Experiment: Understandability of Goal Modeling with ARMOR

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

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

1 Context

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 benefit from modeling 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 requires an extension of EA modelling languages with concepts such as business goal and business value, and support for tracing business goals to EA components.

In previous work [2, 7] we defined a goal-oriented language called ARMOR, based on the concepts found in goal-oriented requirements engineering (GORE). ARMOR has become part of the Open Group EA modeling language standard ArchiMate [9].

1.1 Goals

The main knowledge goal of our experiment is to assess the understandability of ARMOR modeling concepts for its intended users. This should increase our understanding of goal modelling concepts in other GORE languages too. Using this knowledge we would like to improve our course material for ARMOR and to improve the practice of using ARMOR.

This is a replication of an earlier experiment, with important differences. The previous version of the experiment was performed with enterprise architects that had at least five years of experience, and took place during a 8 day course spread out over a period of 12 weeks [3]. The current experiment takes only 90 minutes and participants may not have the same background as those of our previous experiments. We may therefore observe different outcomes in this replication.

1.2 Current Knowledge

This research is part of a larger design cycle in which the ArchiMate language was extended with the ARMOR goal-modeling technique [2]. We performed two initial technical action research studies to validate the ARMOR extension in practice [4]. We learnt that there is a small set of concepts that is easily understood by the subjects and a number of other concepts that are not understood as intended, not used, or misused. The best understood concepts were those of stakeholder, goal and requirement.

In subsequent work we performed two quasi-experiments in which exercises done by practitioners were analyzed for mistakes [3]. These experiments confirmed that practising enterprise architects understand the concepts of stakeholder, goal, and requirement, and do not understand the other concepts very well. The concepts of driver and assessment were often used as a goal. The concept of decomposition was often used as an influence relation. The concept of requirement was also often used as a goal. It is these experiments that we want to replicate, this time with the REFSQ audience.

Houy et al. [5] surveyed the literature for definitions of model understandability and classified them in five types. Houy mainly measures passive understandability, that is the ability to read a model and answer questions about it. We require active understandability, namely the ability to create correct models. We define the understandability of a concept by a set of language users in this proposal 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.

1.3 Benefits to the community

ARMOR incorporates goal-oriented requirements engineering techniques and is used in the practice of EA modeling as part of ArchiMate. If we are able to confirm our results from previous work with a different sample of subjects then this would improve the generalizability of our work. Because each concept in ARMOR occurs in at least one other GORE language, this will reflect on the other GORE languages as well. Identification and explanation of understandability problems will help us improve our teaching methods for ARMOR and other GORE languages.

2 Research Problem

The ARMOR language contains the concepts of stakeholder, driver, assessment, goal, requirement, and the relations of realization, decomposition, influence and association. We operationalize the concept of understandability as the percentage of language users who understand the concept correctly. Our research questions are, then:

Q1: How understandable is ARMOR?
Q2: Which concepts are understood correctly? Why?
Q3: Which concepts are not understood? Why?
Q4: What kind of mistakes are made? Why?

Note that we do not only want to answer the journalistic question what is the case, but also the theoretical question why it is the case.

In our previous work our samples were taken from the population of enterprise architects who work with ARMOR. The sample of this experiment will be more diverse as any participant in REFSQ can participate, and participants consist of junior researchers, senior researchers, and practitioners. We will therefore do a brief pre-test to measure the composition of the sample.

3 Research Design

3.1 Treatment design

The treatment will start with a short pre-test to measure the experience of the subject. The treatment consists of 1 small lesson (30 minutes) of goal modelling. The next 50 minutes are used for modelling assignments. The modelling assignment will comprise of a single assignment with sub questions. We will ask the subjects to model stakeholders, concerns, goals, requirements, the decomposition relation and the influence relation. We will emulate the previous experiments as closely as possible and ask the subjects to construct different model views. One that focuses on stakeholders and goals and one on goal refinement. A post-test will measure the self-perception of participants about how well they now understand the concepts. The subjects will hand in this post test themselves (we will give them the test at the same time as the assignments).

We will provide the subjects with a written case and the subjects will have to elicit goals, requirements and apply the relations. The treatment will end with a short debriefing where we record and link the results from the briefing with the modelling assignments.

3.2 Measurement design

Operationalization To measure experience of a subject we will use the scale:

- no experience
- an understanding of the concepts
- can read diagrams
- can create diagrams
- can teach a requirements modelling technique.

Correctness of the models will be graded by comparing the use of concepts in a diagram with the standard definition of the concepts used in the diagram, and by comparing the solutions of the subjects to a solution we defined earlier One experimenter and two independent experts will correct the results. The corrections will be compared and differences resolved. These independent experts are experienced ARMOR trainers.

We will also measure the time needed to complete an assignment. We will record the starting time of every participant. As soon as they hand in their assignments we will record the end time. We will only record the total time.

We will use a smartphone with clock function to measure the time needed to complete the assignment.

3.3 Inference design

We will provide descriptive statistics of the number of concepts used and the number of mistakes made for each concept. Each of the correctors will also classify the mistakes into types, and we will compare and try to merge these classifications. We will also provide examples of assignments and solutions to the analysis.

We will provide explanations based on semantic analysis to determine the likely reasons why certain mistakes were made. In addition we will use the theories provided by Siau [8].

- Theory on cognition and learning. Without proper assistance the learning curve is too steep.
- Information processing theory. New Knowledge is always mapped to existing knowledge.
- Human cognition theory. Humans are restricted by the size of their short term memory and can only remember a certain set of concepts.

We will use the pre-test and post-test questionnaires to identify possible factors that could have contributed to misunderstandings.

Since this is not a random sample, we will not perform any statistical inferences. Instead, we will try to generalize by similarity to a population of similar subjects. The explanations given earlier will give important information about relevant similarity.

3.4 Required equipment

We will use the following measurement instruments:

- an entry questionnaire to determine the experience of the subjects
- the assignment containing the assignments,
- an exit questionnaire to determine why a subject found a concept hard / easy to use, to record the time
- the assignments
- the solutions to the assignments
- scoring card for each subject
- smartphone with a clock function

The participants need pencil and paper to do the exercises. We will provide pencil, paper and erasers for the participants.

3.5 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.

Our definition refers to the number of mistakes made when building models, and 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.

Internal validity is the support for our causal explanations of the phenomena. Of relevance are all the factors of influence on the outcome. Can subjects have misunderstood some concepts for other reasons than the ones we will hypothesize? For example because they lack experience or because they we explained the constructs badly in the training?

External validity is the support for generalization from our experiment. Our work uses a simplified example, any understandability issues would be even worse in real life. Also if the results from this experiment matches our previous results, this would improve external validity as well.

4 Future Work

- Guidelines for using a lightweight version of ARMOR
- lightweight version of ARMOR for traceability between business objectives and enterprise architecture.

References

1. Carvallo, J.P., Franch, X.: On the use of i* for architecting hybrid systems: A method and an evaluation report. In: The Practice of Enterprise Modeling, pp. 38–53. Springer (2009)
2. 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 (February 2011)
3. Engelsman, W., Wieringa, R.: Understandability of goal-oriented requirements engineering concepts for enterprise architects. In: Advanced Information Systems Engineering (CAiSE), 26th International Conference. Springer (2014)
4. Engelsman, W., Wieringa, R.: Goal-oriented requirements engineering and enterprise architecture: two case studies and some lessons learned. In: Requirements Engineering: Foundation for Software Quality, pp. 306–320. Springer (2012)
5. 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`
6. Matulevičius, R., Heymans, P.: Comparing goal modelling languages: An experiment. In: Requirements Engineering: Foundation for Software Quality, pp. 18–32. Springer (2007)

7. 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, Auckland, New Zealand. pp. 3–13. IEEE Computer Society Press, Los Alamitos (2009)
8. Siau, K., Loo, P.P.: Identifying difficulties in learning UML. Information Systems Management 23(3), 43–51 (2006)
9. The Open Group: ArchiMate 2.0 Specification. Van Haren Publishing (2012)