Elicitation Awareness in Conceptual Modeling: The Role of Transparency

i*’15

Julio Cesar Sampaio do Prado Leite
Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
(PUC-Rio)

Support from
Main Goal

Bring elicitation to the front
Main Premise

Elicitation is taken for granted
Facts

• Bias towards Modeling
  - “However, in our educational system it is institutionalized that students get a ‘perfect’ problem description and don’t have to align with anyone in order to solve the problem (other than the odd question of clarification to the instructor).” (Sikkel, Klaas and Daneva, Maya (2011) Getting the client into the loop in information systems modelling courses. In: REET 2011)

• Elicitation through Modeling
  - "In requirements acquisition a preliminary model for the specification of the entire composite system is elaborated and expressed in a “rich” language. This language needs a variety of built-in concepts to structure requirements about the composite system in terms of the kind of abstractions usually found in requirements documents, such as objectives and constraints to be met by the composite system, entities, relationships, events, and actions taking place in it, agents controlling the actions, responsibilities assigned, possible scenarios of system behavior, and so forth." (Dardenne, van Lamsweerde, Fickas, Goal-directed requirements acquisition, In Science of Computer Programming 20 (1993) 3-50 Elsevier (2063 citations)
The role of Transparency

“Transparency is an interesting quality because it makes it necessary to attach requirements models to software”

John Mylopoulos
The role of Elicitation

“There's no sense in being precise when you don't even know what you're talking about.”

John von Neumann
The RE Formula

\[ S, K \vdash R \]
Jackson View

Figure 1.

Specified machine behaviour is about these shared phenomena only.

Requirements are typically about these private phenomena.

Environment properties relate all of these shared and private phenomena, and so relate the requirements to the machine behaviour.

M. Jackson, *The meaning of requirements*
Reinterpretation of Jackson View

Universe of Discourse

The Environment

The Machine

Time

R

K

S

K
The Elicitation Task

*Elicit K and R in order to write S*
Well ... Problems...

- The complexity of the Requirements Problem (Finding a “suitable” S given R and K)
- Qualitative versus Quantitative
- “The Problem is not at the Interface”
- The mappings of NAT and REQ (R and K)
Problems ...Ubiquity

Ubiquitous Software Systems

- Main Ubiquitous Software Systems characteristics:

  Sensible, invisible, sometimes tolerant, heterogeneous, decentralized and interoperable...

  and we still need to assure their quality...

G. Travassos’s keynote at SBQS 2015, Manaus
Problems ... An instance of Ubiquity

Mapping quality requirements for pervasive mobile games
Luis Valente • Bruno Feijó • Julio Cesar Sampaio do Prado Leite
Problems ... The Interface Problem

DFD

( the "problem" at the interface ) M.J.

( the "problem" is not at the interface ) M.J.
Opposite Poles

Independent Elicitation  Model Driven Elicitation
Independent Elicitation

- Use of several MTT to gather information about the Universe of Discourse. Examples: Interviews, Questionnaires, Document Reading, Observation, Ethnography, Reverse Engineering, Reuse ..
- Universe of Discourse has different information sources, ranging from humans to devices.
- Repository for K and R: making the distinction.
- The Completeness Fallacy
Repository for Independent Elicitation
Elicitation Tricks
Model Driven Elicitation

- Use of MTT to fill in the model
- The model operators and operands / nodes and edges are the “things” to elicit.
- K and R are implicit, since S is the target
- Depends on the richness of the model: MER, Use Case Diagram, KAOS, .... Each has a different set of operands and operators / nodes and edges.
- Usually there is more than one model (language) per S.
ER Diagram

https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model
DFD Diagram

Dardenne, van Lamsweerde, Fickas, *Goal-directed requirements acquisition*,
Opposite Poles

- No one is best
- Pros and Cons
  - Time
  - Previous Knowledge Available
  - Another team’s task
  - Coverage
  - Multiple languages for S
- Possible compromise policies
  - Evolution driven
  - Concurrent Engineering
  - Learning Organization
However ...

How to maintain the focus on the triplet K, S, R?
That is: If S is seen as the primary “object” how to avoid loosing contact with both K and R?
Is this just a traceability issue?

We need more.
How to be aware of K and R in S?
Definitions

- "Awareness is the ability to perceive, to feel, or to be conscious of events, objects, thoughts, emotions, or sensory patterns." (merriam-webster)
- Awareness is a fundamental requirement for software that needs to adapt itself to some degree.
  - Self-adapting provides to software the ability to deal with changes in the environment in which the software is inserted.
- The requirement of awareness, in its turn, provides the software the abilities to perceive what is happening in the environment, and "understand" how the environment changes, and how changes in the environment affect its proper functioning.
- Actors playing the role of a modeler needs to be aware about elicitation.
SIG for Awareness

Elicitation Awareness

- Elicitation Awareness by the modeler
- Elicitation Awareness by the model
- Question: how to bring this awareness to the process as well as to the models?
How Transparency may be of Help?

- Quality related to information access
- To be aware: there is a need to be informed
- Transparency improves efficacy of communication, thus helping sensing by humans and tagging the models
- Processes need to be more transparent as to improve human / agent awareness about elicitation
- S needs to be more transparent as to improve its awareness towards K, and R.
Transparency Network

[Diagram of a network with nodes such as Transparency, Usability, Informativeness, Auditability, Accountability, Validity, Controllability, Traceability, Verifiability, Understandability, Detailing, Composability, Decomposability, and dependencies between them.]

---

Portability
Accessibility
Availability
Publicity
Uniformity
Simplicity
Operability
Intuitiveness
Adaptability
Clarity
Completeness
Correctness
Current
Comparability
Accuracy
Integrity
Conciseness
Composability
Decomposability
Help
Focus on Citizens

Focus on Consumers (2)

Focus on Stakeholders (1)

Scope

Stakeholders: shareholders, customers, employees, suppliers.

• Consumers who use goods or service targeted by transparency policies.
Transparency Ladder

- Accessibility
- Usability
- Informativeness
- Understandability
- Auditability

Transparency
Transparency Maturity Model

Transparency Network
So What?

- “Software Engineering is Big Science” V. Basili
- Several of you here have been working in ways to promote more elicitation awareness.
- More is needed, both on the modeling side as well as on the process side.
- I will go over some ideas we have been working.
Software Oriented
Actor Oriented
The Strategic Actor Model

SA Model

∀ agent occupies position X → agent plays
∀ role covered by position X.
Out-of-Bounds Feedback
Information Source Diagram
Intentionality Panel

Antonio de Padua Albuquerque Oliveira: Engenharia Intencional: Um Método de Elicitação, Modelagem e Análise de Requisitos
An Example of Process
Another Process

EXTENDING i* WITH AWARENESS MODELING CONSTRUCTS

• Our proposal is to add, to i* models, abstractions that help software to perceive the environment with its inherent changes and to relate these new abstractions to other elements in the models that determine the software behavior:

**Situation** - is “the state of the real world at a certain moment or during an interval in time at a certain location”[12].

**Context** - is “a mechanism to describe situations by their defining features and group them into one unit” or in other words: “a context is a description of the current situation on an abstract level that can be matched against previously specified situations”[12].

• A description can be constituted by “a number of conditions that can be evaluated to true or false, possibly with an assigned certainty”


Reusing non-functional patterns in i* modeling; H Cunha, JC Sampaio do Prado Leite Requirements Patterns (RePa), 2014 IEEE 4th International Workshop on, 25-32
Example of a Question Pattern for Context-Awareness

Reusing non-functional patterns in i* modeling; H Cunha, JC Sampaio do Prado Leite
Requirements Patterns (RePa), 2014 IEEE 4th International Workshop on, 25-32
EXTENDING i* WITH AWARENESS MODELING CONSTRUCTS

SRConstruct for the Awareness Requirement

Context element

Analysis

Data Acquisition

Data Interpretation

Alternative action 1 (mean)

Alternative action 2 (mean)

Situation A

Situation B

Context Awareness Acquisition

Awareness [Topic]

Goal

agent

role

Plays

Reusing non-functional patterns in i* modeling; H Cunha, JC Sampaio do Prado Leite

Requirements Patterns (RePa), 2014 IEEE 4th International Workshop on, 25-32
## The Pattern Language

<table>
<thead>
<tr>
<th>Awareness requirement name</th>
<th>&lt;The name should follow the rule: Awareness [Topic] &gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Topic description:</strong></td>
<td>&lt;Brief description of the topic (problem domain) related to the awareness requirement&gt;</td>
</tr>
<tr>
<td><strong>Goal:</strong></td>
<td>&lt;The goal impacted by the context&gt;</td>
</tr>
<tr>
<td><strong>Awareness subtype:</strong></td>
<td>&lt;The awareness subtype (from awareness catalog) which is related to this requirement&gt;</td>
</tr>
<tr>
<td><strong>Suggested operationalization</strong></td>
<td>&lt;Suggested operationalization to this requirement. Some of them can be found in awareness catalog &gt;</td>
</tr>
<tr>
<td><strong>Alternative actions:</strong></td>
<td>&lt;The alternative means to achieve the goal impacted by the context&gt;</td>
</tr>
<tr>
<td><strong>Entity:</strong></td>
<td>&lt;The entity in which the context awareness element is anchored (what the context is about)&gt;</td>
</tr>
<tr>
<td><strong>Source of entity data:</strong></td>
<td>&lt;The source from where the entity data will be acquired&gt;</td>
</tr>
<tr>
<td><strong>Context description</strong></td>
<td>&lt;List of variables that enables the situations identification&gt;</td>
</tr>
</tbody>
</table>

### Domains of variable

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Domain</th>
</tr>
</thead>
</table>

### Context situations specification

<table>
<thead>
<tr>
<th>Situation name</th>
<th>Specification</th>
</tr>
</thead>
</table>

### Alternative action choice

<table>
<thead>
<tr>
<th>Situation</th>
<th>Alternative action</th>
<th>Impact</th>
</tr>
</thead>
</table>

---

Reusing non-functional patterns in i* modeling; H Cunha, JC Sampaio do Prado Leite
Requirements Patterns (RePa), 2014 IEEE 4th International Workshop on, 25-32
Transparency Architecture
Thanks

- This is a team work
- Several present and past collaborators
- Special thanks to the i* community
- www.inf.puc-rio.br/~julio