
SOFTWARE ENGINEERING PROGRAMME
UNIVERSITY OF OXFORD
www.softeng.ox.ac.uk



ASSESSMENT

Student: Mayur Pant

Course: Extensible Markup Language

Date: 9th March 2015

Grade: 81

REPORT

This was a very good assignment, and with a few improvements could have reached model solution quality. I liked your summary in the abstract and reflection on what you had learned. A pervasive issue is the lack of example snippets in the report, which meant I had to work through the appendices to fully understand what you had done in many cases. There is also some scope for improvement of your XSLT style.

Starting task 1 with a mention of 3 root nodes was disturbing, given that an XML document can have only a single root. From the text it was hard to determine what your document structure actually was; an example snippet would have been very welcome, perhaps with the text content omitted for brevity, or at least a diagram outlining the structure. Thankfully from the appendix I discovered that you did have a single root element. There was also some inconsistency in your naming convention (contrast personalInformation and email-address). Nevertheless, your discussion clearly shows that you have thought about the problem domain and how to represent different kinds of information, with consideration of alternative approaches and a justification of the chosen one. It was also good to show an appreciation for XSD and XSLT considerations in designing your structure. Using xs:unique and/or xs:key would have been better than xs:id, but this is a relatively minor point.

You chose a sensible XSD layout for task 2, however you should have justified the choice of design pattern – Venetian Blind is not always the best! (It is reasonable in this case however.) Your sources are cited well, and again the discussion of your design is well argued. There is consideration of alternative approaches and the needs of the other tasks. XSD features are used with skill, upon examining the appendix. Again, you are overly sparse in not including example snippets. Comments are used well.

Such general comments apply equally to task 3. I particularly liked you experimenting with different output formats in order to show mastery of a range of XSLT techniques. There is some scope for improving your style, although you are clearly competent. Generic select attributes on apply-templates could have been used more, and indeed omitting the select when appropriate. You also overuse xsl:if; it is redundant for displaying the middle-name, for instance, since no value will be displayed if the element does not exist and multiple spaces are collapsed in rendered HTML. Applying tem-

plates to “GB-address|US-address” would also have been better than using `xsl:if`. The XPath expression “`.//*`” can be replaced by the simpler “`*`”.

For task 4 the approach to modularity and discussion thereof is good, although would have benefited from a comparison of the merits of `xs:import` and `xs:include`; you clearly understand and use them well, however. Sharing common types was particularly welcome. You have an impressive range of kinds of question and test, and physical units. The definition of `query1ResponsePairType` (and similar types) shows a couple of minor issues: it is normal to use sequence rather than choice for a single child, and `xs:group` is overkill here since there is only one use of the group and it comprises the entire content of the type.

Your discussion of the purpose of testing is good, as is the concise presentation of your test cases, and balancing precision with effort. There is a good range of testing included, particularly on the limits of regexp matches. Some repetition could be reduced – if you have checked occurrence constraints for one part of the document, further checks of this kind can be glossed over in the report. This would then give more scope for expanding on key tests, such as those asserting the correct answer type was given for particular trials. You included some good reflection on data structures, choice of simple types, and the limitations of XSD 1.0.

In task 6 you have a reasonable discussion of the domain implications for evaluating trials. However, there are no samples of evaluation criteria encoded, or even details of exactly what criteria are permitted in your design, which makes assessing it difficult. The structure in the appendix looks more like sample examinations, especially as they have dates! If these are just thresholds, better names for the elements would be sensible, even if this reduces schema reuse opportunities – clarity being better than convenience. How do you specify which things should be increasing or decreasing?

More is included in the task 7 section, and I like the sample output there. It would have benefited from some high level discussion alongside, rather than figuring out the details from the technical discussion later. The comparison of XSLT 1 & 2 features is a nice touch, and demonstrates insight. Some of the terminology used initially is suspicious (e.g. referring to a variable switching) but from a later comment it is clear that you do understand that variable values cannot change; this understanding should have been demonstrated throughout. The narrative flow of the final section suffers a little, but it becomes comprehensible once it has been worked through. At first I

did not follow what the terms ‘initial’, ‘final’ and ‘evaluation’ were referring to. A high-level overview of what your solution is trying to accomplish, as indicated above, would have helped. Your function names were also overly brief, which didn’t help with understanding their purpose. I was impressed at the inclusion of simple units conversion, which went beyond what was expected from submissions. Converting qualitative measurements to a scale value using the schema was another nice touch. Recursion was used well to evaluate your criteria. You could make the `steadily` function tidier by creating variables holding the parameters to use in the recursive call, and hence having just a single call-template; however this is a relatively minor point.