



MÄLARDALENS HÖGSKOLA

Visual Architecture Final Project Report

Version 1.0

Visual Architecture	Version: 1.0
Final Project Report	Date: 2011-01-13

Revision History

Date	Version	Description	Author
2011-01-12	0.8	Initial version	PH
2011-01-13	1	Rounded up and with additional data	DL

Visual Architecture	Version: 1.0
Final Project Report	Date: 2011-01-13

Table of Contents

1. Introduction.....	4
1.1 Purpose of this document.....	4
1.2 Intended Audience.....	4
1.3 Scope.....	4
1.4 Definitions and acronyms.....	4
1.5 References.....	4
2. Background and Objectives.....	4
3. Organization.....	4
3.1 Project Manager.....	4
3.2 Project Group.....	5
3.3 Steering Group.....	5
3.4 Customer.....	5
3.5 Others.....	5
4. Milestones.....	5
4.1 Remarks.....	5
5. Project Results.....	5
5.1 Requirements.....	5
5.2 Work Products and Deliverables.....	6
6. Project Experiences.....	6
6.1 Positive Experiences.....	7
6.2 Improvement Possibilities.....	7
7. Financials.....	7
7.1 Project Cost Summary.....	7
7.2 Work per Member.....	7
8. Metrics.....	7
8.1 Milestone Metrics.....	7
8.2 Effort Metrics.....	7

Visual Architecture	Version: 1.0
Final Project Report	Date: 2011-01-13

1. Introduction

1.1 Purpose of this document

The purpose of this document is to describe how the 'Visual Architecture' project is finished. This includes what and how we have done things, milestones, how long it took us and what the costs would have been.

1.2 Intended Audience

The Visual Architecture project intends to address the following audience

- The Customer
- The Steering group
- The Project Team, and
- All those with a vested interest in the project

This document could also be used as a starting point for groups that want to continue on our work.

1.3 Scope

The scope of the document includes the responsibilities assigned to members in the team, the customer and steering group discussions and guidance, the requirements of the whole project in metrics form so that we will be able to see the figures and the actual performance of the members and the management by the team leaders and the way in which project got to its final destination. The figures and the metrics will also show the total cost on the project and total man hours that are invested on the project. It also includes the actual results and the requirement.

1.4 Definitions and acronyms

1.4.1 Definitions

Keyword	Definitions
Xtext	A language development framework that permits the development of programming and domain specific languages
Assembly	Low-level programming language for computers, microprocessors, and microcontrollers

1.4.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions
GMF	Graphical Modeling Framework
EMF	Eclipse Modeling Framework
CPU	Central Processing Unit
ALU	Arithmetic Logic Unit
IDE	Integrated Development Environment
SVN	Subversion, a source version control system.
MDH	Mälardalens University
FER	University of Zagreb (Faculty of El. Engineering and Computing)

Visual Architecture	Version: 1.0
Final Project Report	Date: 2011-01-13

1.5 References

Our webpage with all the documentation, video previews and the program itself:

http://www.fer.hr/rasip/dsd/projects/visual_architecture

2. Background and Objectives

In the current digital age, computers are very complex, possessing numerous intricate components. In order to keep abreast with emerging software trends, there is a growing need for developers to understand what goes on 'under the hood' of a digital system.

With the Visual Architecture plugin a new outlook into the representation of low-level programming can now be explored, as it enables the virtual creation of simple digital computer components. Due to the extensive nature of Eclipse plugins, it is hoped that future implementations of this tool will be used to demonstrate more complex systems.

The Visual Architecture plugin solution is a framework in Eclipse that allows a teacher to create a diagrammatic representation of some component architecture using 'drag-and-drop' graphical features. Once the architecture has been defined the students can then use this design to develop their own assembly code implementations. The student implementations are developed using the normal Eclipse IDE, powered by Xtext.

This solution also provides a Debugging feature that enables the visualization of developed programs. It uses a 'step-into' and 'step-over' protocol to animate execution sequences in the architecture, giving the student visual feedback for each step executed.

Objectives are to create:

- A diagram editor with compartments and custom computer architecture graphics
- Rich editors for coding with code completion for assembly and diagram-to-command mapping
- Interactive debugger that prints out register and memory values after each step

3. Organization

The work was mostly divided into three parts:

- the teacher's architecture defining environment
- the student's programming environment
- the student's debugging environment that connects all the other environments

The work could also be divided according to the frameworks being used:

- X-text (Assembler programming, Microcode mapping)
- EMF
- GMF (visualizing)
- Acceleo (Generating microsteps)
- Debugger
- Interpreter

3.1 Project Manager

Visual Architecture	Version: 1.0
Final Project Report	Date: 2011-01-13

The project manager collected regular status updates on weekly meetings. Here he also divided the work and made sure everybody was up to date and knew what to do.

3.2 Project Group

Name	Initials
Dražen Lučanin	DL
Prashanta Paudel	PP
Ilijana Ame	IA
Peter van Heck	PH
Sarah Njeri Kuria	SK
Vedran Palajić	VL

The distribution of work in the project was mostly done according to this table:

Activity Owner(s)	Activiy (technology)
All	Team meetings (Skype, TeamViewer, Google) and documentation writing (Google docs, Ms Word, OO Writer)
DL	Launcher (Eclipse Launch), Debugger (Eclipse Debug)
VP, IA	Assembly editing environment (xText), Interpreter (Java)
PP, SK	Diagram editor (GMF, EMF), Diagram Highlighting (GMF)
PH	Architecture-Command Mapping Environment (xText), Microcode Generation (Acceleo)

3.3 Steering Group

Marin Orlić, Dražen Lučanin and Prashanta Paudel

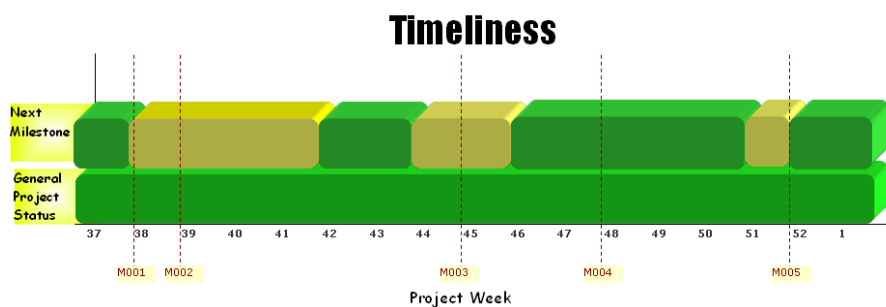
3.4 Customer

Marin Orlić

3.5 Others

Ivica Crnković (MDH), Thomas Leveque (MDH), Aneta Vulgarakis (MDH), Séverine Sentilles (MDH), Mario Žagar (FER), Igor Čavrak (FER), Ivana Bosnić (FER), Ana Petričić (FER).

4. Milestones



Visual Architecture	Version: 1.0
Final Project Report	Date: 2011-01-13

Id	Milestone Description	Responsible Dept./Initials	Finished week				Metr	Rem
			Plan	Forecast		Actual		
				Week	+/-			
M-001	Basic understanding	All	37	37	0	37		
M-002	Writing helloworlds	All	38	40	2	40	1	
M-003	Alpha prototype	All	44	46	2	46		
M-004	Beta prototype	All	47	47	0	47		
M-005	Release candidate	All	51	1	2	1	2	
M-006	Final release	All	2	2	0	2		

Remark Id	Description
1	It took the team some time to get the “initial swing”, due to a lot of documentation that had to be delivered at the beginning of the project.
2	The debugger, key element to connect the teacher's and the student's part, proved to be very difficult to implement, hence the delay of the Release Candidate.

5. Project Results

5.1 Requirements

5.1.1 Requirement Compliance Matrix

Id	Requirement Description	completed	Rem
	Visual architecture design application/process		
VAD-1	Allowing teacher(architecture designer) to assemble graphical architecture	T	
VAD-1.1	Providing tools to drag and drop predetermined architecture parts	T	
VAD-1.2	Providing tools for connecting parts	T	
VAD-1.3	Providing means to open, edit and save graphical architecture	T	
VAD-3	Allowing teacher(architecture designer) to design parts	T	
VAD-3.1	Providing tools for drawing parts	T	
VAD-3.2	Providing tools for describing parts	T	
VAD-4	Visual architecture should be developed using Eclipse GMF and EMF	T	
	Code input and editing application/process		
CIE-1	Allowing student(code runner)to enter code	T	
CIE-1.1	Providing means to make new, open, edit and save code	T	
CIE-1.2	Providing tools for code building(compiling)	T	
CIE-1.3	Providing tools for easier code editing, syntax highlighting, autocomplete.	T	
CIE-3	Tools for code input and editing should be developed using EMF and xtext	T	
	Visual code running application/process		

Visual Architecture	Version: 1.0
Final Project Report	Date: 2011-01-13

VCR-1	Allowing student(code runner) to debug code	T	
VCR-1.1	Providing means of selecting or importing the code for debugging	T	
VCR-1.2.1	Providing tools for simple(step)progressing trough lines of code	T	
VCR-1.2.2	Providing tools for advanced(break-point) progressing trough lines of code	F	
VCR-1.3.1	Showing debug information(registers, variables) trough console	T	
VCR-1.3.2	Showing advanced debug information (registers, variables) in special window	F	
VCR-1.4	Showing visual debug information(graphical architecture)	F	
VCR-2	Allowing student(code runner) to select running architecture	T	
VCR-3	Visual code running part should be developed using EMF and Java debugger	T	

Dropped requirements:

VAD-2	Allowing teacher(architecture designer) to make new language for student and its microcode	Dropped
VAD-2.1	Providing tools for writing down language grammar	Dropped
VAD-2.2	Providing tools for saving or exporting grammar(language)	Dropped
CIE-2	Providing means of selecting the language used	Dropped

Completed: *T (completely implemented)*

F (not implemented at all)

Partially (partially implemented, more description under Remarks subsection)

Unknown (completion status not known)

Dropped (requirement was dropped during the course of the project)

5.1.2 Requirements Compliance Summary

Total number of requirements	22
Number of requirements implemented	19
Requirements partially fulfilled	0
Requirements not fulfilled	3
Requirements dropped	4

5.2 Work Products and Deliverables

To	Output	Planned week	Promised week	Late +/-	Delivered week	Rem
Project Group	Technical Document, SVN guidelines	38	38	0	38	
Steering group	Project vision	38	38	0	38	
Project Team	Summary Week Report	Every Monday by 23:59				
Project Poll	"How does everyone feel?"	Every Tuesday by 23:59				
Steering group	Project plan presentation and document	39	39	0	39	
Project Team	Requirements Definition document	39	39	0	39	

Visual Architecture	Version: 1.0
Final Project Report	Date: 2011-01-13

Steering group, Customer	Requirements Definition and Design Description	40	40	0	40	
Project Team	Technical Document, Coding conventions and guidelines	40	40	0	40	
Steering group, Customer	Design Description document	40	40	0	40	
Steering group, Customer	Milestone - Alpha prototype	44	44	2	46	
Steering Group, Customer	Final project presentation	44	44	0	44	
Steering Group, Customer	Milestone - Beta prototype	47	47	0	47	
Steering Group	Acceptance test plan	49	49	0	49	
Steering Group	Milestone – Release candidate	51	51	2	1	
Steering Group, Customer	Test report	2	2	0	2	
Customer	Final Project Report, and final versions of existing reports documents, other project-related documentation	2	2	0	2	
Customer	Final product (installation, source code, etc.)	2	2	0	2	

6. Project Experiences

6.1 Positive Experiences

During this project we learned a lot. Not only that distance is a problem, but also the cultural differences. This started, as expected, a bit hard. This isn't a bad thing because now we got an excellent experience on how to deal with this.

The distance gives two problems. The first one is communication and the second one is data sharing. To overcome both problems we started to use several communication technologies: Skype, Team Viewer, Live Messenger, Google talk, E-mail, google-groups, Google-docs, Dropbox, SVN and Facebook. We learned how to use this software efficiently and we saw how it can make our life much easier.

We also learned a lot about other cultures, because we were had people from four different countries in our project group. Before we didn't think it was necessary, but we made some unwritten rules about how to communicate with each other so there wouldn't be misunderstandings and other kind of trouble. This turned out to be a good thing.

6.2 Improvement Possibilities

For this project we had to learn how to use a lot of plug-ins. This meant that we had to work through a lot of tutorials. This appeared to be pretty hard, because finding a good tutorial was rather hard. It is not really possible to change this fact, but this was definitely the hardest part of the project, and having to move on, dealing with advanced topics when something elementary is still a bit unclear is rather difficult.

7. Financials

As the work on this project was done pro bono for educational purposes, we decided that the best currency to use is how much man-hours we have spent. Therefore the following project cost summary is expressed in man-hours.

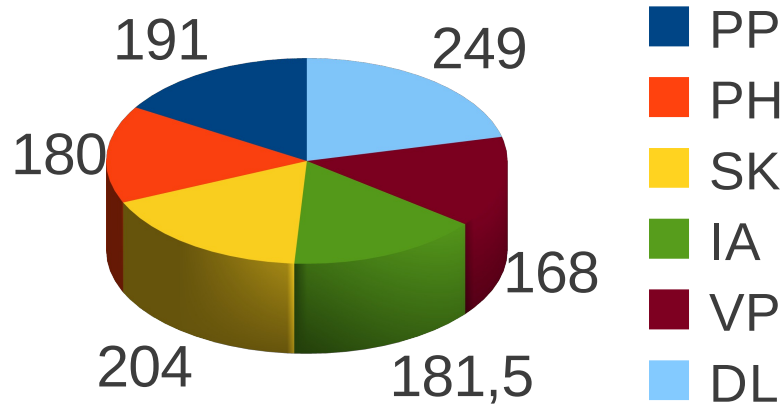
Visual Architecture	Version: 1.0
Final Project Report	Date: 2011-01-13

7.1 Project Cost Summary

Planned Cost	1440
Actual Cost	1173

7.2 Work per Member

Work was more or less evenly distributed as can be seen on this graph:



The concrete details can be read from the following table:

Member	W37	W38	W39	W40	W41	W42	W43	W44	W45	W46	W47	W48
PP	9,5	15,5	11	16	12	9	4	14	14	10	9	7
PH	19	15	16	9	10	12	3	19	11	11	10	5
SK	16	9	4	8	16	10	8	18	37	14	10	11
IA	6,5	11	14	10	11	7	7	7	13	14	7	5
VP	12	10	13	17	9	8	8	8	14	14	4	4
DL	16	12	11	12	17	20	12	14	22	8	20	6
Total	78,75	72,5	69	72	75	66	42	80	111	70	60	38

Member	W49	W50	W51	W52	W1	Total
PP	14	9	9	12	16	191
PH	15	7	8	5	5	180
SK	12	12	3	4	13	204
IA	18	15	7	8	21	181,5
VP	14	4	4	6	19	168
DL	24	19	4	15	17	249
Total	97	66	35	50	91	1174

8. Metrics

8.1 Milestone Metrics

Visual Architecture	Version: 1.0
Final Project Report	Date: 2011-01-13

Completed as planned or earlier	Total	Timeliness
3	6	50,00%

8.2 Effort Metrics

Mostly the differences between planned and actual effort arose in the requirements analysis & definition due to maybe a bit too big value planned in the beginning due to the iterative development method (more time planned for requirements analysis than program development is a bit unbalanced) and in the documentation and preparation phases (there was a lot of documentation to cope with – especially in the beginning of the project).

Activity	Actual Effort	Planned Effort	Deviation (%)
Project preparations	40	61	52,00%
Requirements analysis & definition	100	310	210,00%
Project planning	180	183	2,00%
Program development	340	305	10,00%
Program testing	200	305	0,00%
Documentation development	300	248	52,00%

When we look at the development activity in specific – some of it's subactivities (certain technologies) proved to be quite difficult.

Activity	Actual Effort	Planned Effort	Deviation (%)
Launcher (Eclipse Launch), Debugger (Eclipse Debug)	180	150	17,00%
Assembly editing environment (xText), Interpreter (Java)	100	100	0,00%
Diagram editor (GMF, EMF), Diagram Highlighting (GMF)	250	150	40,00%
Architecture-Command Mapping Environment (xText), Microcode Generation (Acceleo)	100	100	0,00%

Effort estimation accuracy (%) <i>(100*(1 - abs(Actual - Planned)/Actual))</i>	78%
--	-----