Surpassing the Function Perspective: The Complexity of Goal Modeling

A Research Celebration for Professor John Mylopoulos

June 27, 2009

Julio Cesar Sampaio do Prado Leite
Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
(PUC-Rio)
I'll need to know your requirements before I start to design the software.

First of all, what are you trying to accomplish?

I'm trying to make you design my software.

I mean what are you trying to accomplish with the software?

I won't know what I can accomplish until you tell me what the software can do.

Try to get this concept through your thick skull: the software can do whatever I design it to do!

Can you design it to tell you my requirements?
A Very Hard Problem

We are dealing with a very difficult problem, as expressed by Fred Brooks:

"The hardest part of building a system is deciding what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all interfaces to people, to machines and to other software systems. No other part is more difficult to rectify later"

Software Quality

Peter Freeman’s taxonomy

basic quality
- functionality,
- reliability,
- ease of use,
- economy and
- safety

extra quality
- flexibility,
- reparability,
- adaptability,
- reparability ,
- understandability,
- documentation and
- enhanceability
Quality is Job One

• Several accounts regarding software failure point to the lack of attention in dealing with other qualities.

• An analysis of the LAS is typical. Several problems can be traced for the lack of consideration of quality issues at planning time.

• Other disciplines focus on quality as the customer requirements.
LAS – Case Study

“Multitude of Opinions – the social organization surrounding the development of the LAS system was a very complex one. A great number of actors were involved, to some degree, with the system and its deployment. To name just a few there were ambulance crews, ground staff and management, each party with different opinions in relation to the LAS system. This section takes a deeper look into this situation and its consequences. There are 19 passages referenced in this section.

Evolution – in this section we take a closer look at the system in order to understand the evolution of events that ultimately resulted in its downfall. Based on some key issues we tried to uncover the rationale behind design decisions and take a look at documentation. Issues such as changes to the software, specifications (both hardware and software) and technology trade-offs are among the ones addressed. There are 17 passages referenced in this section.

Environmental aspects – this section concentrates in the social aspects surrounding the development and deployment of the system. Issues such as company policies, regulations and the impact the system might have in the organization are central to this discussion. There are 19 passages referenced in this section.

Non-Functional aspects – this section is concerned with aspects that although not directly related to the system, played an important role in its outcome. We believe one of the most significant ones is the communication problem. The system did not take into account that in an environment as crowded as London, radio communications are sometimes difficult, and that resulted in a series of reported malfunctions. Less critical issues, such as cost and the trade-off between desired performance and user interface, among other problems, are identified and addressed in this section. There are 28 passages referenced in this section.”

QFD

“In the area of engineering and management, the well known QFD (Quality Function Deployment) strategy [7] distinguishes positive quality from negative quality: “QFD is quite different in that it seeks out both "spoken" and "unspoken" customer requirements and maximizes "positive" quality (such as ease of use, fun, luxury) that creates value. Traditional quality systems aim at minimizing negative quality (such as defects, poor service)”. One of the techniques used by QFD strategies is the House of Quality [8], in which the process starts "...with the customer, whose requirements are called customer attributes (CA´s) - phrases customers use to describe products and product characteristics...". Incidentally, none of the examples of the CA´s in [8] is related to functionality or just functionality alone.”

Function First

- Most of the representations for software specification stress functionality.
- Software requirements is anchored on functionality.
- Even when using non-procedural paradigms the center piece is functionality.
- Representation is “flat” (one dimension).
The Legacy of Automata Theory

“(3) Assuming that formal descriptions focus on actions, it is essential to identify which actions are controlled by the environment, which actions of the environment are controlled by the machine, and which actions of the environment are shared with the machine. All types of actions are relevant to requirements engineering, and they might need to be described or constrained formally. If formal descriptions focus on states, then the same basic principles apply in a slightly different form.”

Figure 10. Alternatives: "Button-based interface provided"
Figure 11. Button-based interface provided

Qualitative goals have been added to the previous diagram in order to analyse several alternate designs.
NFR - UML
NFR - UML

(a) LightGroup

(b) Safety
- [Dim Light]
- [Dim Light, illumination >= 14 lux]
- [Light Group, Calculate LuxValue]
- [Light Group, LuxValue]

(c) LightGroup

(d) Safety
- [Control System]
- [Control System, Control System Active]
- [Control System Active, T4]
- [Control System Active, Light Group Dimmable lights to 100%]

QFD

Dr. Crawford's Slides on QFD

http://www.me.utexas.edu/~me366j/QFD/Notes.html
### QFD

#### House of Quality

**Customer Needs**
- Easy to close from outside: 5
- Stays open on a hill: 3
- Easy to open from outside: 2
- Doesn't kick back: 2
- Doesn't leak in rain: 2
- Isolation: 2
- No road noise: 1

**Engineering Requirements**
- Torque to close door
- Closing force, level ground
- Closing force on 10% slope
- Torque to open door
- Peak closing force
- Door seal resistance
- Acoustic trans. window
- Road noise reduction
- Water resistance

**Correlation**
- High positive
- Positive
- High negative
- Negative

<table>
<thead>
<tr>
<th>Customer Needs</th>
<th>Engineering Requirements</th>
<th>A s car</th>
<th>B s car</th>
<th>Our car</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to close from outside</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Stays open on a hill</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Easy to open from outside</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Doesn't kick back</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Doesn't leak in rain</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Isolation</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>No road noise</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

**Object Target Values**
- ft-lb
- lb
- lb-ft
- %
- dB
- psi

<table>
<thead>
<tr>
<th>Objective Measures</th>
<th>Our car door</th>
<th>A s car door</th>
<th>B s car door</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective Measures</td>
<td>11 12 6 10 18</td>
<td>9 12 6 9 13</td>
<td>9.5 11 7 11 14</td>
</tr>
<tr>
<td>Technical importance</td>
<td>Absolute</td>
<td>Relative</td>
<td></td>
</tr>
<tr>
<td>Technical measures</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Defect Detection and Prevention

Requirements, Failure Modes, and PACTs (Preventions, Analyses, process Controls, and Tests).

Quantitative Risk-based Requirements Reasoning; M.S. Feather & S.L. Cornford; Requirements Engineering Journal (Springer); Vol 8 No 4, 2003 pp 248-265
Defect Detection and Prevention

Fig. 8 Topology of a DDP model

Quantitative Risk-based Requirements Reasoning; M.S. Feather & S.L. Cornford; Requirements Engineering Journal (Springer); Vol 8 No 4, 2003 pp 248-265
Information Models Ontologies

- Static ontology – checking constraints (spatial reasoning)
- Dynamic ontology – simulation finite state machine, model checking
- Intentional ontology – goal (and/or) graphs analysis
- Social ontology – means-ends analysis

The Evolution

- Roussopoulos, N., Mylopoulos, J., "Using Semantic Networks for Database Management", Proceedings of the First International Conference on Very Large Databases, September 1975, 144-172. (PDF)


- Greenspan, S., Mylopoulos, J. and Borgida, A., "Capturing more world knowledge in the requirements specification", in the Proceedings of the Sixth International Conference on Software Engineering, Tokyo, 1982. (PDF)


Languages – The Evolution

- Taxis
- RML
- Telos
- i*
- Tropos
Aspects

V-Graph

Aspects (cross)

Fig. 30. Security model

Lyrene Fernandes da Silva, Julio Cesar Sampaio do Prado Leite
An Aspect-Oriented Approach for Requirements Modeling (submitted)
Variability

Variability

Table 3: Propagation algorithm values. $SG_x$ are softgoals, and $G_x$ are correlated goals [7].

<table>
<thead>
<tr>
<th></th>
<th>$SG_3$, $SG_4$</th>
<th>$SG_5$, $SG_6$</th>
<th>$SG_7$, $SG_8$</th>
<th>$SG_9$, $SG_10$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sat ($SG_1$)</strong></td>
<td>Sat ($SG_1$)</td>
<td>Sat ($SG_1$)</td>
<td>Sat ($SG_1$)</td>
<td>Sat ($SG_1$)</td>
</tr>
<tr>
<td><strong>Den ($SG_1$)</strong></td>
<td>Den ($SG_1$)</td>
<td>Den ($SG_1$)</td>
<td>Den ($SG_1$)</td>
<td>Den ($SG_1$)</td>
</tr>
</tbody>
</table>

Variability

Figure 19. With almost 4000 variants the comparison is very complex (presented with minimum zoom to show all).

From a Goal Model to Design Views
... to a Software Architecture
An Actor Dependency Model

- Initiator
  - ContributeToMtg
  - UsefulMtg (actor)
  - CalendarInfo
  - ScheduleMtg (resource)

- Participant
  - AttendMtg (task)
  - SuitableTime

- Scheduler

John Mylopoulos, Goal-Oriented Requirements Engineering: Part II
14th IEEE Requirements Engineering Conference
Minneapolis, September 15, 2006
Impact Analysis

Herbet de Souza Cunha: Uso de estratégias orientadas a metas para modelagem de requisitos de segurança, Dissertação de Mestrado, PUC-Rio, 2007.
Impact Analysis

Figure 3. The Business of Technology with Malicious Parties

Softgoal Driven Model

Figure 2. Section of the i* model for the General Chair actor
∀ agent occupies position X → agent plays
∀ role covered by position X.
Process Transparency
The Rationale

Software is deemed transparent if it makes the information it deals with transparent (information transparency) and if it, itself, is transparent, that is it informs about itself, how it works, what it does and why (process transparency).
Softgoal Interpededence Graph
The role of Requirements Engineering

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

Professor John Mylopoulos
WHY?
"... We're not sure," the pet-shop owner replied. "But the other two call him Professor."
WHY?
Looking Beyond
Aigaio Pelagos

By Marcel Germain

Quality in a classical Greek sense is how to live with grace and intelligence, with bravery and mercy.

Theodore White
http://www.flickr.com/photos/8140750@N06/1237871758/