

# **Path Planning in Face of Uncertainty**

Potkuri-group

Helsinki December 12, 2008

Software Engineering Project

UNIVERSITY OF HELSINKI

Department of Computer Science

**Course**

581260 Software Engineering Project (6 cr)

**Project Group**

Veera Hoppula  
Mikko Kuusinen  
Jesse Paakkari  
Tobias Rask  
Timo Tonteri  
Eero Vehmanen

**Client**

Valentin Polishchuk

**Project Masters**

Sampo Lehtinen

**Homepage**

<http://www.cs.helsinki.fi/group/potkuri>

**Change Log**

Version	Date	Modifications
1.0	29.10.2008	Many spelling mistakes fixed
0.5	16.10.2008	Added risk
0.4	29.9.2008	Added risk, fixed spelling mistakes, moved vocabulary to common
0.3	22.9.2008	Fixed spelling mistakes. Altered size/cost expenses
0.2	16.9.2008	Updated the project plan
0.1	5.9.2008	First draft

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Vocabulary</b>	<b>2</b>
<b>3</b>	<b>Project Organization</b>	<b>3</b>
3.1	Members and roles . . . . .	3
3.2	Instructor . . . . .	3
3.3	Customer . . . . .	3
<b>4</b>	<b>Risk Analysis</b>	<b>4</b>
4.1	Risks . . . . .	4
<b>5</b>	<b>Hardware and software requirements</b>	<b>8</b>
<b>6</b>	<b>Size (and cost) estimates</b>	<b>9</b>
<b>7</b>	<b>Distribution of work</b>	<b>10</b>
<b>8</b>	<b>Project Schedule</b>	<b>11</b>
8.1	General timetable - Important dates . . . . .	11
8.2	Detail project schedule . . . . .	11
8.3	Gantt chart . . . . .	12
<b>9</b>	<b>Follow-up and reporting methods</b>	<b>13</b>

# 1 Introduction

There's a need to design a program to help air traffic management guide multiple airplanes safely to ground in presence of hazardous weather systems. The purpose of the project is to produce program which can be used to simulate airplane landings with real or simulated forecasts.

Calculating the routes will be a challenge since the routes must be planned ahead and there is a certain probability involved with different forecasts which must be taken into account. In the core of the program is the "Shortest path problem"-calculation for the planes moving from one merge point to the other before landing the plane. The goal is to avoid the different hazardous weather systems.

This project will also give hands-on experience to the project group. The purpose of the course is to learn about project work and learning about different roles present in the software project. The project group will do all the work involved in software project as assigned by the project manager. Weekly meetings will be held twice a week controlling and checking the schedule and resources of the project.

## 2 Vocabulary

**Airport** Airport is where arrival tree begins, in the middle of the map.

**Arc** Arcs are circles at a determined radius distance of the airport. The merge points are located into these arcs.

**Arrival tree** A binary tree consisting of paths. Has a root at the airport.

**Checkstyle** Java code review for Eclipse.

**dbZ** dBZ stands for decibels of Z. It is a meteorological measure of equivalent reflectivity (Z) of a radar signal reflected off a remote object.

**EclEmma** Java Code Coverage for Eclipse.

**Flight plan** Every plane has a flight plan which describes its path.

**FMI** Finnish Meteorological Institute.

**Integration Testing** Integration testing purpose is to assure that integrated classes do all those services they are planned to do in requirement document.

**Java2D** Display and print 2D graphics in Java programs.

**JAR** Runnable Java archive, which based on the ZIP file format.

**JUnit** JUnit testing framework.

**Map** A map from somewhere in the world used in this product.

**Merge point** A point on the map where two paths merge into one path.

**nmi** nautical mile (=1,8520km)

**Path** A route to the airport that should avoid storms.

**PGM** Portable Gray Map, a graphics file format.

**Plane** An airplane that tries to land at an airport along a path avoiding storms.

**PMD** Java code review for Eclipse.

**Storm** A set of pixels with a dBZ-value over a certain threshold (that is a parameter) close each other on the map. Indicated with red color on the map.

**System testing** System testing purpose is to assure that software corresponds it's requirements.

**User** A person using the product to watch animations on aircrafts landing at an airport in presence of hazardous weather systems.

**Unit testing** Unit testing purpose is to assure that certain class or unit do all those services it is planned to do in requirement document.

## **3 Project Organization**

### **3.1 Members and roles**

Jesse Paakkari	Project manager
Mikko Kuusinen	Requirements manager
Timo Juhani Tonteri	Design manager
Veera Hoppula	Document manager
Eero Antero Vehmanen	Code manager
Tobias Matias rask	Testing manager

### **3.2 Instructor**

Sampo Lehtinen works as project instructor

### **3.3 Customer**

Valentin Polishchuk from the University of Helsinki Computer Science Department.

## 4 Risk Analysis

The purpose of risk analysis is to foresee possible threats about the development of the product, whether the question is about personal, project, complexity of the program or some other external risks.

Risk probability is measured with the scale of 1 - 3. 1 having the least risk. The severity of the risk is measured with the scale of 1 - 3 as well. 1 having the least impact to the project schedule and to functioning of the program itself.

### 4.1 Risks

#### **Sickness of person**

Probability	2
Severity	2
Impact	The work will be halted for the period of sickness
Preventive measures	Eat C-vitamin
Corrective action	The person being sick must notify project group about the situation. The work must be divided and reassigned, if the sickness will last a long time. Project schedule or requirements must be adjusted in the worst case

#### **Lose of interest**

Probability	1
Severity	3
Impact	The work will be halted
Preventive measures	Keep in touch with the project personell
Corrective action	Motivate the person in question to continue to work

**People will run out of time**

Probability	2
Severity	3
Impact	The work will be halted or at least delayed
Preventive measures	Follow the agreed schedule and inform project group of any delays. Reserve enough time in project to different tasks. Schedule all exams and vacations beforehand
Corrective action	Check the individual schedules and find the spirit of teamwork again. Remember that this project will over just in few weeks

**Somebody will quit the project**

Probability	1
Severity	3
Impact	Fatal
Preventive measures	Have good spirit and team effort, help each other and have distinct work assignments. Leave space for innovation and personal ideas
Corrective action	Rearrange project responsibilities

**Narrow area of expertise**

Probability	1
Severity	2
Impact	Work cannot continue because there is no knowledge how to proceed
Preventive measures	Allow asking of questions, helping each other. Have meaningful meetings where all the problems will be notified and handled
Corrective action	More communication. Search for experts

**Customer does not know or cannot describe what he/she wants**

Probability	1
Severity	2
Impact	The end result will not satisfy the customer
Preventive measures	Meet with the customer and review the progress of the work. Simulate the program and its user interface to customer
Corrective action	Communicate more with the customer

**Failure in documentation**

Probability	2
Severity	3
Impact	The program will not be something the customer has wanted
Preventive measures	Document everything in orderly manner. Have regular time to inspect the work which has been done
Corrective action	Correct the documentation

**Resources of the equipment is insufficient**

Probability	1
Severity	3
Impact	The program will not be able to run in real-time
Preventive measures	Stress test the equipment with the program as early as possible
Corrective action	Faster computer, more memory, close other programs

**All the material gets lost**

Probability	1
Severity	3
Impact	Fatal
Preventive measures	Use of version control helps to save several copies of the work. Make backups regularly
Corrective action	Begin to follow preventive measures if not doing so already

**Some phase of work will take longer that anticipated**

Probability	3
Severity	2
Impact	The project will not be final in time
Preventive measures	Closely follow the work in progress and reach after agreed delay
Corrective action	Several project members will help to solve the problem

**The project is too large to be completed in time**

Probability	2
Severity	2
Impact	The project will not get finalized
Preventive measures	Consider all the resources carefully, being a bit pessimistic
Corrective action	Project plan must be reevaluated and changed, contact the customer and project management

**Real-Time weather data service, Helsinki testbed, will no longer be available**

Probability	3
Severity	1
Impact	Practically none
Preventive measures	Real-Time weather data is not used in project
Corrective action	None

**CVS will mess up binary files**

Probability	1
Severity	2
Impact	Great
Preventive measures	Learn to use CVS and how it handles binary files. Make separate backups regularly.
Corrective action	Fix CVS wrapper information.

## **5 Hardware and software requirements**

The program will be run on the customers personal computer using Java-language. There will be connection to the data source using TCP/IP (Weather information). Eclipse is used to develop the program. The source code, as well as the documentation, will be stored in version control system.

## 6 Size (and cost) estimates

There will be no direct expenses nor salary involved for the project group. Salaries of the other persons involved in the project (as well as the customers) are not accounted for in the calculations.

The program has 4 major parts (Weather data, plane position, calculation and graphics displayed) and some general functions (eg. setting the starting values, handling messages...). The size estimate for each is as follows:

Part	Estimated lines of code
Graphics	1500
Fetching the weather data	500
Plane position	1000
Calculation	2000
General	500

Estimates are based on experience and guesswork. The estimates have been exaggerated so that project can be finalized by the agreed time and so that any surprises are not fatal to the project schedule.

## **7 Distribution of work**

All the members have assigned responsibilities. As a team effort all the members of the project will participate in all the phases of the project. At the minimum commenting of the work done, is the least any one can do. In coding the program has been divided into different classes and objects. In project plan each member has been assigned to one of the different sections of the program but the code manager has the main responsibility.

Project manager will conduct all the general meetings and in case of any specific meetings or inspections, the person responsible will conduct the meeting in question.

## 8 Project Schedule

In the project we will have general inspections for the Requirements, Design and Test plan documents. Inspections will be held one week before the document is due and documents will be frozen one week after the due date. There will be one formal inspection of code in schedule.

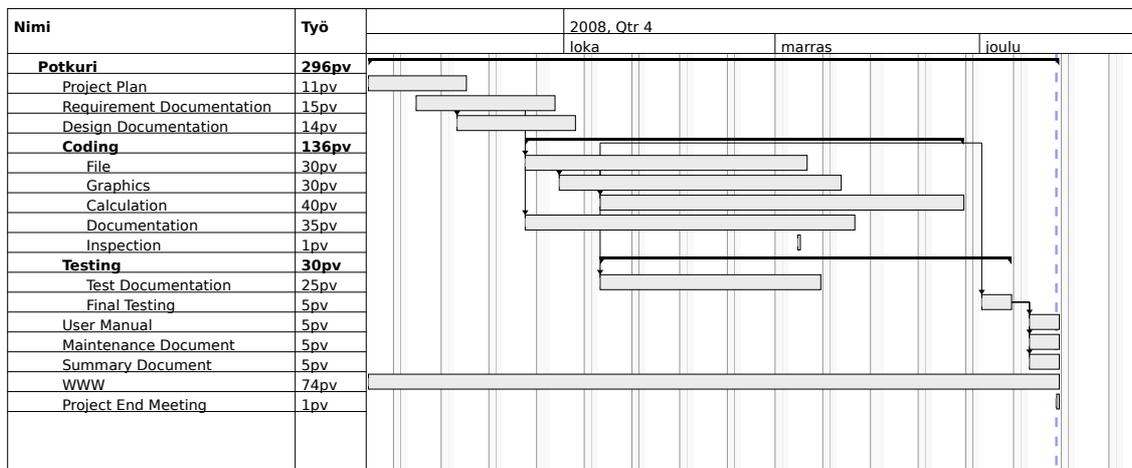
### 8.1 General timetable - Important dates

Project start	2.9.2008
Preliminary project plan	8.9.2008
Requirements document start	9.9.2008
Requirements document inspection	29.9.2008
Design document starts	15.9.2008
Requirements document frozen	29.9.2008
Design document inspection	6.10.2008
Design document frozen	9.10.2008
Coding begins	10.10.2008
1st code testing	9.11.2008
Formal inspection of code	13.11.2008
Coding ends	10.12.2008
All documents are ready excl. summary	11.12.2008
Summary document	12.12.2008
Project end meeting	12.12.2008

### 8.2 Detail project schedule

WBS	Nimi	Aloitus	Päätyminen	Työ	Kesto	Liikkumavara	Kulut	Annettu tehtäväksi
1	<b>Potkuri</b>	<b>2. syyskuuta</b>	<b>12. joulukuuta</b>	<b>296pv</b>	<b>74pv</b>		<b>0</b>	
1.1	Project Plan	2. syyskuuta	16. syyskuuta	11pv	11pv	63pv	0	
1.2	Requirement Documentation	9. syyskuuta	29. syyskuuta	15pv	15pv		0	
1.3	Design Documentation	15. syyskuuta	2. lokakuuta	14pv	14pv	51pv	0	
1.4	<b>Coding</b>	<b>25. syyskuuta</b>	<b>28. marraskuuta</b>	<b>136pv</b>	<b>47pv</b>		<b>0</b>	
1.4.1	File	25. syyskuuta	5. marraskuuta	30pv	30pv		0	
1.4.2	Graphics	30. syyskuuta	10. marraskuuta	30pv	30pv		0	
1.4.3	Calculation	6. lokakuuta	28. marraskuuta	40pv	40pv		0	
1.4.4	Documentation	25. syyskuuta	12. marraskuuta	35pv	35pv		0	
1.4.5	Inspection	4. marraskuuta	4. marraskuuta	1pv	1pv		0	
1.5	<b>Testing</b>	<b>6. lokakuuta</b>	<b>5. joulukuuta</b>	<b>30pv</b>	<b>45pv</b>	<b>5pv</b>	<b>0</b>	
1.5.1	Test Documentation	6. lokakuuta	7. marraskuuta	25pv	25pv	25pv	0	
1.5.2	Final Testing	1. joulukuuta	5. joulukuuta	5pv	5pv		0	
1.6	User Manual	8. joulukuuta	12. joulukuuta	5pv	5pv		0	
1.7	Maintenance Document	8. joulukuuta	12. joulukuuta	5pv	5pv		0	
1.8	Summary Document	8. joulukuuta	12. joulukuuta	5pv	5pv		0	
1.9	WWW	2. syyskuuta	12. joulukuuta	74pv	74pv		0	
1.10	Project End Meeting	12. joulukuuta	12. joulukuuta	1pv	1pv		0	

### 8.3 Gantt chart



## **9 Follow-up and reporting methods**

All the work of the project group will be individually entered in to the system maintained by the University of Helsinki Computer Science Department. All the information has to be entered by the end of the week so it can be reviewed by project manager prior to the first weekly meeting.

Project schedule will be checked every week on Monday.

Bugs reports and other reports should be kept in a spreadsheet and include date, reporting person, responsible person, error / notes, severity of the item, what and if something was done.