

## **Summary document**

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	12.12.2008	Corrections and added summary.
0.1	10.12.2008	1st draft.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Vocabulary</b>	<b>2</b>
<b>3</b>	<b>Information on Product</b>	<b>3</b>
<b>4</b>	<b>Document Abstracts</b>	<b>4</b>
4.1	Project documentation . . . . .	4
4.2	Requirements documentation . . . . .	4
4.3	Design documentation . . . . .	5
<b>5</b>	<b>Project Analysis</b>	<b>6</b>

# 1 Introduction

As a part of studies in Computer Science, Potkuri-group took part in project to build and document a program which would find arrival-tree for planes approaching airport.

## 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 Information on Product**

The Chopper-program calculates the arrival tree for planes approaching the airport and shows visually the storm patterns, arrival tree and air planes on screen. The program was coded with Java and accepts command-line parameter pointing to file which has parameters to run the program. There is no user intervention required while the program is running.

## **4 Document Abstracts**

### **4.1 Project documentation**

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.

### **4.2 Requirements documentation**

Requirements document functions as an agreement between the customer and the project group of what project will produce. Document lists vocabulary, describes interest groups and data used, explains use cases, lists all the requirements and explains system models and architecture that belong to the system. Document is also used as basis for planning.

Purpose of the program is to model and create safe paths for airplanes to land at an airport in changing weather. Goal is to create this to be as visual as possible so the user can see "a movie" of planes following flight routes to the airport. Changing weather in this project means storm centers that the planes have to avoid if storms are too intense. Both planes and weather can be randomly generated or generated by user. Weather data can also be acquired from some real-time weather source like Testbed site of Finnish Meteorological Institute.

### 4.3 Design documentation

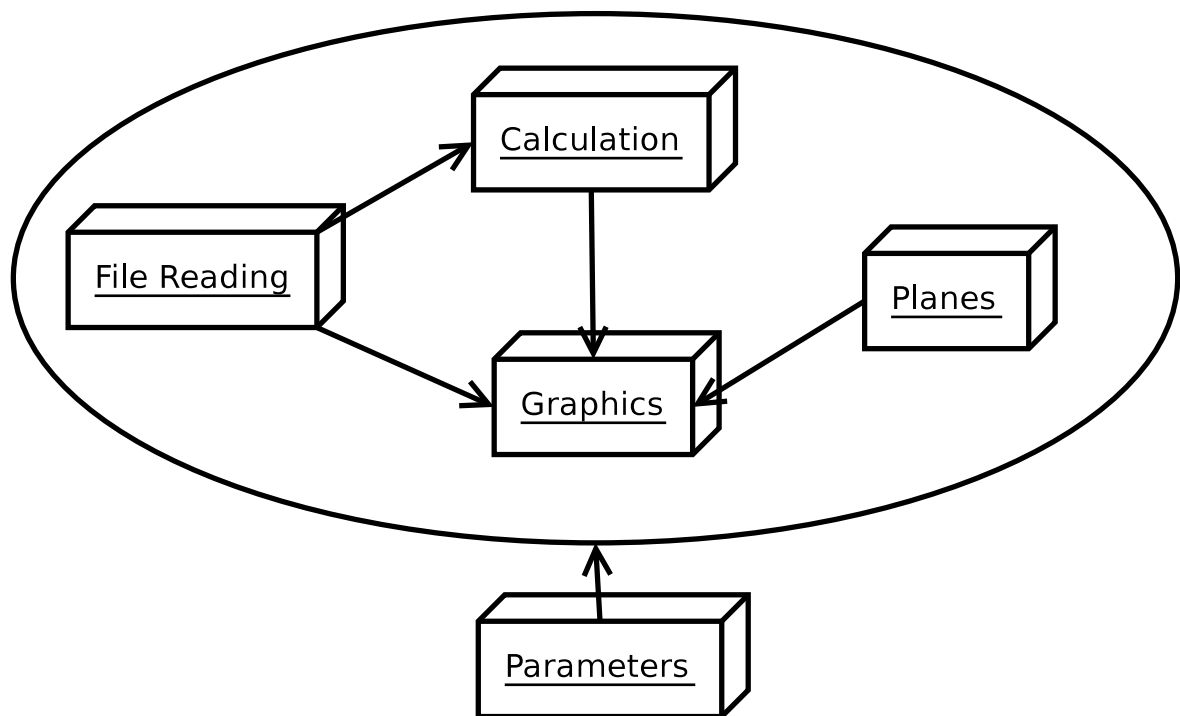
Design document describes the architecture design of the program.

The four parts of the system are *file reading*, *calculation*, *graphics presentation* and *planes*. All of these will get their parameters from the programs *parameters* class. The interface between these parts is the *DataCollection* class. In the *file reading* part, the data files will be read, and converted to a format suitable for the calculation to use. It is also responsible for updating the weather data.

The *calculation* system works as an independent part. It reads the current weather matrix, and updates the graph. After that it calculates (if needed) the new arrival tree for air planes. A recursive algorithm using A\* algorithm will be used to find the arrival tree with the shortest paths.

An interface between the user and program is the *graphics* part. It will use the weather map read by the file reading part, and the tree from the calculation. Together these will form a map for the airplanes to follow.

*Airplanes* part is a set of planes following the tree. It does not really have any intelligence in it, other than the planes following parent nodes.





## 5 Project Analysis

We took part in very interesting project. All the members of the project team agreed to do their best to achieve total customer satisfaction. We committed to write all the required documentation as well as to document the code and all the work we performed. We have kept detailed notes on meeting minutes and following up the progress of the project has been quite easy.

Very heart of the project is the project plan in which the length of the project was tighten up a bit (2 weeks shorter than required). This was done so that we'd have enough time at the end if there would arise any greater difficulties. Also, we listed a number of probable risks, of which none realized.

Responsibilities for all different documents was divided among the group members and all knew their responsibilities. Writing the documents was a group effort, so that nobody was left alone without any help. The common outlook of the documents was also one persons responsibility.

The documentation took a bit longer than anticipated and coding began week later than planned, but the project itself will Finnish by the deadline for the course. It was a smart plan to have a tight schedule.

Coding has been a breeze and there has been only few bugs. All of us have also took part in coding. Java was at least known to all of us, so there was no need to learn a new programming language from scratch.

The group spirit has been very high. There are no isolated group members and communication has been great among the members of the group. We had a afternoon when we went to Pizza-Rax and just had good time. All members also feel equal to each other and get their opinion heard.

All in all this was a great experience of project work. Naturally everybody was learning so some of the things were not perfectly done, but it has not stopped our enthusiasm.