UNIVERSITY OF OXFORD
SOFTWARE ENGINEERING PROGRAMME

Wolfson Building, Parks Road, Oxford OX1 3QD, UK
Tel +44(0)1865 283525 Fax +44(0)1865 283531
info@softeng.ox.ac.uk www.softeng.ox.ac.uk

Part-time postgraduate study in software engineering

Extensible Markup Language, XML

9th – 13th March 2015
ASSIGNMENT

The purpose of this assignment is to test the extent to which you have achieved the learning objectives of the course. As such, your answer must be substantially your own original work. Where material has been quoted, reproduced, or co-authored, you should take care to identify the extent of that material, and the source or co-author.

Your answers to the questions on this assignment should be submitted to:

Software Engineering Programme
Department of Computer Science
Wolfson Building
Parks Road
Oxford OX1 3QD

Alternatively, you may submit using the Software Engineering Programme website — `www.softeng.ox.ac.uk` — following the submission guidelines. The deadline for submission is 12 noon on Tuesday, 28th April 2015. If you have not already returned a signed assignment acceptance form, you must do so before the deadline, or your work may not be considered. The results and comments will be available after the next examiners' meeting, during the week commencing Wednesday, 1st July 2015.

ANY QUERIES OR REQUESTS FOR CLARIFICATION REGARDING THIS ASSIGNMENT SHOULD, IN THE FIRST INSTANCE, BE DIRECTED TO THE PROGRAMME OFFICE WITHIN THE NEXT TWO WEEKS.

Clinical Trial Information System

Clinical trials of medical therapies constitute one of the main sources of new medical knowledge. They are conducted to allow safety and efficacy data to be collected for new drugs or devices. These trials can only take place once satisfactory information has been gathered on the treatment being researched. Traditionally, clinical trials are published only in text-based form, which requires trained humans to translate reported trials into a representation usable for knowledge-based information systems. This assignment aims to allow you to develop the technology underpinning such clinical trials. In order to do so patient data will need to be stored and each visit with the clinician recorded so that the treatment can be monitored. We will not concern ourselves with issues such as the confidentiality of the data; we are primarily interested in capturing it. However we will consider issues such as value domains, units, and variability of data over time.

1 Patient Records

Patient records will be referred to by a number and contain three parts. The first captures the static information regarding a patient, the second part captures information about the drug trial, and the third consists of questions or tests that the doctor will ask the patient each time they visit.

1. Personal Information might include name, contact details, date of birth, and blood group.

2. Drug Trial Information would include the drug name, dosage information, and a description of the desired effect.

3. Each Medical Examination in the sequence would need to record the date, list of medical questions/tests, and patient answers/results.

The list of medical questions will be specific to the drug trial. Consequently you will not be able to say all that much about them initially. However some of the questions might have simple `Yes`, `No`, `Maybe` responses. For instance `"Did you sleep well?"` or `"Have you suffered any pain?"` Other questions might require a value and a unit of measurement; consider `"Weight of Patient?"`. Others might also need some information on how they were measured; consider `"Temperature?"` as an example.

Task 1: Design an XML document structure for a *PatientRecord*. Provide a sample of three patients, each undergoing a different clinical trial with a variety of medical examinations. Assume that medical examinations are ordered by date.

Task 2: Develop a Schema for a *PatientRecord*. Discuss the software engineering aspects of your design.

Task 3: Write a presentation XSLT Stylesheet that generates (X)HTML to display a *PatientRecord* document. Demonstrate its action on your sample document.

2 Validating Medical Questions

In the previous section you developed an XML structure that captured the patients' data clearly but was rather generic in how it handled medical examinations. Hence, there is a need for modularity in the validation process that allows one to capture the more specific requirements of a given clinical trial.

Task 4: Develop accurate Schemas for the medical examinations of your three clinical trials. Each trial needs an associated Schema which ensures that the correct questions are asked, and answers have the appropriate forms. Using modularity, extend your solution to Task 2 so that top-level Schemas can be written that accurately describe a *PatientRecord*. Note: features of XSD 1.1 are *not* required for this assignment.

Task 5: Demonstrate how modular Schemas enable your sample of three patients to be accurately validated. Test your solutions on a variety of incorrect documents. Try to show completeness in your fault detection, demonstrating how it deals with various units and modes of measurement for instance.

3 Evaluating Drug Trials

It is best practice in trial design to specify evaluation criteria at the start, in order to reduce potential sources of bias that may arise if criteria are determined having already observed some results. In this section you will

therefore extend your trial designs with evaluation criteria related to an individual patient, and develop a stylesheet to process these automatically. (Note that you need not consider the overall evaluation of the trial on multiple patients.)

The criteria should enable classification of outcomes such as 'positive effect,' 'no effect' and 'negative effect.' For instance, someone taking Erythropoietin would see a continual rise in their red blood cell count. So we might want to state that a particular value will be increasing throughout the trial, or that 'Rash Size' (under some measurement) has decreased by the end of the trial, or that 'Quality of Sleep' is equal to 'Good' or above. Moreover you might want to specify more complex interactions using boolean and relational operators, combining multiple criteria to obtain a final classification. A new treatment, for instance, might be expected to treat the disease successfully whilst minimising side effects.

Task 6: Develop a *DrugTrialEvaluation* XML document structure. Argue your design decisions and choice of constructs. Add evaluation criteria to each of your three drug trials.

Task 7: Write a Stylesheet that evaluates your *DrugTrialEvaluation* criteria with respect to a *PatientRecord*, to determine what the trial outcome was for that patient. (Tip: the processing of evaluation criteria will be similar to the *MathML* exercise from the course.) Demonstrate the evaluator working on your sample of patients and evaluation criteria.

Submission

Submit a document which lists for each task both a brief summary of the approach taken to solve it, and an argument justifying the chosen design along with crucial code fragments (of XML documents, Schemas and Stylesheets). Leave the full commented listings, including of test cases, for the appendix. Both practical effectiveness and readability are important: your report should be well presented. XML documents must be clear, and the comments appropriate.

Your solutions should demonstrate good software engineering principles. XSLT Stylesheets should not be tightly coupled to the input XML document structure. Modifications to the XML document structure should require minimal changes to the Schema and Stylesheet documents. Sample

documents and test cases should be chosen to demonstrate your skill with the course material, and that your solutions behave as claimed.

The submitted report should be in the region of 20 pages (maximum 25), excluding appendices. Provide an abstract that lists your main achievements and those parts of the assignment that you managed to accomplish. Include your XML documents, Schema documents and Stylesheet transformations in an appendix using a small font (8 point will do).

Assessment Criteria

This assignment is intended to evaluate:

1. your ability to design a simple XML document structure, provide a Schema for it, and write a Stylesheet to transform it into HTML;
2. your ability to use intricate XPath expressions to search and select relevant parts of an XML document;
3. your ability to use Stylesheets to perform complex tasks such as composition and staged computation;
4. your ability to create relevant examples to test your Schemas and Stylesheets;
5. your ability to describe and justify the prominent aspects of your designs and implementations.