Very Lightweight Requirements Modeling

Martin Glinz
Department of Informatics, University of Zurich, Switzerland
glinz@ifi.uzh.ch

Abstract—**We argue for the creation and use of a very lightweight requirements modeling language as an alternative to textual and pictorial requirements specifications.**

I. Introduction and Motivation

Despite all effort that went into the development of requirements modeling languages, the vast majority of requirements specifications created today are still written in natural language, augmented with tables, pictures, and, increasingly, some model diagrams. This situation is not just due to the inability of industry to adopt modeling technology. It is a strong indicator that heavyweight modeling languages such as UML don't fit the needs of industrial requirements engineers [1]. Moreover, requirements at an early stage are by their very nature mainly narrative and pictorial.

This situation motivates us to propose the creation and use of a very lightweight modeling language (or VLML, for short), a small language with little formal expressive power, but one that easily integrates with natural language, helps structure a natural language specification and provides simple modeling constructs for those things that people hate to express textually: structure, relationships, influence, and flow. On the other hand, the envisaged language shall be constructed such that powerful tool support for editing, navigating [6], and analyzing specifications is possible.

We don't aim at replacing heavyweight modeling languages such as UML or ADORA [2] or competing against lightweight formal modeling languages such as Alloy [5], but at improving the huge number of textual specifications that have no structure beyond a section-subsection classification and are not analyzable beyond careful reading.

In this position paper, we make a case for such a modeling language and present a case study. For more details see [3].

II. The Case For Very Lightweight Modeling

A VLML must provide strong support for writing textual requirements and drawing pictures. However, it also shall harness the power of modeling for overcoming the greatest weakness of text and pictures: their unstructuredness and total informality. We illustrate the power of a VLML with a case study (Fig. 3), using a preliminary VLML design as a sample. Fig. 1 summarizes the visual syntax.

A model in this language is a set of *objects* that may be specialized by *modifiers* and can have *relations* among each other. Technical items can have a *hierarchical* inner *structure*. The *context* of an object is given by its embedding

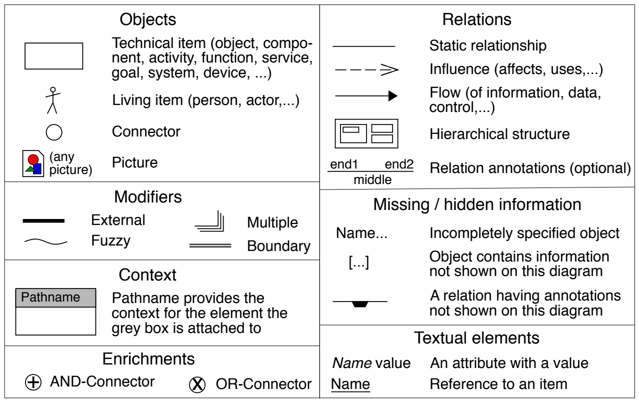
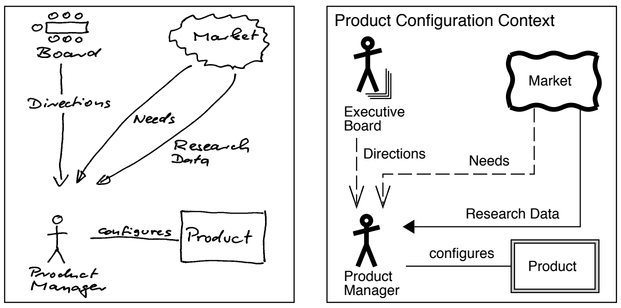

Objects
Technical item (object, component, activity, function, service, goal, system, device, ...)
Living item (person, actor,...)
Connector
(any picture)
Picture
Modifiers
External
Fuzzy
Multiple
Boundary
Context
Pathname
Pathname provides the context for the element the grey box is attached to
Enrichments
AND-Connector
OR-Connector
Relations
Static relationship
Influence (affects, uses,...)
Flow (of information, data, control,...)
Hierarchical structure
end1 end2
middle
Relation annotations (optional)
Missing / hidden information
Name...
Incompletely specified object
[...]
Object contains information not shown on this diagram
A relation having annotations not shown on this diagram
Textual elements
Name value
An attribute with a value
Name
Reference to an item

Figure 1. The visual syntax of a sample VLML



Automatic inference of items such as Product Manager should be possible, while others (e.g., recognizing Board as a group of actors) will require human guidance.

Figure 2. Converting sketches into VLML models

in a hierarchical structure or an explicit context name path. An object may have multiple contexts. *Attributes* provide information analyzable by tools. The basic objects and relations may be *enriched* with additional semantics. Hierarchical structure and options for hiding information from diagrams provide powerful *abstraction* capabilities that can be exploited by tools [2] [6]. We also envisage tool support for converting *sketches* into models (Fig. 2).

Our work on very lightweight requirements modeling is still in a rather preliminary stage. We hope that this position paper will stir the discussion on VLMLs and motivate other researchers to contribute critique and ideas.

References

[1] M. Glinz, Problems and Deficiencies of UML as a Requirements Specification Language, *10th Int'l Workshop on Software Specification and Design*. San Diego, 11–22, 2000.

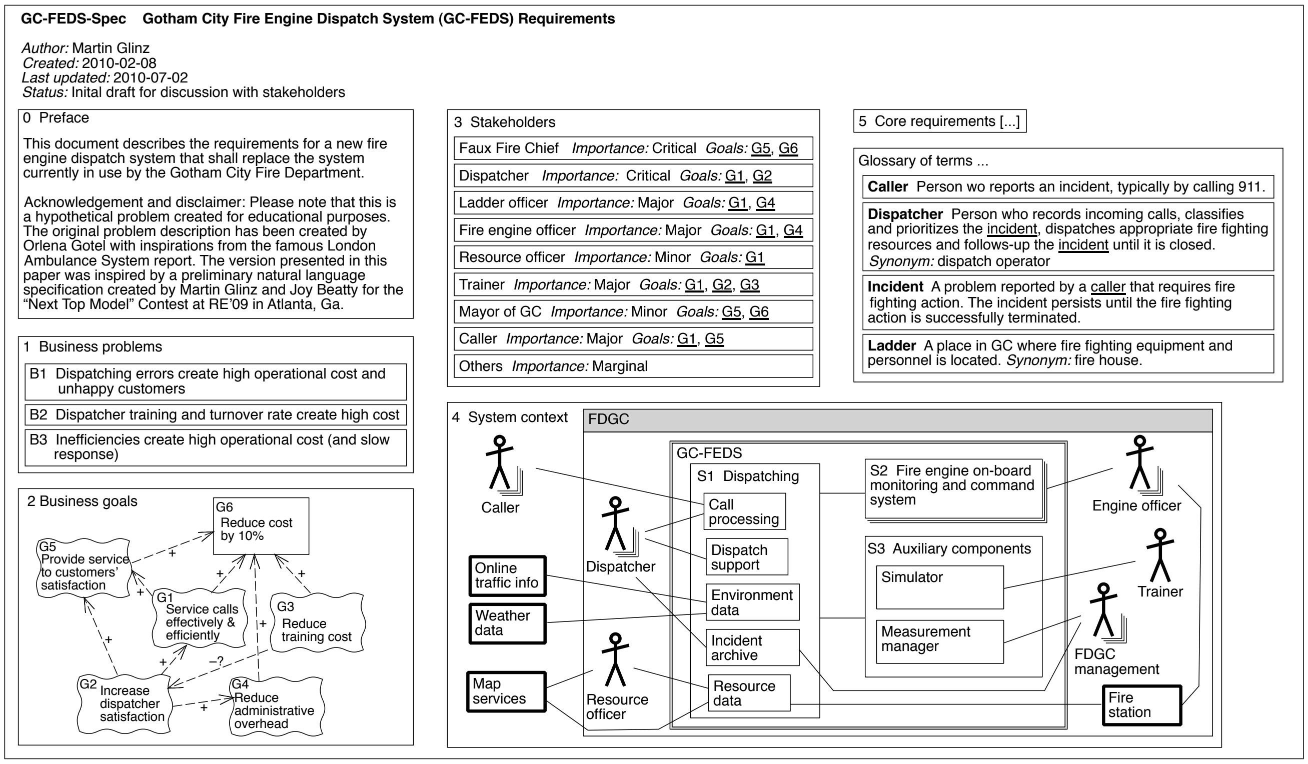
[2] M. Glinz, S. Berner, and S. Joos, Object-Oriented Modeling with ADORA, *Information Syst.*, vol. 27, no. 6, 425–444, 2002.

[3] M. Glinz, D. Wüest, A Vision of an Ultralightweight Requirements Modeling Language. TR IFI-2010.06, Univ. of Zurich.

[4] O. Gotel, GC-FEDS *Replacement System: Request for Requirements Modelers*. Case description distributed at the Next Top Model Contest at RE'09. http://www.gotel.net/ntm/brief.htm

[5] D. Jackson, Alloy: A Lightweight Object Modelling Notation. *ACM TOSEM* vol. 11, no. 2, 256–290, 2002.

[6] T. Reinhard, S. Meier, R. Stoiber, C. Cramer, M. Glinz, Tool Support for the Navigation in Graphical Models. 30th Int'l. Conf. on Software Engineering (ICSE'08), 823–826, 2008.



GC-FEDS-Spec **Gotham City Fire Engine Dispatch System (GC-FEDS) Requirements**

Author: Martin Glinz
Created: 2010-02-08
Last updated: 2010-07-02
Status: Inital draft for discussion with stakeholders

0 Preface

This document describes the requirements for a new fire engine dispatch system that shall replace the system currently in use by the Gotham City Fire Department.

Acknowledgement and disclaimer: Please note that this is a hypothetical problem created for educational purposes. The original problem description has been created by Orlena Gotel with inspirations from the famous London Ambulance System report. The version presented in this paper was inspired by a preliminary natural language specification created by Martin Glinz and Joy Beatty for the "Next Top Model" Contest at RE'09 in Atlanta, Ga.

1 Business problems

B1 Dispatching errors create high operational cost and unhappy customers

B2 Dispatcher training and turnover rate create high cost

B3 Inefficiencies create high operational cost (and slow response)

2 Business goals

G6 Reduce cost by 10%
G5 Provide service to customers' satisfaction
G1 Service calls effectively & efficiently
G3 Reduce training cost
G2 Increase dispatcher satisfaction
G4 Reduce administrative overhead
+
+
+
+
+
+
+
+
−?

3 Stakeholders

Faux Fire Chief	*Importance:* Critical	*Goals:* G5, G6
Dispatcher	*Importance:* Critical	*Goals:* G1, G2
Ladder officer	*Importance:* Major	*Goals:* G1, G4
Fire engine officer	*Importance:* Major	*Goals:* G1, G4
Resource officer	*Importance:* Minor	*Goals:* G1
Trainer	*Importance:* Major	*Goals:* G1, G2, G3
Mayor of GC	*Importance:* Minor	*Goals:* G5, G6
Caller	*Importance:* Major	*Goals:* G1, G5
Others	*Importance:* Marginal	

5 Core requirements [...]

Glossary of terms ...

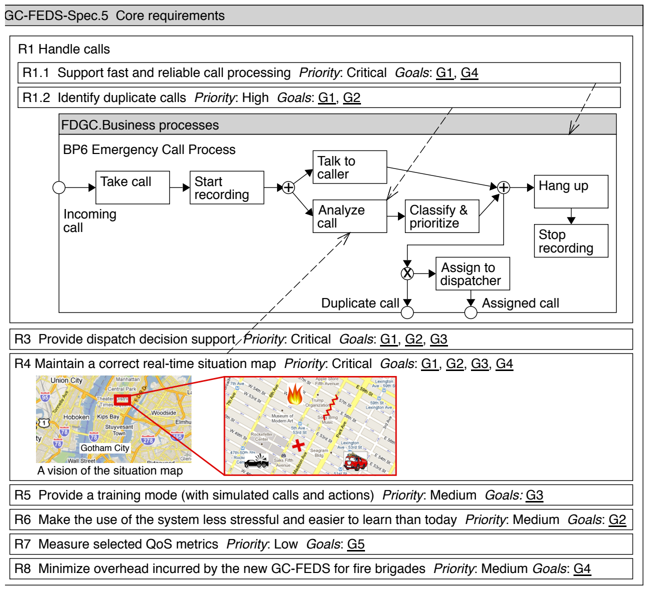
Caller Person wo reports an incident, typically by calling 911.

Dispatcher Person who records incoming calls, classifies and prioritizes the incident, dispatches appropriate fire fighting resources and follows-up the incident until it is closed. *Synonym:* dispatch operator

Incident A problem reported by a caller that requires fire fighting action. The incident persists until the fire fighting action is successfully terminated.

Ladder A place in GC where fire fighting equipment and personnel is located. *Synonym:* fire house.

4 System context



GC-FEDS-Spec.5 Core requirements

R1 Handle calls

R1.1 Support fast and reliable call processing *Priority*: Critical *Goals*: G1, G4

R1.2 Identify duplicate calls *Priority*: High *Goals*: G1, G2

FDGC.Business processes

BP6 Emergency Call Process

R3 Provide dispatch decision support *Priority*: Critical *Goals*: G1, G2, G3

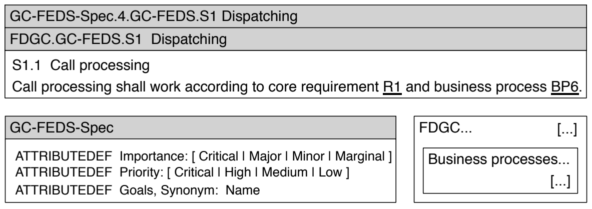
R4 Maintain a correct real-time situation map *Priority*: Critical *Goals*: G1, G2, G3, G4

R5 Provide a training mode (with simulated calls and actions) *Priority*: Medium *Goals*: G3

R6 Make the use of the system less stressful and easier to learn than today *Priority*: Medium *Goals*: G2

R7 Measure selected QoS metrics *Priority*: Low *Goals*: G5

R8 Minimize overhead incurred by the new GC-FEDS for fire brigades *Priority*: Medium *Goals*: G4



GC-FEDS-Spec.4.GC-FEDS.S1 Dispatching

FDGC.GC-FEDS.S1 Dispatching

S1.1 Call processing

Call processing shall work according to core requirement R1 and business process BP6.

GC-FEDS-Spec

```
ATTRIBUTEDEF  Importance: [ Critical | Major | Minor | Marginal ]
ATTRIBUTEDEF  Priority: [ Critical | High | Medium | Low ]
ATTRIBUTEDEF  Goals, Synonym:  Name
```

FDGC... [...]

Business processes... [...]

EXPLANATIONS

GC FEDS-Spec is an object representing a document. Its top-level nested objects serve as an organizational structure.

In the Business goals object, nested objects are arranged in a diagram, in this case a goal graph. The 'fuzzy' modifier is used to denote soft goals. Relation annotations are used to indicate positive and negative influence.

The details of the Core requirements object are hidden from the overview diagram. The '[...]' marker indicates that more details are available.

The Glossary of terms object is incomplete (name followed by '...'). The currently available information is displayed in full (no '[...]' marker).

Italics indicate *attributes* that are followed by values. Underlining a word (e.g., G2) indicates a reference to an item with that name.

In the System context object, modifiers mark the context boundary of the GC-FEDS system and indicate which elements are external to it.

The object BP6 belongs to two hierarchies: (i) to Core requirements by embedding, (ii) to FDGC.Business processes by an explicit context path.

The Call processing object is shown in isolation. It is contextualized by the context paths of the two hierarchies that this object is embedded in.

This specification is based on a hypothetical letter in which the Fire Chief of the Gotham City Fire Department asks a requirements engineering consultancy company for help [4]. He wants a replacement for his current fire engine dispatch system. The specification captures the initially provided information for a stakeholder meeting.

Figure 3. Case Study: The Gotham City fire engine dispatch problem