1. INTRODUCTION

1.1 About this document

This document is a project plan ("Plan van aanpak"), as defined in the "Stagehandleiding"[1] of the Division of Mathematics and Computer Science of the Faculty of Exact Sciences, Vrije Universiteit Amsterdam. The general conditions and financial agreements for the internship (the "stageovereenkomst") have already been defined in the contract between Chess-iT and Guus Bosman (August 1st 2002). If there would be any additional arrangements necessary they will be formulated in a separate document.

1.2 Status of my internship

My official agreement with Chess-iT is dated August 1st 2002, although at that moment I didn't have an official internship assignment yet. This document serves as a starting point of my internship.

There is one university course I didn't finish yet, a practical assignment. The Examencommissie officially allowed me to start my internship[2].

2. INTERNSHIP DESCRIPTION

2.1 Background

Chess develops products that are a combination of hardware and software. When designing embedded products a choice has to be made about what to program in software, and what to implement in hardware. The observation made at Chess is that the decision what to implement where (distribution between soft- and hardware) is often made too early in the design process, and is often based on rules-of-thumb by senior engineers. The hardware and software are usually produced in two separate processes, by two separate teams.

Chess would be interested in a development method that would allow a better-structured decision to be taken and that would allow this decision to be taken in a later phase. Specifically in the prototyping phase of the software cycle it would be very useful if hardware can be defined using a software definition.

In VDM, a language Chess currently uses, parallelism is hard to specify. Chess considers this a major weakness. They would like to find a (specification) language designed for modeling concurrency.

A number of candidates for this kind of language exist. CSP is an older one; IOA (from MIT) is from a more recent date. Chess is looking for a language that can replace VDM (the language used now) and that can be used in 2, 3 years from now.

However, having one specification language is not enough for the complete process from specification to implementation. Often a combination of languages and tools is necessary. I will call such a combination of tools and languages a *development method*. Typically a development method consists of 1 high level language (UML for example), an implementation language (C for example), and perhaps an intermediate language.

UML
SDL
VHDL

An example of a development method.

[1] http://www.cs.vu.nl/stage/documenten/handl-stage-nl.html
[2] e-mail R. van Veldhuizen, 6 Sep 2002

2.2 Assignment

2.2.1 Development methods

In my assignment I will select 3 development methods. This selection will be based on:

- Complete coverage of the process from design to implementation,
- Having a representative candidate of each class of developments methods,
- Expectations on which language is the most likely candidate to become an industry standard in the field.

These methods will be chosen in close coordination with my internship supervisors. A refinement will have to be made for some languages. For example UML is such a broad language that a description of the parts of UML used has to be given.

I will investigate the relevance and relationships between these languages in the track from specification to implementation. I'll research to which degree existing development and implementation methods fit into a single paradigm.

2.2.2 Scenarios

I'll carry out two small experiments using these methods, to get a good understanding of the problem of distribution. These experiments will be based on existing problems Chess faces and should have different 'natural' distributions of hardware and software. The following two projects have been given by Chess:

- Design and implement a DSP chip that compresses or encrypts information (streaming) (A)
- Design and implement a controller that is used in a fuel processing system to send information back to the gas-company (B)

2.2.3 Methods vs. scenarios

The methods and scenarios can be seen in this matrix:

	Method 1	Method 2	Method 3
DSP chip	A1	A2	A3
Fuel processing system controller	B1	B2	B3

For each combination I will try to do the whole process of specification to an (abstract) implementation in hardware and software. The hardware will be made using a FPGA.

2.2.4 Comparisons

I hope to discover the merits of the various development methods. I will compare the theoretical aspects of the new languages with each other, and address the issues that arise from the experiments. I hope to be able to link the theoretical results to the practical experiments.

3. APPROACH

3.1 Time schedule

Week	Activity	Details	Deliverables
46	Writing internship assignment, project plan.		• Internship assignment • Project plan
47	Creating 2 abstract project requirements (scenarios) Finding 3 methods		• Stable useful descriptions of 2 scenarios • 3 methods • Text on the initial selection of the methods
48	A1		
49			
50			
51	B2		
52	(Holidays)		
1		Presentation about the results of the first 2 experiments in the beginning of January.	
2	A3		
3			
4	B1		
5			
6	Creating a second version of A1, B2, A3 or B1.		
7	Extra week.		
8	Writing thesis.		
9			
10			• My thesis.

3.2 Prototypes

In week 50, 1, 3 and 5 I will be ready with a method/scenario combination. I will deliver a working prototype; an (abstract) implementation in both software and hardware.
In week 6 I will redo one of the selected combinations. This time I will try to give a different hardware/software mixture. If the first time I had an 80% software and 20% hardware implementation, I will now try to implement it the other way around.

3.3 Holidays

In week 52 I planned a week off; my supervisors won't be there either.

In the beginning of January Bert Bos will be unavailable for about a week. In the second half of January Ralf Lämmel will be unavailable.

4. REQUIREMENTS

4.1 Technical

Chess-iT already provided me with a computer and has been very helpful in arranging practical matters. Additionally, to be able to carry out the experiments it will be necessary for me to have some extra hardware available (i.e. board, microprocessor, FPGA).

If there is external software necessary it is important to order or request it on time. If there are any costs involved this will be discussed with my supervisors.

4.2 Advisory

It is mandatory for an internship to have both an external internship supervisor and a university lecturer. Ralf Lämmel, assistant professor at the Vrije Universiteit will be my internship supervisor. External coaches are Bert Bos (Chess-it) and Petra Eussen (Chess-iT).

5. RESULTS

The final results of this internship will be:

- A thesis on the differences between the selected languages for modeling hardware in relationship to the distribution between software and hardware;
- A few small prototypes;
- A presentation of the results established in the paper
- Material for writing a scientific paper on the subject, to be done after the internship.