Software Architecture Project Info (Requirement & Architecture phase)

Course staff February, 2010

This document contains info about the main task of the software architecture project, as well as templates for the first delivery. As you should be well acquainted with the COTS (Khepera simulator, XNA, Android and iPhone) by now, we will not go into details about that.

1. The Task

1.1 The robot simulator

The task is to create a robot that will collect all balls on a map and place them near the light. The map to be used is maze.
The following information is given:

- There is one light
- There is at least 4 balls, but the exact number is unknown
- Balls are not placed in awkward positions, like in corners, or in extensions to wall ends.
- The robot can be set to start anywhere, and can start facing left/right/up/down. It will not start at an angle.
- The navigation does not need to be carried out using a map.

1.2 A Game in Android or XNA or iPhone

The task is to make a functioning game using XNA , Android or iPhone, based on your own defined game concept. However, the game must be design according to a specified software architecture. *The XNA game needs to be a multiplayer game*, but it can be turn-based or similar (does not need to be a network game). The Android/iPhone game can be single-player.

2. Group Specific Tasks

Requirements of the quality attributes focus are following:
The group should develop robot controller / game with focus on a primary and secondary quality attribute:

- **Primary quality attribute**: Modifiability (The software architecture and implementation should be easy to change in order to add or modify functionality.)
- **Secondary quality attribute** (you can choose one): Testability (all), Availability (XNA), Safety (Robot, avoid the robot getting stuck), Usability (XNA, Android, iPhone).

3. Your Solution

The perfect implementation is not the ultimate quest (holy grail) of this project, and it counts together with all the documents towards the final grade. Also, the thing to remember is that the implementation and the architectural description should reflect each other, which may lead to the architecture being changed as the finer points of implementation, is realized. The final code and implementation should be well documented, easy to compile-and-run. Much hassle and too much time spent on trying to get a system to run will only irritate the course staff and possible degrade the result.

4. About the Documents

The document should be written in English and must be delivered in the Pdf format. The project will be graded on the final delivery, but documentation with big holes during the project can also be penalized.

4.1 Requirements Document

In this document you are to list the requirements for your application. We should use the following structure:

- Front page where the group name, group members, chosen COTS and domain and quality attribute must be present.
- Introduction: Info about what it contains.
- Functional Requirements: Detail the requirements that you have to fulfill in order to complete the task.
- Quality Requirements: Write at least one scenario for the most relevant quality attributes (e.g. modifiability, testability, availability etc.). Use (textual/table) scenarios of the type used in chapter 4 of the book. Remember to make them measurable/testable by a concrete limit, or by, in some circumstances, a simple true/false evaluation. Set hard time limits, even if you are not fully aware of what is realistic. Avoid phrases like “reasonable time”. And create something nice, table like to write them in, remember... readability. (I repeat, do not write them in the horribly spacious figures of the inside cover of the book).
- COTS Components and Technical Constraints: Detail the constraints given by the COTS (Khepera, XNA, or Android,iPhone), relevant for your project. (If you find other constraints, they can of course be added as well
- References: A list of any references to books, articles, web-pages, documents etc. that you have used.
- Issues: Optional point of issues you have with this document.
- Changes: To be used for evaluating the project at the final delivery in order to tell us what have changed (improved) since the original delivery.

4.2 Architectural Description Document

This is your main document, with the architectural description for the project. The document should contain (based recommendations from IEEE 1471):

- Front page where the group name, group members, chosen COTS and domain and quality attribute must be present.
- Introduction: Info about what the document is, and how it is structured.
- Architectural Drivers: The main drivers that affect the system mostly, including the attribute on which you focus.
- Stakeholders and Concerns: The stakeholders of the system, and their concerns.
- Selection of Architectural Viewpoint: A list of the viewpoints used, their purpose, target audience, and from of description. Places to look for possible viewpoints include the book, and the 4+1 article by Kruchten.
- Architectural Tactics: For all attributes, but more detailed for the one you are focusing on.
- Architectural Patterns: The major patterns of your architecture, both architectural and major design ones.
- Views: A section for each of your chosen views (at least two views). Each should apply to a viewpoint.
- Consistency Among Views: Discuss the consistency between each described view.
- Architectural Rationale: In this section, and sub-sections, remember to add why things are chosen.
- Issues: Optional point of issues you have describing the architecture.
- References: A list of any references to books, articles, web-pages, documents etc. that you have used.

- Changes: Again for the 21st of April. Describe what changes have been made and why. It is not unlikely that the architecture will change slightly during implementation.

(4.3 Code Skeleton) - optional

If time, we would like you implement and hand in a code skeleton of your architecture. This is to emphasize the importance of starting the implementation early, and to ensure that you can implement your architectural design within the constraints given.

4.4 Template for references

Here are some short examples of how to write correct references to books, articles and web- pages respectively:

Book:

- Georg Busk, “Conquer the world in a few week – nukes are fun”, White House Press Limited, Second Edition, March 2005.

Article:

- G. Bush, “Frankly I don´t care – The history of Iraq”, First Conference on Global Assassination (FCGA’06), Bagdad, Iraq, Feb. 23-12, 2006.

Web-page/site:

- Bush Company Inc, “Texas and Guns – The homepage for Rights to shoot your neighbor in the face or another place”, accessed 23. February 2004