Research Applicant Tracking System
COMP9116 Project
Draft Requirements

July 30, 2008

Due date  Week 14

Please note: this is a draft assignment specification.
Please notify any problems in the specification.

1 The project

The project is to construct requirements, specification, refinement and implementation of a Research Applicant Tracking System (RATS) starting from informal requirements documents. A proposal for this system is a current fourth year undergraduate thesis proposal by Albert Nymeyer, see the project proposal on the class Project page. This project is also being used as a requirements exercise in SENG1031.

1.1 Brief Overview

RATS is a system that is intended to track the various activities required for managing research students within the School of Computer Science & Engineering. A document describing the current, manual system can be found on the class Project page.

1.2 First Project Step

The first step is to build a set of requirements for the system based on the informal documents. In our case we want to present the requirements as:

Concepts to be modelled: the actors and events with which the system is concerned.

Actions: the actions (operations) that the system is required to carry out. The actions need to identify the system components on which they depend.

2 Requirements

2.1 Concepts to be Modelled

The system must model the following:

Administrator the person who runs that system, otherwise known as Albert.
Applicants are potential postgraduate students who have applied to undertake a research project.

Proposals are submitted by applicants when they apply.

Offers of acceptance are given to an applicants.

Students are enrolled postgraduate students.

Supervisors supervise students.

Reviews of students must be arranged at regular intervals.

Coordinators coordinate student reviews.

Courses are the courses in which Students are enrolled.

Theses are submitted by students prior to completion.

2.2 Operations

1. applicant ←− Apply(personalinfo, proposal, program): a person makes an application to enter a program, providing personal information, personalinfo, a proposal.

2. offer ←− Offer(applicant): an applicant is made an offer.

3. student ←− Enrolment(offer): a student accepts an offer.

4. ReEnrolment(student): a student re-enrols.

5. approval ←− Approval(supervisor, applicant): a supervisor gives approval/disapproval to supervise an applicant.


7. review ←− ScheduleReview(student, coordinator, date): schedules a review for a student, with review coordinator.

8. report ←− ReportReview(coordinator, review): reports the result of a particular review conducted by coordinator.

9. TransferProgram(student, program): transfer a student to another program.

10. UpgradeStudent(student, program): upgrade a student’s program, for example from Masters to PhD.

11. DowngradeStudent(student, program): downgrade a student’s program, for example from PhD to Masters.

12. RequestDiscontinuation(student): a student requests discontinuation.

13. RequestProgramLeave(student): a student requests program leave.

2.3 Modelling Sets, Constants and Definitions

Context machines should be used for sets, constants and definitions.

You will need to model date, and time.
3 What you should do

1. Develop “low-level” machines with preconditioned, fragile operations.
2. Animate the specifications and develop animation scripts that illustrate the checking of the various conditions in the requirements.
3. Discharge the proof obligations.
4. Develop robust specifications.
5. Discharge the proof obligations.
6. Develop refinements.
7. Discharge proof obligations.
8. Develop implementations.
9. Discharge as many proof obligations as possible.

4 The Challenge

The challenge of this assignment is to resist any temptation to rush to implementation, and to work your way through the refinements, implementations and associated proof obligations. This way you will experience the novelty of generating code that must be correct, modulo the correctness of the specification.

   Good luck!