Understandability of Goal-Oriented Requirements Engineering Concepts for Enterprise Architects

Wilco Engelsman[1,2] and Roel Wieringa[2]

[1] BiZZdesign
w.engelsman@bizzdesign.nl
[2] University of Twente
r.j.wieringa@utwente.nl

Abstract. ArchiMate is a graphical language for modelling business goals and enterprise architecture. In previous work we identified possible understandability issues with the goal-oriented notations in ArchiMate. [**Problem**] We investigated how understandable the goal-oriented concepts really were in two quasi-experiments with practitioners. [**Principal ideas/results**] Only three concepts were understood by most or all subjects; the stakeholder concept, the goal concept and the requirement concept. The other concepts were misunderstood by most of our subjects. We offer explanations for these (mis)understandings. [**Contribution**] This paper provides new insights into the understandability and hence usability of goal-oriented concepts by practicing enterprise architects.

Keywords: GORE, ArchiMate, Evaluation.

1 Introduction

In large companies the gap between business and IT is usually bridged by designing and maintaining an enterprise architecture (EA). An enterprise architecture is a high-level representation of the enterprise, used for managing the relation between business and IT.

Large organizations must model their enterprise architectures in order to coordinate IT projects and the management of IT costs. In addition, in recent years EA is used to increase the flexibility of the organization and to justify the contribution of IT to business goals. This means that EAs need to be used not only to coordinate IT projects, but also for the following kinds of analysis:

- to determine the impact of changes in the business environment on the organizational goals and the EA,
- to determine the value of a certain architectural element and
- to assess which projects that implement the architecture have the most business value.

This requires an extension of EA modelling languages with concepts such as business goal and business value, and support for tracing business goals to EA

M. Jarke et al. (Eds.): CAiSE 2014, LNCS 8484, pp. 105–119, 2014.

components. In this paper, we empirically investigate such an extension on understandability.

The context of our work is the ArchiMate language for enterprise architecture modelling [27]. In previous work [9,23] we defined a set of goal-oriented concepts based on the concepts found in goal-oriented requirements engineering (GORE) and extended ArchiMate with these concepts. These goal-oriented extensions have been adopted in the Open Group standard for enterprise architecture modelling [27]. In subsequent work we provided initial empirical validation of the usability of this extension [10]. This validation showed that some users of the language experienced difficulty in understanding the extension to ArchiMate, and we proposed a simplification of the goal-oriented extension. In this paper, we present and analyze further data about understandability of goal-oriented concepts by enterprise architects, and we present explanations of the understandability issues. We present tentative generalizations about goal-oriented concepts in the context of enterprise architecture. We believe that the population of enterprise achitects have no difficulty in using the stakeholder, goal and requirement concept. Regarding the relations, the influence relation is the best understood relation. These findings should also be true for the languages we based the motivation extension of ArchiMate on, namely i*, Tropos, KAOS and GBRAM.

We start with listing the research questions in the next section. Next we describe our research methodology in section 3. We detail our conceptual framework in section 4. Section 5 presents our data and analyzes the implications of these data for goal-oriented requirements concepts. In section 6 we provide answers to our research questions. Section 7 discusses related work and in section 8 we discuss some implications for practice and for further research.

2 Research Problem

We want to know how understandable the goal-oriented requirements extension to an enterprise architecture language is for practicing enterprise architects. So our population of interest is the population of enterprise architects, and in our research we investigate a small sample of them, and we investigate the understandability of the goal-oriented extension of the ArchiMate language. At the end of the paper we discuss the generalizability of our results to the larger population of interest. Here we state our research questions:

Q1: How understandable is the motivation extension of ArchiMate by enterprise architects?
Q2: Which concepts are understood correctly? Why?
Q3: Which concepts are not understood? Why?
Q4: What kind of mistakes are made? Why?

We will define the concept of understandability, used in Q1, as the percentage of language users who understand the concept correctly. Q2 and Q3 ask which concepts are understood by all users or misunderstood by at least some users, respectively. For the concepts that are misunderstood, Q4 asks what kinds of mistakes are made.

In all cases, we want to know not only an answer to the journalistic question what is the case, but also the research question why it is the case.

3 Research Methodology

In terms of design research methodology, our empirical study is an evaluation of a technology implemented in practice, namely ArchiMate [29]. However, ArchiMate (version 2.0) has only recently been implemented and although it is used, it has not been used long enough on a large scale to make an evaluation by survey possible. Moreover, although surveys may reveal large-scale trends, they are inadequate at providing the detailed data about understandability that we need.

Our data comes from two groups of practitioners who followed a course on ArchiMate. Their homework provided the material we needed to assess understandability of goal-oriented concepts to enterprise architects, and to answer our research question above. The first group had 7 participants, and the second group had 12 participants. Their homework was an excercise based on an actual problem within the organization. These were real requirements engineering or EA design problems and therefore a fair representation of the difficulty level.

The participants of the two groups self-selected into the course, and so they may be more motivated or more talented than the "average" enterprise architect. They were also highly motivated to pass the course, since they were sent by their employer. They had to pass their homework exercises in order to get a certificate. Not passing the exam would have relfected badly on the subject and weakened their position in the organization.This would make understandability problems all the more telling.

All participants had at least 5 years of experience as an enterprise architect (or a similar role) and all had at least a bachelors degree (not necessarily in computer science or software engineering). The median experience is based on the linkedin profiles of the subjects. They have had some modelling experience, since this is common in their role of architect or business analyst. Since we did not do random sampling, and the groups are too small for statistical inference anyway, we cannot draw any statistical inferences from our results. We can only give descriptive statistics of our sample, but not draw statistical conclusions about the population of enterprise architects.

A controlled experiment would have given us more flexibility, but this is beyond our budget to do such an experiment with practitioners (i.e. we would have to pay them commercial fees).

However, because we have detailed data from the homework done by the participants, we will analyze possible causes for (mis)understanding goal-oriented concepts in ArchiMate, and then consider whether these explanations provide a reason for expecting (mis)understanding of goal-oriented concepts to occur outside our sample in a similar way that it happened in our sample. We will also compare our results with those in the published literature to see if results similar to ours have been found in other studies too, which would strengthen the plausibility of generalizations.

The first author (Engelsman) has been a teacher in the first but not in the second course. On both courses, he did the correcting of student assignments. The assignments were relatively small compared to real-world enterprise architecture concepts. That reduces the generalizability of our results, but in a useful direction: We expect that in larger, real-world projects, understandability problems would increase, not decrease compared to what we have observed in our courses. This is useful because our findings provide suggestions for improvement of teaching and using goal-oriented concepts in enterprise architecture in practice.

The first author was involved in the definition of GORE concepts in ArchiMate. Correction was done twice, and we assume that few mistakes in correcting the assignments have been made. A sample of the corrections of the student exercises have been discussed between the two authors of this paper, and no mistakes were found. However, later we will see that even if we would increase or decrease the percentages (in)correct in the gradings with as much as 10 points, this would not change our explanations and qualitative generalizations.

4 Defining Understandability

Many authors operationalize understandability in terms of the time needed to understand a model [8,12] or the number of mistakes made in answers to questions about a model [16,21,22,25]. Houy et al. [14] surveyed these definitions of model understandability and classified them in five types:

- Recalling model content. Subjects are given a model, and are given time to study the model. Afterwards they have to recall how the model looked like.
- Correctly answering question about model content. Subjects are given a model and are given time to study the model. Afterwards they are presented with a questionnaire and have to answer questions about the information in the model (e.g. the constructs used in a model).
- Problem solving based on the model content. Subjects are given a model to analyze, and are asked to solve problems (answer questions) based on this model. For example, if the model were a route for a bus, they were asked questions about the route of the bus.
- Verification of model content. Subjects are given a model and a textual description. They have to answer questions regarding the correctness of the model content based on the problem description.
- Time needed to understand the model. Subjects are given a model to study. The time needed to answer questions about the model is measured.

Another interesting approach to measuring understandability, not mentioned by Houy et al., is that of Caire et al. [5], who measured the ability of subjects to guess the definition of an i* construct by looking at the icons.

All of these measures indicate a passive form of understanding because they concern the understanding of a given model. We are however interested in a more active kind of understanding of a modeling language, this is needed when an analyst uses the language to build models. Such an active concept of understanding is used by, for example, Carvallo & Franch [6] and by Matulevičius

& Heymans [17], who measured the number of mistakes made in constructing i* models, and by Abrahao et al., who measured the time needed to build a model [1].

We define the understandability of a concept for a set of language users in this paper as the percentage of language users who, whenever they use the concept when building a model, use it correctly. Understandability is thus relative to a set of language users. In this paper we measure the understandability of goal-oriented concepts in ArchiMate 2.0 in a sample of 19 language users. From our observations, we draw conclusions about understandability of ArchiMate in general, and of goal-oriented concepts in general, for enterprise architects.

5 Data Analysis

Table 1 summarizes the scores that the 19 enterprise architects received on their homework. The numbers are the percentage of correctly used concepts by each subject. When a subject did not use a concept at all, the corresponding cell contains "na". Subject 1-7 are the subjects from 2011 and subject 8 - 19 are the subjects from 2012. The avg column shows the percentage of users that always uses the concept correctly. Looking at this column we see that only four concepts were used correctly by at least half of the subjects: the concepts of stakeholder, goal, requirement and influence. We now discuss our findings in detail.

Table 1. Understandability of goal-oriented concepts in ArchiMate by a sample of 19 practitioners. Row *i* column *j* shows the percentage of times that practitioner *i* used concept *j* correctly.

Practitioner	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	avg
Stakeholder	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Driver	66,6	100	100	100	77	na	69	50	38	100	55	100	100	100	69	33	100	85	100	47
Assessment	25	8,3	100	44	100	na	13	50	100	100	100	71	100	100	83	90	100	97	100	47
Goal	94	82	100	95	100	92	98	100	100	50	100	100	100	100	100	100	100	100	100	68
Requirement	100	100	100	75	100	na	100	100	0	80	100	91	62	100	100	100	100	95	85	57
Decomposition	0	na	na	100	100	83	24	na	62	na	100	100	100	50	na	na	79	57	15	26
Influence	100	50	na	100	100	100	100	na	100	na	100	100	100	100	100	100	100	100	100	79
Realization	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

5.1 Description of Model Complexity

In total the 19 subjects constructed 246 diagrams and on average the models contained 9 concepts. However, complexity of the models varied. Some diagrams contained as little as 2 concepts, and others contained 35 concepts. Not every diagram contained every concept. This is because ArchiMate uses views to reduce model complexity. There are roughly three kind of views. The first is a stakeholder view, showing the stakeholders, drivers, assessment and initial goals. The second is a goal refinement view showing the modeling of goals, goal influence, goal decomposition and goal realization through requirements. The third view shows the realization of requirements by architecture components. Figure 1

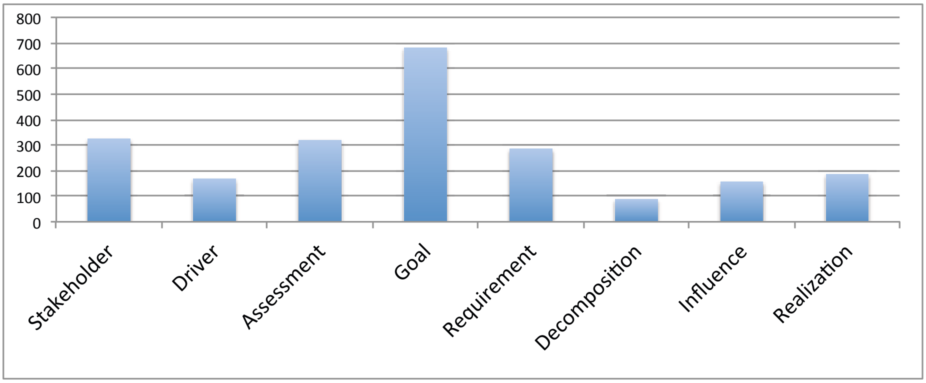

Fig. 1. Frequency of use of goal-oriented concepts in 246 EA models

shows the frequency with which different concepts were used. The most frequently used concepts are those of goal, stakeholder and assessment. We have included an median sized model in the appendix A. This is an actual (translated) model constructed by one of the subjects. It contains more than nine concepts, but that is just an average. The example illustrates the size of the models.

5.2 Analysis of GORE Concepts and Relations in ArchiMate

Stakeholder. The first concept under analysis is the stakeholder concept. This concept is based on definitions from TOGAF, i* and Tropos. TOGAF defines a *stakeholder* as an individual, team, or organization (or classes thereof) with interests in, or concerns relative to, the outcome of the architecture [26]. This seems more general than the definition of actor in i* and Tropos as entities with intentional properties such as goals, beliefs, abilities, and commitments [30]. The motivation extension of ArchiMate adopted the more general definition of TOGAF [9, 27].

In our experiments the stakeholder concept was perfectly understood by every student. There was not a single mistake made in all instances of use. This can be explained by the fact that the TOGAF stakeholder concept is a well known concept by the subjects. Its definition is very clear, and it is substantially different than the other concepts used in the motivation extension, so that it is not easy to confuse the stakeholder concept with any other concept. For these reasons we think this finding will generalize to other ArchiMate users too. To the extent that the concept of actor in i* and Tropos coincides with that of TOGAF stakeholder, we expect that users of i* and Tropos will find the actor concept unproblematic too, and to be able to use it without mistakes.

Driver. The driver concept is not found in the GORE literature, but used in the EA literature. TOGAF defines driver as the key interests that are crucially important to the stakeholders in the system, and determines the acceptability of the system [26]. The motivation extension of ArchiMate adopted this definition [9, 27]. In our experiments only nine subjects (47%) understood the driver concept correctly. The most common mistake made with driver was that it was used as a goal. For example, subjects modeled the goal 'to improve Financial Information' as a driver. But the driver corresponding to this goal is the key interest 'Financial Information'. Apparently the definition of driver is not very clear, and it is so close to the concept of a goal that it generates more confusion than clarity. We see no reason why this would not confuse other practicing enterprise architects, so we think that this finding will generalize to other ArchiMate 2.0 users as well.

Assessment. The concept of an assessment too is based on definitions found in the EA literature. The Business Motivation Model (BMM) defines an assessment as a judgment about some influencer that affects the organization's ability to employ its means or achieve its end [4]. The motivation extension of ArchiMate attempted to make this more concrete by defining an assessment as the *outcome of the analysis of some stakeholder concern* [9, 27]. In our experiments nine subjects used the concept perfectly. The most common mistake was that the assessment concept was used as a goal. For example, correct use of an assessment would be 'the financial information is incorrect'. This is a possible outcome of an analysis of the key stakeholder interest 'financial information'. However the subjects used the concept often to denote a goal like 'improve financial information', just as we saw with the driver concept above. The distinctions between a key interest, an analysis of a key interest, the outcome of the analysis, and the goal motivated by this outcome, were lost on most of our subjects. Moreover, the outcome of an analysis of a key interest can indeed be to 'improve something'. For these reasons we think this confusion will be present in other enterprise architects who use ArchiMate, as well as in users of the assessment concept in BMM.

Goal. The ArchiMate definition of goal is based on a combination of the EA literature and the GORE literature. KAOS defines goals as desired system properties that have been expressed by some stakeholder(s) [28]. This seems more technical and solution oriented than the i* and Tropos concepts of a goal as the intentions of a stakeholder i* [30]. BMM defines goal as a state or condition of the enterprise to be brought about or sustained through appropriate means BMM [4]. ArchiMate defines goal as *some end that a stakeholder wants to achieve* [9, 27]. In our experiments 13 subjects understood the goal concept perfectly. The most common mistakes made were that a goal was either used as a driver or as a requirement. For example, the subjects would write down 'financial information' as a goal, but it should actually be something like 'improve financial information'. When it was used as a requirement, it was written down as 'the system should have 100% availability'. We can reuse our explanation that the distinction between driver, goal and requirement were lost by the subject in that instance. The concept of goal can therefore be understood and used by practicing enterprise architects, but mistakes are made too. Since the

ArchiMate concept of a goal is similar to that in KAOS, i* Tropos and the BMM we expect that this will happen in users of those languages too. This calls for clearer guidelines in the application of the goal concept.

Requirement. We based our definition of the requirements concept on the GORE literature. In KAOS a requirement is a goal assigned to an agent of the software being studied [28]. In GBRAM a requirement specifies how a goal should be accomplished by a proposed system [2]. The ArchiMate motivation extension defines requirement as *some end that must be realized by a single component of the architecture* [9, 27]. In our experiments in total 11 subjects (57%) perfectly understood the concept. In general the requirement concept was reasonably well understood. This can be explained by the fact that it is a well known concept already known in practice by the subjects.

However, there were quite a large number of mistakes. Many subjects specified requirements that were goals not yet allocated to a system. For example, instead of the 'the system should have a financial reports function', they specified the goal 'improve financial reports'. We see again that semantically close concepts are confused by practitioners, even though the definitions of the concepts are clear. We expect this confusion to be present in other users of ArchiMate as well.

The decomposition relation. ArchiMate 2.0 based this relation on a combination of concepts from the EA and GORE literature. i* defines a decomposition as an element that is linked to its component nodes. [30]. BMM uses a similar definition, but it is more aimed at goals. BMM defines decomposition as an end that includes an other end BMM [4]. In Tropos, a parent goal is satisficed if all of its chlidren goals are satisficed [3]. In KAOS the conjunction of all the subgoals must be a sufficient condition entailing the goal KAOS [28].

The motivation extension of ArchiMate defines decomposition as a *some intention that is divided into multiple intention.* [9, 27]. This was understood correctly by only five subjects (26%). When the decomposition relation was used incorrectly, it was used as a influence relation, which in ArchiMate is defined as a contribution relation. For example, correct decomposition of the goal 'improve correctness financial information' should be 'improve correctness financial information regarding outstanding debt AND improve correctness financial information sales'. This decomposes a goal into more detailed goals. However, many subjects decomposed the goal 'improve correctness financial information' into the component goal 'acquire a financial reports system that records sales information'. But this is an influencer, i.e. a new goal that contributes to the original goal. The confusion is probably caused by the fact that satisfaction of an influencer increases the satisfaction of the influenced goal, just as satisfaction of the components increases the satisfaction of the composite goal. Based on this analysis we expect other users of ArchiMate to have similar problems.

The influence relation. In i* a contribution is a link of elements to a soft goal to analyze its contribution [30]. Tropos defines contribution analysis as goals that can contribute positively or negatively in the fulfillment of the goal to be analyzed [3]. ArchiMate defines this as a goal G1 contributes to another goal G2 if satisfaction of G1 influences the satisfaction of G2 positively or negatively [9, 27]. The influence relation was understood correctly by 15 subjects (79%). In the cases were the relation was not used correctly, the subjects linked requirements with goals on a 1 on 1 basis, which amounted to stated the same goal twice. Others used a standard ArchiMate association relation where they should have used an influence relation. To further reduce the misunderstandings of the influence relation, better guidelines must be found.

The realization relation. This relation is based on relations found in the GORE literature. i* defines a means-end relation, which is a relation between an end and a means [30]. KAOS defines a relation for linking requirements to operations [28]. The ArchiMate motivation extension defines the realization relation as a relation that some end that is realized by some means [9, 27]. All subjects used the realization relation correctly. This can be easily explained by the fact that the support tool only allows connecting a requirement to a goal and an architecture element to a requirement so that the relation cannot be used incorrectly. 100% correct use therefore has no implication for (mis)understanding of the concept by tool users.

6 Answers to Research Questions

Q1: How understandable is the motivation extension of ArchiMate by enterprise architects? As shown by the last column of table 1, not all of the motivation extension is understood very clearly.

Q2: Which concepts are understood correctly? Why? Only the stakeholder, goal, requirement concepts and the influence relation were understood by the majority (scoring more than 55%). However the requirement concept was a borderline case where a lot of mistakes were made. Our explanation of this level of understanding is that they are well known concepts already used in practice, and that they have a semantic distance that prevents confusion. However, the distance between requirement and goal is smaller than the other concepts and immediately we saw an a drop in understandability.

Q3: Which concepts are not understood? Why? The concepts of driver, assessment and decomposition were not very well understood. They were often confused with other concepts, such as that of a goal. Our explanation is that drivers, assessments and goals are very closely related and may even overlap, and that the definition of the decomposition relation overlaps with the definition of the influence relation.

Q4: What kind of mistakes are made? Why? The subjects made two types of mistakes. Drivers and assessments were modelled as goals. A driver is related to a stakeholder and an interest area of the stakeholder. A goal is a statement of desire about this interest area. This makes goals and drivers conceptually very close and created confusion in our subjects.

The same is true for the assessment concept. An assessment is the outcome of some analysis. It is not defined what this outcome should be. It can very well be a goal or something else, which is confusing again. The use of the requirement concept to model a goal is similar. Because both concepts are closely related, the difference between desired functionality from the viewpoint of a stakeholder is very much similar to the stated functional need of a system. The only difference is the perspective.

The second type of mistake is that an influence relation was expressed by means of a decomposition relation. Again, the definitions turn out to be too close to each other for many of our subjects.

7 Related Work

The Business Rules Group has published a model that relates business goals and elements found in EA, called the business motivation model [4], which is now an OMG standard. The difference with ArchiMate is that the BMM provides no concrete modelling notations. It provides plans and guidelines for developing and managening business plans in an organized manner, all related to enterprise architecture.

Clements & Bass extend software architecture modelling with GORE, but remove all notational conventions of GORE techniques and return to a classic bulleted list of possible goals and stakeholders [7]. This makes goal-oriented modelling usable for requirements and architecture engineering workshops with practitioners, but does not help to support the kinds of analysis that we mentioned earlier in the introduction.

Stirna et al. describe an approach to enterprise modelling that includes linking goals to enterprise models [24]. However they do not describe concrete modelling notations that are needed to extend existing EA modelling techniques. Jureta and Faulkner [15] sketch a goal-oriented language that links goals and a number of other intentional structures to actors, but not to EA models. Horkhoff and Yu present a method to evaluate the achievement of goals by enterprise models, all represented in i* [13].

An important obstacle to applying GORE to real-world problems is the complexity of the notation. Moody et al. [20] identified improvements for i* and validated the constructs of i* in practice , based on Moody's theory of nottions [18].

Caire et al. [5] also investigated the understandability of i*. They focussed on the ease of understanding of a concept by infering its definition by its visual representation. They had novices design a new icon set for i* and validated these icons in a new case study. This contrasts with our work because they focus on notations and we focus on concepts.

Carvallo & Franch [6] provided an experience report about the use of i* in architecting hybrid systems. They concluded that i* could be used for this purpose for stakeholders and modellers, provided that i* was simplied. Our work extends on these findings. We also found out that related concepts are hard to distinguish (i.e the distinction between driver,assessment,goal, the distinction between requirement and goal and the distinction between decomposition and influence).

Matulevičius & Heymans [17] compared i* and KAOS to determine which language was more understandable. The relevant conclusions for this work were that the GORE languages had ill defined constructs and were there hard to use, GORE languages also lacked methodological guidelines to assist users in using the languages. These conclusions were also found in our work.

Another contrast is that most of the empirical studies of the usability of GORE languages have been done with students, while we do our empirical studies with practitioners.

8 Discussion

8.1 Generalizability

To which extent are our results generalizable beyond our sample of practitioners? In our experiments every subject had at least five years experience, the minimal of a bachelors degree. Enterprise architects usually have the same educational background as our subjects. Our subjects were responsible for translating business strategy and business goals into requirements models and they had to design an enterprise architecture based on these requirements. This is similar to the tasks enterprise architects have to perform in general.

Moreover, the results from this study match with our previous research. In our previous work [10] we reported about a real-world project in which practicing enterprise architects used ArchiMate to redefine an enterprise architecture and link it to changed business objectives. They used the stakeholder and goal concepts as intended. They had some trouble understanding the requirement concept and often formulated requirements as if they were business goals. We also saw that the subjects had a difficult time to see the difference between goals and drivers. The driver concept was too general to use for the subjects. The same was true for the distinction between driver, goal and assessment. Those finding and their explanations agree with the ones reported about in this paper.

All of this justifies the claim that other enterprise architects may understand and misunderstand goal-oriented ArchiMate concepts in the same way as our subjects did. This is a weak generalization, as it says “this can happen more often” without giving any quantification how often it could happen [11]. But such a quantification is not needed to draw some implications for practice, as we do below.

Because the goal-oriented concepts that we used have been taken from other existing goal-oriented languages, we hypothesize that our conclusions may be generalized to those languages too. Again, we cannot quantify this beyond the weak claim that this may happen in those languages too. But we do claim that our findings are sufficiently generalizable to motivate similar research for those languages.

8.2 Validity

Construct validity is the extent to which theoretical constructs are applied and measured correctly in our study. The only theoretical construct that we use is that of understandability, and we defined it in section 4. Our definition agrees with that used by other authors [6, 17] but with that of all other authors. Our definition refers to the number of mistakes made when building models, and not the the amount of time (indicator of effort) required to build the models. Other definitions refer to the number of mistakes or the amount of time needed to answer questions about the models. Comparison of our results with that of studies that use another definition of understandability should be done with caution.

Internal validity is the support for our causal explanations of the phenomena. Could subjects have misunderstood some concepts for other reasons than the ones we hypothesize? For example because they lack competence or because they were explained badly in the training? We cannot exclude these other explanations, but find them less plausible because all subjects had similar background and experience, and because the teachers similarly have several years of experience teaching these concepts. And even if these explanations were true for some subjects, this would not invalidate our explanation in terms of semantic closeness of concepts.

External validity is the support for generalization from our quasi-experiment. Because our explanations do not refer to particular properties of our sample but are stated in terms of the language itself, and because other practitioners are relevantly similar in background and experience to those in our sample, we think our conclusions are generalizable. But we do not claim that they are generalizable to the entire population of practicing enterprise architects, nor to all other goal-oriented languages.

8.3 Implications for Practice

ArchiMate 2.0 is now an Open Group standard, and the concepts we investigated in this paper will remain present in the language. However, one practical implication of this paper is that in future training programs we will not teach all concepts anymore. We will make a distinction between the recommended minimal concepts and less important concepts. We will recommend that future users of the language at least should use the stakeholder concept, the goal concept and the requirement concepts.

A second implication is that we need more practically usable guidelines for the use of the concepts that we do recommend, because other than the goal and realization concepts, we expect that many practitioners will misunderstand and incorrectly apply the basic concepts of goal and requirement and the relations of influence and decomposition. This is a topic for future research.

8.4 Future Research

In addition to work on the guidelines mentioned above, we intend to combine our work with the results of usability and understandability of notations done by Moody [18, 19], Caire [5] and Heymans [17]. The focus of Moody and Caire was on the understandability of the notation. Heymans focussed more on the conceptual use of GORE concepts. If we combine their work with ours we will improve understandability the conceptual and notational level. This could lead to more clearly defined and usable GORE languages.

We also intend to investigate the utility of goal-oriented concepts in ArchiMate as well. They have been added in order to facilitate traceability between business goals and enterprise architecture. Are they actually used this way in practice? We plan to do additional surveys and experiments with practicing enterprise architects to investigate this.

References

1. Abrahão, S., Insfran, E., Carsí, J.A., Genero, M.: Evaluating requirements modeling methods based on user perceptions: A family of experiments. Information Sciences 181(16), 3356–3378 (2011)
2. Anton, A.I.: Goal-based requirements analysis. In: Proceedings of the Second International Conference on Requirements Engineering, pp. 136–144. IEEE (1996)
3. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An agent-oriented software development methodology. Autonomous Agents and Multi-Agent Systems 8(3), 203–236 (2004)
4. Business Motivation Model: Business motivation model version 1.0. Standard document (2007), http://www.omg.org/spec/BMM/1.0/PDF (September 22, 2009)
5. Caire, P., Genon, N., Moody, D., et al.: Visual notation design 2.0: Towards user-comprehensible re notations. In: Proceedings of the 21st IEEE International Requirements Engineering Conference (2013)
6. Carvallo, J.P., Franch, X.: On the use of *i** for architecting hybrid systems: A method and an evaluation report. In: Persson, A., Stirna, J. (eds.) PoEM 2009. LNBIP, vol. 39, pp. 38–53. Springer, Heidelberg (2009)
7. Clements, P., Bass, L.: Using Business Goals to Inform a Software Architecture. In: 18th IEEE International Requirements Engineering Conference, pp. 69–78. IEEE Computer Society Press (2010)
8. Cruz-Lemus, J.A., Genero, M., Manso, M.E., Morasca, S., Piattini, M.: Assessing the understandability of UML statechart diagrams with composite states: A family of empirical studies. Empirical Software Engineering 14(6), 685–719 (2009)
9. Engelsman, W., Quartel, D.A.C., Jonkers, H., van Sinderen, M.J.: Extending enterprise architecture modelling with business goals and requirements. Enterprise Information Systems 5(1), 9–36 (2011)
10. Engelsman, W., Wieringa, R.: Goal-oriented requirements engineering and enterprise architecture: Two case studies and some lessons learned. In: Regnell, B., Damian, D. (eds.) REFSQ 2011. LNCS, vol. 7195, pp. 306–320. Springer, Heidelberg (2012)
11. Ghaisas, S., Rose, P., Daneva, M., Sikkel, K.: Generalizing by similarity: Lessons learnt from industrial case studies. In: 1st International Workshop on Conducting Empirical Studies in Industry (CESI), pp. 37–42. IEEE Computer Society Press (2013)

12. Hadar, I., Reinhartz-Berger, I., Kuflik, T., Perini, A., Ricca, F., Susi, A.: Comparing the comprehensibility of requirement models expressed in use case and Tropos: Results from a family of experiments. Information and Software Technology (2013)
13. Horkoff, J., Yu, E.: Evaluating goal achievement in enterprise modeling – an interactive procedure and experiences. In: Persson, A., Stirna, J. (eds.) PoEM 2009. LNBIP, vol. 39, pp. 145–160. Springer, Heidelberg (2009)
14. Houy, C., Fettke, P., Loos, P.: Understanding understandability of conceptual models - what are we actually talking about? - supplement. Tech. rep., UniversitÃts- und Landesbibliothek, Postfach 151141, 66041 SaarbrÃcken (2013), `http://scidok.sulb.uni-saarland.de/volltexte/2013/5441`
15. Jureta, I.J., Faulkner, S.: An agent-oriented meta-model for enterprise modelling. In: Akoka, J., Liddle, S.W., Song, I.-Y., Bertolotto, M., Comyn-Wattiau, I., van den Heuvel, W.-J., Kolp, M., Trujillo, J., Kop, C., Mayr, H.C. (eds.) ER Workshops 2005. LNCS, vol. 3770, pp. 151–161. Springer, Heidelberg (2005)
16. Kamsties, E., von Knethen, A., Reussner, R.: A controlled experiment to evaluate how styles affect the understandability of requirements specifications. Information and Software Technology 45(14), 955–965 (2003)
17. Matulevičius, R., Heymans, P.: Comparing goal modelling languages: An experiment. In: Sawyer, P., Heymans, P. (eds.) REFSQ 2007. LNCS, vol. 4542, pp. 18–32. Springer, Heidelberg (2007)
18. Moody, D.: The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering. IEEE Transactions on Software Engineering 35(6), 756–779 (2009)
19. Moody, D., van Hillegersberg, J.: Evaluating the visual syntax of UML: An analysis of the cognitive effectiveness of the UML family of diagrams. In: Gašević, D., Lämmel, R., Van Wyk, E. (eds.) SLE 2008. LNCS, vol. 5452, pp. 16–34. Springer, Heidelberg (2009)
20. Moody, D.L., Heymans, P., Matulevičius, R.: Visual syntax does matter: improving the cognitive effectiveness of the i* visual notation. Requirements Engineering 15(2), 141–175 (2010)
21. Nugroho, A.: Level of detail in UML models and its impact on model comprehension: A controlled experiment. Information and Software Technology 51(12), 1670–1685 (2009)
22. Purchase, H.C., Welland, R., McGill, M., Colpoys, L.: Comprehension of diagram syntax: an empirical study of entity relationship notations. International Journal of Human-Computer Studies 61(2), 187–203 (2004), `http://www.sciencedirect.com/science/article/pii/S1071581904000072`
23. Quartel, D.A.C., Engelsman, W., Jonkers, H., van Sinderen, M.J.: A goal-oriented requirements modelling language for enterprise architecture. In: Proceedings of the Thirteenth IEEE International EDOC Enterprise Computing Conference, EDOC 2009, pp. 3–13. IEEE Computer Society Press, Los Alamitos (2009)
24. Stirna, J., Persson, A., Sandkuhl, K.: Participative enterprise modeling: Experiences and recommendations. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 546–560. Springer, Heidelberg (2007)
25. Storrle, H.: On the impact of layout quality to understanding UML diagrams. In: 2011 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 135–142. IEEE (2011)
26. The Open Group: TOGAF Version 9. Van Haren Publishing (2009)
27. The Open Group: ArchiMate 2.0 Specification. Van Haren Publishing (2012)

28. van Lamsweerde, A.: From system goals to software architecture. In: Bernardo, M., Inverardi, P. (eds.) SFM 2003. LNCS, vol. 2804, pp. 25–43. Springer, Heidelberg (2003)
29. Wieringa, R.J.: Design science as nested problem solving. In: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, Philadelphia, pp. 1–12. ACM, New York (2009)
30. Yu, E.: Towards modelling and reasoning support for early-phase requirements engineering. In: Proceedings of the Third IEEE International Symposium on Requirements Engineering, pp. 226–235. IEEE Computer Society Press (2002)

A Example Model

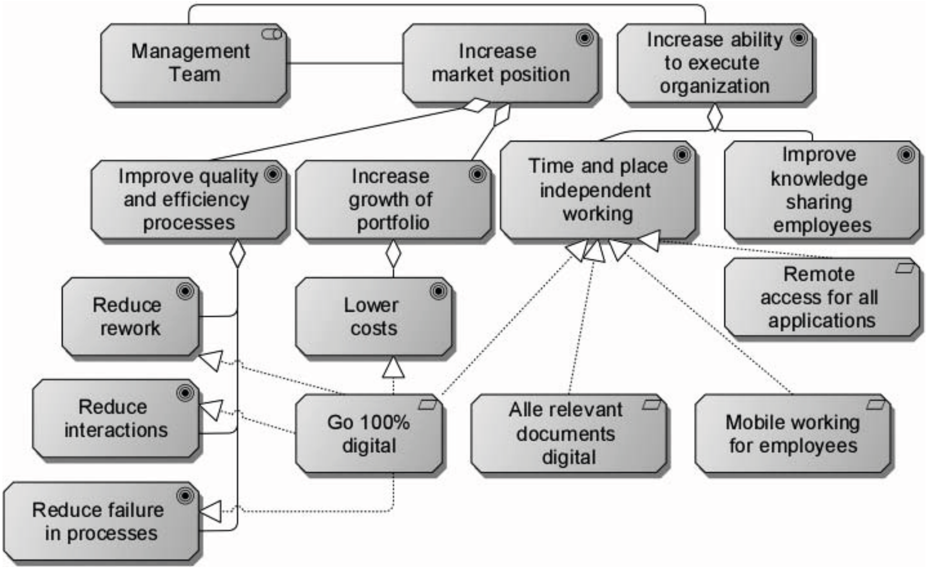
Management Team
Increase market position
Increase ability to execute organization
Improve quality and efficiency processes
Increase growth of portfolio
Time and place independent working
Improve knowledge sharing employees
Reduce rework
Lower costs
Remote access for all applications
Reduce interactions
Go 100% digital
Alle relevant documents digital
Mobile working for employees
Reduce failure in processes

Fig. 2. Example model