBVR

CONCEPT	DEFINITION	SOURCE
Object Type		
Goal	An end state that an actor wishes to achieve and that is to be brought about or sustained through appropriate operations.	Goal (The Open Group 2012; OMG 2010)
Refinement	Groups lower-level goals that all have to be fulfilled in order to fulfil a higher-level goal. Different refinements for one higher-level goal express different alternatives.	Refinement (Van Lamsweerde 2009)
Actor	An organizational entity that is capable of performing operations.	Business actor (The Open Group 2012)
Human Actor	A human being who is capable of performing operations.	Human actor (The Open Group 2012)
Role	The responsibility for the performance of specific operations, to which a human actor can be assigned who performs the role.	Business role (The Open Group 2012)
Software Actor	A software system or part of a software system that encapsulates its behaviour and data to perform operations.	Business actor + Application component (The Open Group 2012)
Device	A hardware resource or physical equipment that is capable of performing operations.	Business actor + Device (The Open Group 2012)
Operation	Internal behaviour that needs objects as input and produces objects as output, in order to operationalize goals. It can be a process or project.	Adapted from Operation (Van Lamsweerde 2009)
Process	A behaviour element that groups behaviour based on an ordering of activities with the objective of carrying out work. It is intended to produce a defined set of products or business services.	Process (OMG 2011a) + Business process (The Open Group 2012)
Project	A temporary endeavour undertaken to create a unique product, service or result.	Project (Project Management Institute 2013)
Object	A passive element that has relevance from a business, information, or technological perspective. It corresponds to a real world counterpart that may or may not be physical.	Business object (The Open Group 2012) + Object (Snoeck et al. 1999)
Entity	An autonomous and passive object.	Entity (Van Lamsweerde 2009)
Fact Type		
OR-Ref	Refines a higher-level goal in alternative refinements.	OR-refinement (Van Lamsweerde 2009)
AND-Ref	Expresses that an alternative refinement of a higher-level goal can be satisfied by satisfying all its subgoals.	AND-refinement (Van Lamsweerde 2009)
Conflict	Interconnects goals to capture potential conflicts among them.	Conflict (Van Lamsweerde 2009)
Wish	Captures the fact that an actor would like a goal to be achieved.	Wish (Van Lamsweerde 2009)
Assignment	An actor is assigned to a goal if it is required to restrict its behaviour so as to achieve the goal.	Responsible (Van Lamsweerde 2009)
Operationalization	Refers to the process of mapping goals (ends) to operations (means) realizing them.	Operationalization (Van Lamsweerde 2009) + Realization (The Open Group 2012)
Concern	Connects goals to the objects to which they refer.	Concern (Van Lamsweerde 2009)
Division	Indicates that an Actor groups a number of other Actors.	Aggregation (The Open Group 2012)
Supervision	A supervisee reports to a supervisor.	Supervises (OMG 2009)
Performs	Links roles with human actors that fulfil them.	Assignment (The Open Group 2012)
Performance (RACI)	Links operations with active elements (actors) that perform them or more specifically that are responsible, accountable, consulted, or informed.	Assignment (The Open Group 2012) + RACI (ISACA 2012)
Monitoring	An actor monitors an object if it can use the object, without changing it.	Monitoring (Van Lamsweerde 2009)
Control	An actor controls an object if it can create or transform the object.	Control (Van Lamsweerde 2009)
Includes	Groups suboperations in the superoperations of which they are part.	Aggregation (The Open Group 2012)
Input	Designates an object to which the operation applies.	Input (Van Lamsweerde 2009)
Output	Designates an object on which the operation acts.	Output (Van Lamsweerde 2009)
Association	Models a relationship between objects that is not covered by another, more specific relationship.	Association (The Open Group 2012)
Aggregation	Indicates that an object groups a number of other objects.	Aggregation (The Open Group 2012)
Specialization	Indicates that an object is a specialization of another object.	Specialization (The Open Group 2012)

an EA overview for the SME's CEO or managers. As such, it could be a starting point for further detailed elaboration and analysis.

7.4 CHOOSE constraints using OCL

Finally, the metamodel is completed by adding constraints (see Appendix 2: Table 3 for a full list of the constraints). These constraints are meta-constraints as they constrain metamodel components. They are to be determined at metamodel definition time, checked at model-building time when enough model elements are available in each view, and rechecked at model evolution time when the linked items are changed. Most rule-based checks can be fully automated through queries on a model database structured according to the metamodel, for instance in further software tool development efforts (Bernaert et al. 2013a; Dumeez et al. 2013; Ingelbeen et al. 2013). The main advantage of having constraints is that constraint violations drive models towards structural consistency. Further, since missing items are often revealed, these constraints address structural completeness as well (Paige et al. 2007). Next to these universal consistency rules, a model may also be further constrained by business-specific rules at M1-level.

Within the constraints, a distinction can be made between hard or soft ones, on the one hand, and intra-view or inter-view ones, on the other hand. Hard constraints (1, 4, 13–20, 25–26) refer to those that must never be violated, while soft constraints (2–3, 5–12, 21–24) can be seen as recommendations for the SME in order to arrive at a more balanced and complete enterprise model. The latter make it possible for the user to figure out what remains to be done at any step of the model-building method. The distinction between intra-view and inter-view constraints involves the extent to which the whole model is checked or not. Intra-view constraints (1–2, 13–20, 25–26) are related to only one of the four viewpoints of CHOOSE and are marked by the corresponding colour. They check the structural consistency, completeness, and correctness within just one of the views. Inter-view constraints (3–12, 21–24), in contrast, are related to at least two of the four viewpoints and are also marked by the corresponding colours. These constraints are not limited to just one viewpoint, but check the structural consistency, completeness, and correctness of the whole model. As mentioned before, this improves the cohesion of the four viewpoints and enhances the integration and traceability of the different domains of a company.

Some constraints (2–3, 5, 7–8, 10–12, 21, 23–24, 26) are based on earlier KAOS constraints (Van Lamsweerde 2009), but often required some alterations because of the changed metamodel, as mentioned earlier (e.g., *Actors* can be linked to more *Goals*, *Operations* can be performed by more *Actors* and can also be linked to non-leaf *Goals*). Some new constraints (1, 4, 6, 9, 13–20, 22, 25) had to be developed since the KAOS metamodel was adapted to form the basis of the CHOOSE metamodel and inconsistent or incomplete models were discovered during the action research.

The constraints are expressed using the object constraint language (OCL), a standard of OMG (2012a) that can easily be used with the other OMG standards UML and SBVR (Warmer and Kleppe 2003). In order for the constraints to be tested and validated on instantiations of the metamodel, a UML-based specification environment tool (USE) was used, which was developed to test OCL constraints on UML models (Gogolla et al. 2007). Of course, this was not presented to the SMEs, since this software tool is rather meant to support the CHOOSE metamodel development effort and is not adapted to the characteristics of SMEs and EA. In Table 3 of Appendix 2 the metamodel including a full list of all constraints is presented as the text file serving as input for the USE tool. An example of resolving a constraint violation is given in Fig. 10 of Section 8.1.

Although this set of constraints proved to be sufficient for developing the EA model of the action research SME, it can definitely be extended. A possible future area of research could involve other relevant constraints and queries, for example to assist in conflict management (van Lamsweerde et al. 2002) or reasoning about alternative options (Heyse et al. 2012; Mylopoulos et al. 1992).

7.5 Model viewpoints

It became clear during the action research that even though the CHOOSE metamodel contains few elements, the CHOOSE models became quite large, even in small SMEs (see Fig. 8).

Therefore, queries can be used on the model database to extract other model views and visualize these for dedicated analyses, in order to be able to keep an overview of the EA model. For instance, if an *Object* is *Output* of an *Operation* under the *Performance* of an *Actor* and *Input* for an *Operation* under the performance of another *Actor*, an implicit relationship exists between the *Actors* since they are dependent on each other. This provides a useful, direct view of mutual interfaces among *Actors*. Another example is load analysis, to see what *Operations* each *Actor* is *Performing*, or a RACI chart, if the *Performance Types* are according to RACI. Problematic situations can be spotted where a *Human Actor* appears overloaded.

A sufficient set of viewpoints will be further developed with the help of additional case studies.

8 CHOOSE metamodel evaluation

The evolving CHOOSE metamodel was evaluated through the different rounds of the action research programme. The results of this evaluation will be summarized in this section, and it will be determined whether the final metamodel

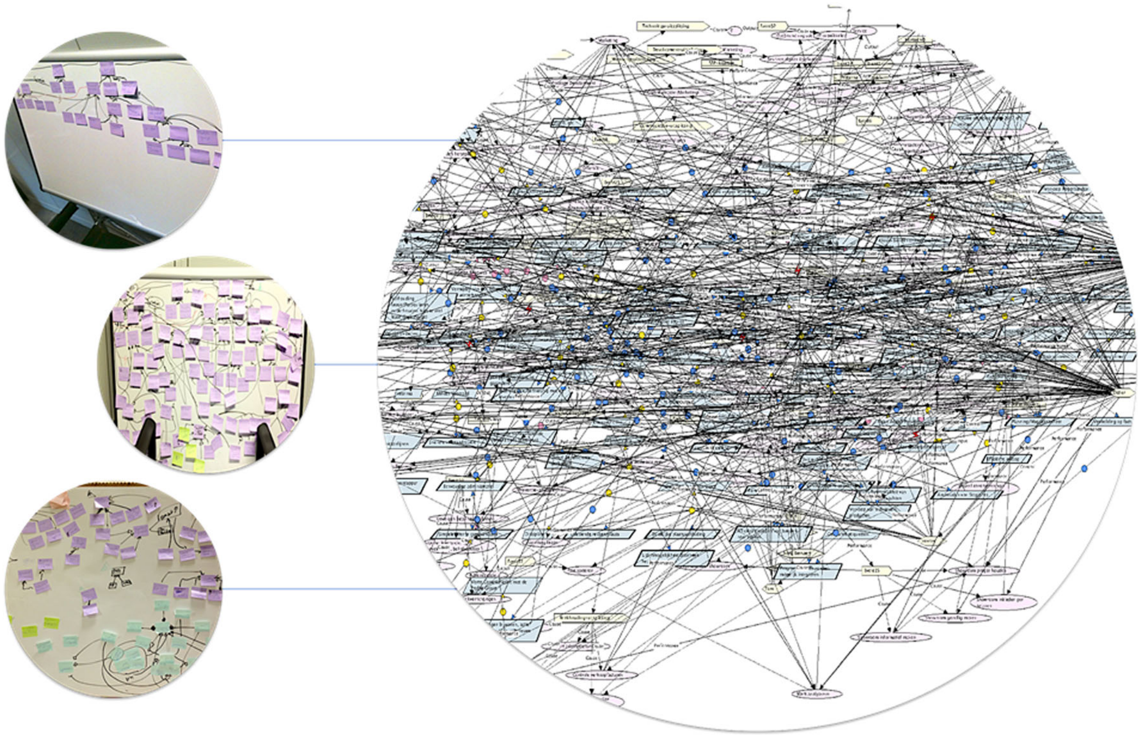

Fig. 8 The CHOOSE model of the action research SME (using post-its and afterwards inserting it in the Objectiver tool for KAOS) became quite large

supports the essential EA dimensions (Section 4.2) and meets the requirements for EA for SMEs (Section 2.2).

8.1 Action research evaluation and example

The action research effort demonstrated that CHOOSE enables the development and management of an EA model for SMEs. It made the CEO think about his SME, how things work, why things are done, who is involved in and responsible for what, what the conflicting goals of different stakeholders are, and how balanced decisions should be made between these conflicting goals. In this respect, one specific advantage was that the CEO of the action research SME became able to assess which operations could be executed by software instead of by the employees that executed them up to that moment. For example, because of some insights from the CHOOSE model, the CEO decided to purchase an extra module for the ERP system. This module allowed him to automatically link payments with the correct customer, an operation that he used to have to do himself and that was very time-consuming.

In general, it is safe to say that the CHOOSE model enabled a better control of the SME, with improved communication and interaction, by offering a holistic overview, in which elements are part of a bigger picture. The approach was primarily used in a top-down manner (i.e. from *Goals* to *Operations*), thus increasing the CEO's control of the SME. At the same time, though, CHOOSE also increased communication and interaction among employees and other stakeholders, as it was also used to discuss parts of the model with them. Although the terminology may not be clear to all users right now, this will definitely be remedied by the software tools we are developing (Bernaert et al. 2013a; Dumeez et al. 2013; Ingelbeen et al. 2013). A final advantage could be that employees may become more motivated if they know how their role is situated within the bigger picture of the whole SME. This was not yet visible in the SME, but longer-term evaluation will undoubtedly provide more insight into this type of benefits.

In order to illustrate most of CHOOSE's concepts and relationships, Fig. 9 shows an extract from the SME's CHOOSE model, modelled in the USE tool. In this example from the action research SME, the CEO wished to increase the customer base by increasing visibility in one of two possible ways. First, he could open a new store, but this conflicts with decreasing the costs, an objective of the bookkeeper. Second, he could improve the signage of the building and enhance online visibility. This second alternative was chosen. Since signage can be a pricy affair and thus conflicts with decreasing the cost, first online visibility was enhanced. In this particular SME, the CEO also performs the role of a marketing expert and is part of the SME's back office together with the bookkeeper, who is supervised by the CEO. As marketing expert, the CEO is assigned

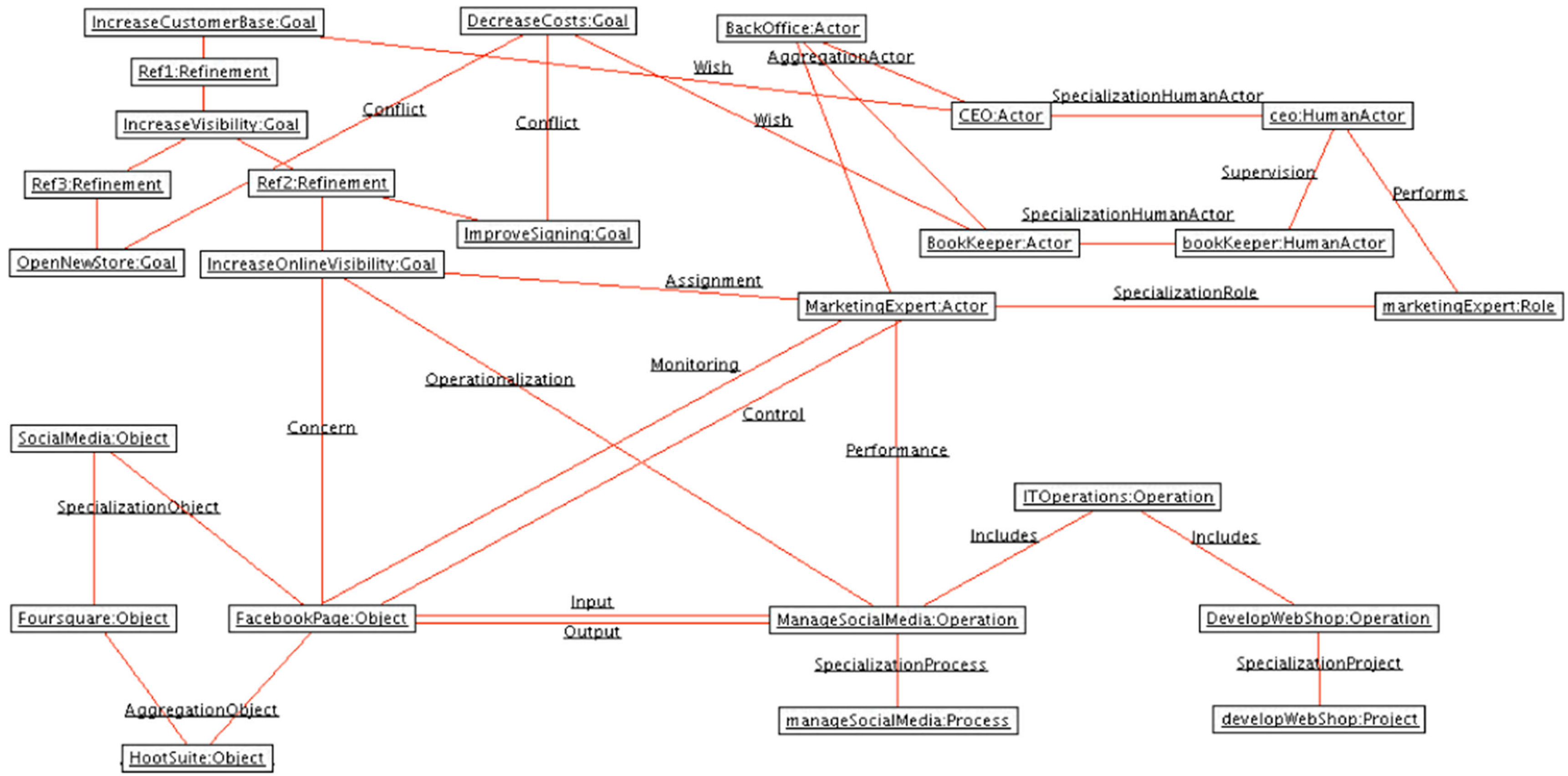

Fig. 9 Extract from the CHOOSE model from the action research SME

to the goal of increasing the online visibility. This is operationalized by managing the social media. More specifically, the SME's Facebook page will be managed and the marketing expert can see (i.e. *Monitor*) and even change (i.e. *Control*) this page. Managing social media is part of the IT operations in this SME, like for example also the project of the web shop development. The company also has a Foursquare page as a kind of social media, however, nothing is currently being done with this page. Both Facebook and Foursquare are part of the SME's Hootsuite account, in which different social media platforms can be managed.

Five OCL constraints are violated in this model extract (left part of Fig. 10), which could guide the SME to make the CHOOSE model more consistent and complete. For example, *ASSIGNMENTAGGREGATION* (constraint 5) failed because the *Actor* BackOffice has no *Assignment* relation with any of the higher-level *Goals* of IncreaseOnlineVisibility, which is *Assigned* to MarketingExpert, one of the *subActors* of BackOffice. This could be resolved by *Assigning* BackOffice to IncreaseCustomerBase (right part of Fig. 10).

8.2 Support of essential EA dimensions

The CHOOSE metamodel conforms to the essential dimensions of EA frameworks as identified after analysis of important EA frameworks in Section 4:

- The CHOOSE metamodel covers and integrates the four essential EA focuses: *why* through the goal viewpoint, *who* through the actor viewpoint, *how* through the operation viewpoint, and *what* through the object viewpoint. Relationships are defined to relate concepts from different viewpoints.
- The CHOOSE metamodel *blends* the three EA layers (*business*, *IS*, *IT*) by providing *Actors* for each layer (*Human Actor*/*Role*, *Software Actor*, *Device*) and enabling the other three viewpoints to be related to it. *Goals*, *Operations*, and *Objects* could also originate from the three different EA layers, as seen in the EA model of the SME during the action research. Yet, for this SME, no explicit specialization was needed.
- The CHOOSE metamodel provides a means to analyze the strategy space without worrying about any constraints of operations beforehand, as it separates *Goals* from *Operations* via *Operationalization* links.

8.3 Meeting the requirements for EA for SMEs

The CHOOSE metamodel conforms to the EA requirements listed in Section 2.2.1:

1. By providing a means to analyze the SME by using a metamodel, control was increased for the CEO. Constraints in OCL that are generally applicable are presented. SME-specific queries can be made on the EA model.
2. By conforming to the essential parts of EA frameworks, a holistic overview can be provided, but the SME is not obliged to make a global model. If necessary, the models can be made for one project at a time (Ross et al. 2006). The *when* and *where*-focus can be considered to be part of the operation viewpoint, which could be elaborated by business process modelling languages. The SME did not need separate *Operation* attributes for these two focuses,

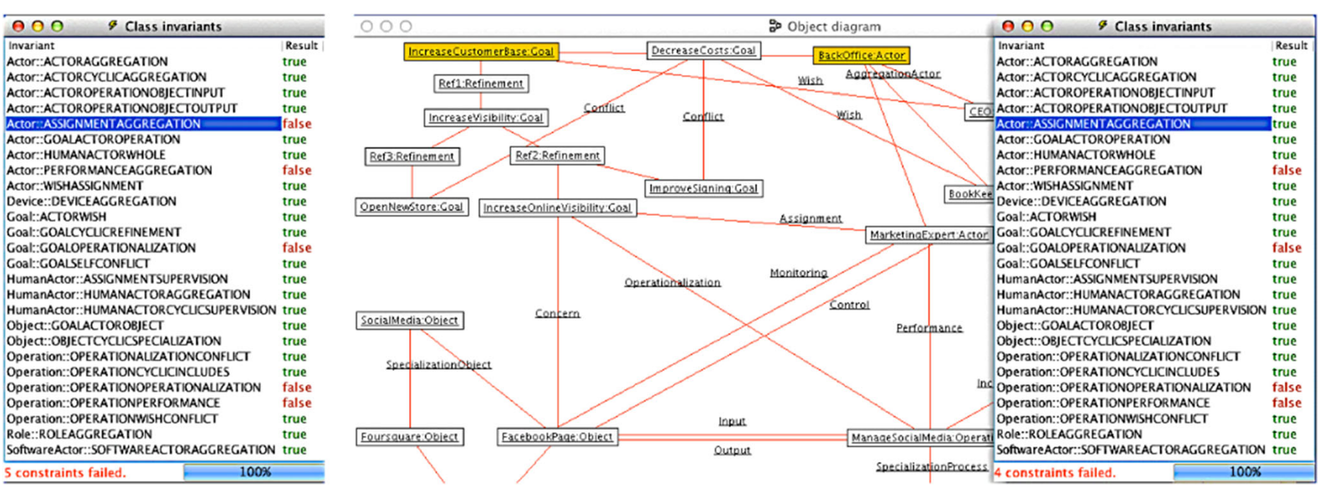

Fig. 10 OCL constraints failed (*left*) and one constraint resolved (*right*)

since the *Description* attribute was sufficient to describe details.
3. Since the CHOOSE metamodel is based on goal refinements, the requirement regarding objectives is fulfilled.
4. This requirement (fit for the target audience) was split up into SME-specific requirements (see further).
5. Since CHOOSE is based on the essential dimensions from EA frameworks used for modelling enterprises, it provides an enterprise overview.

As the fourth requirement of EA is related to SMEs as a target audience (and, more specifically, to the CEOs or managers of SMEs), the requirements for adoption and successful use of IT in SMEs can be discussed (Section 2.2.2):

4.1. To allow the CEO to work more efficiently, the CHOOSE metamodel is kept to the bare minimum (e.g., a comparison can be made between the number of metamodel elements and relationships in CHOOSE and ArchiMate). Nevertheless, a metamodel by itself did not appear to offer the CEO enough flexibility to work whenever and wherever he had the time for it. Further software tool support (i.e. research step 5 in Fig. 1) should developed to make this possible (Ernst et al. 2006). In the fourth round of the action research programme, a prototype CHOOSE software tool was installed in this SME.

4.2. To make the approach accessible to people with few IT or modelling skills, the metamodel is kept as simple as possible (including some optional parts that do not have to be used), with just four viewpoints that each contain only one central concept. The CEO was able to work with CHOOSE and is now also using the software tool. Still, a longer-term evaluation and further case studies are needed to improve the CHOOSE approach and software tool support.

4.3. Throughout the different rounds of the action research programme, the researchers guided the CEO in the development of the EA model. After the fourth round, the CEO started working with the software tool himself. The ultimate goal is to further develop the CHOOSE approach so that any need for external help is reduced to a minimum.

4.4. A process overview can be built with the operation viewpoint. Processes (or projects) can be elaborated by using a business process management approach (or project management approach) and linking this to the corresponding process (or project) in the CHOOSE model. In the SME of the action research, no processes were elaborated. As this could be the case in other SMEs, further research is still needed on how to easily link process models to the process overview (e.g., by providing attachment options in the software tool support).

4.5. The CEO was involved in developing the CHOOSE model, as he possessed the required knowledge to make an overview of the SME. The CHOOSE model is an instantiation of the CHOOSE metamodel that is developed and further refined throughout the action research cycles, based on the problems the CEO and the researchers encountered.

4.6. In terms of complexity, the number of metamodel concepts and relationships of CHOOSE is considerably lower than in other EA frameworks and kept to the bare minimum. The main benefits in the SME from the action research were threefold. First, because the EA was built from scratch, this offered considerable insight into the structure and inner workings of the SME. It was clear that the CEO became very enthusiastic after he had explicated his goals for the SME, because he experienced this entire process as a steep learning curve. Second, when managing the EA, the CHOOSE metamodel helped store decisions of meetings in one place (i.e. in the EA model). Third, CHOOSE provided the SME a platform for analysis and guided change, especially because of the built-in traceability by integrating four viewpoints into one metamodel. Among other things, it became possible to predefine analyses and enabled easy querying. In the SME, it was the OCL constraints that gave the most guidance. However, additional benefits are now becoming apparent while the SME is actually using the software tool. Some functions have already been programmed (e.g., Excel output, different viewpoints, querying) and others will mainly be developed in line with the feedback from the case study research that is conducted in other SMEs.

9 Conclusion

This paper presents the design of the CHOOSE metamodel as the first effort to develop an EA approach specifically tailored to SMEs. The CHOOSE metamodel is designed according to the requirements for EA in an SME context (Bernaert et al. 2013b). This is achieved by means of an action research programme in one specific SME, complemented by case study research in five more SMEs. The resulting metamodel is expressed as a UML class diagram, and extended with concept and relationship definitions in SBVR and intra- and inter-view constraints in OCL.

As the action research SME implemented certain changes according to the insights gained from the EA model, it was clear that the CHOOSE metamodel was indeed very valuable.

In fact, CHOOSE is still used in the SME, with the help of a software tool to support it.

Nevertheless, further work is still required. A first limitation is that the scope of the research was limited to a single company, which is typical of action research. However, five more case studies were concurrently performed in different kinds of SMEs, as research indicates that SMEs differ significantly in size, sector, and other factors (Bernaert et al. 2013b). The input, management, and output of CHOOSE models are hence tested in multiple SME settings. These case studies serve as input for the development and refinement of a method with step-by-step guidelines, for a further evaluation of the metamodel presented in this paper, and for the evaluation of the benefits of EA for SMEs. For example, the explicit representation of business rules in the CHOOSE metamodel has to be further examined if the need arises in further case studies. Possibilities of how this representation could be achieved can be found in (Businska et al. 2012).

Another area for future research involves software tool support for different platforms. This would enable an easier interface for SMEs to input, adjust, and analyze their EA model. Prototypes for PCs (Ingelbeen et al. 2013), smartphones, and tablets (Bernaert et al. 2013a; Dumeez et al. 2013) have already been developed and are currently being tested in different case studies. At present, we are also working on different possibilities to make as-is and to-be models and analyses, and are testing which best meet the needs of the SMEs. Moreover, the cognitive effectiveness of alternative notations for CHOOSE models (Boone et al. 2014) is being researched so as to provide a more efficient and effective visualization, since this also influences usability (Henderson-Sellers et al. 2012; Moody 2009). Finally, an integration with ArchiMate is being developed. This would allow users to switch from CHOOSE to ArchiMate if a more elaborate EA approach is needed to increase effectiveness for experienced EA users (Rossi and Brinkkemper 1996) (e.g., if a more detailed representation of the IT architecture would be needed), or to switch from ArchiMate to CHOOSE (Roose et al. 2013).

Appendix 1: KAOS metamodel

Each viewpoint is discussed separately and then the integrated KAOS metamodel is presented in Fig. 11. Concepts from the goal, agent, operation, object, and behaviour viewpoints will be respectively coloured in yellow, red, purple, green, and grey. Attributes between square brackets are optional attributes.

KAOS goal viewpoint

The central element of the KAOS goal viewpoint is a *Goal*. A *Goal* is a prescriptive statement of intent that the system should satisfy through the cooperation of its *Agents*. The formulation is declarative, unlike *Operational* procedures to implement it. A *Goal* can be of a specific type (*SoftGoal* or *BehaviourGoal* (*Achieve* or *Maintain/Avoid*)) and of a specific *[Category]* (functional or non-functional).

Goal Refinement is enabled by refining higher-level *Goals* in zero or more *Refinements* (*OR-Ref*) that group (*AND-Ref*) one or more lower-level *Goals*. An *AND-Ref* (*OR-Ref*) means that the parent *Goal* can be satisfied/satisficed by satisfying/satisficing all (one or more) child *Goals* in the *Refinement*. A *LeafGoal* is a *Goal* that can be under the *Responsibility* of exactly one *Agent* and is a *Requirement* or *Expectation*, depending on the type of *Agent* that has a *Responsibility* relationship with it (respectively *SoftwareToBeAgent* and *EnvironmentAgent*).

Domain properties (*DomInvar*) or hypotheses (*DomHyp*) are descriptive statements (*DomDescript*) holding regardless of how system *Agents* behave. Domain properties typically correspond to physical laws that cannot be broken.

Goals can be *ObstructedBy Obstacles* or can be a *Resolution* for *Obstacles*. *Obstacles* can be refined by *O-Refinements* in the same way as *Goals* can be refined by *Refinements*. *Conflict* links may interconnect *Goal* nodes to capture potential *Conflicts* among them. They are not explicitly represented in the metamodel, but are captured in the *Divergence* relation, which captures a potential *Conflict*, where some statements become logically inconsistent if a *BoundaryCondition* becomes true.

KAOS agent viewpoint

The central element of the KAOS agent viewpoint is an *Agent*. *Agents* are active system *Objects* that are responsible for the *LeafGoals* in a goal model. An *Agent* is responsible for a *Goal* by a *Responsibility* relationship if restricting its individual behaviour by adequate control of system items is sufficient for ensuring *Goal* satisfaction/satisficing.

From an operational standpoint, an *Agent* can be defined as a processor that performs (*Performance*) *Operations* under restricted conditions to satisfy the *Goals* for which it is responsible (*Responsibility*). For an *Agent* to be assigned (*Assignment*) to a *Goal*, the *Goal* must be realizable by the *Agent* in view of its capabilities. *Agent* capabilities are defined in terms of *Object Attributes* and *Associations* that the *Agent* can *Monitor* or *Control*. *Monitor* means that an *Agent* can get the values of the *Attribute* or can evaluate whether the *Association* holds, while *Control* means that an *Agent* can set values for this *Attribute* or can create or delete an *Association*.

An *Agent* can be decomposed into finer-grained ones with finer-grained *Responsibilities* through the recursive *Aggregation* relationship. An *Agent* may be related to other *Agents* through *Dependency* links. A *depender* depends on a *dependee* for a *Goal*, if a *dependee's* failure to get this *Goal* satisfied/satisficed can result in a *depender's* failure to get one of its *Assigned Goals* satisfied/satisficed.

An agent model defines the boundary between the software-to-be and its environment, as an *Agent* can be a *SoftwareToBeAgent* or an *EnvironmentAgent*. An *Agent* can be of a different *[Category]*, while this is not explicitly visible in the metamodel: *NewSoftwareAgents* to be developed, *ExistingSoftwareAgents* with which the software-to-be will have to interoperate, *Devices*, and *HumanAgents* playing specific *Roles*.

The *Wish* meta-relationship is not shown in the metamodel. It links *Goal* and *HumanAgent* and captures the fact that this *HumanAgent* would like the *Goal* to be satisfied/satisficed.

KAOS operation viewpoint

The operation viewpoint captures the functional services that the target system should provide in order to meet its *Goals*. An *Operation* is a binary relation over system *States*. Is has a tuple of *Input* variables and a tuple of *Output* variables defining its signature. An *Input* variable designates an *Object* instance to which the *Operation* applies. The *State* of this instance affects the application of the *Operation*. An *Output* variable designates an *Object* instance on which the *Operation* acts. The *State* of this instance is changed by the application of the *Operation*. An *Input* variable can be an *Output* variable for the same *Operation*. A particular application of the *Operation* yields a *State Transition* from a *State* in *InputState* to a *State* in *OutputState*.

An *Agent* performs (*Performance*) an *Operation* if the applications of this *Operation* are activated by instances of the *Agent*. Every *Operation* is *Performed* by exactly one *Agent. Operationalization* refers to the process of mapping *LeafGoals*, under the *Responsibility* of single *Agents*, to *Operations* ensuring them. Each such *Operation* is performed (*Performance*) by the responsible (*Responsibility*) *Agent* under restricted conditions for satisfaction/satisficing of its *Goals*. While a single *Operation* may operationalize (*Operationalization*) multiple *Goals*, a single *Goal* will in general be operationalized (*Operationalization*) by multiple *Operations*.

It is important to notice the difference between a *Goal* and an *Operation*. A *Goal* is an intentional specification: it leaves the *Operations* realizing it implicit, whereas an *Operation* is an operational specification: it leaves the intentions underlying it implicit. A *Goal* has a higher stability than an *Operation* (van Lamsweerde et al. 1995). A *Goal* captures an objective that the system should satisfy and is specified declaratively. An *Operation* captures a functional service that the system should provide to satisfy such an objective and maybe others and is specified by conditions characterizing its applicability and effect. Semantically speaking, a *BehaviouralGoal* constrains entire sequences of system *State Transitions*, while an *Operation* constrains single *State Transitions* within such sequences.

In KAOS, *Operations* are atomic and cannot be decomposed into finer-grained ones. *Goal Refinements* will be favoured, from which fine-grained *Operations* are derived, over *Goal*-free *Operation* refinement in an operational model (Letier and van Lamsweerde 2002).

KAOS object viewpoint

The object viewpoint provides a structural view of the system and is represented by entity-relationship diagrams using the UML class diagram notation. *Entities* and the structural features of *Events* and *Agents* will be represented as *Operation*-free UML classes and *Associations* will be represented as UML associations. The object model gathers all concept definitions and domain properties used in the goal, agent, operation, and behaviour models and introduces a common vocabulary to refer to. The object model can later on provide a basis for generating a database schema and for elaborating a software architecture.

A conceptual *Object* is a discrete set of instances of a domain-specific concept that are manipulated by the modelled system. These instances are distinctly identifiable, can be enumerated in any system *State*, share similar features, and may differ from each other in their individual *States* and *State Transitions*. The set of instances that are members of an *Object* will thus generally change over time. The semantic *InstanceOf* relation is kept implicit in the metamodel. This built-in semantic relation allows determining which individuals are instances of the *Object* in the current *State*.

An *Object* can be an *Association*, an *Entity,* an *Event*, or an *Agent*. An *Entity* is an autonomous and passive *Object*. An *Association* is a passive *Object* dependent on other *Objects* that it *Links* and it is also used under the synonymous term relationship. Each *Linked Object* plays a specific *Role* in the *Association*. An *Event* is an instantaneous *Object*. An *Agent* is as already mentioned an autonomous and active *Object*. It is important to notice that an *Agent* is a subtype of an *Object* and inherits the relationships of an *Object* (Dardenne et al. 1993). An *Association* can *Link* two or more *Objects*,

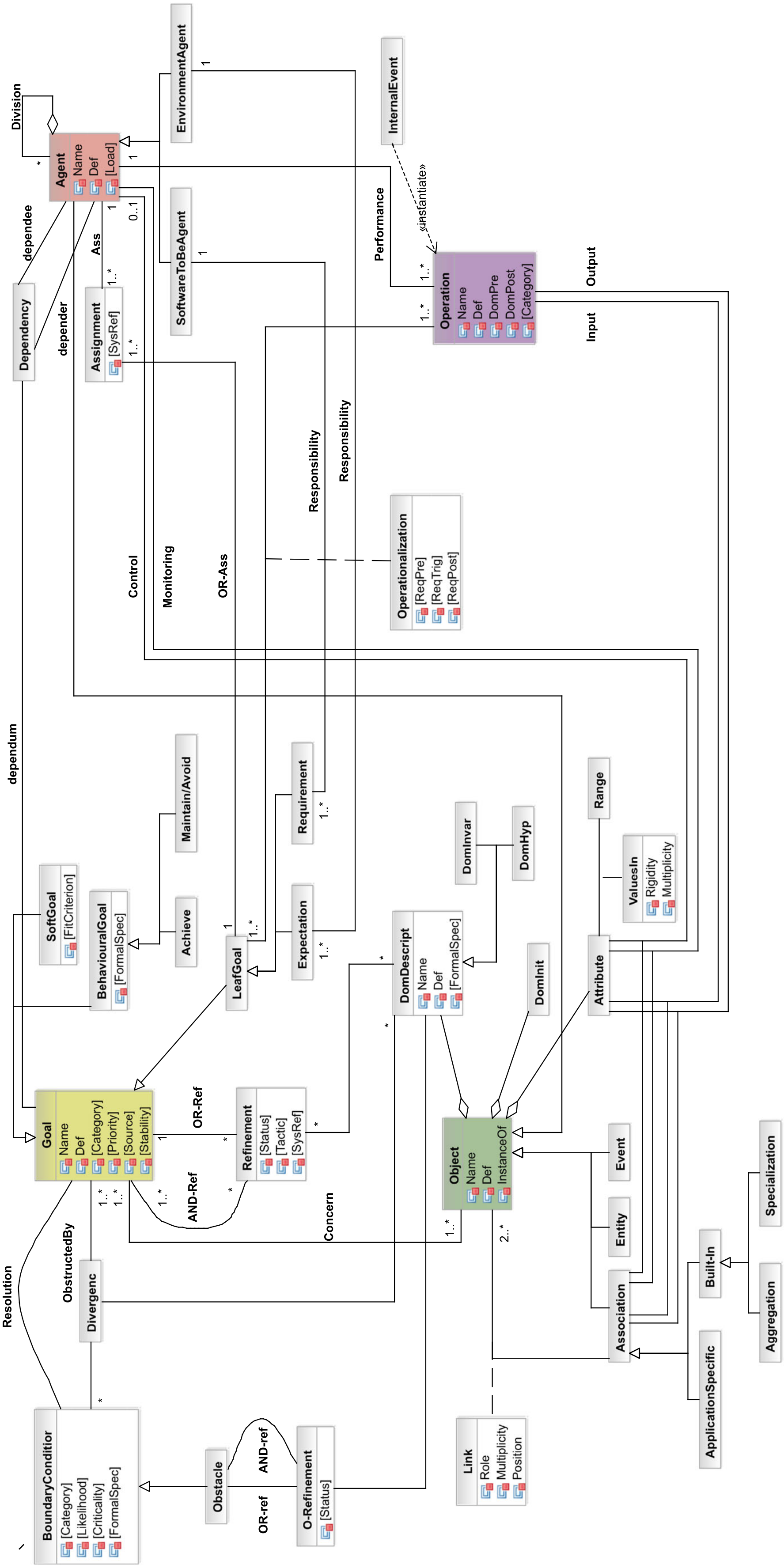

Fig. 11 KAOS integrated metamodel

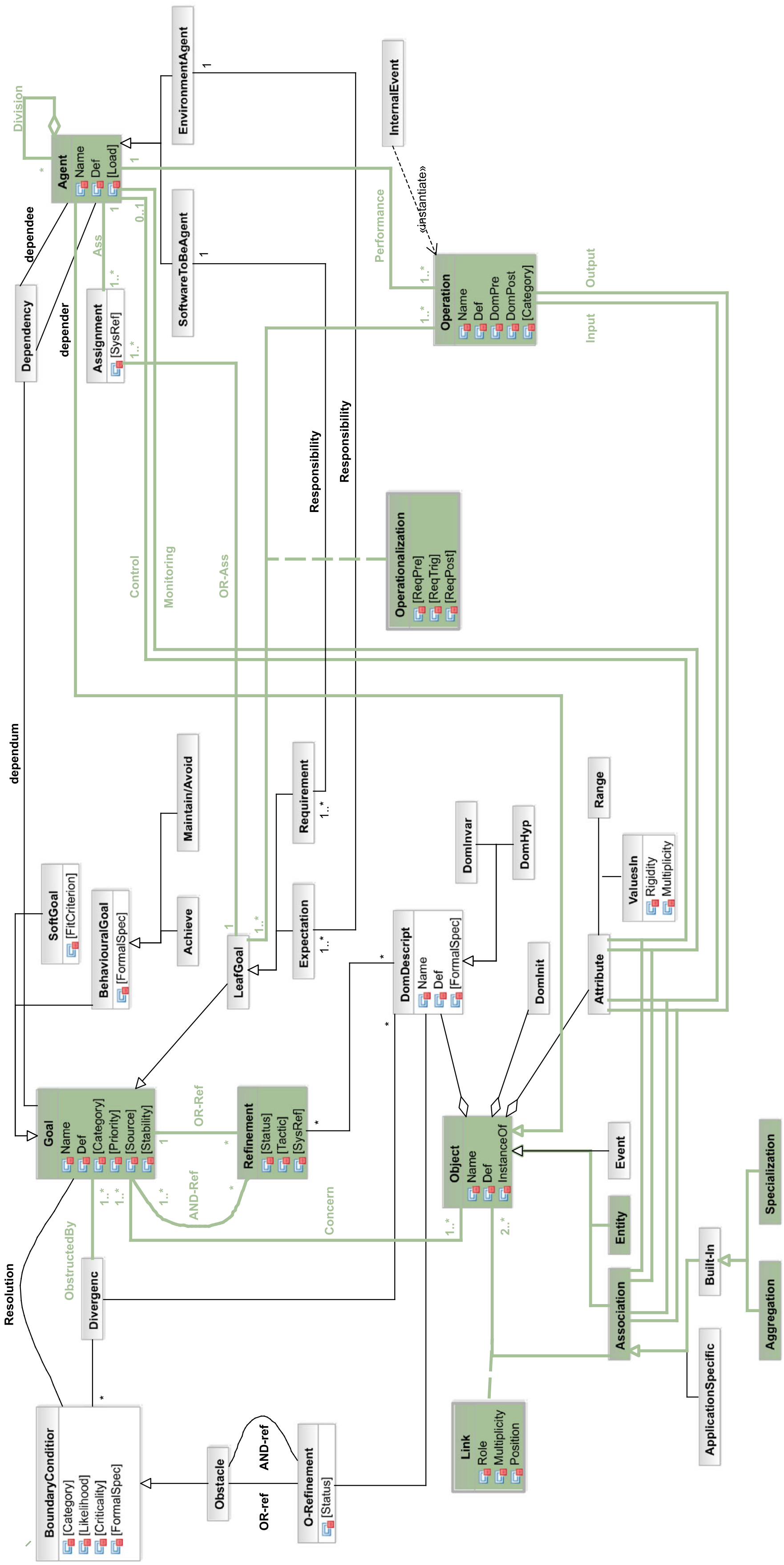

Fig. 12 KAOS elements being retained in CHOOSE

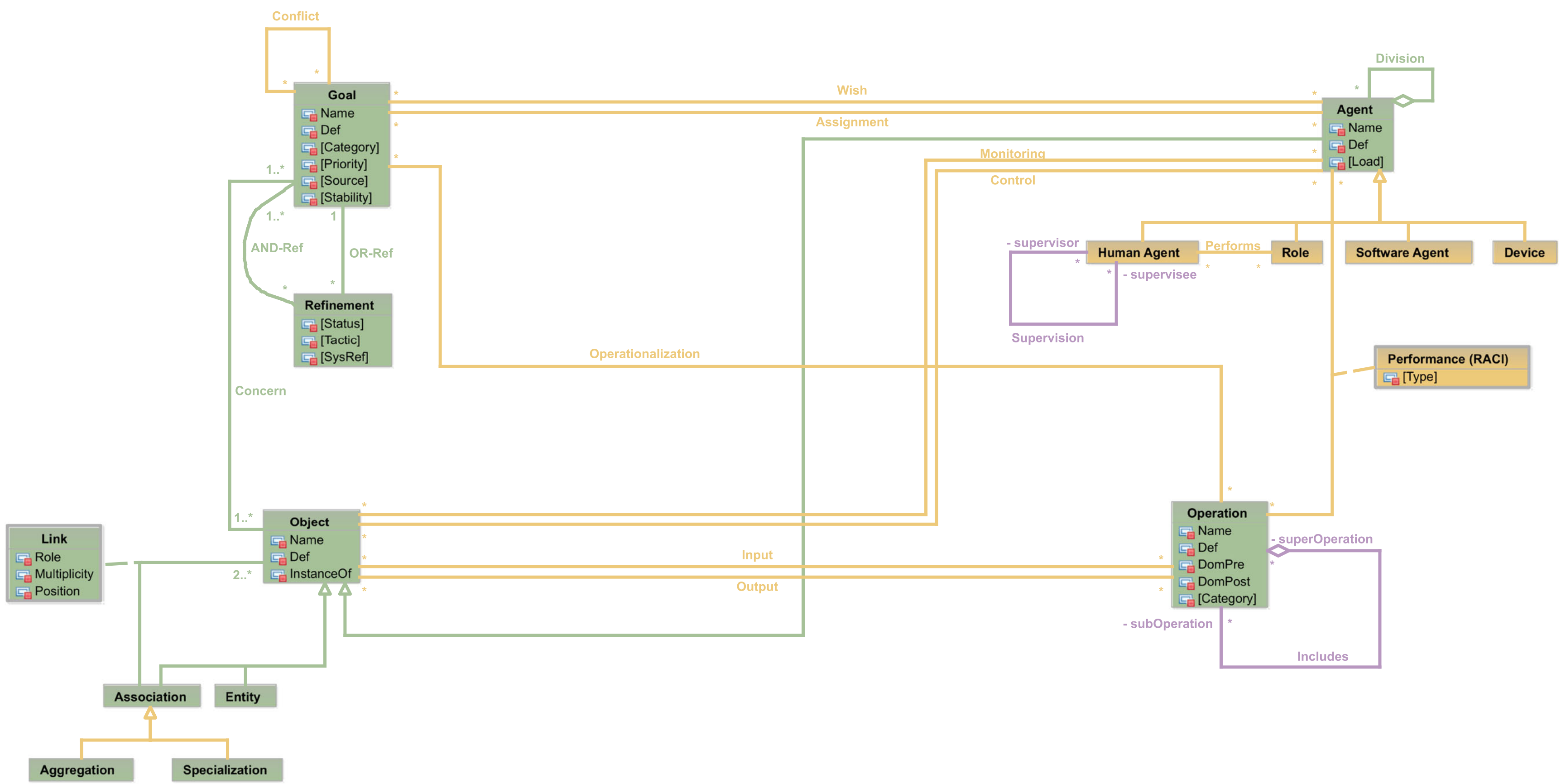

Fig. 13 Adjusting and adding elements from KAOS to CHOOSE

can be reflexive, can have different *Multiplicities* and can be a *Specialization*, an *Aggregation*, or an *ApplicationSpecific* type. An *Association* can have a *Name*, so a user can define different *Associations* with different *Names*. *Concern* links connect *Goal* nodes to the *Objects* to which they refer.

An *Attribute* is an intrinsic feature of an *Object* regardless of other *Objects* in the model. It has a *Name* and a *Range* of values.

KAOS behaviour viewpoint

The behaviour viewpoint completes the static representation of system functionalities by capturing the required system dynamics. An operation model focuses on classes of *Input–output State Transitions*, an object model declares and structures the variables undergoing *State Transitions*, and an agent model indicates which variable is controlled by which *Agent*.

Since this behaviour viewpoint will not be included in the CHOOSE metamodel, it is not further explained.

Integrated KAOS metamodel and adaptation to the CHOOSE metamodel

The viewpoints (excluding the behaviour viewpoint) can be combined to form the integrated KAOS metamodel (Fig. 11). The core element is each time represented in the corresponding colour. In Fig. 12, the green parts of the KAOS metamodel are the ones that were retained in the CHOOSE metamodel. Figure 13 depicts how these elements were either used as such (green) or adapted (orange), or where new elements were added (purple) to form the CHOOSE metamodel.

Appendix 2: OCL constraints

The complete CHOOSE metamodel's classes and associations were input in the USE tool (Fig. 14). Next, constraints were added and tested by instantiating the metamodel in the tool. In Table 3 the metamodel including constraints is presented as a text file serving as an input for the USE tool. The objectified relationships *Association*, *Aggregation*, and *Specialization* are defined as normal relationships and the association class of *Link* and *Performance* is not shown. Due to tool limits, both aggregation and specialization relationships are modelled as normal associations. If interested, this text file can be used directly as an input for the USE tool following the guidelines on (Database Systems Group 2013) to test the metamodel and OCL constraints.

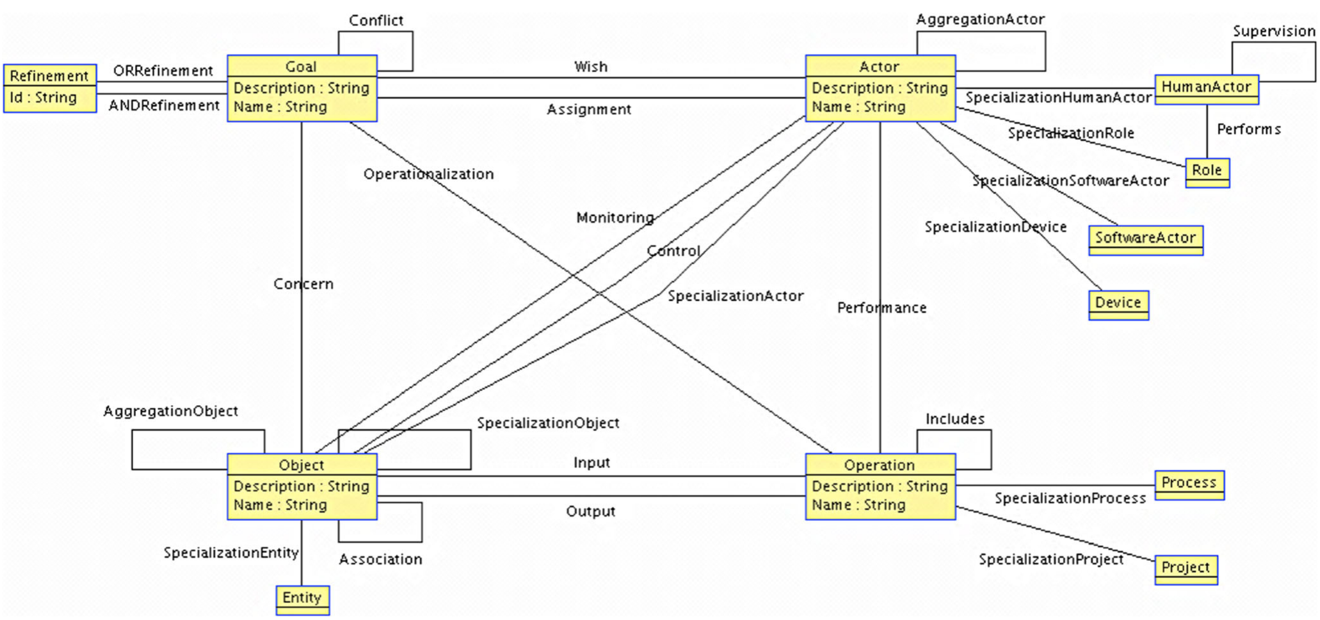

Fig. 14 CHOOSE metamodel in USE tool

Table 3 CHOOSE metamodel and constraints as input for the USE tool

```
model CHOOSE
-- CLASSES
        -- GOAL
class Goal
attributes
        Name : String
        Description : String
--Define the recursive upward operation to include all higher-level Goals
operations
        closureGoal(s : Set(Goal)) : Set(Goal) =
                if s->includesAll(s.ANDRefinement.ORGoal->asSet) then s
                else closureGoal(s->union(s.ANDRefinement.ORGoal->asSet)) endif
        allHigherGoals() : Set(Goal) = closureGoal(self.ANDRefinement.ORGoal->asSet)
end
class Refinement
attributes
        Id : Integer
end
        -- ACTOR
class Actor
attributes
        Name : String
        Description : String
-- Define the recursive upward operation to include all Whole Actors
operations
        closureActor(s : Set(Actor)) : Set(Actor) =
                if s->includesAll(s.WholeActor->asSet) then s
                else closureActor(s->union(s.WholeActor->asSet)) endif
        allWholeActors() : Set(Actor) = closureActor(self.WholeActor->asSet)
end

class HumanActor
-- Define the recursive upward operation to include all Supervisors
operations
        closureHumanActor(s : Set(HumanActor)) : Set(HumanActor) =
                if s->includesAll(s.Supervisor->asSet) then s
                else closureHumanActor(s->union(s.Supervisor->asSet)) endif
        allSupervisors() : Set(HumanActor) = closureHumanActor(self.Supervisor->asSet)
end
class Role
end
class SoftwareActor
end
class Device
end
        -- OPERATION
class Operation
attributes
        Name : String
        Description : String
-- Define the recursive upward operation to include all SuperOperations
operations
        closureOperation(s : Set(Operation)) : Set(Operation) =
                if s->includesAll(s.SuperOperation->asSet) then s
                else closureOperation(s->union(s.SuperOperation->asSet)) endif
        allSuperOperations() : Set(Operation) = closureOperation(self.SuperOperation->asSet)
end
class Process
end
class Project
end
```

Table 3 (continued)

```
            -- OBJECT
class Object
attributes
            Name : String
            Description : String
-- Define the recursive upward operation to include all upper SuperObjects
operations
            closureObject(s : Set(Object)) : Set(Object) =
                        if s->includesAll(s.SuperObject->asSet) then s
                        else closureObject(s->union(s.SuperObject->asSet)) endif
            allSuperObjects() : Set(Object) = closureObject(self.SuperObject->asSet)
end
class Entity
end
-- ASSOCIATIONS
            -- GOAL
association ORRefinement between
            Goal[1..1] role ORGoal
            Refinement[*] role ORRefinement
end
association ANDRefinement between
            Goal[1..*] role ANDGoal
            Refinement[*] role ANDRefinement
end
association Conflict between
            Goal[*] role ConflictGoal1
            Goal[*] role ConflictGoal2
end
            -- GOAL-ACTOR
association Wish between
            Goal[*] role WishGoal
            Actor[*] role WishActor
end
association Assignment between
            Goal[*] role AssignmentGoal
            Actor[*] role AssignmentActor
end

            -- GOAL-OPERATION
association Operationalization between
            Goal[*] role OperationalizationGoal
            Operation[*] role OperationalizationOperation
end
            -- GOAL-OBJECT
association Concern between
            Goal[*] role ConcernGoal
            Object[*] role ConcernObject
end
            -- ACTOR
association AggregationActor between
            Actor[*] role WholeActor
            Actor[*] role PartActor
end
association SpecializationHumanActor between
            Actor[*] role SuperHumanActor
            HumanActor[*] role SubHumanActor
end
association SpecializationRole between
            Actor[*] role SuperRole
            Role[*] role SubRole
end
association SpecializationSoftwareActor between
            Actor[*] role SuperSoftwareActor
            SoftwareActor[*] role SubSoftwareActor
end
association SpecializationDevice between
            Actor[*] role SuperDevice
            Device[*] role SubDevice
end
```

Table 3 (continued)

```
association Supervision between
        HumanActor[*] role Supervisor
        HumanActor[*] role Supervisee
end
association Performs between
        HumanActor[*] role PerformsHumanActor
        Role[*] role PerformsRole
end
        -- ACTOR-OPERATION
association Performance between
        Actor[*] role PerformanceActor
        Operation[*] role PerformanceOperation
end
        -- ACTOR-OBJECT
association Monitoring between
        Actor[*] role MonitoringActor
        Object[*] role MonitoringObject
end
association Control between
        Actor[*] role ControlActor
        Object[*] role ControlObject
end
        -- OPERATION
association Includes between
        Operation[*] role SuperOperation
        Operation[*] role SubOperation
end
association SpecializationProcess between
        Operation[*] role SuperProcess
        Process[*] role SubProcess
end
association SpecializationProject between
        Operation[*] role SuperProject
        Project[*] role SubProject
end

        -- OPERATION-OBJECT
association Input between
        Operation[*] role InputOperation
        Object[*] role InputObject
end
association Output between
        Operation[*] role OutputOperation
        Object[*] role OutputObject
end
        -- OBJECT
association Association between
        Object[*] role AssociationObject1
        Object[*] role AssociationObject2
end
association AggregationObject between
        Object[*] role WholeObject
        Object[*] role PartObject
end
association SpecializationObject between
        Object[*] role SuperObject
        Object[*] role SubObject
end
association SpecializationEntity between
        Object[*] role SuperEntity
        Entity[*] role SubEntity
end
association SpecializationActor between
        Object[*] role SuperActor
        Actor[*] role SubActor
end
```

Table 3 (continued)

-- CONSTRAINTS

```
        -- GOAL
--Constraint 1) Hard constraint: a Goal cannot have a Conflict with itself
context Goal
inv GOALSELFCONFLICT: (self.ConflictGoal1->union(self.ConflictGoal2))->excludes(self)

--Constraint 2) Soft constraint: the Goal model may not contain Refinement cycles
context Goal
inv GOALCYCLICREFINEMENT: self.allHigherGoals()->excludes(self)

        -- GOAL-ACTOR
--Constraint 3) Soft constraint: favour Assignments of Goals to Actors Wishing one of the related higher-level Goals
context Actor
inv WISHASSIGNMENT:
        if self.AssignmentGoal->notEmpty
        then
                if self.WishGoal->notEmpty
                then self.AssignmentGoal.allHigherGoals().WishActor->union(self.AssignmentGoal.WishActor)-
>includes(self)
                else true endif
        else true endif

--Constraint 4) Hard constraint: a Role, SofwareActor or Device cannot have a Wish relationship with a Goal
context Goal
inv ACTORWISH: self.WishActor.SubRole->isEmpty and self.WishActor.SubSoftwareActor->isEmpty and
self.WishActor.SubDevice->isEmpty

--Constraint 5) Soft constraint: if an Actor has an Assignment relationship with a Goal and is part of another Actor, then
the encompassing Actor should have an Assignment relationship with the same or a related higher-level Goal
context Actor
inv ASSIGNMENTAGGREGATION:
        if self.AssignmentGoal->notEmpty
        then
                if self.WholeActor->notEmpty
                then ((self.AssignmentGoal.allHigherGoals().AssignmentActor-
>union(self.AssignmentGoal.AssignmentActor))->intersection(self.allWholeActors()))->notEmpty
                else true endif
        else true endif

--Constraint 6) Soft constraint: if a HumanActor has an Assignment relationship with a Goal, then one of its Supervisors
should have an Assignment relationship with the same or a related higher-level Goal
```

```
inv GOALACTOROPERATION:
        if self.AssignmentGoal->isEmpty
        then true
        else (self.AssignmentGoal.OperationalizationOperation->asSet) = (self.PerformanceOperation->asSet) endif

--Constraint 11) Soft constraint: avoid allocating an Operation to an Actor if the Operation, or a child of it,
Operationalizes a Goal that itself or a related higher-level Goal Conflicts with the Goals Wished by the Actor
context Operation
inv OPERATIONWISHCONFLICT:
        if self.OperationalizationGoal->isEmpty or self.PerformanceActor->isEmpty
        then true
        else (self.OperationalizationGoal.allHigherGoals().ConflictGoal1.WishActor.PerformanceOperation-
>union(self.OperationalizationGoal.ConflictGoal1.WishActor.PerformanceOperation))-
>union(self.OperationalizationGoal.allHigherGoals().ConflictGoal2.WishActor.PerformanceOperation-
>union(self.OperationalizationGoal.ConflictGoal2.WishActor.PerformanceOperation))->excludes(self) endif

        -- GOAL-ACTOR-OBJECT
--Constraint 12) Soft constraint: if an Object is referred to by a Goal under the Assignment of an Actor, the Object must be
Monitored or Controlled by this Actor
context Object
inv GOALACTOROBJECT:
        if self.ConcernGoal.AssignmentActor->isEmpty
        then true
        else (self.ConcernGoal.AssignmentActor->asSet)=(self.MonitoringActor->union(self.ControlActor)->asSet)
endif

        -- ACTOR
--Constraint 13) Hard constraint: the HumanActor model may not contain Supervision cycles (a HumanActor cannot
Supervise itself)
context HumanActor
inv HUMANACTORCYCLICSUPERVISION: self.allSupervisors()->excludes(self)

--Constraint 14) Hard constraint: a HumanActor cannot Aggregate other Actors
context Actor
inv HUMANACTORWHOLE: self.WholeActor.SubHumanActor->isEmpty
```

Table 3 (continued)

```
--Constraint 15) Hard constraint: an Actor can only be Aggregated by other unspecialized Actors
context Actor
inv ACTORAGGREGATION:
        if self.SubHumanActor->isEmpty and self.SubRole->isEmpty and self.SubSoftwareActor->isEmpty and
self.SubDevice->isEmpty
        then self.WholeActor.SubHumanActor->isEmpty and self.WholeActor.SubRole->isEmpty and
self.WholeActor.SubSoftwareActor->isEmpty and self.WholeActor.SubDevice->isEmpty
        else true endif

--Constraint 16) Hard constraint: a HumanActor can only be Aggregated by unspecialized Actors
context HumanActor
inv HUMANACTORAGGREGATION: self.SuperHumanActor.WholeActor.SubHumanActor->isEmpty and
self.SuperHumanActor.WholeActor.SubRole->isEmpty and self.SuperHumanActor.WholeActor.SubSoftwareActor-
>isEmpty and self.SuperHumanActor.WholeActor.SubDevice->isEmpty

--Constraint 17) Hard constraint: a Role can only be Aggregated by other Roles or unspecialized Actors
context Role
inv ROLEAGGREGATION: self.SuperRole.WholeActor.SubHumanActor->isEmpty and
self.SuperRole.WholeActor.SubSoftwareActor->isEmpty and self.SuperRole.WholeActor.SubDevice->isEmpty

--Constraint 18) Hard constraint: a SoftwareActor can only be Aggregated by other SoftwareActors or unspecialized
Actors
context SoftwareActor
inv SOFTWAREACTORAGGREGATION: self.SuperSoftwareActor.WholeActor.SubHumanActor->isEmpty and
self.SuperSoftwareActor.WholeActor.SubRole->isEmpty and self.SuperSoftwareActor.WholeActor.SubDevice->isEmpty

--Constraint 19) Hard constraint: a Device can only be Aggregated by other Devices or unspecialized Actors
context Device
inv DEVICEAGGREGATION: self.SuperDevice.WholeActor.SubHumanActor->isEmpty and
self.SuperDevice.WholeActor.SubRole->isEmpty and self.SuperDevice.WholeActor.SubSoftwareActor->isEmpty

--Constraint 20) Hard constraint: the Actor model may not contain Aggregation cycles (an Actor cannot contain itself)
context Actor
inv ACTORCYCLICAGGREGATION: self.allWholeActors()->excludes(self)
        -- ACTOR-OPERATION
--Constraint 21) Soft constraint: every Operation should be under the Performance of at least one Actor
context Operation
inv OPERATIONPERFORMANCE: self.PerformanceActor->notEmpty

--Constraint 22) Soft constraint: if an Actor has a Performance relationship with an Operation and is part of one or more
Actors, then at least one of those other Actors should have a Performance relationship with the same Operation, or one of
its SuperOperations
context Actor
inv PERFORMANCEAGGREGATION:
        if self.PerformanceOperation->notEmpty
        then
                if self.WholeActor->notEmpty
                then ((self.PerformanceOperation.allSuperOperations().PerformanceActor-
>union(self.PerformanceOperation.PerformanceActor))->intersection(self.allWholeActors()))->notEmpty
                else true endif
        else true endif
        -- ACTOR-OPERATION-OBJECT
--Constraint 23) Soft constraint: the Inputs of an Operation Performed by an Actor should be Monitored by the Actor
context Actor
inv ACTOROPERATIONOBJECTINPUT:
        if self.PerformanceOperation.InputObject->isEmpty
        then true
        else (self.MonitoringObject->intersection(self.PerformanceOperation.InputObject)-
>asSet)=(self.PerformanceOperation.InputObject->asSet) endif

--Constraint 24) Soft constraint: the Outputs of an Operation Performed by an Actor should be Controlled by the Actor
context Actor
inv ACTOROPERATIONOBJECTOUTPUT:
        if self.PerformanceOperation.OutputObject->isEmpty
        then true
        else (self.ControlObject->intersection(self.PerformanceOperation.OutputObject)-
>asSet)=(self.PerformanceOperation.OutputObject->asSet) endif
        -- OPERATION
--Constraint 25) Hard constraint: the Operation model may not contain Includes cycles (Operation cannot Include itself)
context Operation
inv OPERATIONCYCLICINCLUDES: self.allSuperOperations()->excludes(self)
        -- OBJECT
--Constraint 26) Hard constraint: the Object model may not contain Specialization cycles (Object cannot Specialize itself)
context Object
inv OBJECTCYCLICSPECIALIZATION: self.allSuperObjects()->excludes(self)
```

References

Aarabi, M., Saman, M. Z. M., Wong, K. Y., Beheshti, H. M., & Hemdi, A. R. (2011). The effect of enterprise architecture for enterprise resource planning in small and medium enterprises: A review and research direction. In *IESS, Indonesia* (pp. 159–163).

Andersson, B., Johannesson, P., & Zdravkovic, J. (2009). Aligning goals and services through goal and business modelling. *Information System and e-Business Management, 7*(2), 143–169.

Anton, A. I. (1996). Goal-based requirements analysis. In *ICRE* (pp. 136–144).

Anton, A. I., McCracken, W. M., & Potts, C. (1994). Goal decomposition and scenario analysis in business process reengineering. In *CAiSE* (Vol. 811, pp. 94–104, LNCS). Utrecht: Springer.

Balabko, P., & Wegmann, A. (2006). Systemic classification of concern-based design methods in the context of enterprise architecture. *Information Systems Frontiers, 8*(2), 115–131.

Baskerville, R., & Myers, M. D. (2004). Special issue on action research in information systems: making IS research relevant to practice: foreword. *MIS Quarterly, 28*(3), 329–335.

Bernaert, M., & Poels, G. (2011). The quest for know-how, know-why, know-what and know-who: Using KAOS for enterprise modelling. In *CAiSE international workshops* (Vol. 83, pp. 29–40, LNBIP): Heidelberg: Springer.

Bernaert, M., Maes, J., & Poels, G. (2013a). An android tablet tool for enterprise architecture modeling in small and medium-sized enterprises. In *The practice of enterprise modeling* (pp. 145–160). Heidelberg: Springer.

Bernaert, M., Poels, G., Snoeck, M., & De Backer, M. (2013b). Enterprise architecture for small and medium-sized enterprises: A starting point for bringing EA to SMEs, based on adoption models. In *Information systems and small and medium-sized enterprises (SMEs): State of art of IS research in SMEs*. Berlin: Springer.

Bhagwat, R., & Sharma, M. K. (2007). Information system architecture: a framework for a cluster of small- and medium-sized enterprises (SMEs). *Production Planning and Control, 18*(4), 283–296.

Bidan, M., Rowe, F., & Truex, D. (2012). An empirical study of IS architectures in French SMEs: integration approaches. *European Journal of Information Systems, 21*(3), 287–302.

Bittler, R. S., & Kreizmann, G. (2005). Gartner enterprise architecture process. *Evolution, 21*.

Boman, M., Bubenko, J. A., Johannesson, P., & Wangler, B. (1997). *Conceptual modelling*. London: Prentice-Hall, Inc.

Boone, S., Bernaert, M., Roelens, B., Mertens, S., & Poels, G. (2014). Evaluating and improving the visualisation of CHOOSE, an enterprise architecture approach for SMEs. In *The practice of enterprise modeling* (Vol. 197, pp. 87–102, LNBIP). Berlin: Springer.

Braun, C., & Winter, R. (2005). A comprehensive enterprise architecture metamodel and its implementation using a metamodeling platform. In *EMISA* (pp. 64–79).

Bubenko, J. (1993). Extending the scope of information modelling. In *DAISD* (pp. 73–97).

Business Transformation Agency. (2009). Vocabulary-Driven Enterprise Architecture Development: Guidelines for DoDAF AV-2: Design and Development of the Integrated Dictionary.

Businska, L., Kirikova, M., Penicina, L., Buksa, I., & Rudzajs, P. (2012). Enterprise modeling for respecting regulations. In *PoEM* (pp. 106–117).

Clark, T., Barn, B. S., & Oussena, S. (2011). LEAP: A precise lightweight framework for enterprise architecture. In *ISEC* (pp. 85–94). ACM.

Daneva, M., & van Eck, P. (2007). What enterprise architecture and enterprise systems usage can and can not tell about each other. *International Journal of Computer Science and Applications, 4*, 93–109.

Dardenne, A., Fickas, S., & van Lamsweerde, A. (1991). Goal-directed concept acquisition in requirements elicitation. In *IWSSD* (pp. 14–21). Los Alamitos: IEEE.

Dardenne, A., van Lamsweerde, A., & Fickas, S. (1993). Goal-directed requirements acquisition. *Science of Computer Programming, 20*(1–2), 3–50.

Database Systems Group. (2013). USE: A UML based Specification Environment. http://sourceforge.net/projects/useocl/.

Denzin, N. K. (2006). *Sociological methods: A sourcebook*. Transaction Publishers: Piscataway.

Devos, J. (2011). *IT governance for SMEs*. Ghent: University of Ghent.

DoD. (2010). DoDAF Architecture Framework Version 2.2. http://dodcio.defense.gov/dodaf20.aspx.

Dumeez, J., Bernaert, M., & Poels, G. (2013). Development of software tool support for enterprise architecture in small and medium-sized enterprises. In *CAiSE international workshops* (LNBIP). Springer.

Engelsman, W., & Wieringa, R. (2012). Goal-oriented requirements engineering and enterprise architecture: Two case studies and some lessons learned. In *REFSQ* (Vol. 7195, pp. 306–320, LNCS). Heidelberg: Springer.

Engelsman, W., Quartel, D., Jonkers, H., & van Sinderen, M. (2011). Extending enterprise architecture modelling with business goals and requirements. *Enterprise Information Systems, 5*(1), 9–36.

Erickson, J., & Siau, K. (2007). Can UML be simplified? Practitioner use of UML in separate domains. In *EMMSAD* (pp. 89–98). Citeseer.

Eriksson, H.-E., & Penker, M. (2000). *Business modeling with UML: Business patterns at work*. New York: Wiley.

Ernst, A. M., Lankes, J., Schweda, C. M., & Wittenburg, A. (2006). Tool support for enterprise architecture management—Strengths and weaknesses. In *EDOC* (pp. 13–22)

European Commission. (2011). Are EU SMEs recovering from the crisis? *Annual Report on EU Small and Medium sized Enterprises 2010/2011*.

French, W. L., & Bell, C. H. (1973). *Organization development: Behavioral science interventions for organization improvement*. Englewood Cliffs: Prentice-Hall.

Gartner (2012). Gartner's 2011 Global Enterprise Architecture Survey: EA Frameworks Are Still Homemade and Hybrid.

Georgiadis, G. (2015). Development of an ontology and risk-led method for selecting and tailoring Enterprise Architecture Frameworks. In W. Paper (Ed.), Working Paper, Ghent University.

Glissman, S., & Sanz, J. (2009). *A comparative review of business architecture*. San Jose: IBM.

Gogolla, M., Büttner, F., & Richters, M. (2007). USE: a UML-based specification environment for validating UML and OCL. *Science of Computer Programming, 69*(1–3), 27–34.

Hannon, P. D., & Atherton, A. (1998). Small firm success and the art of orienteering: the value of plans, planning, and strategic awareness in the competitive small firm. *Journal of Small Business and Enterprise Development, 5*(2), 102–119.

Hegge, H. M. H., & Wortmann, J. C. (1991). Generic bill-of-material: a new product model. *International Journal of Production Economics, 23*(1–3), 117–128.

Henderson, J. C., & Venkatraman, N. (1993). Strategic alignment: leveraging information technology for transforming organizations. *IBM Systems Journal, 32*(1), 4–16.

Henderson-Sellers, B., Low, G., & Gonzalez-Perez, C. (2012). Semiotic considerations for the design of an agent-oriented modelling language. In *EMMSAD* (Vol. 113, pp. 422–434). Springer LNBIP.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75–105.

Heyse, M., Bernaert, M., & Poels, G. (2012). *Keuzes Maken binnen Processen: Het Vermijden van een Russische Roulette voor de Organisaties*. Ghent: University of Ghent.

Hoogervorst, J. (2004). Enterprise architecture: enabling integration, agility and change. *International Journal of Cooperative Information Systems, 13*(3), 213–233.

IEEE Computer Society. (2000). IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. *IEEE Std 1471–2000*.

IFEAD. (2005). Trends in Enterprise Architecture 2005: How are Organizations Progressing?

IFEAD. (2006). Extended Enterprise Architecture Framework Essentials Guide.

Ingelbeen, D., Bernaert, M., & Poels, G. (2013). Enterprise architecture software tool support for small and medium-sized enterprises: EASE. In *AMCIS*.

ISACA. (2012). COBIT 5. http://www.isaca.org/COBIT/Pages/COBIT-5-Framework-product-page.aspx.

Jacobs, D., Kotzé, P., van der Merwe, A., & Gerber, A. (2011). Enterprise architecture for small and medium enterprise growth. In *EEWC* (Vol. 79, pp. 61–75, LNBIP). Heidelberg: Springer.

James, G. A., Handler, R. A., Lapkin, A., & Gall, N. (2005). Gartner Enterprise Architecture Framework. *Evolution*.

Järvinen, P. (2007). Action research is similar to design science. *Quality & Quantity, 41*(1), 37–54.

Jonkers, H., Lankhorst, M. M., ter Doest, H. W. L., Arbab, F., Bosma, H., & Wieringa, R. J. (2006). Enterprise architecture: management tool and blueprint for the organisation. *Information Systems Frontiers, 8*(2), 63–66.

Kerzner, H. R. (2013). *Project management: A systems approach to planning, scheduling, and controlling* (Vol. 11). Hoboken: Wiley.

Kroon, B., Voorde, K., & Timmers, J. (2012). High performance work practices in small firms: a resource-poverty and strategic decision-making perspective. *Small Business Economics, 1*, 1–21.

Lankhorst, M. (2013). *Enterprise architecture at work: Modelling, communication and analysis* (Vol. 3). New York: Springer.

Leist, S., & Zellner, G. (2006). Evaluation of current architecture frameworks. In *SAC* (pp. 1546–1553). New York: ACM.

Letier, E., & van Lamsweerde, A. (2002). Deriving operational software specifications from system goals. In *SIGSOFT* (pp. 119–128). New York: ACM.

Letier, E., & van Lamsweerde, A. (2004). Reasoning about partial goal satisfaction for requirements and design engineering. In *SIGSOFT* (pp. 53–62). Newport Beach: ACM.

Lindström, A., Johnson, P., Johansson, E., Ekstedt, M., & Simonsson, M. (2006). A survey on CIO concerns-do enterprise architecture frameworks support them? *Information Systems Frontiers, 8*(2), 81–90.

Loucopoulos, P., & Kavakli, E. (1995). Enterprise modelling and the teleological approach to requirements engineering. *International Journal of Cooperative Information Systems, 4*(1), 45–79.

Luftman, J., & Ben-Zvi, T. (2011). Key issues for IT executives 2011: cautious optimism in uncertain economic times. *MIS Quarterly Executive, 10*(4), 203–212.

Lybaert, N. (1998). The information use in a SME: its importance and some elements of influence. *Small Business Economics, 10*(2), 171–191.

Moody, D. L. (2003). The method evaluation model: A theoretical model for validating information systems design methods. In *ECIS*.

Moody, D. (2009). The physics of notations: toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering, 35*(6), 756–779.

Mostow, J. (1985). Towards better models of the design process. *AI Magazine, 6*(1), 44–57.

Mylopoulos, J., Chung, L., & Nixon, B. (1992). Representing and using nonfunctional requirements: a process-oriented approach. *IEEE Transactions on Software Engineering, 18*(6), 483–497.

OMG. (2008). Semantics of Business Vocabulary and Business Rules (SBVR) (v1.0). http://www.omg.org/spec/SBVR/1.0/.

OMG. (2009). Organization Structure Metamodel (OSM). http://www.omg.org/cgi-bin/doc?bmi/09-08-02.

OMG. (2010). Business Motivation Model (BMM) (v1.1). http://www.omg.org/spec/BMM/1.1/.

OMG. (2011a). Business Process Model and Notation (BPMN) (v2.0). http://www.omg.org/spec/BPMN/2.0/.

OMG. (2011b). OMG Unified Modeling Language (OMG UML), Infrastructure (v2.4.1). http://www.omg.org/spec/UML/2.4.1/.

OMG. (2011c). OMG Unified Modeling Language (OMG UML), Superstructure (v2.4.1). http://www.omg.org/spec/UML/2.4.1/.

OMG. (2012a). OMG Object Constraint Language (OCL) (v2.3.1). http://www.omg.org/spec/OCL/2.3.1/.

OMG. (2012b). OMG's MetaObject Facility. http://www.omg.org/mof/.

OMG. (2013). MDA—The Architecture Of Choice For A Changing World. http://www.omg.org/mda/.

O'Regan, N., & Ghobadian, A. (2004). The importance of capabilities for strategic direction and performance. *Management Decisions, 42*(2), 292–313.

Paige, R. F., Brooke, P. J., & Ostroff, J. S. (2007). Metamodel-based model conformance and multiview consistency checking. *ACM Transactions on Software Engineering and Methodology, 16*(3).

Project Management Institute. (2013). Project Management Body of Knowlegde (PMBOK). http://www.pmi.org/PMBOK-Guide-and-Standards.aspx.

Radeke, F. (2011). Toward understanding enterprise architecture management's role in strategic change: Antecedents, processes, outcomes. In *WI* (Vol. 2011).

Rescher, N. (1977). *Methodological pragmatism: Systems-theoretic approach to the theory of knowledge*. Oxford: Blackwell Publishers.

Respect-IT. (2007). A KAOS Tutorial. http://www.respect-it.be/.

Roose, D., Vansteenlandt, J., Bernaert, M., & Poels, G. (2013). *Development of a common base for enterprise architecture: Building the bridge between CHOOSE and ArchiMate*. Ghent: University of Ghent.

Ross, J. W., Weill, P., & Robertson, D. (2006). *Enterprise architecture as strategy: Creating a foundation for business execution*. Boston: Harvard Business Review Press.

Rossi, M., & Brinkkemper, S. (1996). Complexity metrics for systems development methods and techniques. *Information Systems, 21*(2), 209–227.

Sandkuhl, K., Stirna, J., Persson, A., & Wißotzki, M. (2014). *Enterprise modeling: Tackling business challenges with the 4EM method*. Berlin: Springer.

Scheer, A.-W. (2000). *ARIS—Business process modeling* (Vol. 3). Berlin: Springer.

Schekkerman, J. (2006). *How to survive in the jungle of enterprise architecture frameworks: Creating or choosing an enterprise architecture framework*. Victoria: Trafford Publishing.

Sessions, R. (2007). *Comparison of the top four enterprise architecture methodologies*. Microsoft.

Small Business Administration. (2011). How Important are Small Businesses to the U.S. Economy? http://www.sba.gov/advo.

Smith, J. A. (1998). Strategies for start-ups. *Long Range Planning, 31*(6), 857–872.

Snoeck, M., Dedene, G., Verhelst, M., & Depuydt, A.-M. (1999). *Object-oriented enterprise modelling with MERODE*. Leuven: Leuven University Press.

Stirna, J., & Persson, A. (2007). Ten years plus with EKD: Reflections from using an enterprise modeling method in practice. In *EMMSAD* (pp. 99–108).

Susman, G. I., & Evered, R. D. (1978). An assessment of the scientific merits of action research. *Administrative Science Quarterly, 23*(4), 582–603.

Tamm, T., Seddon, P. B., Shanks, G., & Reynolds, P. (2011). How does enterprise architecture add value to organisations? *CAIS, 28*(1)

The Open Group. (2009). TOGAF Version 9. http://www.opengroup.org/togaf.

The Open Group. (2012). ArchiMate 2.0. http://www3.opengroup.org/subjectareas/enterprise/archimate.

The White House OMB. (2012). The Common Approach To Federal Enterprise Architecture.

The White House OMB. (2013). Federal Enterprise Architecture. http://www.whitehouse.gov/omb/e-gov/FEA.

Urbaczewski, L., & Mrdalj, S. (2006). A comparison of enterprise architecture frameworks. *Issues in Information Systems, 7*(2), 18–23.

Van Lamsweerde, A. (2009). *Requirements engineering: From system goals to UML models to software specifications* (Vol. 3). New York: Wiley.

van Lamsweerde, A., Dardenne, A., Delcourt, B., & Dubisy, F. (1991). The KAOS project: Knowledge acquisition in automated specification of software. In *AAAI Spring Symposium Series, Stanford University* (pp. 59–62).

van Lamsweerde, A., Darimont, R., & Massonet, P. (1995). Goal-directed elaboration of requirements for a meeting scheduler: Problems and lessons learnt. In *RE* (pp. 194–203). IEEE.

van Lamsweerde, A., Darimont, R., & Letier, E. (2002). Managing conflicts in goal-driven requirements engineering. *IEEE Transactions on Software Engineering, 24*(11), 908–926.

van't Wout, J., Waage, M., Hartman, H., Stahlecker, M., & Hofman, A. (2010). *The integrated architecture framework explained: Why, what, how*. Berlin: Springer.

Veasey, P. W. (2001). Use of enterprise architectures in managing strategic change. *Business Process Management Journal, 7*(5), 420–436.

Wagter, R., van den Berg, M., Luijpers, J., & van Steenbergen, M. (2005). *Dynamic enterprise architecture: How to make it work*. New York: Wiley.

Warmer, J., & Kleppe, A. (2003). *The object constraint language: Getting your models ready for MDA* (Vol. 2). Addison-Wesley Professional.

Wegmann, A., Regev, G., Rychkova, I., Lê, L.-S., de la Cruz, J. D., & Julia, P. (2007). Business and IT alignment with SEAM for enterprise architecture. In *EDOC.*

Weske, M. (2012). *Business process management: Concepts, languages, architectures* (Vol. 2). Berlin: Springer.

Winter, R., & Fischer, R. (2007). Essential layers, artifacts, and dependencies of enterprise architecture. *Journal of Enterprise Architecture, 3*(2), 7–18.

Wißotzki, M., & Sonnenberger, A. (2012). Enterprise architecture management—State of research analysis & a comparison of selected approaches. In *PoEM* (pp. 37).

Yu, E. S. K. (1993). Modelling organizations for information systems requirements engineering. In *RE* (pp. 34–41). IEEE.

Zach, O. (2012). *ERP system implementation in small and medium-sized enterprises*. Agder: University of Agder.

Zachman, J. A. (1987). A framework for information systems architecture. *IBM Systems Journal, 26*(3), 276–292.

Zachman International. (2011). Zachman Version 3.0. http://www.zachman.com.

Maxime Bernaert's expertise situates at the frontier of strategy, business architecture, enterprise architecture governance and organization, and economics. Maxime is currently consultant at inno.com (a consulting firm specialized in enterprise architecture and strategic ICT projects), responsible for inno.com institute's academic "Master of ICT Enterprise Architecture" and PhD student at Ghent University. During his PhD in Management Information Science as a Commercial Engineer at Ghent University, Maxime gathered a lot of theoretical and practical experience in enterprise architecture. He developed the CHOOSE approach, up till now the first and only enterprise architecture alternative for small and medium-sized enterprises (SMEs). CHOOSE is based on an extensive analysis of existing enterprise architecture approaches and excels in simplicity and usability. Maxime worked together with a team of 25 Master thesis students during the development of this CHOOSE approach.

Geert Poels is full professor at Ghent University's Faculty of Economics and Business Administration. He is founder and director of the UGentMIS Business Informatics research group. He was (co-)supervisor of 6 completed PhD's and is (co-)supervisor of 11 ongoing PhD's. He is (co-)author of 84 publications listed in Web of Science, including 38 articles. In 2012 he was program chair of the sixth International IFIP Conference on Research and Practical Issues in Enterprise Information Systems (CONFENIS 2012). In 2014 he was conference chair of the eight European Conference on Information Systems Management and Evaluation (ECIME 2014). Prof. Poels is the developer of the DISTANCE approach to software measurement. He published widely on topics related to functional size measurement and quality of conceptual models. At Ghent University he initiated the Business Informatics research and developed the conceptual modeling, business ontology, and business modeling research lines. His current research also involves Enterprise Architecture, Business Process Management, and Service Science. He is editorial (review) board member of three academic journals.

Monique Snoeck is full professor at the KU Leuven, Research Center for Management Informatics (LIRIS), and visiting professor at the UNamur. Her research focuses on enterprise information systems engineering, enterprise modeling, requirements engineering, model-driven engineering and business process management. Main guiding research themes are the integration of different modelling approaches into a comprehensive approach, the quality of models through formal grounding, models to code transformations and educational aspects of conceptual modelling. She has published over 80 peer-reviewed papers.

Prof. Manu De Backer graduated as Master in Applied Economic Sciences and obtained the degree of Doctor in Applied Economic Sciences in 2007 (KULeuven). He is an assistant professor at Ghent University (75% of FTE) and part-time assistant professor in the Department of Management Information Systems of the University of Antwerp. His research focuses on the verification of business process models and the consistency of business processes using mathematical models such as Petri Nets and Finite State Machines. More recently his research was more oriented towards the strategic use of business processes and the integration of business process models with other conceptual models (DMN, SBVR, etc.).