New York University
Computer Science Department
Courant Institute of Mathematical Sciences

Course Title: Software Engineering
Course Number: g22.2440-001
Instructor: Jean-Claude Franchitti
Session: 4

(Individual) Assignment #2 – Requirements Analysis - Requirements Modeling - High-Level Business Architecture Analysis and Design

I. Due

Thursday March 20 2014 at the beginning of class.

II. Objectives

1. Learn how to analyze a problem scenario.
2. Learn how to document and analyze requirements on an ongoing basis.
3. Learn how to get started on requirements model engineering, high-level analysis and design, and detailed level analysis and design.
4. Learn about blueprinting techniques, the User Requirements Notation (URN), the Business Process Modeling notation (BPMN), and the Unified Modeling Language.

Disclaimer: The artifacts created in this assignment will be used to drive forward subsequent technical planning in terms of deliverables, and most importantly the actual form of the object model you will use to represent the architecture. It is therefore important to create and record the artifacts as formally as possible.

III. References

1. Slides and handouts posted on the course Web site.
2. IBM Rational UML resource center: http://www-306.ibm.com/software/rational/uml/
3. IBM Rational systems and software design: http://www-01.ibm.com/software/rational/offerings/architecture/softwaredesign.html

IV. Software Required

1. IBM Rational RequisitePro
2. jUCMNav URN Editor

3. Architecture Modeling Tool with appropriate BPMN plugin (e.g., SparxSystems Enterprise Architect, IBM Rational Software Architect, IBM Rational Software Modeler, etc.)
1. Win Zip as necessary

V. Assignment

1. Problem Statement:

   Problem Scenario #1:

   As the head of information systems for a college you are tasked with developing a new student registration system. The college would like a new client-server system to replace its much older system developed around mainframe technology. The new system will allow students to register for courses and view report cards from personal computers attached to the campus LAN. Professors will be able to access the system to sign up to teach courses as well as record grades.

   Due to a decrease in federal funding, the college cannot afford to replace the entire system at once. The college will keep the existing course catalog database where all course information is maintained. This database is an Ingres relational database running on a DEC VAX. Fortunately the college has invested in an open SQL interface that allows access to this database from college's Unix servers. The legacy system performance is rather poor, so the new system must ensure that access to the data on the legacy system occurs in a timely manner. The new system will access course information from the legacy database but will not update it. The registrar's office will continue to maintain course information through another system.

   At the beginning of each semester, students may request a course catalogue containing a list of course offerings for the semester. Information about each course, such as professor, department, and prerequisites, will be included to help students make informed decisions.

   The new system will allow students to select four course offerings for the coming semester. In addition, each student will indicate two alternative choices in case the student cannot be assigned to a primary selection. Course offerings will have a maximum of ten students and a minimum of three students. A course offering with fewer than three students will be canceled. For each semester, there is a period of time that students can change their schedule. Students must be able to access the system during this time to add or drop courses. Once the registration process is completed for a student, the registration system sends information to the billing system so the student can be billed for the semester. If a course fills up

during the actual registration process, the student must be notified of the change before submitting the schedule for processing.

At the end of the semester, the student will be able to access the system to view an electronic report card. Since student grades are sensitive information, the system must employ extra security measures to prevent unauthorized access.

Professors must be able to access the on-line system to indicate which courses they will be teaching. They will also need to see which students signed up for their course offerings. In addition, the professors will be able to record the grades for the students in each class.

Problem Scenario #2:

You have been tasked to develop the architecture for a mine pump control system, designed to monitor and pump flood water out of mine shafts. As underground mining operations take place far below the water table, flooding into mine galleries and shafts is an ever-present danger. Excessive flooding is clearly a safety hazard for workers, but also has profitability implications ranging from equipment damage to productivity delays, to mine closures in extreme circumstances.

The system to be developed will be required to monitor the water level in a given mine shaft using two sensors. A high water sensor that measures the maximum acceptable level of flooding in a shaft before pumping begins, and a low water sensor, which measures the minimum level of acceptable flooding and pumping stops. These sensors are used to start a mine pump. When the flooding level exceeds the level determined by the high water sensor the pump is switches *on*. When the water has been pumped out and the minimum level of acceptable flooding has been reached, as measured by the low water sensor, the pump switches *off*.

In addition to flooding mining is often hindered by methane pockets, where gas seeps into the shafts and galleries triggering an evacuation. Again this is a safety hazard, the mining staff won't be able to breathe, and even more critically, operating equipment may generate sparks which will cause the methane to ignite. Therefore the system will include a methane sensor that will be used to trigger an evacuation alarm in the presence of dangerous levels of methane (measured in N parts per million), and also switch *off* the pump regardless of the current water level.

The system is used by two key roles. The first is the *Operator*. This role is required to log in to the system with a username and password. Following a successful login the operator is able to start or stop the pump if, and only if, the

water level is between the high and low sensor limits. The second role is the *Supervisor*. A supervisor must verify their security credential as per the operator above. Following a successful login they are able to switch the pump on, or off at any time. For example a supervisor could run the pump after the flood level has dropped below the level set by the low water sensor. They could also switch the pump off if the water level goes over the maximum high level of flooding. In these cases the supervisors' actions override the automatic behavior of the pump. A supervisor is required to "reset" the pump system in order to re-establish automatic behavior.

Finally to meet Federal monitoring standards a persistent log is required to capture the following events:

- ✓ Pump switched on by high water sensor
- ✓ Pump switched off by low water sensor
- ✓ Pump switched on or off by operator or supervisor
- ✓ Evacuation alarm triggered by methane sensor
- ✓ The reading of the methane sensor every 30 minutes

The reading of the methane sensor (for the last 24 hours) can be read by the operator. All readings (up to 30 days) can be read by the supervisor. The supervisor also has the capability to add a "note" to any specific log event that occurs within 24 hours.

2. Your company has decided to utilize "best practices" in requirements elicitation and iterative development practices. As a consequence, for each of the problem scenarios described in the problem statement section, perform a problem analysis and capture high-level requirements in RequisitePro. You may use the RequisitePro EAMF template for this purpose. Perform a business solution/system investigation, and analyze each business solution/system to refine your requirements as needed.

3. Perform high-level architecture analysis and design and come up with a set of blueprints that are representative of business solutions suitable for each of the problem scenarios described in the problem statement section.

4. Perform requirements model engineering and specify a set of essential and real use cases describing the functionality that your business solution/system will exhibit. To start with, using URN GRL diagrams, identify goals/subgoals, actors and their dependencies, and high-level tasks. Perform scenario modeling using URN UCM diagrams and group scenarios as needed to identify essential use cases. Using essential/real Use Case diagrams identify the following:

   - ✓ Actors – who/what interacts with the business solution/system?
   - ✓ Use Cases – what behaviors does the business solution/system exhibit?

5. For each use case you have identified, fill out the Use Case Specification (Description) template, one for each use case. Document the Use Case Specifications/Descriptions using Microsoft Word processing software. Give each Use Case an ID and name each Use Case description file (e.g., for the first problem scenario this could be **UC1_Login.doc, UC2_Register.doc**, etc.) **Save each file as a Word document**. A template of the Use Case Description can be found on the next page and is also available on the course Web site under Handouts.

6. Draw the URN GRL and UCM diagrams using jUCMNav and the UML Use Case Diagrams using a UML modeling tool of your choice.

7. Develop preliminary business process diagrams based on the UCM scenarios identified so far.

8. Establish proper requirements traceability in RequisitePro all the way down to Use Case requirements. Please integrate the Use Case template mentioned in question 4 with the Use Case specification document used in RequisitePro.

9. What additional relationships such as extends, generalization-specialization etc can be used to refine your Use Case models?

10. Save the model files as XMI files (or other available formats) and place them in a zip file named **SE_Sp11_*firstname_lastname*_hw_2.zip** (e.g., "SE_Sp11_john_doe_hw_2.zip)**.**

11. Email your assignment file to the course grader, and submit a hard copy of the requirements specification file to the professor by the due date. Use the following subject line naming convention when submitting emails related to assignment submissions: "SE Spring 2011 – *firstname lastname* – Assignment #" (e.g., "SE Spring 2011 – John Doe – Assignment 2"). Please note that all files/archives submitted must include your name.

Use Case Specification

Use Case ID: *{This should be coded to identify the owning team and the level of the use case}*

Use Case Name: *{Short descriptive phrase}*

Relevant requirements: *{Reference to other relevant requirements documents.}*

Primary Actor: *{Main sub-system/entity that initiates use}*

Pre-conditions: *{Requirements on the state of the system prior to this use being valid.}*

Post-conditions: *{This describes the state of the system following the successful completion of this use. Affects on other systems and actors may also be described.}*

Basic Flow or Main Scenario: *{Numbered flow of events: 1 The user initiates an action by… 2 The system responds by**…}***

Extensions or Alternate Flows: *{This section presents variations on this use case. It presents those use cases that have an extends relation with the current use case.}*

Exceptions: *{This section describes all error conditions that can arise in the use case.}*

Related Use Cases: *{use cases that are either usually performed just before or after the current use.}*

Revision History -- *{Follow the standard corporate document versioning template}*

Date	Description	By

VI. Deliverables

1. Electronic:
   Your assignment files (**SE_Sp11*_firstname_lastname_*hw_2.zip, *RequisitePro projects, URN-related files, and* UC*_*.doc files***) must be emailed to the TA. The files must be sent by the beginning of class. After the class period, the homework is late. The email clock is the official clock.

2. Written:
   Printout of each diagram and each Use Case description file.
   Printout of RequisitePro project reports.
   The cover page supplied on the next page must be the first page of your assignment file
   Fill in the blank area for each field.

NOTE:

The sequence of the hardcopy submission is:

1. **Cover sheet**
2. **RequisitePro Project Reports**
3. **Blueprints**
4. **URN GRL and UCM Descriptions**
5. **UML Use Case Descriptions**
6. **UML Analysis Model**

Name _____ Username: _____ Date: _____
(last name, first name, username is SID)
Section: _____

Assignment 2 Assessment

Assignment Layout

☐ Assignment is neatly assembled on 8 1/2 by 11 paper.
☐ Cover page with your name (last name first followed by a comma then first name), username and section number with a signed statement of independent effort is included.
☐ Problem and system analysis documentation is correct.
☐ File names are correct.

Requirements Definitions

☐ RequisitePro projects.
☐ Completeness of requirements definitions.
☐ Correctness and appropriateness of requirements traceability hierarchies.

Use Case Description(s)

☐ Follow standard Use Case Specification Template.
☐ Assumptions provided when required.
☐ Completeness of descriptions.

Blueprints, URN, UML, and BPMN Diagrams

☐ Blueprints, GRL, UCM, BPMN, and essential/real Use Case Diagram(s).
☐ Completeness of diagrams.

Total points _____
Professor Comments:

Affirmation of my Independent Effort: _____
(Sign here)