

UpWind plug-in: Long and short term route planning

Pre-study report

Juha-Matti Hurnasti
Anu Pramila
Tomi Sarni
Andrei Vainik

1. Introduction

UpWind is an Open Source Software project initiated and coordinated by the UpWind team at Department of Information Processing Science at University of Oulu. Since 2006 there has been several project teams designing and developing an advanced navigation software for sailboats. Today, the software includes all essential navigation features and can be used as such in real boats.

The previous projects teams have build the software by building code on top of each other. This has resulted in code which is difficult to maintain and understand. Therefore a new plug-in based architecture model was introduced. Most of the functionality for the plug-in based architecture have already been implemented and long and short term route planning are the few last pieces to be ported to the new architecture.

In this work, our task is to implement a long and short term route planning for UpWind-system by creating the functionality by following new plug-in based architecture model. The task is developed within Qt cross-platform application framework and implemented with C++ programming language. The project is managed with Scrum method.

In order to complete the task, information about the various subjects are needed. The aim of this document is to provide necessary articles and references for the implementation work. However, due to the novelty and the nature of the implementation environment, scientific references and publications are difficult to find. Therefore the research relies heavily on guides and articles found from Internet.

The articles have been divided in the following according to the subject. First Qt (officially pronounced as “cute”) Application Framework is introduced. This framework is an open source cross-platform application framework for developing application software with graphical user interface. Qt uses standard C++ but has also a special code generator (Meta object Compiles) and several macros to enrich the language.

Qt supports coding with plug-ins and thus the plug-in architecture and design pattern related to it are also presented here. In order to be able to take full advantage of the architecture we naturally need a deeper understanding of the long and short term route planning. Therefore some publications about the subject are included in this work.

Finally, the work is developed in the end by applying Agile programming and Scrum. Understanding these methods are vital for reaching a successful conclusion.

2. Qt Application Framework

Article	http://developer.qt.nokia.com/wiki/QtWhitepaper#112e0760e6bdfa1eeadf4e3c27cdf081
Description	This whitepaper is an introduction to Qt framework. It describes several features such as: Graphical User Interfaces, Qt Designer, Graphics and Multimedia, Databases, Plugins and the Meta-Object System Building Qt Applications, and Qt's Architecture. It also contains examples and sample code for Qt applications.

Article	http://developer.qt.nokia.com/doc/qt-4.8/classes.html
Description	The Qt API serves as support material for Qt development.

3. C++ programming language

Article	Stroustrup, B. (2000) The C++ Programmig Language. Addison-Wesley Series in Computer Science, Special Edition.
Description	Works as a quick reference for C++ language.

Article	Meyers, S. (2005) Effective C++: 55 Specific Ways to Improve Your Programs and Designs, Addison-Wesley Professional
Description	"The book is organized around 55 specific guidelines, each of which describes a way to write better C++. Each is backed by concrete examples."

Article	Wilson Greg & Oram Andy. 2007. Beautiful code. O'Reilly Media Inc. USA.
Description	This book offers "beautiful" solutions to many common problems face by programmer's in their every day work. Each problem is also described with pseudo-code examples, so the information is applicable in various environments. This book contains examples that can be of practical help in this project. As the book is electronically available through Nelliportaali, searching for solutions is quick and easy.

4. Plug-in Architecture and design patterns

Article	Schleinzer, B., Cabac, L., Moldt, D., Duvigenau, M. (2008). <i>From Agents and Plugins to Plugin-Agents, Concepts for Flexible Architectures</i> . New Technologies, Mobility and Security, 2008.
Description	This paper characterises software Agents and Plugin metaphores approaches. Agent technology provides autonomous behavior, pro-activity, distribution and even mobility, while plugin technology provides well established concepts for lightweight extensible architectures.

Article	Jun Zhu, Quan Yi; Rui Zhu, Changguo Guo, Huaimin Wang, Quanyuan Wu, (2008). <i>A Plugin-Based Software Production Line Integrated Framework</i> . International Conference on Computer Science and Software Engineering
Description	This paper provides the definitions and differentiations of software product line, software production line and plugin architecture. A platform independent model of the software production line integrated framework, that is based on the plugin architecture, is put forward.

Article	Sametinger, J., (1997). <i>Software Engineering with Reusable Components</i> . Springer, 1997.
Description	The book provides a clear understanding of what software reuse is, where the problems are, what benefits to expect, the activities, and its different forms. The reader is also given an overview of what software components are, different kinds of components and compositions, a taxonomy thereof, and examples of successful component reuse. An introduction to software engineering and software process models is also provided.

Article	Freeman Eric T., Robson Elisabeth, Bert Bates & Sierra Kathy. 2004. <i>Head first design patterns</i> . O'Reilly Media Inc. USA.
----------------	--

Description	A real hands-on guide on design patterns. The book presents various example problems which programmers face when designing software systems. The book describes these problems from a layman's perspective, which is often the case when a programmer tries to make sense of user requirements for instance. Its a good background reading on good design practices. We chose this because it connects user requirements and design patters through understandable examples. It will help us in understanding structures and design program structures.
--------------------	---

Article	http://developer.qt.nokia.com/doc/qt-4.8/plugins-howto.html
Description	The article describes QT Plug-In mechanism.

5. Understanding Sailboat Route Planning: Short and Long Term Planning

Article	Díez S. M. (2010). Long term route planning for sailboats along the navigation lines. Oulu, Finland: University of Oulu.
Description	Master's thesis from the University of Oulu about route planning along the navigation lines for a specific sailboat at certain wind conditions. Based on the UpWind project.

Article	Stelzer, R. and Pröll, T. (2008) Autonomous sailboat navigation for short course racing, Robotics and Autonomous Systems 56(7), pp. 604-614.
Description	A compact method to calculate a suitable route for a sailboat in order to reach any specified target.

Article	Herrero Pau, Jaulin Luc, Veh Joseph ´ & Sainz Miguel. 2005. Inner and outer approximation of the polar diagram of a sailboat. University of Girona.
Description	This study article describes the mathematics behind sail boat polar diagram. The sail boat polar diagram is a key component in calculating short term route for the sail boat. Even though this study heavily math oriented, it will help us understand more about the set of output variables which we need to handle in the project.

Article	Warmerdam Frank (2008). Open Source Approaches in Spatial Data Handling. <i>Advances in Geographic Information Science</i> . 2 (2), 87-104.
Description	This article sums up what is the GDAL library and history of it. GDAL forms the core of the UpWind project as it contains sea charts with important information such as navigational lines, bridges, coast lines, water depth charts and such. This document will help us understand what does the database contain and how is the data represented in the data base.

6. Understanding Scrum

Article	Abrahamsson, P., Warsta, J., Siponen, M.T., and Ronainen, J. New Directions on Agile Methods: A Comparative Analysis. <i>Proc. 25th International Conference on Software Engineering</i> , May 2003, pp. 244-254
Description	"This paper reports results from a study, which aims to organize, analyze and make sense out of the dispersed field of agile software development methods."

Article	Deemer, P. and Benefield, G. The Scrum Primer - An Introduction to Agile Project Management with Scrum. retrieved 8.2.2012 URL: http://www.rallydev.com/documents/scrumprimer.pdf
Description	The publication presents the basics of Scrum in easily understood format.

Article	Coram, M. & Bohner, S. (2005). <i>The Impact of Agile Methods on Software Project Management</i> . Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05). pp. 363-370
Description	This study will improve the project group's understanding about agile methods and their use. The article studies the impacts of agile methods on the people involved a project and the process itself.

Article	Haugen, N.C. (2006). <i>An Empirical Study of Using Planning Poker for User Story Estimation</i> . Proceedings of AGILE 2006 Conference (AGILE'06).
----------------	---

Description	Since user stories are used in the project to gather requirements, this article will shed more light into the meaning and effect of the user stories. It might also help the project group to improve the user stories and their use. The study argues that although group estimation of requirements is superior to individual estimation, it could be utilized even better.
--------------------	---

Article	Rising, L. & Janoff, N. S. (2000). The Scrum Software Development Process for Small Teams. IEEE Software, 17(4). pp. 26-32.
Description	Gathers the experiences of the paper's authors about using Scrum to react changing requirements during the product life cycle. Again, will improve the project group's understanding about Scrum and Agile philosophy.