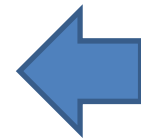


Understandability of Goal Concepts by Requirements Engineering Experts

Wilco Engelsman (BiZZdesign)

Roel Wieringa (University of Twente)

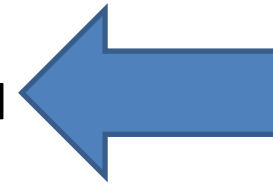


Motivation

- Enterprise Architecture (EA) is a map of the business and its IT used for managing and coordinating IT projects
 - ArchiMate is a commercial EA language
 - ARMOR is an extension with Goal-Oriented Requirements Engineering (GORE) concepts
 - The goal of ARMOR is to facilitate traceability between EA and business goals
- Goal of this research
 - Find out how usable ARMOR is for enterprise architects

Operationalizations of understandability

- Ability to recall model content
- Ability to correctly answer questions about a model
- Time needed to answer questions about a model
- Ability to solve problems using the model
- Ability to verify a model
- Number of mistakes made in constructing a model
- Time needed to construct a model



Previous research

- Two case studies
 - Engelsman used the Armor language in an EA project
 - Architects did so too in another EA project
 - Interviewing the professionals in the case studies revealed understandability problems
- Two quasi-experiments
 - Two groups of architects learned the language and did exercises
 - Correcting the solutions revealed similar understandability problems
- Current paper:
 - Are usability problems similar for RE experts?
 - If they have similar problems for similar reasons, then this is surely due to the language

Two experiments at REFSQ 14

- Two times 90 minutes
 - 30 minutes explanation of the language
 - 50 minutes exercise
 - 10 minutes exit survey
- 18 participants in total
 - 2 BSc, 7 MSc, 9 PhD
 - Some were teaching GORE

ARMOR

- Stakeholder: individual or organization interested in the outcome of the architecture
- Influence: Satisfaction of G1 contributes to satisfaction of G2.
- Goal: Some end that a stakeholder wants to achieve.
- Assessment: Outcome of analysis of stakeholder concern.
- Realization: Some end realized by some means.
- Requirement: End that must be realized by architecture component
- Driver: Key interest of stakeholder
- Decomposition: Intention that is divided into multiple intentions.

Results of REFSQ 14 experiments

- Stakeholder: individual or organization interested in the outcome of the architecture
 - Used correctly by all subjects. Explanation: common concept with universally shared meaning.
- Influence: Satisfaction of G1 contributes to satisfaction of G2.
 - Of those who used it, 89% used it correctly. Subjects found it hard to distinguish from decomposition
- Goal: Some end that a stakeholder wants to achieve.
 - Used correctly by 88% of the subjects. Subjects found it hard to distinguish from the concepts of driver, and of requirement.
- Assessment: Outcome of analysis of stakeholder concern.
 - Used correctly by 83% of subjects. Subjects found it hard to distinguish from a goal.

- **Realization:** Some end realized by some means.
 - Of those who used it, 83% used it correctly. Most subjects found it easy to use.
- **Requirement:** End that must be realized by architecture component
 - Of those who used it, 69% used it correctly. Subjects found goals and requirements hard to distinguish. This was the most common mistake.
- **Driver:** Key interest of stakeholder
 - Used correctly by 67%. Subjects found it hard to distinguish from goal; this was the most common mistake.
- **Decomposition:** Intention that is divided into multiple intentions.
 - Of those who used it, 19% used it correctly. 11 subjects found this concept easily understandable, but only 3 subjects used it correctly.

Comparison with earlier results

- Concept of stakeholder used correctly by all.
- Concept of realization used correctly by all practitioners, but academics had some problems
- Concepts of stakeholder, influence, goal, requirements were best understood by all (in this order)
- Concept of decomposition least understood by all.
- All found requirements goals, drivers and assessments hard to distinguish

Explanations

- Too many concepts creates problems for novices to GORE (the practitioners)
- Small semantics distance among some concepts of ARMOR
- Cognitive fit
- Internal validity threat: problems caused by teacher, not by language

Generalizations: Statistical

- **Statistical generalization** infers properties of a population from properties of a random sample of the population.
 - Our population is the set of RE researchers
 - To get a random sample from a population, you must have a list of elements and then select randomly from this.
 - Our sample is not random but self-selected. Unknown bias.
- There is an unknown population of researchers who would similarly self-select into the sample.
- Our sample is a random selection from that population
- Samples of researchers with a similar bias as ours, will show similar statistics.
 - We could even give 95% confidence intervals for this, but this is not very meaningful.

Generalizations: Extreme case reasoning

- Our sample consisted of experts.
 - So in the entire population of researchers, results will be worse.
- Our sample consisted of researchers
 - So the results will be worse for practitioners

Generalizations: Similarity

- We replicated the ordering of understandability observed in earlier experiments with practitioners.
 - Concepts of stakeholder, influence, goal, requirements were best understood by all (in this order)
 - Concept of decomposition least understood by all.
 - Etc.
- So we can generalize these rankings to the heterogeneous population of practitioners and researchers.

Generalization: Underlying mechanisms

- We have given explanations of these orderings in terms of general mechanisms
 - Semantic closeness
 - Cognitive fit
- This ordering of understandability is as general as the underlying mechanisms that produce it.

Impact on practice

- Archimate now contains Armor.
 - This is an Open Group standard and it will not change.
- So:
 - Improve teaching to practitioners
 - Improve guidelines for practitioners

Impact on research

- Results about Armor may generalize to other GORE languages.
- Simplified GORE language:
 - *Stakeholder* to reason about system scope
 - *Goal* to reason about stakeholder goals and system requirements
 - *Influence* to reason about satisfaction and trade-offs of goals
- Decomposition is an architectural concept