

**MONASH UNIVERSITY**  
**Clayton School of Information Technology**

**CSE3308 - Software Engineering: Analysis and Design**  
**Assignment 2 - Semester 1, 2006**  
**Group Analysis and Design Project**

**Computer-Aided Software Engineering Tool**

**Introduction**

The Group Analysis and Design project requires that you form yourselves into groups of four or five students and create a set of Object-Oriented Analysis and Design (OO A&D) documents in UML. You must use the methodologies presented to you in the lectures to prepare the OO A&D documents.

Structured Analysis (SA) emerged during the 1970s and 1980s, and developed into a number of forms. It was the dominant analysis paradigm for software development during the eighties and into the nineties. In 1989, Ed Yourdon published "Modern Structured Analysis", which introduced a standardised notation for SA. This facilitated the development of Computer-Aided Software Engineering (CASE) tools supporting SA.

Structured Analysis has recently been superseded in popularity by Object-Oriented A&D. Nevertheless, large amounts of existing software, that must still be maintained and/or re-engineered, was developed using SA. Consequently, software engineers today need an understanding of SA.

**1. Assignment Task**

Your task is to produce a preliminary Object-Oriented Analysis and Design for a CASE tool to do Structured Analysis, using Yourdon notation. Your reference for the specification of Yourdon notation is to be *Just Enough Structured Analysis* by Ed Yourdon. You should access this by using the link on the Resources page of the unit web site:

<http://www.csse.monash.edu.au/courseware/cse3308/html/resources.html>

For this preliminary OO A&D, only a subset of all possible SA diagrams and documents need be supported. Users of the CASE tool will employ it to analyse systems. For a given system, the CASE tool must allow the user to create:

- A Context Diagram (CD)
- A set of levelled Data Flow Diagrams (DFDs)
- Process Specifications (PSPECs) for the lowest level processes in the DFDs. Your model must support the existence of several kinds of PSPEC (Structured English, Pre- and Post-Conditions and Decision Tables), and be extensible to support other kinds in future.

- An Entity Relationship Diagram (ERD) for the datastores in the model (follow the lecture notes in preference to Yourdon for these)
- A Data Dictionary (DD) for all the dataflows, entities and datastores in the model.

Consider what happens when an architect designs a house. The architect has a mental model of the house and draws plans to reveal aspects of this model. There is one mental model but there may be front, plan and side elevations as well as perspective views. There may also be a schematic plan giving a lay-out of the rooms. All of these views have to be consistent with the architect's overall mental model.

The emphasis of this assignment is not on the design of a GUI. You should focus on the idea that the CASE tool is allowing someone to design a system. The designer has a mental model of the system and the different kinds of diagrams reveal particular views of this system. Thus, the Context Diagram, the Data Flow Diagrams and Process Specifications allow us to see the system from the point of view of the data flows between modules while the Entity-Relationship Diagrams highlight the relationships between records in the datastores and the Data Dictionary focuses on the data that flows between modules.

A CASE tool is more than a drawing tool. It does not simply allow a user to create diagrams using certain symbols and labels - it understands the rules for using those symbols: which symbols may appear in which diagrams, and what kinds of relationships that they should have. The CASE tool can and should assist users to maintain the completeness, correctness and consistency of their SA diagrams and documents, e.g.:

- Processes must have both incoming and outgoing dataflows. If the user's model contains a process that does not obey this rule, this should be indicated to the user.
- Every dataflow in the DFDs must have a corresponding entry in the DD. When the user creates a dataflow, the tool should assist them to select from the existing DD entries, or to create a new DD entry.
- Levelled DFDs must be balanced-the lower-level DFD corresponding to a process at the level above must have the same incoming and outgoing dataflows as that process.

These are just a few examples. There are many such rules about the necessary relationships between the elements of SA diagrams and documents, and between the diagrams and documents themselves. They are described in the lecture notes, and also in *Just Enough Structured Analysis*. An important part of the assignment work will be reading and understanding these sources so that you understand the necessary relationships between the SA diagrams and documents well enough to model them in your OO A&D.

You will also need to think about the various ways in which a user could interact with the CASE tool to produce and edit their analysis. This will help you to develop your use cases. Experimenting with other tools, such as Rational Rose, may help you to get a feel for this.

If you are unsure about a requirement, and cannot resolve your question through reference to the lecture notes or *Just Enough Structured Analysis*, you should post a question about it on the newsgroup on the unit web site.

**2. Timeline Week 5 (30<sup>th</sup> March)** - Groups formed and Group Information Sheet submitted to the lecturer by the end of the Thursday lecture.

Analysis and design documents are submitted through through the Third Year assignment box outside the Clayton School of IT office in building 75, with completed Group Assignment Cover Sheet (see <http://www.csse.monash.edu.au/subjects/2006/clay1.html>).

Minutes of Group Meetings and Technical Reviews that occurred in the development of a deliverable should be submitted with that deliverable.

If during the course of the assignment you need to change any analysis and design documents that may have already been submitted and assessed, the updated versions of those documents should still be submitted as one of the assessment criteria will be the extent to which all your analysis and design documents are consistent.

**Start of Week 10 (Monday 8<sup>th</sup> May)** - Part A: Use cases, minutes and technical reviews due.

**Start of Week 12 (Monday 22<sup>nd</sup> May)** - Part B: Complete class diagram, without methods, minutes and technical reviews, as well as any updated documents, due. You do not need to specify basic utility classes.

**Start of Week 13 (Monday 29<sup>th</sup> May)** - Part C: Submission of the completed analysis and design documents as well as any updated documents.

### 3. Allocation of Marks

This assignment is worth 45% of the marks for the subject. All group members will receive the same mark for their assignment, modified by the average evaluation they receive from the other group members. All members of the group will rate the contribution of all other members of the group, and these ratings will modify the mark that each individual receives, but not by more than 20% of the group mark.

The week 10 submission will be worth 18% of the marks for the subject.

The week 12 submission will be worth 18% of the marks for the subject.

The week 13 submission will be worth 9% of the marks for the subject.

The assignment will be marked using the following criteria:

**Presentation:** A professional level of presentation is expected for all documentation. The following are the minimum requirements:

- All documentation should be typed
- All documentation should be adequately bound
- All pages should be numbered
- A Table of Contents should be provided
- All documents must clearly indicate their creator(s)
- Diagrams may be hand-drawn, but this is not recommended.

**Suitability of the Analysis and Design:** This is measured by how well the analysis and design matches the requirements of the system. Not only must all requirements be covered, but the analysis

must model the requirements correctly. Requirements must be consistently supported across all the design documents.

**Extensibility of the Analysis and Design:** This is measured by how easily the analysis and design could be extended to handle relevant new requirements of the system in the future.

**Correctness of the Use of the Modeling Technique:** This is measured by how well your analysis and design follows the standards of the techniques applied (e.g. OO A&D using UML). It is vital that the various components of the OO A&D model correspond to each other, and that all relationships between the model components are correctly and consistently maintained.

**Minutes of Formal Technical Reviews:** You must submit the minutes of the technical reviews you carried out for each of the work products. These will be assessed for completeness and adherence to standards.

#### **4. Tutor Inspections**

Each group will be allocated a tutor, who will be their main point of contact for this assignment throughout the semester. In particular, you must get your tutor to inspect your expanded use cases, class diagram, interaction diagrams and state diagrams. You should seek a tutor inspection as soon as you have done a walk-through of the product in your group and are reasonably satisfied with it. The tutor will provide feedback, and keep records of having seen the deliverables. Note that you are still allowed to change these products after inspection.

#### **5. Group Work Problem Resolution**

It is quite common for groups to have significant difficulties working together, in both industry and at university. It is therefore necessary for students to behave in a professional manner while doing this assignment. If your group is having difficulty working together, here is the appropriate method for dealing with it:

- (1) Approach the person with whom you are having a problem and attempt to solve the problem on a one-to-one basis.
- (2) Attempt to solve the problem as a group.
- (3) If there is still a problem, then the group should approach the lecturer and ask him or her to arbitrate. You **must** do this as soon as it becomes apparent that the problem can not be solved by the group. Do **not** leave it until there is a submission deadline to raise the problem with the lecturer. A claim that a student did not contribute his or her fair share will not be considered if it is made just prior to the submission of the assignment, or after submission.

#### **6. Extensions**

There will be no extensions for this assignment. As there will be a minimum of four students working on the assignment and the assignment is over a period of 2 months, the illness of one student does not justify any extensions.

## 7. Submission Requirements

The following documents must be submitted:

- Expanded Use Cases
- Use Case Diagrams which summarize the expanded use cases
- Class Diagram(s)
- Package Diagram (if necessary due to having a large number of classes)
- Operation Specifications: a specification should be provided for each non-trivial operation of each class. You may use pseudo-code, or pre- and post-conditions as appropriate. "get" and "set" operations are trivial.
- Interaction Diagrams (either Collaboration or Sequence Diagrams. You choose.)
- State Diagrams for objects with interesting life cycles. An object that is simply created, has a single state, and is then destroyed does not have an interesting life cycle.
- Minutes of Group Meetings and Technical Reviews

## 8. Additional Notes on Assignment Scope

- **It is not necessary to implement the system in any fashion.**
- **Design of the user interface is not required.** You do not need to provide screen layouts.
- **You do not need to design operations or classes to handle user interface details.** You may assume that an all-encompassing user interface package exists that is able to send output to the user, obtain input from the user, and send messages to the objects in your design. You are only concerned with the contents of this input and output, not its presentation. Classes corresponding to diagram elements should have an operation such as `drawSelf()`, which is called whenever an object of that class needs to be rendered on the screen. No specification is required for these operations.

## 9. Attachments

You will find the following attachment:

- **Peer Assessment Sheet.** Each student must complete this sheet and hand it in independently of the other members of the group. They are used to assess the contribution that each member made to the team. This sheet is to be handed in with the final submission in week 13.

### Attachments

Peer assessment sheet