Analysis beyond UML

Christiane Stutz, Johannes Siedersleben, Dörthe Kretschmer, Wolfgang Krug
sd&m Research

1. Introduction

In spite of being a de facto standard for analysis and design, UML has some obvious shortcomings: the UML definition is at best semi-formal, the set of result types is far too large and heterogeneous, the tool-support is not satisfactory.

If this is true - how do people get on with UML? How do they use it in their every day work? At sd&m (a medium size software company in Munich, Germany) we conducted a study of best practices. The study was restricted to analysis; we considered neither requirements specification (gathering requirements) nor design issues. The aim of the study was to answer the following questions:

- What does a typical analysis documentation contain?
- Which parts of UML are really used?
- What kinds of non-UML documentation is used?

Here is the surprising result: We identified 17 modules a typical analysis documentation consists of. These will be briefly described in the following sections. Only 3 out of 17 modules use UML at all. For the other modules, projects have defined their own standards, sometimes supported by cute home-made tools, while text takes up the major part of the documentations.

2. Overview of the analysis modules

Organized by sections the analysis modules are:

- Functional Requirements: *Business Processes*, *Use Cases* and *Analysis Functions*
- Data: *Data Model* and *Data Types*
- User Interface: *Dialogs*, *Printed Outputs* and *Batch*
- Environment: *Interfaces to Up/Downstream Systems*, *Data Migration* and *Cut Over*
- Essentials: *Goals, Scope, Constraints (GSC)*, *Non-Functional Requirements*, *Cross-Cutting Concerns*, and *Technical Architecture*
- Miscellaneous: *Reading Instructions* and *Glossary*

Please notice that these modules reflect the requirements for developing large information systems as this is the focus of projects at sd&m.

In the functional requirements sections, only UML use case diagrams are used. These diagrams are supported by text describing the preconditions, normal flow, variances, fault scenarios and postconditions.

We do not use activity diagrams for business processes. A business process is a workflow involving different parts of the organization and crossing software boundaries. Using objects to indicate used data and having no elements for systems involved in the process, activity diagrams are not suitable for the analysis. Hence, we prefer the ARIS notation.

In compliance with object oriented analysis UML provides object methods only but no functions outside of objects. Keeping in mind that the goal of the analysis is a proper description of the requirements that is readable for users and developers, it is not relevant to specify the functions as methods for a class. The definition of an analysis function suits better for this purpose.

In the data section we use the UML class diagram with classes, attributes and associations for the data model. As we do not use methods during analysis the data models resemble entity-relationship models.

UML does not provide special elements for data types. You may use stereotypes to indicate data types in a class model but we rather recommend a text description in this module.

The only UML support in the user interface section are interaction diagrams for the specification of the dynamics of dialogs. There are still many more aspects in dialog specification like layout of the screens and used data for which we need a description with layout graphics and text.

For all other modules, UML does not provide suitable description elements. Text templates have been developed for their documentation.

3. Conclusions

The analysis consists rather of text documents than UML models. We have to meet the challenge of managing large semi-formal documents. Rules for the transformation from informal documents to formal ones are needed. With the analysis modules we provide a documentation framework for large software projects as an instrument for documentation and communication between teams. A framework like this is needed in large projects with a heavy-weight process as well as in agile software development with many teams.

Proceedings of the IEEE Joint International Conference on Requirements Engineering (RE'02)
1090-705X/02 $17.00 © 2002 IEEE