Research proposal
Predictable Real-Time Scheduling

Bart Kerkhoff
For Océ Venlo

Venlo, 24 februari 2006

Inhoudsopgave

1 Introduction

This document is a research proposal. Its purpose is to specify to all involved parties how I will undertake my research project about predictable real-time scheduling. This research project will be the master thesis of Bart Kerkhoff, studying at the Radboud University Nijmegen. Bart studies computer science, with the graduation theme 'Embedded Systems' and the track 'Management and Technology'. The project is an assignment of, and will be done at, Océ Venlo[1] (see appendix A for a short description of the company).

So the parties involved in this research are;

- Bart Kerkhoff as the researcher
- Océ where this research will take place
- Both the computer science and the management department of the Raboud University, who will be supervising the research

The assignment is to integrate an experimental scheduling technique into the software development process at Océ. Doing that will ensure that the software generated, automatically satisfies real-time properties. Nowadays this is ensured manually, which costs time, resources and is more susceptible to errors. Using the new technique could possibly negate these things, but it could also improve the performance of Océ's products or lower the price of them. This research will deliver the tools required for the integration, an analysis of the effect of these tools and an advice to Océ, about whether or not to use these tools.

I start with explaining the context of this assignment; what is real time, what is predictable scheduling and why is it important? Then, in chapter three, I will show the relevance of this research. After that, the details of the research will be explained in the chapters four to seven; its objectives, the questions that need to be answered, how the research will be conducted and the planning.

In the appendices you can find additional information. Appendix A contains short description of Océ and the original assignment description by Océ can be found in appendix B.

1. See 'http://www.oce.nl' for Océ's website

2 Context

I will start with describing the context of the research. What does real time mean, what is predictable scheduling and why is it important? In the following section, this will all be related to Océ.

2.1 Real Time

Real time is a property of a certain system. That system can consist of hardware, software or a combination of both. A real-time system is a system that must oblige some timing constraints. This means that whenever something happens to the system, it has to react to that within certain time bounds.

Consider for example the lighting of a room. When someone pushes the light switch, the light should switch on within one second, not after one minute. And another example, a word processor. When the user types a character on his keyboard, it should appear on his screen within several milliseconds.

The printers and copiers Océ produces also have many of these timing constraints. Their interface must react immediately to user commands, users demand jobs to be finished within a certain time, etc... But also the internal hardware poses timing restrictions. When the paper gets stuck or another malfunction occurs, the machine must react quickly too prevent further damage. If it reacts to late, it could require a mechanic to come by, which renders the machine inoperable for quite a while and costs a lot of money. Or when printing starts, getting the job into the printer's memory has a high priority, it determines when printing can start and thus influences the time printing takes. When the paper arrives in the printing section, printing should start quickly. And when the machine is turned on, it should be ready to print/scan/copy within a certain time.

2.2 Predictable scheduling

Predictable scheduling refers to software. In a computer, there are many pieces of software that must share a limited amount of hardware. There can be multiple programs that want to access the Internet over a single connection, or there can be hundreds of programs being executed on a single processor. The system must ensure that all pieces of software can use the desired hardware at a certain point, the system has to make a schedule for the hardware. It determines when some software can use a certain piece of hardware. That is called scheduling.

When scheduling is predictable, it means that some properties with regard to scheduling can be proven. It can, for example, be guaranteed that a program has to wait at most one second before it gets access to the processor. Other examples are that a certain computation requires at most ten seconds to complete, that a system can run one hundred programs without any problems. And in our case, that it can be guaranteed that all real-time requirements are met.

2.3 The importance

As I have shown above, predictable scheduling can be used to prove that a system will meet its real-time requirements. These requirements must be met to ensure the quality of the system. When the scheduling algorithm is not predictable, but you do want to be sure that the real time constraints are met, you have to resort to other ways. You can extensively test the system to be sure, you can carefully analyse and tune the system, or you can use over specified hardware(twice as fast processor). All the additional options are either time consuming or expensive. Furthermore they only make it more likely that the constraints are met, they still offer no guarantees. Therefore predictable scheduling is the best way to ensure real-time constraints.

3 Relevance for Océ

As can been seen in appendix A, Océ produces printers and copiers. These printers and copiers contain lots of real time constraints, as discussed above. An important aspect of creating these machines, is developing the software that controls them. This software is also subjected to those real-time constraints. In my research I will focus at the development of that particular software.

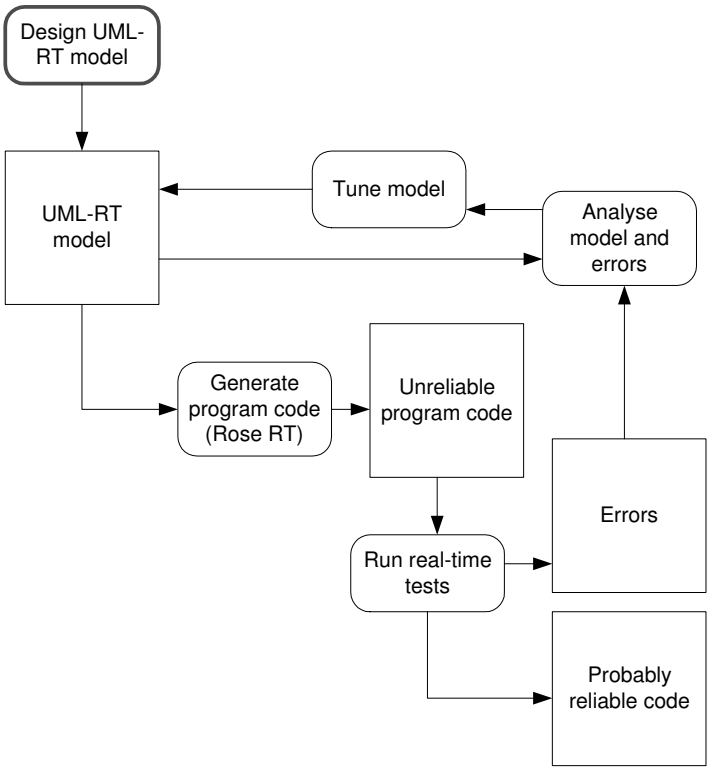
Lets first look at the way things are currently done and after that, what can be improved.

3.1 Current software development process

For the design of their software Océ uses UML-RT[1], an extension of UML[2], based on ROOM[3]. UML is a set of standards and diagrams that can be used to create a model of a system. It shows what a system should do, which components are in, how these components interact, how the software is deployed over the system, etc... In UML-RT the timing aspects of real time are added. To create these UML-RT models and its diagrams, Oce uses a software tool; IBM Rational Rose Real Time[4]. That tool, I will simply call it *Rose RT* from now, can automatically generate the code, the actual software program, from the UML-RT model. After that has happened, you already have a working piece of software. However, this software is not predictable with regard to scheduling.

To make the software more predictable, Océ has to take additional steps. The system has to be closely analysed, insight is requires in which parts of the system are active, what messages are being sent by these active parts, how much time is required to process these message, etc... A lot of complex work with a high risk of errors. Using this information the system can be tuned. After that, the real-time requirements will be met most of the time, however there is still no proof. Long and extensive testing is used after that, to detect whether or not there are still flaws, but this still doesn't offer hard guarantees.

Figure 1 graphically depicts the current software development process. You see that the generated code needs to be tested for its real-time properties.



Figuur 1: Software development process - before

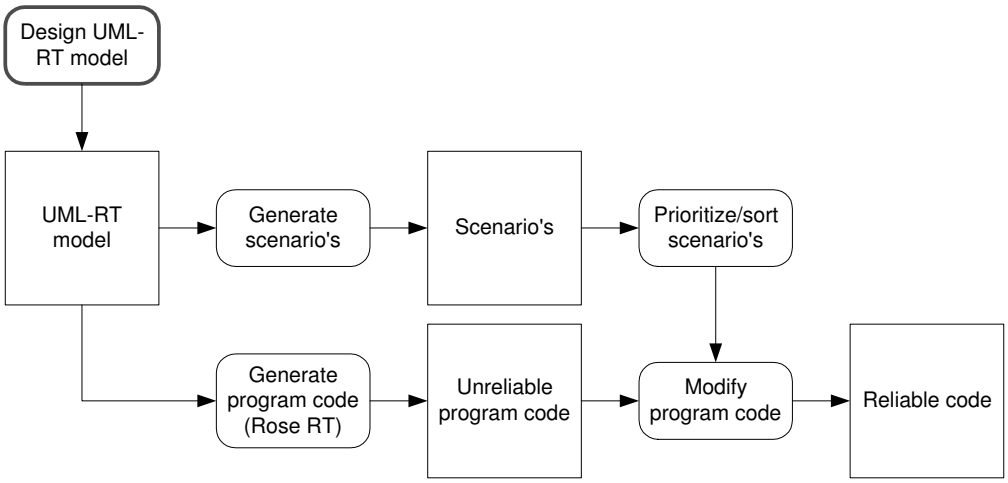
Someone needs to devise these tests and they should be run. To see whether or not an error can occur, these test should often run for quite a while. When an error occurs, the model and the error need to be analyzed; what was the exact cause of the error and how can the model be changed to prevent this error. Doing this analysis requires extensive user interaction. This test/analyze/fix loop, normally has to be repeated several times.

3.2 What will I try to improve?

At the 'Seoul National University (SNU)', a new experimental scheduling technique has been developed[5, 6]. They have also made an implementation, called SISA, to prove the concept. What I am going to do, is to integrate this experimental scheduling technique into the software development process at Océ.

What will change by doing that? The developers still have to create the UML-RT models. After that, the new scheduling will be able to derive the 'use case scenarios'. This was previously done by hand. Tuning these scenario's will also be done by the scheduler. And finally, the code(program) will be generated. But this time, it should be predictable and it can be ensured that the real time constraints are met.

So the second phase of the original software development process will be



Figuur 2: Software development process - after

done automatically, or at least in assistance with, the experimental scheduling technique. That should save time by removing the need to establish the scenarios and doing the tuning by hand. Futhermore, doing this automatically should be less error prone and results in better response times. Finally, we should have a predictable system.

See figure 2 for a representation of the new software development process. The only phase in which user interaction is required is, besides creating the model, is in the phase where the scenario's are prioritized and grouped. Note that this should only be done once, there is no loop.

4 Research objective

In this research, I am going to integrate the scheduling technique from Seoul in the software development process at Océ. I will also evaluate whether or not its an improvement to the process.

First, the technical part. The existing implementation has to be adopted and upgraded to fit within the software development process at Océ. The deliverables of this phase, are the *working tool*, an *example project*, *accompanying documentation* and a *report of the integration*. The *example project* will be a previous Océ product, that has been re-created using the newly developed tool. That should show the technical feasibility.

And second, I am going to evaluate the new scheduling technique. The original software development process will be compared to the new one using the scheduling technique. For this purpose, I will derive a model based on literature about software development and real-time systems. This model will be made specific for the situation at Océ. Based on that model, a comparison of both the software development processes will be made. From this comparison, I will

come to an advice for Océ whether or not to adopt the tool developed in the first phase of this research. In this phase, I will deliver a report that includes the important aspects of the software development process tuned for Océ, to what extend the original and the new process meet these demands and an advice about whether or not to use the new tool.

5 Research questions

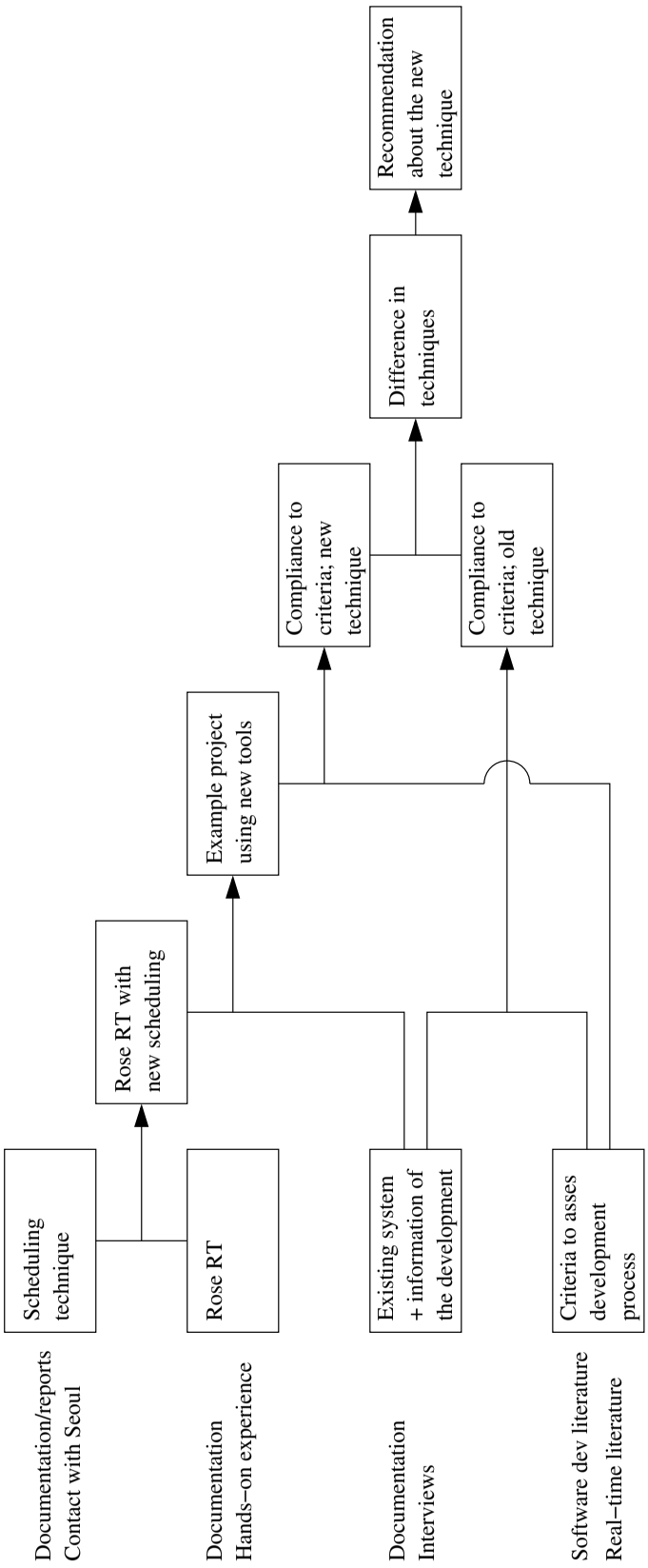
In order to reach the objectives stated above, two questions have to be answered. One with regard to the technical feasibility and another one with regard to the usefulness.

1. What is the best way to integrate the new scheduling technique into the software development process at Océ?
   (a) What will be the impact of integrating the experimental scheduling technique in the software development process?
   (b) What are the limitations/restrictions of, SISA, the current implementation?
   (c) To what extend is SISA scalable? Can it be use for large project?
   (d) Which modifications can be made to the current implementation to reduce the limitations/restrictions?
2. What is the advice about using the scheduling technique in the software development process?
   (a) Which criteria should be used to evaluate the software development process and the resulting real-time system; how can we determine the quality of a real-time system?
   (b) To what extend does the original situation meets these criteria?
   (c) To what extend does the new situation meets these criteria?
   (d) What are the differences between both situations with regard to the criteria?

6 Research strategy

I have created an strategy to tackle the questions and to reach the objectives. You can see a graphical representation of this strategy in figure 3. This figure and its steps will be explained below.

I should start with an analysis of SISA, the underlying scheduling technique and some alternative techniques. Furthermore, I need to acquire an understanding of how UML-RT and Rose RT work. The article describing the scheduling technique and Rose RT's manual and tutorials can be used for that. With this knowledge in mind, the scheduling technique has to be made into a tool that works in conjunction with Rose. I start by looking at SISA, all its



Figuur 3: Research strategy

restrictions and limitations will be documented. Next, it will be modified to operate in the software development process at Océ. And, if possible, improvements will be made to reduce the restrictions/limitations.

After that, I have to search for an Océ project which has been handled according to the original software development process. Probably my supervisor at Océ has some general ideas for that project, documentation of that project can be used for further information. The Rose model of this project has to be adopted to utilize the new tool. A working example project should be created.

To determine how successful the adjusted development process is, I have to create a model to assess the process. Literature on *software development* and *real-time systems* will primarily be used. The model needs to be adopted to Océ's specific needs.

Both the original project and the 'new' project have to be assessed according to the previous criteria. These assessments have to be compared.

Finally, I have to give a recommendation to Océ about the new scheduling technique. Does the new technique improves the software development process?

7 Planning

The planning of this research will follow the discussed framework. In the first month I will gather information about the scheduling techniques and *Rose RT*. The month after that, I will create the tool that uses the scheduling technique. I anticipate that there will not be a clear transition between these months, but iterative switching between the tasks. At the end of the second month, I will deliver the implemented tool, accompanying documentation and a document with all the implementation decisions and alternatives.

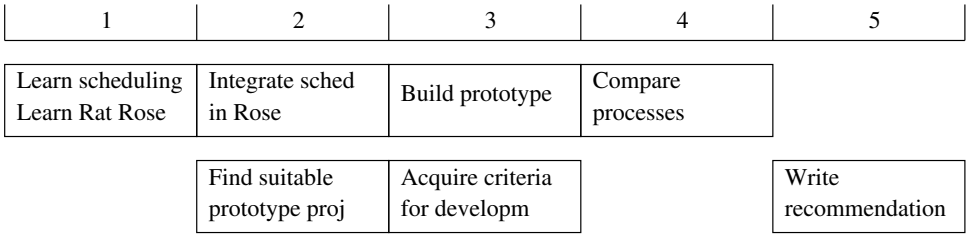
During the making of the tool, it will become clear what kind of project can be used to turn into an example during the research. I will start searching for a feasible project. Required are a project and information about its development process.

When there is a suitable project, after 2 months, I start building the example project. This will roughly take one month. The milestone will be the deliverance of the working example.

Parallel to the creation of the example, I can start with determining the criteria for the software development process. These criteria will be documented and are available after this phase.

In the fourth month, the analysis of both processes will start. This analysis with respect to the previously discussed criteria, will be documented. After that, I will come up with the recommendation and my thesis. The presentation of the thesis will be somewhere in the beginning of July, the exact date will determined in May.

Below you can find a graphical representation of the planning. In figure 4 you can find a table summarising the deadlines and corresponding milestones.



8 Organization

This research project will be the master thesis of Bart Kerkhoff. He studies at the computer science department of the *Radboud University Nijmegen*. His graduation theme is 'Embedded Systems' and his graduation variant is 'Management and Technology'. Bart will be conducting the research and is responsible for a good result.

Jozef Hooman will supervise the technical, computer science, aspects of this research. With regard to the technical aspects, Frits Vaandrager will act as a referent. Pepijn Vos will act as a supervisor for the management aspects of this research. Both the supervisors at the university are going to receive a weekly status update.

Océ issued the original assignment. From Océ, this research will be supervised by Johan van de Hee. Furthermore will Ton Janssen assist in the use and working of Rose RT. The research will be conducted at Océ Venlo. They will offer me a place to work and the necessary equipment. The duration of the research will be from the first of February till the end of June.

All three supervisors will assist Bart on the issues with regard to their specializations. They should be regularly available for questions. These three supervisors, and the referent, will also be responsible for the final grading of Bart's master thesis. The date of the final presentation of the thesis will be set in may.

Deadline	Task	Deliverables	Status
28-feb	Write research proposal	The final research proposal	Review
1-mrt	Gather information about scheduling	Document describing several scheduling techniques and their pros/cons	Review
		Document describing how the scenario based scheduling technique from Korea works	Review
1-mrt	Gather information about Rose RT	Document shortly describing what Rose RT is, and how it works	Review
1-apr	Adopt SISA for Oce	A working toolset, based on SISA, that utilizes scenario based scheduling	Active
		Document describing the original problems/limitations of SISA, what I have changed and the remaining problems/lmitations	Active
1-apr	Find suitable example project	Document describing an example project that will be implemented using the toolset (Cicero?) Document describing the problems/limitations of the current implementation	
1-mei	Do example project	UML model and source code of the example project developed with the toolset Document describing my experience with the toolset and its discovered limitations/problems	
1-mei	Assessment criteria	Document describing a model that will be used to assess real-time projects and their development process	
1-jun	Assess scenario based scheduling	Document describing how the 'orignal' example project scores with regard to the assessment model Document describing how the 'new' example project scores Document explaining the differences	
15-jun	Recommendation	Document in which a recommendation is given about wheter or not Oce should use scenario based scheduling and the toolset	
15-jun	Write thesis v1	First version of the thesis	
30-jun	Write final thesis	The final thesis	

Figuur 4: Milestones

Referenties

[1] Bran Selic and Jim Rumbaugh. Using uml for modeling complex real-time systems. Technical report, ObjecTime, 1998.

[2] Object Management Group (OMG). Unified modelling language (uml). http://www.uml.org.

[3] Bran Selic, Garth Gulekson, and Paul T. Ward. *Real-Time Object-Oriented Modeling*. John Wiley & Sons, New York, NY, 1994.

[4] IBM Rational Software Corporation. *Rational Rose RealTime User Guide*, 2003.06.00 edition.

[5] Saehwa Kim, Jiyong Park, and Seongsoo Hong. Scenario-based multitasking for real-time object-oriented models. *Journal of Information and Software Technology*, 2005.

[6] Jamison Masse, Saehwa Kim, and Seongsoo Hong. Tool set implementation for scenario-based multithreading of uml-rt models and experimental validation. In *Proceedings of the 9th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'03)*, 2003.

APPENDIX

A Short description of Océ

In this section I will give a short description of the company Océ. The information is directly taken from their website, `http://www.oce.nl` and is in dutch.

De Océ organisatie

Profiel

Océ stelt mensen in staat informatie met elkaar te delen. Daarvoor biedt Océ producten en diensten aan voor de (re)productie, presentatie, distributie en het beheren van documentstromen. Het assortiment omvat software, kopieer- en printsystemen en materialen. Daarnaast biedt de onderneming haar klanten innovatieve diensten op het gebied van consultancy, uitbesteding en financiering.

Océ streeft naar een vooraanstaande positie op de wereldmarkt, met geavanceerde producten die zich onderscheiden door hoge kwaliteit, betrouwbaarheid, productiviteit, duurzaamheid en gebruikers- en milieuvriendelijkheid. Het assortiment wordt grotendeels door Océ zelf ontwikkeld en geproduceerd. Jaarlijks wordt circa 7% van de omzet in onderzoek en ontwikkeling geïnvesteerd.

Door directe verkoop en service biedt Océ in een rechtstreekse relatie met de klant professionele ondersteuning. Daardoor beschikt Océ altijd over de meest actuele marktinformatie. Dit stelt de onderneming in staat adequaat te reageren op de behoeften in haar markten en daarvoor een evenwichtig aanbod van producten en diensten op te bouwen.

Het hoofdkantoor van de Océ Groep is gevestigd in Venlo. Daar is ook het grootste deel van de research, de productie en de internationale marketing geconcentreerd. De Océ Groep heeft wereldwijd ruim 21.000 mensen in dienst en realiseerde in 2004 een jaaromzet van € 2,7 miljard en een nettowinst van € 78 miljoen.

Océ-Nederland B.V. verzorgt in en vanuit 's-Hertogenbosch met ruim duizend werknemers de marketing, verkoop en support van de Océ producten en diensten in Nederland. De klanten van Océ zijn vooral actief in kantooromgevingen en in industriële en grafische sectoren.

B The original assignment by Océ

Below you will find the original assignment description by Océ.

Context

Voor de embedded besturing van de professionele printers van Océ wordt gebruik gemaakt van een object georiënteerd modellering tool, genaamd IBM Rational Rose RealTime. Deze tool is gebaseerd op de Unified Modelling Language (UML). Real-time eisen die gesteld worden aan deze embedded besturing nemen steeds toe en een voorspelbaar real-time gedrag is een must.

In de huidige werkwijze worden 'use case scenarios' opgesteld. Hierbij wordt voor specifieke opdrachten inzichtelijk gemaakt welke berichten er tussen de zogenaamde actieve capsules worden verstuurd en welke acties er worden uitgevoerd in zo'n geval. Het onderliggend Operating System te samen met het Target Run-Time System (TargetRTS) van Rational Rose RealTime verzorgt de scheduling van het berichten verkeer. Dit gebeurt op basis van taak prioriteiten, bericht prioriteiten en 'run-to-completion' mechanismen van het TargetRTS. De scheduling voor het uitvoeren van use cases is niet echt voorspelbaar maar wel te tunen d.m.v. de taak en bericht prioriteiten.

De Seoul National University Korea (SNU) heeft nu een experimentele schedulingsmethode ontwikkeld waarbij men claimt dat de timing requirements van de use cases gegarandeerd kunnen worden. Via IBM Rational worden nu echte klanten gevraagd om mee te doen in een pilot. Deze manier van scheduling zou interessant kunnen zijn voor Océ. Meer informatie over deze methode is te vinden op: http://redwood.snu.ac.kr/PAPERS/source/03-rtas.pdf . Door middel van een afstudeeropdracht willen we de geschiktheid van de methode evalueren.

Opdracht

Het doel van de opdracht is om:

- De werking en toepasbaarheid van de schedulingmethode te begrijpen.
- Door de ontwikkeling van een prototype model, in samenwerking met de SNU, ervaring opdoen met de schedulingmethode.
- In de embedded software van een bestaand product use cases aanpassen met timing requirements om vervolgens de werking te testen in een simulatie omgeving en op een echte target.
- Door middel van documentatie een aanbeveling te doen of deze methode toepasbaar is voor het Océ applicatie domein.

Omdat de pilot via de SNU wordt begeleid is het maken van een externe rapportage (of white-paper) ook een vereiste. Studenten met affiniteit voor scheduling en real-time concepten genieten voor deze opdracht de voorkeur.

Interessegebied

Real Time, Scheduling, Modelling