# Understandability of Goal Concepts by Requirements Engineering Experts

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

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

**Abstract.** ARMOR is a graphical language for modeling business goals and enterprise architectures. In previous work we have identified problems with understandability of goal-oriented concepts for practicing enterprise architects. In this paper we replicate the earlier quasi-experiments with experts in requirements engineering, to see if similar problems arise. We found that fewer mistakes were made in this replication than were made in the previous experiment with practitioners, but that the types of mistakes made in all the concepts were similar to the mistakes made in our previous experiments with enterprise architects. The stakeholder concept was used perfectly by our sample, but the goal decomposition relation was not understood. The subjects provided explanations for understandability problems that are similar to our previous hypothesized explanations. By replicating some of our earlier results, this paper provides additional support for the generalizability of our earlier results.

## 1 Introduction

In large organizations the gap between business and IT is usually bridged by an enterprprise architecture (EA). An EA is a high-level representation of the enterprise, used for managing the relation between business and IT and to coordinate IT projects. An EA usually contains models of aspects of the business, of IT applications, of the IT infrastructure aspects and of relations between all of these. In addition, in recent years EA has been used to increase the flexibility of the organization and justify the contribution of EA to business goals. This means that EAs are are not only used to manage the relation between business and IT and to coordinate IT projects, but also to determine the impact of changing business goals on the EA and vice versa.

This requires an extension of EA modelling languages with concepts like goals, and support for tracing business goals to EA. In previous work, we have extended the EA modeling language Archimate [18] with concepts from goal-oriented requirements engineering (GORE) [7]. The extension is called ARMOR, and the result of extending Archimate with ARMOR is called Archimate 2.0. So ARMOR is the GORE part of Archimate 2.0. This paper evaluates the understandability of ARMOR.

M. Indulska and S. Purao (Eds.): ER Workshops 2014, LNCS 8823, pp. 97–106, 2014.

In previous work we have investigated the understandability of the ARMOR extension by two case studies [9] and two quasi-experiments with practicing enterprise architects [8]. The results showed that practitioners find ARMOR very complex and use only a few of the concepts of ARMOR correctly.

To test the generalizability of these findings, we have replicated the experiment with participants of the REFSQ '14 conference that can be considered experts in GORE languages[1]. We additionally asked the subjects for the perceived understandability of ARMOR concepts in an exit survey. The results confirm our earlier findings about understandability problems in goal-oriented notations.

We start with listing the research questions in the next section. Next we describe our research methodology in section 3. Section 4 describes our conceptual framework. The results from the experiment, the exit survey and the comparison with our previous results are described in section 5. Answers to the research questions are summarized in section 7 and section 8 discusses related work. Section 9 describes some implications for practice and further research.

## 2 Research Problem

In our courses teaching ARMOR to practitioners we saw that there were understandability issues regarding the concepts. Therefore we started to investigate this problem. This work is a replication of our prevous studies. Our research questions are the same as in our previous quasi-experiments, extended with two more questions. We added a question to compare subjects' perception of understandability with the understanding they exhibited during the experiment. We also added a fifth question in which we ask about the comparison across all quasi-experiments.

- Q1: How understandable is the ARMOR language?
- Q2: Which concepts are understood correctly and why?
- Q3: Which concepts are not understood? Why? Does this agree with subjects' perceptions of understandability?
- Q4: What kind of mistakes are made? Why?
- Q5: How much do our findings differ from our previous samples and why?

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

We performed two identical experiments at REFSQ'14 of 90 minutes each. We could not control any information flow from the first experiment to the second experiment, and we depended on the integrity of the participants, all researchers, to refrain from creating such a flow.

[1] http://refsq.org/2014/live-experiment/

**Table 1.** Entry questionnaire

- What is your highest level of completed education?
- What is your daily function?
- How many years of experience do you have in this function?
- How experienced are you with a (any) requirements modeling notation? (select one: I have no experience / I understand the concepts / I can read diagrams / I can create diagrams / I can teach a requirements modeling technique.)

Subjects self-selected into the experiments, and to be able to assess the influence of previous knowledge of GORE concepts, we measured the knowledge and experience of the participants with GORE notations in a short entry-questionnaire (table 1).

Each experiment started with a very short lecture (30 minutes) on ARMOR. Next, the participants had to construct simple goal models of a case. To allow answering the research questions, the case required all ARMOR constructs to model. But to fit the restricted time available for the modeling exercise (50 minutes), the case was very easy compared to the actual real world problems of our previous experiments.

Finally, before leaving the room, each participant filled in an exit questionnaire in which for each of the GORE concepts used in the assignment, it was asked (1) whether they found the concept easy, normal or hard to use, and (2) to optionally explain their answer.

During data analysis, the answers were graded by the first author in the same way as in the previous experiments. The first author compared the used concepts to intended use of the concepts and marked if the concepts were used incorrectly. Results were discussed with the second author.

## 4 Defining Understandability

In a survey of definitions of understandability of conceptual models, Houy et al. [11] identified five types of definitions: the ability to recall model content, the ability to correctly answer questions about a model,the time needed to answer questions about the model, the ability to solve problems using the model, and the ability to verify a model.These are however measures of model understandability, whereas we are interested in measures of language understandability. An example of a measure of language understandability is the ability of subjects to guess the definition of a language construct by looking at the icons. Caire et al. [5] measured this for i*.

However, these are all measures of passive understanding, whereas we are interested in a more active form of understanding that is closer to the concept of ease of use. How easy is it to construct a model in a language? This concept of understanding is used by, for example, Carvallo & Franch [6] and by Matulevičius & Heymans [13], 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]. Our concept of understandability is close to the first of these, and we define the understandability of a language construct as *the percentage of users that can use a concept correctly.*

*Construct validity* is the validity of the operationalizations of a construct. Note that our definition of understandability is close to that of ease of use, and that our results are therefore about a different concept of understandability than that used when studying understandability of a conceptual model. Our definition agrees with that used by other authors [6, 13], but of the two known operationalizations, correctness of use and time to use, we have selected the first one only. This should be taken into consideration when comparing our results with those of others.

## 5 Observations

There were 18 participants in total, about evenly spread over the two experiments. Two subjects had a bachelor's degree, seven had a master's degree and nine a PhD degree. Furthermore, the majority of the subjects considered themselves experts in requirements engineering in either industry or academia. According to the entry survey 9 out of 18 subjects had the ability to teach requirements engineering notations.

Combining this high level of expertise with the relative simplicity of the assignment, we would not expect any serious understandability problems with GORE notations.

**Table 2.** Data about correct construct usage by the 18 participants

| **Practitioner** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Stakeholder | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| Influence | 94 | 100 |  |  | 100 | 100 | 100 | 100 | 100 | 0 |  | 100 | 100 | 100 |  |  |  |  | 89 |
| Goal | 69 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 0 | 100 | 100 | 100 | 100 | 88 |
| Assessment | 100 | 100 | 85 | 100 | 100 | 100 | 40 | 100 | 100 | 100 | 100 | 100 | 100 | 0 | 100 | 100 | 100 | 100 | 82 |
| Realization | 0 | 100 |  | 100 | 100 | 0 | 100 | 100 | 100 | 100 | 100 | 100 | 33 | 100 |  |  | 100 | 100 | 78 |
| Requirement | 0 | 86 |  | 100 | 100 | 20 | 50 | 100 | 100 | 100 | 67 | 100 | 100 | 100 |  |  | 100 | 100 | 73 |
| Driver | 0 | 100 | 33 | 100 | 100 | 0 | 100 | 100 | 0 | 100 | 100 | 100 | 50 | 100 | 100 | 100 | 100 | 100 | 71 |
| Decomposition | 33 | 100 |  | 0 | 50 | 0 | 12 | 29 | 0 | 25 | 0 | 50 |  | 100 |  | 100 | 67 | 0 | 19 |

Table 2 lists the ARMOR constructs on the left and summarizes the scores that the subjects received on their assignments. Row $i$ column $j$ shows the percentage of times that practitioner $i$ used concept $j$ correctly. 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 is empty. The avg column shows the percentage of users that always used the concept correctly. The rows are ordered from best understood to least understood construct.

Table 3 summarizes the scores of the subjective evaluation of understandability, ordered in the same way as table 2. The numbers are the total number of subjects that found a certain concept easy, normal or hard to use. The final

**Table 3.** Summary of the exit survey

| | Easy | Normal | Hard | Most common explanation |
|---|---|---|---|---|
| Stakeholder | 16 | 1 | 1 | A very common and well known concept. |
| Influence | 5 | 7 | 6 | Unknown when to use it. |
| Goal | 6 | 7 | 5 | Hard to distinguish from driver. Hard to distinguish from requirement. Common concept. |
| Assessment | 7 | 6 | 5 | Difficult to distinguish from a goal. |
| Realization | 5 | 5 | 3 | What is a full realization? |
| Requirement | 5 | 8 | 5 | Very similar to goal. Common concept |
| Driver | 6 | 9 | 3 | Very difficult to distinguish from a goal. |
| Decomposition | 4 | 7 | 7 | Unknown when to use it. |

column summarizes the most frequently occurring explanations provided by the subjects. We now discuss our findings in detail.

The *stakeholder* concept is based on definitions from TOGAF, i* and Tropos [3, 17, 20]. All subjects used this concept correctly and we conclude that the stakeholder concept is an easy to use concept. This is supported by the subjective evaluation of the exit questionnaire. The explanation the subjects provided is that it is a common concept.

The next best understood construct was that of *influence,* defined in ARMOR as a positive or negative influence of satisfaction of one goal on the satisfaction of another goal. This definition is based the influence concept on i* and Tropos [3,20]. 89% of the subjects used the influence relation correctly, but only 5 out of 18 users found the relation easy to use. Participants found it difficult to choose between the decomposition and influence relation. The most common mistake was also that it was used instead of a decomposition.

ARMOR defines a *goal* as some end that a stakeholder wants to achieve, a definition common in the GORE literature [4, 19, 20]. 89% of the subjects used the goal concept correctly. The subjective evaluation shows that subjects still had a hard time using the concept. They found it hard to distinguish from the concepts of driver and of requirement. This is consistent with the types of mistakes made as sometimes drivers or requirements were stated as goals.

ARMOR defines an *assessment* as the outcome of the analysis of some stakeholder concern, a definition based on that of BMM [4]. 83% of the subjects used the assessment concept correctly. However, subjects found the concept was too close to a goal. This is supported by the types of mistakes made by the subjects, assessments were confused with goals.

ARMOR defines the *realization* relation as a relation that some end that is realized by some means, a definition found too in i* and KAOS [19, 20]. 79% of the subjects used the realization relation correctly. This is consistent with the subjective evaluation, where only three subjects found it hard to use. The most common mistake was that it was used to relate two requirements.

ARMOR defines *requirement* as some end that must be realized by a single component of the architecture, a definition found also in KAOS and GBRAM [2, 19]. 69% of the subjects used the requirements concept correctly. The most common mistake was that goals were modeled as requirement. This is consistent with the explanations the subjects provided, that goals and requirements were difficult to distinguish.

A *driver* in ARMOR is that it is a key interest of a user, a definition that is taken from TOGAF [17]. Only 67% of the subjects used the concept correctly, which is consistent with the subjective evaluation. The subjects found it very similar to the concept of a goal. The most common mistake made was indeed that a goal was modelled as a driver.

The ARMOR concept of a *decomposition* is a combination of concepts from the EA and GORE literature [3, 4, 20]. ARMOR defines it as a some intention that is divided into multiple intention. Only 19% used the decomposition relation correctly. This is consistent with the subjective evaluation where only five users found it easy to use. The subjects found it difficult to choose between decomposition and influence.

Some of the data in table 2 are consistent with the subject evaluations of the exit questionnaire. For example, when a subject subjectively found a concept hard to use, often they would not use the concept all. The subjects provided an explanation that the relations were sometimes hard to identify. We believe that therefore they just picked one. This is also the case with the other concepts which were very similar, for example the goal and requirement concept.

There are also discrepancies. For example, 11 subjects found the decomposition relation not hard to use, but only 3 subjects used the relation correctly. Conversely, only 5 users found the influence relation easy to use, but most participants used it correctly. Apparently, perceived understandability does not coincide with understanding.

## 6 Discussion

*Comparison With Our Previous Results.* The level of understanding exhibited by the participants was much higher than in our previous study with practitioners [8]. In our earlier study, only 5 concepts were used correctly by more than half of the practitioners. This agrees with the higher level of expertise of our current group of participants compared to our previous samples.

However, there is a rough correspondence in the orderings of understandability. In our earlier experiment, the concepts of stakeholder and of realization were used correctly by all practitioners. In our current experiment, the concept of stakeholder was used correctly too, but the concept of realization was used incorrectly by some participants, and they perceived some problems in using it. This may be a consequence of the more academic expertise of the subjects.

In all experiments, the concepts of stakeholder, influence, goal and requirement were the best understood (in that order) and the concept of decomposition was the least understood. And in all experiments, participants had trouble distinguishing requirements, assessments and drivers from goals, and participants wondered why all of these concepts are present in the language.

*Explanations.* Our observations support the explanations of understandability problems listed earlier. The number of concepts in ARMOR is large, making it difficult for novice users to choose among them. Related to this is the second explanation, which is that the semantic distance among some concepts is very small, making it even harder to choose the right concept to use in a modeling problem.

Finally, the distance of ARMOR concepts and the meaning of those concepts in daily practice is large in our previous experiments. This explained problems that practitioners had with assimilating ARMOR concepts. For the academics that participated in the current experiment, this distance is smaller, because they teach GORE concepts or have studied them. This may explain the higher scores that the participants in the current experiment had compared to the practitioners' score in the previous experiments.

One factor that affects the internal validity of these explanations is that the explanation of ARMOR given by the first author may have created understandability problems. However, The first author regularly teaches these concepts to practitioners. And to prepare for the current experiment, he has explained the concepts to university colleagues. This should mitigate the threat that understandability problems have been caused by the instructor rather than by the language.

*Generalizability.* Our sample is too small to do any statistical inference. Moreover, the participants self-selected in the sample, which may have biased the results. However, given the fact that our sample consisted of GORE experts who chose to do an assignment with a GORE language, we think that other academic subjects would at least have the understandability problems that we observed in our sample.

We replicated the findings of earlier experiments about most understandable and least understandable concepts, and this supports generalizability too.

Moreover, our explanations in terms of the large number of concepts and the small semantic distance among some concepts, and the need of language users to assimilate new concepts to existing knowledge, are stated in general terms. To the extent that these explanations are generalizable, the phenomena that they explain are generalizable too.

Whether our results generalize to other GORE languages, must be determined by repeating this experiment for these other languages. The question whether all semantic constructs present in i* are really needed has been raised earlier by Moody et al. [15], but it has not yet been answered by empirical research.

## 7 Answers to Research Questions

Q1: How understandable is ARMOR? The last column of table 2 shows the answer to this. Only the stakeholder concept scored 100% an was perfectly understood. However, the only concept that was not clearly understood was that of the decomposition relation, scoring only 19%. The concepts of driver, assessment and goal were very well understood scoring more than 80%. The concepts of requirement, influence and realization were fairly well understood scoring in the 70% range.

Q2: Which concepts are understood correctly and why? Except for the decomposition relation all concepts were understood (scoring more than 55) This can be explained by that most of the concepts are very common concepts.

Q3: Which concepts are not understood correctly and why? There is a gradation in non-understanding, with the decomposition relationship at the bottom. The decomposition relation is very difficult to distinguish from the influence relation.

Q4: What kind of mistakes are made? Why? Does this agree with subjects' perceptions of understandability? The subjects modeled drivers and assessments as goals, and modeled influence relations by means of decomposition relations Explanations were given above. Apparently perceived understandability does not coincide with actual understandability.

Q5: How much do the results differ and why? The results from this study were roughly similar to the results of our previous work. The major difference is that the subjects scored much better than the subjects in our previous experiments This can be explained by the higher expertise level of the current subjects, and the greater simplicity of the assignment compared to the modeling task in the previous experiments.,

## 8 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.

Stirna et al. describe an approach to enterprise modelling that includes linking goals to enterprise models [16]. However they do not describe concrete modelling notations that are needed to extend existing EA modelling techniques. Jureta and Faulkner [12] 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* [10].

An important obstacle to applying GORE to real-world problems is the complexity of the notation. Moody et al. [15] identified improvements for i* and validated the constructs of i* in practice , based on Moody's theory of nottions [14].

Caire et al. [5] also investigated the understandability of i*. They focussed on the ease of understanding of a concept by asking subjects to infer 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 modelers, provided that i* was simplified. 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 [13] 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.

## 9 Implications

### 9.1 Implications for Practice

ARMOR is part of an Open Group standard [18] 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 make a distinction between the recommended minimal concepts such as the concepts of stakeholder, goal, and requirement, and less important concepts, such as those of driver and assessment, that can safely be ignored in practice.

We also have to improve our training material. When we saw that the level of education went up, the number of understandability issues dropped. Somehow we need to compensate some of this with our training material. This can be with practically usable guidelines for the use of the concepts that we do recommend. These guidelines could be tailored to specific experience levels, e.g. develop guidelines for inexperienced participants and different guidelines for experienced participants.

### 9.2 Future Research

In future work we will focus on the traceability aspects of the ARMOR language. Our design goal was to realize traceability between business goals and enterprise architecture. We want to establish that with a minimalized version this is still achieved. Another interesting connection to explore is the relation with Moody's work on the understandability of notations. That work too seems to point at the need for reducing complexity by reducing the number of concepts to be represented in a language.

## 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. (2007) (September 22, 2009), Standard document: http://www.omg.org/spec/BMM/1.0/PDF
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. 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)
8. Engelsman, W., Wieringa, R.: Understandability of goal-oriented requirements engineering concepts for enterprise architects. In: Jarke, M., Mylopoulos, J., Quix, C., Rolland, C., Manolopoulos, Y., Mouratidis, H., Horkoff, J. (eds.) CAiSE 2014. LNCS, vol. 8484, pp. 105–119. Springer, Heidelberg (2014)
9. 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)
10. 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)
11. 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
12. Jureta, I., Faulkner, S.: An agent-oriented meta-model for enterprise modelling. In: Akoka, J., et al. (eds.) ER Workshops 2005. LNCS, vol. 3770, pp. 151–161. Springer, Heidelberg (2005)
13. 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)
14. 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)
15. 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)
16. Stirna, J., Persson, A., Sandkuhl, K.: Participative enterprise modeling: experiences and recommendations. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007. LNCS, vol. 4495, pp. 546–560. Springer, Heidelberg (2007)
17. The Open Group: TOGAF Version 9. Van Haren Publishing (2009)
18. The Open Group: ArchiMate 2.0 Specification. Van Haren Publishing (2012)
19. 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)
20. 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)