Analysis & Comparison: Four Cross Platform C/C++ GUI framework: Qt, Gtk+, wxWidgets and Fltk

Suprasanna Sarkar
Department of Computer Science
University of Tampere

Background

- For the recent couple of years it does not seem appropriate to constrain applications to a single operating system or computer platform, and there is an increasing need to address programming in a cross platform manner. Cross platform (also known as multi platform) development generally refers to the portability across several hardware and software platforms
- Software portability is a challenge in the present fast-changing era in platform technology . Cross platform application development frameworks contain Application Programming Interface (API) that is very nearly the same on all supported platforms [J Smart et al, 2005] can be used to achieve portability in application development
- Achieving portability is a challenge in large-scale systems at the age of complex platform architecture. Viewpoints in application development frameworks help to achieve portability over these complex systems by taking into account implementation details

Background(contd.)

- Different frameworks prescribe slightly different application viewpoints for achieving portability.
  - Portability feature implementation verification should check the framework architecture and the rationale of different available platforms(completeness and quality in depth)
  - Investigation of architecture and design of framework(Scope of architecture)
- With the increasing need for the portability of applications, it is very important to select a robust, efficient and suitable framework to standardize and bring structure to application development
- There are research works related to frameworks evaluation and many of them are limited to single criteria.

Background(contd.)

- Bishop [2006] on the other hand identified the user interface challenges for multiplatforms by investigating the model-driven frameworks from the 1990s. However she only compared the toolkits from the Qt and gtk GUI frameworks to identify the user interface challenges. This result is limited because it does not compare the Qt and Gtk frameworks from the cross platform portability point of view.
- Packard and Gettys[2003] have made an informal comparison of system performance, quantitative network performance and XServer performance of different applications created using Qt and Gtk+ GUI frameworks . However, the result is not the most recent and limited to run time performance only
- The explosion in types and number of computing devices is driving demand to deliver software applications on more and diverse platforms. To be competitive, an organization must develop the software engineering skills to develop once and deploy everywhere at the lowest cost.

Background(contd.)

- Some of them investigate the possible solutions to the portability problems without considering any GUI frameworks. Most of this research does not necessarily reflect the dynamic nature of portability, but provides the background behind the more recent graphical user interface methodologies
- No studies with ambition of providing an evaluation that deals with application portability, and few of them attempt to review most of the relevant criteria at once.

Research Questions

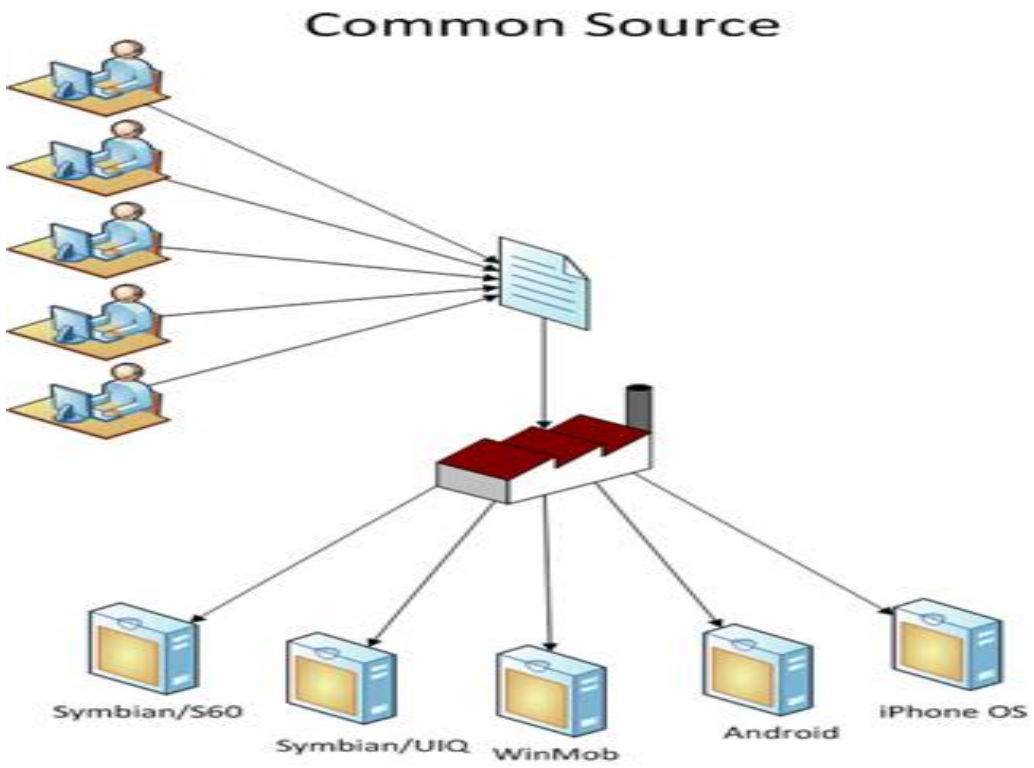
- What are the features and capabilities of popular cross platform C/C++ GUI frameworks?
- What are the ideal features and capabilities of an efficient cross platform C/C++ GUI framework?
- What is the right C/C++ framework for proprietary and open source portable software development?

Defining Portability

- Porting is the act of producing an executable version of a software unit or system in a new environment, based on an existing version. There are many definitions for the attribute of portability. According to Mooney [1997],

  A software unit is portable (exhibits portability) across a class of environments to the degree that the cost to transport and adapt it to a new environment in the class is less than the cost of redevelopment.

- Why Portability
  - A portable design up front can reduce porting costs, and often reduce total costs for the entire life of the product.
  - However, developing portable software adds costs during initial development, in return for benefits which will come later.
  - If the focus is on getting it out the door, portability will receive little consideration. If instead a more holistic view is taken, based on total lifecycle costs, portability should become a standard element of the software process.



Source :
http://sender11.typepad.com/sender11/2008/07/multi-platform.html

Methodology

- Feature comparison
  - The efficiency of cross platform C/C++ application development can be increased by the use of an appropriate framework
  - Choosing a proper framework is dependent on several factors, this part of the thesis chooses some basic but important features to be the standard of the framework
  - Discusses each feature from framework to framework and gives a comprehensive presentation of each framework's portability feature implementation

Methodology(Contd.)

- Selection Criteria
  - “application data and resource files”, “file systems”, “internationalization and localization”, “compilers”, “operating system interfaces”, “user Interfaces”, “build system” and “configuration management”.

Methodology(Contd.)

- **Selection Criteria (Phase 1): Study strategy and processes regarding portability**
  - This phase is divided in two different stages. In the first stage I studied cross platform strategies available over different platforms to achieve portability in application development. This is something that I consider fundamental if I am going to evaluate how well application adapts to different platform considering that strategy.
  - In the second stage, I investigated the process related to that strategy in order to acquire portability over supported platforms. This decision consists on a deep study of each alternative strategy in order to adopt one of them.

Methodology(Contd.)

- **Selection Criteria (Phase 2)Study different platforms from the context of portable application development**
  - Based on the knowledge about the strategies and processes obtained on earlier phase, some minimum strategies candidate are picked
  - conducted a comprehensive study about different available platforms.
  - Obtain enough minimum information on each platform in order to decide whether the suggested strategy could bring portability in each of the platforms.

Methodology(Contd.)

- **Dig into the features of frameworks relating portability**
  - List of strategies, process and platform details obtained in previous phases.
  - More information was obtained regarding different available features of different frameworks by deeply investigating the process and strategies acquired earlier.
  - The list of selection criteria was refined by examining the adaptabilities with each framework.

Methodology(Contd.)

- **Analysis and demonstration of strategy candidates and filter**
  - Gather all the opinions, reviews and refines the list of criteria for portability.
  - Compare overall implementation process related to each selection criteria and checked the suitability with the desired frameworks and supported platforms.
  - Considering the analysis presented in previous phases, I select seven portability criteria.

Methodology(Contd.)

- A Case Study
  - A case study with some sample open source applications originally available from different open source communities
  - Revise and expand these applications to make it as a practical example for this thesis.
  - These are four separate applications that have been implemented with Qt, Gtk+, wxWidgets, and FLTK and then compiled and run in Windows XP, KDE and GNOME of Ubuntu Linux 8.04 (Hardy heron) to check the portability of those applications.
  - Because of the unimportance and less relevance, requirements and functionality descriptions of those applications is not presented.

Methodology(Contd.)

- Feature comparison & analysis
  - Cross platform portability feature implementations of four chosen frameworks are evaluated
  - Advantage and disadvantage
  - The second part sums up with the suggestion for creating a suitable application frameworks can produce portable applications

Methodology(Contd.)

- **Limitation**
  - Some framework features was not be included in this thesis because of knowledge limitation.
  - The framework's feature discussion are restricted by the implementation skill of the thesis author.
  - The cross platform features` shortcoming is limited by the version of the frameworks, the specific cross platform feature could be reinforced in the later version of the framework.
  - The portability of those applications in the case study were not tested and studied under any MAC platform because of the author's knowledge limitation.
  - However, all of the four limitations cannot have severe impact on the correctness of research result.
    - For the first two delimitation, as state before this thesis concentrate on basic but essential portability features, and analyze sample application focus on reflecting the real feature implementation of different frameworks and thus no fabulous programming skill is need at this point.
    - For the third delimitation, the readers could get the newest information from the framework official web site, and easily adjust their options based on the research result of this thesis.

Technology and Infrastructure Investigation

- **GTk+**
  - Gnome's application development framework consists of a suite of libraries, all written in portable ANSI C and intended to be used on UNIX-like systems [Gtk+, 2008]. Libraries which involve graphics rely on the XWindow System.
  - Wrappers are available which export the Gnome API to most of the programming languages including at least three different C++ ,Ada, Scheme, Python, Perl, Tom, Eiffel, Dylan, and Objective C.
  - TK+ was initially developed for and used by the GIMP, the GNU Image Manipulation Program. These days commonly known as GTK+, "The GIMP ToolKit" is used by a large number of applications including the GNU project's GNOME desktop

Technology and Infrastructure Investigation

- **Qt**
  - Qt is a cross-platform, graphical, application development framework that enables developers and technical managers to compile and run applications on Windows, Mac OS X, Linux, different brands of Unix and other embedded platforms [Qt, 2008]
  - A large part of Qt is devoted to providing a platform-neutral interface to everything, ranging from representing characters in memory to creating a multithreaded graphical application.

Technology and Infrastructure Investigation

- **wxWidgets**
  - wxWidgets is a cross-platform C++ framework with a GUI library, offering classes for all common GUI controls as well as a comprehensive set of helper classes for most common application tasks [wxWidget,2008].
  - wxWidgets uses native widgets on all platforms whenever possible and fills missing gaps on some platforms using generic controls written with wxWidgets itself.
  - Link with the appropriate library for specific platform and compiler and the application will adopt the look and feel appropriate to that platform.

Technology and Infrastructure Investigation

- **Fast, Light Tool Kit (FLTK)**
  - FLTK is a cross-platform C++ GUI toolkit provides modern GUI functionality without the extraneous features and supports 3D graphics via OpenGL and its built-in GLUT emulation.
  - FLTK is designed to be small and modular enough to be statically linked, and also works as a shared library.
  - Its GUI design functionality has OpenGL support capability, which focuses on being small and modular with good performance [Fltk, 2008]

Portability feature comparison

- Compare seven important portability features for four chosen framework by investigating feature implementation and presenting corresponding case study and then make a conclusion and evaluation for each framework's portability features

Strengths of the four C/C++ cross platform GUI frameworks

- **Application data and resource files**
  - Developing resources that are easy to move between platforms is just as important as making sure that the code compiles and runs correctly. All of the four frameworks have some kind of custom data or resource files.
  - As if writing code that runs between multiple platforms is not enough of a task by itself, simply trying to share data between platforms can turn into a frustrating experience for developers and users alike.
  - Files may have data stored in different formats and byte ordering and in some cases files themselves may be missing key data lost during the transition from one system to another. Using text files exclusively can mitigate these problems to some degree; however, this is at the cost of larger file size and increase time to load and parse.

Strengths of the four C/C++ cross platform GUI frameworks(contd.)

- **Platform independent build tool**
  - This item covers the important topic of managing and building source code in a portable manner. Even the simplest cross-platform project can become chaotic if attention is not given to how to build and maintain source portably, and it gets even more chaotic when large numbers of people become involved.
  - Portable build tools solve a lot of problems associated with host specific build environments by pushing more of the work onto the developer, thus lessening the burden on the tools
  - Qt, Gtk+, WxWidgets and FLTK provides the facility of using cross platform portable build tools
  - They are often not particularly amenable to automation. Configuring periodic builds across a group of machines can be cumbersome.

Strengths of the four C/C++ cross platform GUI frameworks(contd.)

- **Internationalization and localization**
  - The terms internationalization and localization are sometimes used interchangeably, though they are related process. Internationalization consists of developing and preparing an application so that it can be localized easily. Localization is the process of taking an internationalized application and customizing it for a particular language, region and culture.
  - When taking their application to an international market, the first thing that comes to mind is translation. Developers will need to provide a set of translations for all the strings their application presents in each foreign language that it supports.

Strengths of the four C/C++ cross platform GUI frameworks(Contd.)

- **Compilers**
  - As rigorous as the C and C++ standards may seem, compiler authors have a great amount of leeway when it comes to the interpretation of the standard and the implementation of features [Logan, 2007].
  - The four GUI frameworks support different available C/C++ compilers. They provide compiling the project easily with the fully supported compilers on particular platforms.
  - These supported platforms might not be working on other platform which happens for the community supported compilers or partially community supported compilers. None of these four frameworks are able to completely hide the compiler specific platform dependencies away from developers

Strengths of the four C/C++ cross platform GUI frameworks(contd.)

- **Handling operating system interfaces**
  - As operating system become more all-encompassing, application become more dependent on them for system services such as user interface, memory allocation and mapping, security and privilege access, sound, video and networking. [Hook, 2005].
  - As software packages become larger and more complex, their memory requirements often outstrip the available physical memory installed in typical computer systems.portability issue involves memory mapping and memory protection.

Strengths of the four C/C++ cross platform GUI frameworks(contd.)

- **Separate user Interface from model**
  - One of the more difficult aspects of cross platform software development comes into play when creating applications with GUI. The appearance, behavior and tools used to create UI itself all vary greatly across platforms.
  - All of the four aforementioned GUI frameworks' approach to solve this is a well-defined separation of the code that maintains the data in an application from the code that displays this same data to the user

Strengths of the four C/C++ cross platform GUI frameworks(contd.)

- **Support multiple libraries**
  - At present time software development is less about writing new code than it is about gluing together chunks of preexisting pieces. If developers are dependent on a group of proprietary libraries or APIS, porting to a new platform is comparatively difficult with these four frameworks. But all these four GUI frameworks provide support multiple alternative libraries that accomplish the same result. For example Qt supports OpenGL and Direct3D. All other frameworks supports OpenGL library
- **Cross platform bug reporting and tracking system**
  - A major component of cross platform toolset is the bug reporting and tracking system. A bug system is used by developers and testers to report defects and issues encountered during the software development and testing phases of a project. In general, a bug system should allow the reporter to specify the problem encountered, identify context or state of the system at the time the bug was discovered and the list of the steps required to reproduce the bug. In cross platform development project using any of the four frameworks is compatible with the most of the native bug tracking system

Limitations of the four GUI frameworks

- **Inefficiency of hiding platform dependencies**
  - All the four cross platform GUI frameworks are not always able to hide the arcane details of source code dependencies , compiler and linker switches, platform specific compiler settings from the developers to provide a pleasant , easy to use interface.
  - In case of moving to a new platform or tool chain, the developers need to resolve some of the cross platform dependencies between different files, since these frameworks are unable to hide all the gritty details from developers view in some cases.
  - Frameworks would be more efficient in terms of usability for the developers and technical managers working in large projects targeting multiple platforms, if they can hide platform specific details away from the developers and technical managers

Limitations of the four GUI frameworks(contd.)

- **Incapability of supporting new platform**
  - Beyond libraries are applications frameworks are meant to subsume the entire application in their architecture. The code is absorbed by the framework. As a result there is much larger commitment when choosing and using an application framework. The urgent need of targeting an unsupported platform is not widely supported in these frameworks

Limitations of the four GUI frameworks(contd.)

- **Unavailability of cross platform IDE**
  - IDEs (Integrated Development Environment s) are frequently used by software developers for their daily works in developing software.
  - In order to achieve the goal of developing portable application it is important to make sure that the developers are using different host platforms and tool as is practical.
  - But none of these frameworks provides the common IDE for different platforms or compatibility with other IDEs in order to get a heterogeneous development environment

Limitations of the four GUI frameworks(contd.)

- **Inefficiency of standardized and integrated testing**
  - Porting software involves changing and writing a lot of code that may not be tested on another platform for quite some time, which means that many bugs have a lengthy gestation period.
  - For this reason, it is vital that the developers or testers performs standardized testing to catch bugs as early as possible
  - None of these frameworks provide the integrated testing support to test on multiple platform during the software development process

Limitations of the four GUI frameworks(contd.)

- **Incapability of using variety of compilers**
  - It is important to use different compilers as much as possible during the software development process in order to make the application portable over various compilers.
  - By compiling successfully on a wide range of compilers, it is possible to avoid being left stranded if developer's preferred compiler vendor suddenly disappears or does not provide support to a new unsupported popular platform. This also can ensure the code base does not rely on any compiler specific features

Limitations of the four GUI frameworks(contd.)

- **Incapability of automatic creation of portable document**
  - Portability means flexibility for project development which can be important for projects involving members at different institutions and those which span over many years.
  - None of these four C/C++ GUI frameworks can create automatic portable document by using consistent programming procedures in order to overcome the problems of handing over code to new developers

Limitations of the four GUI frameworks(contd.)

- **Incapability of separating different character encodings**
  - Developers need to make sure that their application can correctly set up the character encodings used by their GUI elements and in their data files
- **Unavailability of any complete platform abstraction library**
  - A platform abstraction library can provide solution to several commonly encountered portability problems. For example, Netscape Portable Runtime Library (NSPR) originally developed by engineers at Netscape and now a part of the Mozilla open source codebase provide abstractions of functionality present on each platform but for which a portable API does not exist

Limitations of the four GUI frameworks(contd.)

- **Incapability of integrate with multiple native IDEs**
  - Portability means flexibility for project development which can be important for projects involving members at different institutions and those which span over many years.
  - None of these four C/C++ GUI frameworks can create automatic portable document by using consistent programming procedures in order to overcome the problems of handing over code to new developers

Suggestions for designing an efficient cross platform GUI framework

- After evaluating the frameworks, guidelines have been indentified for an efficient C/C++ software framework that offers a complete development platform for application domain.
- These guidelines offer instructions to create a framework that enable the exploitation of all the features to easily build cross platform applications for any kind of algorithms and systems.
- These guidelines represent a step forward over existing frameworks in cross platform application development domain. Furthermore, included instructions can be used for achieving the goal of bringing portability in application.

Suggestions for designing an efficient cross platform GUI framework(contd.)

- All of the four GUI frameworks have a problem of sharing a common user interface on most of the platform which can lead to human interface guidelines (HIG) violations for some platforms. This is because currently is not possible to use platform specific classes for drawing interface with all of the four frameworks.
- The nicest way to do cross platform applications come from using a combination of a cross-compiling core and platform specific UI code. Most of the cross-compiling UI libraries from these frameworks are unruly on both platforms. It is required to include a reimplementation of some of the UI libraries of those frameworks to make those portable under human interface guidelines

Suggestions for designing an efficient cross platform GUI framework(contd.)

- User interface designers usually create some prototype with some graphical user interface creation tools (sometimes with photo editing tools) which could be integrated with the framework to provide more flexibility in the area of application development.
- None of these four GUI frameworks are able to hide platform specific dependencies away from the developers and technical managers. This capability in cross platform development is very essential and could be added in these frameworks to give the developers and technical managers a pleasant development environment.

Suggestions for designing an efficient cross platform GUI framework(contd.)

- All four chosen frameworks do not support compatibility with the native IDEs which are useful to improve the applicability and acceptability of the application in that particular platform. An efficient integration should support cross platform development frameworks to surmount a lot of platform specific dependencies
- Software developers should use the four frameworks to achieve the portability of their software application. These frameworks provide tool for creating and simulating portable graphical user interface in software development activities. In spite of providing an integrated environment for software engineers, they do not provide any tool which supports the simulation of whole application in different available platforms. They do not support for automatic configuration for all available platform

Analysis

- What are the features and capabilities of popular cross platform C/C++ GUI frameworks?
  - Major features & capabilities related to portability were discussed with comparison
  - Hardware specific features & capabilities were not mentioned
  - Some features were not mentioned for the knowledge limitations
  - Features & capabilities related to embedded platforms were not included

Analysis

- What are the ideal features and capabilities of an efficient cross platform C/C++ GUI framework?
  - After evaluating the frameworks, guidelines have been indentified for an efficient C/C++ software framework that offers a complete development platform for application domain.
  - These guidelines offer instructions to create a framework that enable the exploitation of all the features to easily build cross platform applications for any kind of algorithms and systems.
  - These guidelines represent a step forward over existing frameworks in cross platform application development domain. Furthermore, included instructions can be used for achieving the goal of bringing portability in application

Analysis

- What is the right C/C++ framework for proprietary and open source portable software development?
  - All the four C/C++ GUI frameworks have done that at achieve the portability. These frameworks bridge the differences between platforms. For the most part, they attempt to abstract and unify access to the window system, input, event handling, multithreading, file I/O, networking, and 2D/3D graphics.
  - These frameworks have tools to build GUI layout and forms, enabling rapid development of comparatively high-performance user interfaces with native look and feel across some supported platforms. T
  - hey also provide a set of tools to ease the effort in internationalization workflow. They also provide other general software development capabilities such as software metrics, change analysis, and quality check to support software developers' tasks.
  - The capabilities and features of the four GUI frameworks have limitations hence they satisfy partly the needs of software developers and technical managers

Conclusion and Future Work

- Firstly, the four chosen framework's infrastructure were investigated separately, the content includes framework introduction, framework key components and the portability issues of the framework. The aim of investigating infrastructure of different frameworks was to reveal the general view of each framework and lays the understanding foundation for the portability features comparison. After the research on the four chosen frameworks I summarized different portability features and some advantages and disadvantages of different platforms
- Secondly, I selected seven essential portability features, which are application data and resource files, file systems, internationalization and localization, compilers, configuration management, operating system interfaces, user interfaces and build system, as the comparison yardstick for cross platform frameworks
- Thirdly, I carried out a comparison of those frameworks based on the above criteria. The advantages and disadvantages of each framework´s portability feature implementation were summarized. Based on the research results of framework infrastructure investigation and portability feature comparison; some suggestions and guidelines were provided for designing a framework for incorporating portability more systematically into software development projects

Future Work

- This thesis provides sufficient grounds for creating criteria for comparing different C/C++ cross platform application development frameworks. A detailed and formal description of the portability domain is obtained, which provides a comfortable and precise framework for reasoning about the involved probabilities. It can help to determine portability requirements clearly that will facilitate more mature evaluations of alternatives frameworks.
- Using the presented analysis and comparison of those frameworks, further research and development can be done. Because of the constant changes in different platforms, the future work of this thesis may focus more on portability feature analysis amendment. Furthermore, the presented recommendations and suggestions for designing an efficient cross platform framework can be used to extend portability features of those available frameworks or to design a new cross platform application development framework

References

- [Bishop, 2006] Multi-platform User Interface Construction –A Challenge for Software Engineering-in-the-Small by Judith Bishop
- [Ezust and Ezust, 2006] An Introduction to Design Patterns in C++ with Qt 4 by Alan Ezust, Paul Ezust
- [Fltk, 2008] FLTK official website, http://www.fltk.org/ , July 2008
- Gtk+, 2008] The Gtk+ project, http://www.gtk.org/ , September 2008
- [Hook, 2005] Writing Portable Code by Brian Hook.
- [J Smart et al, 2005] Cross-platform GUI programming with wxWidgets by By Julian Smart, Kevin Hock, Stefan Csomor, July 2005
- [Jonathan, 2005] Jonathan W. Hoyle Cross-Platform Approaches from a Macintosh Perspective
- [Krause, 2007] Foundations of Gtk+ Development by Andrew Krause

References

- [Logan, 2007] Cross-Platform Development in C++ building Mac OS X, Linux, and Windows Applications by Syd Logan.
- [Mooney, 2001] J.D Mooney, Software Portability Home Page. http://www.cs.wvu.edu/~jdm/research/portability/home.html , September 2008
- [Mooney, 1997] James D. Mooney, Bringing portability to the software process by James D. Mooney West Virginia University
- [Packard and Gettys, 2003] Keith Packard and James Gettys, XWindow System Network Performance, FREENIX Track: USENIX Annual Technical Conference
- [Parnas, 1994] D. Parnas, Software Aging. Proc. 16th Int. Conf. on Softw. Engr., Sorrento, Italy, 994, 279-287.
- [Qt, 2008] Qt homepage, http://trolltech.com/products/ , Setptember 2008
- [Seffah and Javahery, 2003] Seffah, A.; Javahery, H (eds.) Multiple User Interfaces: Cross-platform applications and contextaware interfaces. Wiley & Sons, West Sussex, UK (2003)

Thanks

Questions?