ENES 489P Project Template

Your ENES 489P project should begin with a concise problem statement and follow the system development procedure implied by Figures 3.12 and 3.13 in the class notes. The purposes of this project template are threefold. First, we want to help you keep your project on course. You can use the template as a project checklist. Finally, we hope that the template will result in a consistent look-and-feel for all of the projects developed in this course.

Last Modified: September 18, 2012.

Project Title:
Students:
ISR Advisor:
Industry/Government Advisor(s): (if applicable).

Project Abstract

Provide a 150-200 word summary of your project.

Part 1. Problem Statement

Using easy-to-understand English and pictures, describe the problem area that will be addressed by your project. Your problem statement should set the stage for the content to follow and address "big picture" issues such as:

- Why is this work needed? Who will use it? And where?
- Who are the project stakeholders and, generally speaking, what are their concerns?
- If successful, what are the potential benefits of this project?
- What factors are likely to drive the economics of development?
- Is there anything new in your project?

Note. The conventional metrics for a good design are cost/economics, functionality (what a system does), and performance (how good/reliable is the functionality). However, this list is not exclusive. Some projects will also be measured in terms of their sensitivity to political, social, and ethical concerns, health and safety, manufacturability and sustainability.

Part 2. Use Case Development

Develop a detailed use case model for your project. A few questions that you might ask are:

- Who are the project stakeholders?
- Who are the actors?

- What are the individual functionalities (use cases) that the system will support?
- What does the system boundary look like?
- What is the relationship between each actor and the individual use cases?
- Are there dependencies among the system functionalities (use cases)?

Part 3. Textual Scenarios

For each use case (fragment of system functionality),

- Indicate when the use case will be triggered (i.e., pre-conditions).
- Develop a detailed textual scenario for the normal flow of events when everything works as expected.
- Describe how the system will deal with exceptions and/or failures.
- What will be true after the use case has finished (i.e., post-conditions).

At this point you should:

- Develop a vocabulary of key terms for your project.
- Create a list of objects/subsystems that are implied by the scenarios.
- Create a mapping of use cases to identified objects and subsystems.

Reference: See pages 238-241 of the class notes.

Part 4. Simplified Models of System Behavior

Create activity diagrams and sequence diagrams to visually represent the sequences of events and/or tasks implied by the textual scenarios. (If at all possible use large fonts (16 pts) in your diagrams so that print versions will be readable.)

- Make sure that the flows of events and tasks in your diagrams are consistent with the textual scenarios.
- Organize your activity diagrams into swim lanes so that you can see who will be responsible for each part of the use case execution.
- Make sure that the actors in your sequence diagram are consistent with the actors in your use case diagrams.

Reference: See page 199 of the class notes. Also see Adrian Marsh's Project on the Combat Marking System from 2003.

Part 5. Requirements Engineering

Develop and organize the requirements for your project. I suggest that you organize your requirements into three categories:

- Functional requirements,

- Non-functional requirements,
- Interfaces requirements.

Where possible, translate textual requirements into sets of inequality constraints which can be quantitatively evaluated.

Part 6. System-Level Design

Organize the system objects into a system architecture. I expect that for most projects, the architecture will be a combination of hierarchy and network structures.

For each object, identify sets of attributes that will be related to the measures of effectiveness.

Create at least two system-level design alternatives by assigning elements of system functionality to specific components in your system architecture.

Part 7. Simplified Approach to Tradeoff Analysis

Using either CPLEX or a spreadsheet/tabular organization, conduct a systematic study of design options to determine the trade spaces for your problem domain.

All projects must include cost as a metric and at least two other metrics (e.g., performance, reliability, accuracy, sustainability) relevant to your design.

Part 8. Summary and Conclusions

Briefly summarize the key challenges that you faced in the development of your project. What did you learn? What would you do next if you had an additional semester to work on the project?

Part 9. References

Provide a complete list of references for your project.

Part 10. Credits.

Please indicate the primary contribution to the project by each of the team members.