UML IN DESIGN PHASE OF DECA

TABLE OF CONTENTS

1. INTRODUCTION

This document is intented to describe the use of Unified Modeling Language (UML) in the designing of Document Commenting, Editing and Analysis System (DECA) developed by DevTeamware.

2. UNIFIED MODELING LANGUAGE (UML)

2.1 Overview

The Unified Modeling Language (UML) is a language for specifying, constructing, visualizing, and documenting the artifacts of a software-intensive system.

UML provides users a ready-to-use, expressive visual modeling language, which is a standard way to develop and exchange meaningful models. UML is independent of particular programming languages and development processes.

Since Document Editing, Commenting and Analysis System (DECA) will be implemented in an object-oriented language Java, UML with it's outstanding OO modeling features is a natural choice as a modeling language in this project.

2.2 Tools

There is a wide range of UML tools in the market, almost everything for each platform. Since part of the development team have had some experience in working with Rational Rose, we chose to evaluate it. In addition, a little lighter UML modeling tool, Object Insight's JVision was evaluated.

3. APPLYING UML IN THE DESIGNING OF DECA

3.1 General

UML was applied in the first phase of DECA system. Use case diagrams were constructed to illustrate the use cases specified in the requirement analysis document.

DECA technical architecture design was split into two sections. Since we chose not to use a relational database as the information storage, there was no need to separately design

the database, even though UML has the means to do it. Technical architecture design was focused on creating a functional Java class hierarchy

3.2 Modeling class hierachy

The class hierarchy modeling was an iterative process. The first layout of the object hierarchy was formed in design team meetings by drawing on a white board. These raw figures were transformed into UML.

Class hierarchies were split into two kinds of diagrams, inheritance and dependency diagrams. This made the diagrams much easier to read and understand.

4. EXPERIENCES SO FAR

UML gives a nice visual touch to all the documentation produced in the desiging of the DECA system. Using UML forces designers to define clear interfaces to program modules, which probably will make the developing and testing more efficient and fault-tolerant.

Two UML tools were evaluated in the design process of the DECA system. The designers considered Rational Rose to be too complex an application to be learned in such a short period of time. The second tool, JVision, is a lighter application written in Java. JVision lacked elementary functions which restricted its usage.

The design team ended up with a rather scarce solution. Since the design phase required the designers to draw diagrams which couldn't be expressed with UML notation, and the designers simply didn't have the time and competence to use the provided tools, they used Microsoft PowerPoint for all the drawing.

5. REFERENCES

Using UML and Rational Rose for Data Modeling

http://www.infoadvisors.com/articles/UML/UML_RationalRose_DM.htm

The UML and Data Modeling

http://www.rational.com/products/whitepapers/101516.jsp

Updated use case diagrams of DECA

http://www.hut.fi/~klamminp/dekka_html/dekka.html

Requirement analysis of DECA

http://www.hut.fi/~tamatti2/ohjtyo/Vaatimusmaarittely.html