

Schema Designing to Index Some Available Materials Related to Software Certification

Ishwaree Argade
MEng(Computer Science)
Department of Computing and Software
McMaster University

April 8, 2014

Abstract

Software Certification is the process of certifying the systems scoped to their software part. The report emphasizes on the task of indexing some available material around the various areas of software certification and thus explains the idea of software repository. However, it revolves around Challenge Problems, Course Modules and Certified Software Libraries and Tools which are the three areas related to Software Certification and also are the three components of the repository. The scope of the report is to design and implement schemas using XSDs to index some important information for the above three areas. The report starts with the introduction to the software certification and some approaches followed to perform software certification. It then describes the idea of software repository and the need for schema design to store the data in the repository. The schemas are implemented using XSDs. Thus, the latter chapters explain the process of schema design and present implementations followed by the corresponding XSLs for challenge problems, course modules, libraries and tools respectively. Further part provides an overview of testing process and a representative case study for challenge problem category.

Contents

1	Introduction	3
2	Overview of Software Repository	5
3	Challenge Problems	7
3.1	Schema Design	7
3.2	Implementation	10
3.2.1	Common Schema	11
3.2.2	Schema for Execution Environment	14
3.2.3	Challenge Problem Schema	17
3.3	Viewing Data in Browser	23
4	Course Modules	28
4.1	Schema Design	28
4.2	Implementation	30
4.3	Viewing Data in Browser	38
5	Libraries and Tools	41
5.1	Schema Design	41
5.1.1	Attributes Listing: Libraries	41
5.1.2	Attributes Listing: Tools	43
5.2	Implementation	45
5.2.1	Common Schema for Libraries and Tools	45
5.2.2	Schema for Libraries	47
5.2.3	Schema for Tools	49
5.3	View Data in Browser	53
5.3.1	XSL for Libraries	53
5.3.2	XSL for Tools	56

6	Testing	59
6.1	Test Cases	59
6.2	Case Study	61
7	Conclusion	71

Chapter 1

Introduction

Software Certification deals with the process of certifying a system containing some sort of software inside it but, restricting the certification process to the software aspect only [Ben11]. The certification process ensures the reliability and safety of the software system to be certified listing all the information necessary for its assessment. It encompasses wide range of formal, semi-formal and informal assurance techniques which includes even formal verification of safety policies, system simulation, testing and code reviews [DF05]. Thus, the certificates can have various types and certification process follow various mechanisms.

Most popular approach for software certification is process based certification of systems. The process through which a software system is developed is evaluated rather than evaluating the final product. As many software certifiers find the evaluation of software process easier than evaluation of product itself, process based certification is widely used [AW10]. One reason for this is, it is not possible to test the final product entirely even with the help of huge number of test cases. Hence, the focus is given on certain supportive evidences which would guarantee the quality of the software systems. Secondly, it is difficult to determine the metrics/attributes essential in assessing the final software product, more emphasis is given on the software process instead [AW10]. Some examples of this approach like ISO 9000 and CMMI certify that the proper engineering methods and processes are followed to manufacture the product [Voa14].

Though process based certification is a popular approach, it doesn't guarantee the reliability of the software as it focuses only on the process and not on the individual product. It certifies overall products and not the specific product. Thus, another approach called product based certification is put

forward. A detailed analysis of this aspect of software certification is found out in the paper by Wassyng, Maibaum, and Lawford [AW10]. According to them, the goal of the certification should be to ensure that a product satisfies certain characteristics by assessing some measurable attributes of the product. This approach to the software certification believes that there should be a mandated software development process which would guarantee the quality of development process of the product and then the product can be evaluated without consideration of the actual process followed to develop the specific product [AW10].

Another certification method based on product based approach is proposed by Voas [Voa14]. According to him, by hiring a third party to issue software certification based on end users' feedback provides more unbiased and reliable software certification. Using this concept, he proposes a certification process involving automated methods to assess the behaviour of the software and to avoid the issue of miscertification [Voa14].

Software development nowadays widely follows reusability of components. Reuse of components is an important factor to reduce cost of software development. Thus, the reliability of the component to be reused has to be evaluated. One method to determine reliability of software which builds the structural model and usage profile of software components and then evaluates it against a set of test cases is given by Wohlin and Runeson [CW94] and is applicable to both component as well as system certification.

A software certification management system is used for management of certification [DF05]. It stores the information about different systems and varieties of certificates along with the entire certification of history of the specific system. One of the challenges related to software certification is storing and providing the useful information. Hence, the goal of this report is to create a repository to store some material in various areas related to Software Certification.

This report focuses on three areas related to Software Certification namely Challenge Problems, Course Modules and Certified Software Libraries and verification tools in order to index some available material in the respective areas. The latter chapters in the report would introduce the idea of software repository meant to store available materials, schema designing and testing processes for Challenge problems, Course Modules and Certified Software Libraries finally concluding with future scope.

Chapter 2

Overview of Software Repository

As mentioned earlier, the main idea behind this report is to create a software repository which stores material from various areas around software certification in a schematic format. The prime components of the repository are Challenge Problems, Course Modules, Creation and Maintenance of Libraries, Body of Knowledge and Certified Components which include definitions and managed libraries. The main structure of the repository is shown in 2.1.

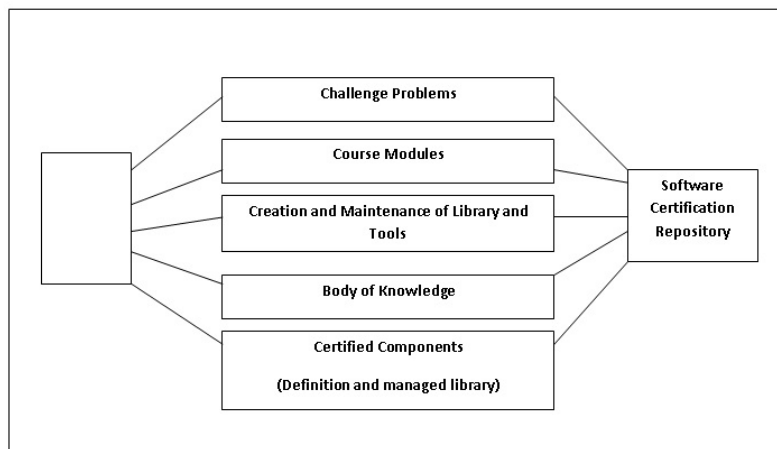


Figure 2.1: Design of Software Repository

The repository would have a schema design for each of these components. The schema would be designed as XSD which would cover all the required parameters to store the required data for that component. The actual data is then stored in XML format after validating it against the corresponding XSD schema. Finally, XSLs designed according to respective schemas of the components of repository, are used to view XMLs in the browser.

This report considers first three components of this repository viz. Challenge Problems, Course Modules and Creation and Maintenance of Library and Tools. The Challenge Problem part of the repository is meant to store some challenge problems in the area of software certification. XMLs added to the repository would contain all the information regarding a particular challenge and would be created according to a general schema designed for challenge problems. Course modules intend to have all the information about the courses involving topics in software certification. This information is again in XML format following schema designed for Course Module component of Software Repository. The last component manages information about libraries and verification tools. Libraries comprises of two types. It can either be a library which is a part of a verification tool or it can be a library which is verified by a verification tool. XMLs representing libraries and tools are validated against two separate schemas for libraries and tools respectively.

The subsequent chapters of the report documents the entire process of schema designing, XML and XSL creation and provide schemas, XSLs and some sample XMLs used for testing the corresponding schemas for above three parts of software repository.

Chapter 3

Challenge Problems

Challenge problems are sets of prototypes of problems in software certification area. The software repository intends to store all the available and relevant challenge problems. This includes both solved and unsolved challenge problems meaning that the solution is also saved if it is available. Challenge problems can be part of various conferences held for the software community. For example, there are challenges called SAT challenge, CADE ATP challenge, Pacemaker challenge, SMT COMP challenge, SV-COMP challenge [SAT,CADb,Pac07,SMT,SV13] etc. have been offered as a part of various conferences and workshops. Each challenge has different dimensions.

The requirements and specifications depend on the actual challenge problem and their organization meaning the committee who is putting forward this challenge. As the purpose of the repository is to collect all the challenge problems, a general schema which would be able to catch all the information of diverse challenge problems is needed. The schema provides a schematic structure to the repository in order to store challenge problems.

3.1 Schema Design

The first step in schema designing process is to analyze various challenge problems and try to find out some attributes which are common in all the challenges. The collective attributes actually are various the parameters in different challenges which make it possible to preserve their information in a structured manner.

As specified earlier, the schema is designed using XSD. Thus, all the information is tracked in the form of tags using XMLs. The XMLs of various challenges are then validated against the same designed schema. The schema

portrays the general structure for all the challenges still allowing users to embed challenge specific information in the XMLs. The Listing 3.1 shows all the attributes derived after analyzing some challenge problems.

Listing 3.1: Attributes for Challenge Problem Schema

- 1 A. Challenge Name
- 2 B. Area
- 3 C. Challenge Description
 - 4 a. Description
 - 5 b. Challenge Date
 - 6 1. To
 - 7 2. From
 - 8 c. Challenge Location
 - 9 d. Associated Conference
 - 10 e. Part of Series
- 11 D. Rules
 - 12 a. Rule
 - 13 1. Rule Category
 - 14 2. Description
 - 15 3. Input Requirements
 - 16 i. Input Requirement
 - 17 4. Output Requirements
 - 18 i. Output Requirement
 - 19 5. Links
- 20 E. Supporting Documents
 - 21 a. Document Name
 - 22 b. Description
 - 23 c. Link
- 24 F. Year
- 25 G. Assessment Description
 - 26 a. Description
 - 27 b. Link
 - 28 c. Jury
 - 29 1. Jury Name
 - 30 2. Description
 - 31 3. Phone
 - 32 4. Email
 - 33 5. Web page Link
 - 34 d. Score Details
 - 35 i. Score
 - 36 1. Points
 - 37 2. Description
- 38 H. Participants
 - 39 a. Participant
 - 40 b. Participant Description Link

- 41 I. Benchmarks
- 42 a. Benchmark
- 43 1. Categories
- 44 2. Description
- 45 3. Format
- 46 4. Timeline
- 47 5. Link
- 48 J. Expected Solution
- 49 a. Allowed Forms
- 50 b. Input Requirements
- 51 i. Input Requirement
- 52 c. Output Requirements
- 53 i. Output Requirement
- 54 d. Execution Environment
- 55 1. Description
- 56 2. Libraries
- 57 i. Library
- 58 A. Name
- 59 B. Link
- 60 3. Compilers
- 61 i. Compiler
- 62 A. Name
- 63 B. Description
- 64 4. Processors
- 65 i. Processor
- 66 A. Name
- 67 B. Memory
- 68 C. Description
- 69 5. OS Used
- 70 i. OS
- 71 A. Name
- 72 B. Version
- 73 e. Deadlines
- 74 1. Deadline
- 75 A. Name
- 76 B. Date
- 77 f. Allowed Submissions
- 78 K. Allowed Tools
- 79 a. Tool
- 80 1. Name
- 81 2. Description
- 82 3. Link
- 83 L. Required Contacts
- 84 a. Contact
- 85 1. Name

86	2.	Description
87	3.	Phone
88	4.	Email
89	5.	Web page Link
90	M.	Results
91	a.	Link
92	N.	Changes from Previous Challenges
93	a.	Change
94	1.	Description
95	2.	Link
96	O.	Solutions
97	a.	Format
98	b.	Description
99	c.	Input Requirements
100	d.	Output Requirements
101	e.	Execution Environment

Attributes' names indicate the purpose of the attribute. Thus, it covers all the required parameters regarding the challenge such as its name, area, its description, associated conference, rules to solve the challenge, documentation, tools allowed, expected solution format, available solutions, assessment details, deadlines and contacts. Some challenge problems also have a set of benchmark problems. Benchmark problems are smaller sets of problems related to main challenge problem. Their formats, rules and solutions vary from challenge to challenge. Some attributes are used to tag this important information about benchmarks as well. Therefore, this listing of attributes provides a hierarchical structure to tag information about various challenge problems and thus makes it convenient to implement this schema design using XSD.

3.2 Implementation

The above schema design is implemented using XSD. The whole schema is broken down into three schemas. The first one has all the elements common to the areas covered in this report meaning that it contains all the attributes common in the areas of challenge problems, course modules, libraries and tools. The second schema contains the elements required to tag detail information regarding execution environment. The third schema is the actual main schema for challenge problem which includes the above two schemas. An online tool is used to validate XMLs against this schema [Bria].

3.2.1 Common Schema

As described earlier, this schema represents all the common attributes to the three areas covered in this report. This is done to avoid redundancy in implementing the entire schema. This schema would be included in all the main schemas for challenge problems, course modules, libraries and tools. The Listing 3.2 displays the code for the common schema.

Listing 3.2: Common Schema

```
1 <?xml version="1.0" encoding="iso-8859-1"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:xlink="http://www.w3.org/1999/xlink">
3 <xs:import namespace="http://www.w3.org/1999/xlink"
   schemaLocation="http://www.w3.org/1999/xlink.xsd"/>
4 <!--Schema Common to Challenge Problems, Course Modules,
   Certified Libraries and Tools-->
5
6 <xs:complexType name="contactType">
7 <xs:sequence>
8   <xs:element name="name" type="xs:string" minOccurs
      ="0"></xs:element>
9   <xs:element name="contactsDescription" type="xs:string"
      minOccurs="0"></xs:element>
10  <xs:element name="phone" type="xs:string" minOccurs="0"
      maxOccurs="unbounded"></xs:element>
11  <xs:element name="email" type="xs:string" minOccurs="0"
      maxOccurs="unbounded"></xs:element>
12  <xs:element name="linkDescription" minOccurs="0"
      maxOccurs="unbounded">
13    <xs:complexType>
14      <xs:simpleContent>
15        <xs:extension base="xs:string">
16          <xs:anyAttribute namespace="http://www.w3.org
             /1999/xlink"/>
17        </xs:extension>
18      </xs:simpleContent>
19    </xs:complexType>
20  </xs:element>
21 </xs:sequence>
22 </xs:complexType>
23 <xs:complexType name="contactsType">
24 <xs:sequence>
25   <xs:element name="contact" type="contactType" maxOccurs
      ="unbounded"></xs:element>
26 </xs:sequence>
```

```

27 </xs:complexType>
28 <xs:complexType name="toolsType">
29     <xs:sequence>
30         <xs:element name="tool" maxOccurs="unbounded">
31             <xs:complexType>
32                 <xs:sequence>
33                     <xs:element name="toolName" type="xs:string"></
                        xs:element>
34                     <xs:element name="toolDescription" type="xs:
                        string" minOccurs="0"></xs:element>
35                     <xs:element name="toolLinkDescription"
                        minOccurs="0" maxOccurs="unbounded">
36                         <xs:complexType>
37                             <xs:simpleContent>
38                                 <xs:extension base="xs:string">
39                                     <xs:anyAttribute namespace="http://www.
                                        w3.org/1999/xlink"/>
40                                 </xs:extension>
41                             </xs:simpleContent>
42                         </xs:complexType>
43                     </xs:element>
44                 </xs:sequence>
45             </xs:complexType>
46         </xs:element>
47     </xs:sequence>
48 </xs:complexType>
49 <xs:complexType name="relatedLinksType">
50     <xs:sequence>
51         <xs:element name="relatedLinkDescription" minOccurs
            ="0" maxOccurs="unbounded">
52             <xs:complexType>
53                 <xs:simpleContent>
54                     <xs:extension base="xs:string">
55                         <xs:anyAttribute namespace="http://www.
                            w3.org/1999/xlink"/>
56                     </xs:extension>
57                 </xs:simpleContent>
58             </xs:complexType>
59         </xs:element>
60     </xs:sequence>
61 </xs:complexType>
62 <xs:complexType name="documentType">
63     <xs:sequence>
64         <xs:element name="documentName" type="xs:string"
            minOccurs="0"></xs:element>

```

```

65     <xs:element name="documentDescription" type="xs:
        string" minOccurs="0"></xs:element>
66     <xs:element name="documentLinkDescription" minOccurs
        ="0" maxOccurs="unbounded">
67         <xs:complexType>
68             <xs:simpleContent>
69                 <xs:extension base="xs:string">
70                     <xs:anyAttribute namespace="http://www.
                        w3.org/1999/xlink"/>
71                 </xs:extension>
72             </xs:simpleContent>
73         </xs:complexType>
74     </xs:element>
75 </xs:sequence>
76 </xs:complexType>
77 <xs:complexType name="supportingDocumentsType">
78     <xs:sequence>
79         <xs:element name="document" type="documentType"
            maxOccurs="unbounded"></xs:element>
80     </xs:sequence>
81 </xs:complexType>
82 <xs:complexType name="descriptionType">
83     <xs:sequence>
84         <xs:element name="name" type="xs:string" minOccurs
            ="0"></xs:element>
85         <xs:element name="description" type="xs:string"
            minOccurs="0"></xs:element>
86         <xs:element name="linkDescription" minOccurs="0"
            maxOccurs="unbounded">
87             <xs:complexType>
88                 <xs:simpleContent>
89                     <xs:extension base="xs:string">
90                         <xs:anyAttribute namespace="http://www.
                            w3.org/1999/xlink"/>
91                     </xs:extension>
92                 </xs:simpleContent>
93             </xs:complexType>
94         </xs:element>
95     </xs:sequence>
96 </xs:complexType>
97 <xs:complexType name="resultsType">
98     <xs:sequence>
99         <xs:element name="resultsLinkDescription" minOccurs
            ="0" maxOccurs="unbounded">
100     </xs:complexType>

```

```

101         <xs:simpleContent>
102             <xs:extension base="xs:string">
103                 <xs:anyAttribute namespace="http://www.
104                     w3.org/1999/xlink"/>
105             </xs:extension>
106         </xs:simpleContent>
107     </xs:complexType>
108 </xs:element>
109 </xs:sequence>
110 </xs:complexType>
111 </xs:schema>

```

3.2.2 Schema for Execution Environment

This schema represents particularly the information specific to execution environment for the challenge problems. It separates all the execution environment data from the other elements of challenge problems such as its description, assessment details, contacts etc. This schema stores data like expected solution format, libraries, compilers, processors and OS permissible to solve the challenge and basically all the elements needed to describe the expected solution as well as existing solutions. The Listing 3.3 shows the code for Execution Environment Schema.

Listing 3.3: Execution Environment Schema

```

1 <?xml version="1.0" encoding="iso-8859-1"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
3   xmlns:xlink="http://www.w3.org/1999/xlink">
4   <xs:import namespace="http://www.w3.org/1999/xlink"
5     schemaLocation="http://www.w3.org/1999/xlink.xsd"/>
6   <xs:complexType name="inputRequirementsType">
7     <xs:sequence>
8       <xs:element name="inputRequirement" type="xs:string"
9         maxOccurs="unbounded"></xs:element>
10    </xs:sequence>
11  </xs:complexType>
12  <xs:complexType name="outputRequirementsType">
13    <xs:sequence>
14      <xs:element name="outputRequirement" type="xs:string"
15        maxOccurs="unbounded"></xs:element>
16    </xs:sequence>
17  </xs:complexType>
18  <xs:complexType name="libraryType">

```



```

16     <xs:sequence>
17         <xs:element name="library" minOccurs="0" maxOccurs="
18             unbounded">
19             <xs:complexType>
20                 <xs:sequence>
21                     <xs:element name="libraryName" type="xs:string
22                         "></xs:element>
23                     <xs:element name="libraryLinkDescription"
24                         minOccurs="0">
25                         <xs:complexType>
26                             <xs:simpleContent>
27                                 <xs:extension base="xs:string">
28                                     <xs:anyAttribute namespace="http://www.
29                                         w3.org/1999/xlink"/>
30                                 </xs:extension>
31                             </xs:simpleContent>
32                         </xs:complexType>
33                     </xs:element>
34                 </xs:sequence>
35             </xs:complexType>
36         </xs:element>
37     </xs:sequence>
38 </xs:complexType>
39 <xs:complexType name="compilersType">
40     <xs:sequence>
41         <xs:element name="compiler" maxOccurs="unbounded">
42             <xs:complexType>
43                 <xs:sequence>
44                     <xs:element name="compilerName" type="xs:string
45                         "></xs:element>
46                     <xs:element name="compilerDescription" type="xs
47                         :string" minOccurs="0"></xs:element>
48                 </xs:sequence>
49             </xs:complexType>
50         </xs:element>
51     </xs:sequence>
52 </xs:complexType>
53 <xs:complexType name="processorsType">
    <xs:sequence>
        <xs:element name="processor" maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="processorName" type="xs:
                        string"></xs:element>
                    <xs:element name="processorMemory" type="xs:

```

```

54         string"></xs:element>
        <xs:element name="processorDescription" type="
            xs:string" minOccurs="0"></xs:element>
55     </xs:sequence>
56 </xs:complexType>
57 </xs:element>
58 </xs:sequence>
59 </xs:complexType>
60 <xs:complexType name="osType">
61     <xs:sequence>
62         <xs:element name="OS" maxOccurs="unbounded">
63             <xs:complexType>
64                 <xs:sequence>
65                     <xs:element name="osName" type="xs:string"></xs
                        :element>
66                     <xs:element name="osVersion" type="xs:string
                        "></xs:element>
67                 </xs:sequence>
68             </xs:complexType>
69         </xs:element>
70     </xs:sequence>
71 </xs:complexType>
72 <xs:complexType name="deadlineType">
73     <xs:sequence>
74         <xs:element name="deadlineName" type="xs:string"></xs
            :element>
75         <xs:element name="submissionDeadline" type="xs:
            dateTime"></xs:element>
76     </xs:sequence>
77 </xs:complexType>
78 <xs:complexType name="deadlinesType">
79     <xs:sequence>
80         <xs:element name="deadline" type="deadlineType"
            maxOccurs="unbounded"></xs:element>
81     </xs:sequence>
82 </xs:complexType>
83 <xs:complexType name="ExecutionEnvironmentType">
84     <xs:sequence>
85         <xs:element name="environmentDescription" type="xs:
            string" minOccurs="0"></xs:element>
86         <xs:element name="libraries" type="libraryType"
            minOccurs="0"></xs:element>
87         <xs:element name="compilers" type="compilersType"
            minOccurs="0"></xs:element>
88         <xs:element name="processors" type="processorsType"

```

```

      minOccurs="0"></xs:element>
89    <xs:element name="OSUsed" type="osType" minOccurs
      ="0"></xs:element>
90  </xs:sequence>
91 </xs:complexType>
92 <xs:complexType name="expectedSolutionType">
93   <xs:sequence>
94     <xs:element name="AllowedForms" type="xs:string"
      maxOccurs="unbounded" minOccurs="0"></xs:element>
95     <xs:element name="inputRequirements" type="
      inputRequirementsType" minOccurs="0"></xs:element>
96     <xs:element name="outputRequirements" type="
      outputRequirementsType" minOccurs="0"></xs:element
      >
97     <xs:element name="executionEnvironment" type="
      ExecutionEnvironmentType" minOccurs="0"></xs:
      element>
98     <xs:element name="deadlines" type="deadlinesType"
      minOccurs="0"></xs:element>
99     <xs:element name="allowedSubmissions" type="xs:string
      " minOccurs="0"></xs:element>
100   </xs:sequence>
101 </xs:complexType>
102 </xs:schema>

```

3.2.3 Challenge Problem Schema

This is the main schema for challenge problems. The structure of the schema is according to the schema design described earlier. It includes the two supporting schemas viz. Common Schema and Execution Environment Schema explained in the above sub sections. The Listing 3.4 displays the code for challenge problem schema.

Listing 3.4: Challenge Problem Schema

```

1 <?xml version="1.0" encoding="iso-8859-1"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xlink="http://www.w3.org/1999/xlink">
3 <xs:include schemaLocation="
  XmlSchema_Execution_Environment_Challenge_Problem_Version_6
  .xsd"/>
4 <!--Importing Common XMLSchema-->
5 <xs:include schemaLocation="D:\MEng_Docs\MEng_Docs\Project\
  Common_Schema\XmlSchema_Common.xsd"/>

```

```

6 <xs:import namespace="http://www.w3.org/1999/xlink"
  schemaLocation="http://www.w3.org/1999/xlink.xsd"/>
7
8 <xs:complexType name="participantsType">
9   <xs:sequence>
10    <xs:element name="participant" maxOccurs="unbounded"
      type="xs:string"></xs:element>
11    <xs:element name="participantDescriptionLink"
      minOccurs="0" maxOccurs="unbounded">
12      <xs:complexType>
13        <xs:simpleContent>
14          <xs:extension base="xs:string">
15            <xs:anyAttribute namespace="http://www.w3.org
              /1999/xlink"/>
16          </xs:extension>
17        </xs:simpleContent>
18      </xs:complexType>
19    </xs:element>
20  </xs:sequence>
21 </xs:complexType>
22 <xs:complexType name="challengeDateType">
23   <xs:sequence>
24    <xs:element name="from" type="xs:date"></xs:element>
25    <xs:element name="to" type="xs:date" minOccurs="0"></
      xs:element>
26  </xs:sequence>
27 </xs:complexType>
28 <!--Declaring ComplexTypes and Groups-->
29 <xs:complexType name="benchmarkType">
30   <xs:sequence>
31    <xs:element name="categories" type="xs:string"
      minOccurs="0"></xs:element>
32    <xs:element name="description" type="xs:string"
      minOccurs="0"></xs:element>
33    <xs:element name="format" type="xs:string" minOccurs
      ="0" maxOccurs="unbounded"></xs:element>
34    <xs:element name="timeline" type="xs:duration"
      minOccurs="0"></xs:element>
35    <xs:element name="benchmarkLinkDescription" minOccurs
      ="0" maxOccurs="unbounded">
36      <xs:complexType>
37        <xs:simpleContent>
38          <xs:extension base="xs:string">
39            <xs:anyAttribute namespace="http://www.
              w3.org/1999/xlink"/>

```

```

40         </xs:extension>
41     </xs:simpleContent>
42 </xs:complexType>
43 </xs:element>
44 </xs:sequence>
45 </xs:complexType>
46 <xs:complexType name="ruleDetailsType">
47     <xs:sequence>
48         <xs:element name="ruleCategory" type="xs:string"
49             minOccurs="0"></xs:element>
50         <xs:element name="ruleDescription" type="xs:string"
51             ></xs:element>
52         <xs:element name="inputRequirements" type="
53             inputRequirementsType" minOccurs="0"></xs:element>
54         <xs:element name="outputRequirements" type="
55             outputRequirementsType" minOccurs="0"></xs:element
56             >
57         <xs:element name="ruleLinkDescription" minOccurs="0"
58             maxOccurs="unbounded">
59             <xs:complexType>
60                 <xs:simpleContent>
61                     <xs:extension base="xs:string">
62                         <xs:anyAttribute namespace="http://www.
63                             w3.org/1999/xlink"/>
64                     </xs:extension>
65                 </xs:simpleContent>
66             </xs:complexType>
67         </xs:element>
68     </xs:sequence>
69 </xs:complexType>
70 <xs:complexType name="ruleType">
71     <xs:sequence>
72         <xs:element name="rule" maxOccurs="unbounded" type="
73             ruleDetailsType"></xs:element>
74     </xs:sequence>
75 </xs:complexType>
76 <xs:complexType name="solutionformatType">
77     <xs:sequence>
78         <xs:element name="solutionForm" type="xs:string"></
79             xs:element>
80         <xs:element name="solutionLinkDescription" minOccurs
81             ="0" maxOccurs="unbounded">
82             <xs:complexType>
83                 <xs:simpleContent>
84                     <xs:extension base="xs:string">

```

```

75         <xs:anyAttribute namespace="http://www.
           w3.org/1999/xlink"/>
76     </xs:extension>
77 </xs:simpleContent>
78 </xs:complexType>
79 </xs:element>
80 </xs:sequence>
81 </xs:complexType>
82 <xs:group name="groupBenchmark">
83     <xs:sequence>
84         <xs:element name="benchmark" type="benchmarkType"
           minOccurs="0" maxOccurs="unbounded"></xs:element>
85     </xs:sequence>
86 </xs:group>
87 <xs:complexType name="solutionType">
88     <xs:sequence>
89         <xs:element name="solutionFormat" type="
           solutionformatType" minOccurs="0"></xs:element>
90         <xs:element name="solutionDescription" type="xs:
           string" minOccurs="0"></xs:element>
91         <xs:element name="inputRequirements" type="
           inputRequirementsType" minOccurs="0"></xs:element>
92         <xs:element name="outputRequirements" type="
           outputRequirementsType" minOccurs="0"></xs:element
           >
93         <xs:element name="executionEnvironment" type="
           ExecutionEnvironmentType" minOccurs="0"></xs:
           element>
94     </xs:sequence>
95 </xs:complexType>
96 <xs:group name="groupSolution">
97     <xs:sequence>
98         <xs:element name="solution" type="solutionType"
           minOccurs="0" maxOccurs="unbounded"></xs:element>
99     </xs:sequence>
100 </xs:group>
101 <xs:complexType name="benchmarksType">
102     <xs:sequence>
103         <xs:group ref="groupBenchmark"></xs:group>
104     </xs:sequence>
105 </xs:complexType>
106 <xs:complexType name="solutionsType">
107     <xs:sequence>
108         <xs:group ref="groupSolution"></xs:group>
109     </xs:sequence>

```

```

110 </xs:complexType>
111 <xs:complexType name="challengeDescriptionType">
112   <xs:sequence>
113     <xs:element name="description" type="xs:string"></xs:
      element>
114     <xs:element name="challengeDate" type="
      challengeDateType" minOccurs="0"></xs:element>
115     <xs:element name="challengeLocation" type="xs:string"
      minOccurs="0"></xs:element>
116     <xs:element name="associatedConference" type="xs:
      string" minOccurs="0"></xs:element>
117     <xs:element name="partOfSeries" type="xs:string"
      minOccurs="0"></xs:element>
118   </xs:sequence>
119 </xs:complexType>
120 <xs:complexType name="scoreDetailType">
121   <xs:sequence>
122     <xs:element name="points" type="xs:string" minOccurs
      ="0"></xs:element>
123     <xs:element name="description" type="xs:string"></xs:
      element>
124   </xs:sequence>
125 </xs:complexType>
126 <xs:complexType name="scoreDetailsType">
127   <xs:sequence>
128     <xs:element name="score" type="scoreDetailType"
      maxOccurs="unbounded"></xs:element>
129   </xs:sequence>
130 </xs:complexType>
131 <xs:complexType name="assesmentDescriptionType">
132   <xs:sequence>
133     <xs:element name="description" type="xs:string"
      minOccurs="0"></xs:element>
134     <xs:element name="assessmentLinkDescription" minOccurs
      ="0" maxOccurs="unbounded">
135       <xs:complexType>
136         <xs:simpleContent>
137           <xs:extension base="xs:string">
138             <xs:anyAttribute namespace="http://www.
              w3.org/1999/xlink"/>
139           </xs:extension>
140         </xs:simpleContent>
141       </xs:complexType>
142     </xs:element>
143     <xs:element name="jury" type="contactsType" minOccurs

```

```

    ="0" maxOccurs="unbounded"></xs:element>
144 <xs:element name="scoreDetails" type="
    scoreDetailsType" minOccurs="0"></xs:element>
145 </xs:sequence>
146 </xs:complexType>
147 <xs:complexType name="changeType">
148 <xs:sequence>
149 <xs:element name="changeDescription" type="xs:string
    "></xs:element>
150 <xs:element name="changeDescriptionLink" minOccurs
    ="0" maxOccurs="unbounded">
151 <xs:complexType>
152 <xs:simpleContent>
153 <xs:extension base="xs:string">
154 <xs:anyAttribute namespace="http://www.w3.org
    /1999/xlink"/>
155 </xs:extension>
156 </xs:simpleContent>
157 </xs:complexType>
158 </xs:element>
159 </xs:sequence>
160 </xs:complexType>
161 <xs:complexType name="changesType">
162 <xs:sequence>
163 <xs:element name="change" type="changeType" maxOccurs
    ="unbounded"></xs:element>
164 </xs:sequence>
165 </xs:complexType>
166 <!-- Main Schema -->
167 <xs:element name="challenges">
168 <xs:complexType>
169 <xs:sequence>
170 <xs:element name="challenge" maxOccurs="unbounded">
171 <xs:complexType>
172 <xs:sequence>
173 <xs:element name="challengeName" type="xs:
    string" />
174 <xs:element name="area" type="xs:string" />
175 <xs:element name="challengeDescription" type
    ="challengeDescriptionType"></xs:element>
176 <xs:element name="rules" type="ruleType"
    minOccurs="0"></xs:element>
177 <xs:element name="supportingDocuments" type="
    supportingDocumentsType" minOccurs="0"></
    xs:element>

```



```

178         <xs:element name="year" type="xs:gYear"
           minOccurs="0"></xs:element>
179         <xs:element name="assessmentDescription" type=
           ="assesmentDescriptionType" minOccurs
           ="0"></xs:element>
180         <xs:element name="participants" type="
           participantsType" minOccurs="0"></xs:
           element>
181         <xs:element name="benchmarks" type="
           benchmarksType" minOccurs="0"></xs:element
           >
182         <xs:element name="expectedSolution" type="
           expectedSolutionType"></xs:element>
183         <xs:element name="allowedTools" type="
           toolsType" minOccurs="0"></xs:element>
184         <xs:element name="solutions" type="
           solutionsType" minOccurs="0"></xs:element>
185         <xs:element name="contactDetails" type="
           contactsType" minOccurs="0"></xs:element>
186         <xs:element name="results" type="resultsType"
           minOccurs="0"></xs:element>
187         <xs:element name="changes" type="changesType"
           minOccurs="0"></xs:element>
188     </xs:sequence>
189 </xs:complexType>
190 </xs:element>
191 </xs:sequence>
192 </xs:complexType>
193 </xs:element>
194 </xs:schema>

```

3.3 Viewing Data in Browser

The data stored in the repository is in the form of XMLs which are validated against the above schema designed for Challenge Problems. Hence, some mechanism is needed to visualize those XMLs in the browser making them easy to interpret and read. This is done using XSLT. XSL is used to transform XML code into HTML code which is read by the browser and thus displays the contents of the XML file in the form of a html page.

The XSL for challenge problem reads the contents from the XML files according to the challenge problem schema and displays the information in proper format if it is present in the XML file. An online tool is used to

transform XML with XSL [Brib]. The tool accepts both the XML and XSL and then transforms XML code into HTML code understood by the browser. This html file is then viewed in the browser. Thus, XSL is used to display the data stored in the repository in readable format and hence is similar to the data retrieving process. The Listing 3.5 depicts some part of the code for XSL for challenge problems.

Listing 3.5: Sample of XSL for Challenge Problems

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org
   /1999/XSL/Transform"
3      xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-
      result-prefixes="msxsl"
4      xmlns:xlink="http://www.w3.org/1999/xlink"
5  >
6  <xsl:output method="xml" indent="yes"/>
7  <xsl:template match="/">
8  <html>
9  <style type="text/css">
10     .alignmentCenter{
11     text-align: center;
12     }
13     .setWidth{
14     width: 50%;}}
15 </style>
16 <body>
17 <xsl:for-each select="challenges/challenge">
18   <h1 class="alignmentCenter">
19     <xsl:value-of select="challengeName"/>
20   </h1>
21   <h2>
22     <xsl:if test="area">
23       <xsl:value-of select="area"/>
24     </xsl:if>
25     <h3>Rules</h3>
26     <xsl:for-each select="rules/rule">
27       <xsl:if test="ruleCategory">
28         <b>
29           <xsl:value-of select="ruleCategory"/>
30           <xsl:text>: </xsl:text> </b>
31         </xsl:if><br/>
32         <xsl:value-of select="ruleDescription"/>
33         <br/>
34         <xsl:if test="inputRequirements">
```

```

35 <b>Input Requirements</b>
36 <br/>
37 <xsl:for-each select="inputRequirements/inputRequirement
    ">
38 <li>
39 <xsl:value-of select="."/>
40 </li>
41 </xsl:for-each>
42 </xsl:if>
43 <xsl:if test="outputRequirements">
44 <b>Output Requirements</b>
45 <br/>
46 <xsl:for-each select="outputRequirements/
    outputRequirement">
47 <li>
48 <xsl:value-of select="."/>
49 </li>
50 </xsl:for-each>
51 </xsl:if>
52 <xsl:if test="ruleLinkDescription">
53 <b>Links</b>
54 <br/>
55 <xsl:for-each select="ruleLinkDescription">
56 <li>
57 <xsl:element name="a">
58 <xsl:attribute name="href">
59 <xsl:value-of select="./@xlink:href" />
60 </xsl:attribute>
61 <xsl:value-of select="."/>
62 </xsl:element>
63 </li>
64 </xsl:for-each>
65 <br/>
66 </xsl:if><br/>
67 </xsl:for-each>
68 <xsl:if test="supportingDocuments" >
69 <h3>Supporting Documents</h3>
70 <table border="1">
71 <tr bgcolor="green">
72 <th>Document Name</th>
73 <th>Description</th>
74 <th>Link</th>
75 </tr>
76 <xsl:for-each select="supportingDocuments/document">
77 <tr>

```

```

78 <td class="setWidth">
79   <xsl:choose>
80     <xsl:when test="documentName">
81       <xsl:value-of select="documentName"/>
82     </xsl:when>
83     <xsl:otherwise>
84       <xsl:text>—</xsl:text>
85     </xsl:otherwise>
86   </xsl:choose>
87 </td>
88 <td>
89   <xsl:choose>
90     <xsl:when test="documentDescription">
91       <xsl:value-of select="documentDescription"/>
92     </xsl:when>
93     <xsl:otherwise>
94       <xsl:text>—</xsl:text>
95     </xsl:otherwise>
96   </xsl:choose>
97 </td>
98 <td>
99   <xsl:choose>
100     <xsl:when test="documentLinkDescription">
101       <table>
102         <xsl:for-each select="documentLinkDescription">
103           <tr>
104             <td>
105               <xsl:element name="a">
106                 <xsl:attribute name="href">
107                   <xsl:value-of select="./@xlink:href" />
108                 </xsl:attribute>
109                 <xsl:value-of select="." />
110               </xsl:element>
111             </td>
112           </tr>
113         </xsl:for-each>
114       </table>
115     </xsl:when>
116     <xsl:otherwise><xsl:text>—</xsl:text>
117   </xsl:otherwise>
118 </xsl:choose>
119 </td>
120 </tr>
121 </xsl:for-each>
122 </table>

```

```
123     </xsl:if>
124 </xsl:if>
125 </xsl:for-each>
126 </body>
127 </html>
128 </xsl:template>
129 </xsl:stylesheet>
```

The testing of the above schemas and XSL along with a case study is described in detail later in the report.

Chapter 4

Course Modules

The software repository collects some course modules related to software certification area [Yor, Wat, MaC, Vic]. Thus, the main objective behind this schema designing process is to identify the essential elements in various courses offered in these areas. This provides a schematic structure in order to be able to tag all the information related to various courses. Some examples of course modules are courses related to safety critical systems, embedded systems and real time systems. Hence, relevant course modules are basically those which are related to some systems containing software configuration.

4.1 Schema Design

Similar to challenge problem schema design, the first step here is again to analyze various course modules and figure out common elements. The elements altogether represent all the information of the course modules. Note that here the main focus is to collect all time independent data. Therefore, the major goal of analysis of various challenges is to find out attributes such as learning outcomes, aim of the module, pre-requisites and contents rather than focussing on deadlines and timings.

This schema is also implemented using XSD. Thus, all the attributes found by analyzing various course modules are organized in a hierarchical structure making the translation to XSD easy. The Listing 4.1 shows all the attributes used to implement course module schema.

Listing 4.1: Attributes for Course Modules Schema

- 1 Modules
- 2 a. Module
- 3 1. School/University

- 4 2. Module Name
- 5 3. Module Code
- 6 4. Professors/Lecturers
- 7 5. Status (Core/Optional)
- 8 6. Required For (Full-time/part-time)
- 9 7. Allowed Tracks
- 10 8. Number of credits
- 11 9. Teaching Term
- 12 10. Pre-requisites
- 13 11. Description
- 14 12. Aims
 - 15 a. Aim
 - 16 i. Description
 - 17 ii. Link
- 18 13. Learning Outcomes
- 19 14. Rules
- 20 15. Workload
 - 21 i. Total lecture hours
 - 22 ii. Each lecture time
 - 23 iii. Total Private study time
 - 24 iv. Assessment Time
- 25 16. Feedback
 - 26 i. Description
 - 27 ii. Link
- 28 17. Content
 - 29 i. Topics covered
 - 30 a. Description
 - 31 b. Link
 - 32 ii. Teaching Material
 - 33 a. Name
 - 34 b. description
 - 35 c. Type (Slides/Case Studies/exercise)
 - 36 d. Link
 - 37 iii. Books
 - 38 a. Name
 - 39 b. Type (Required/Recommended)
 - 40 c. Author
 - 41 d. Title
 - 42 e. Publisher
 - 43 f. Year
 - 44 iv. Deadlines
 - 45 v. Submissions
 - 46 a. Description
 - 47 b. Link
 - 48 c. Start Date

```

49         d. End Date
50     vii. Assessment
51         a. Description
52         b. Assignments
53             1. Assignment
54                 a. Description
55                 b. weight
56                 c. Link
57         c. Exams
58             2. Exam
59                 d. Description
60                 e. weight
61                 f. Link
62         d. Start Date
63         e. End Date
64     viii. Allowed Tools
65         a. Tool Name
66         b. Description
67         c. Link
68     ix. Location
69     x. Results

```

4.2 Implementation

The schema implementation follows the schema discussed in above section. It also includes the common schema implemented given in Listing 3.2. This schema again covers the general structure to tag all the attributes of course modules in diverse areas while allowing to save some module specific data as well. The schema is then tested with some sample course module XMLs by using an online XML schema validation tool [Bria]. The testing details are discussed later in the report. The Listing 4.2 shows the code for course module schema.

Listing 4.2: Course Module Schema

```

1 <?xml version="1.0" encoding="iso-8859-1"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:xlink="http://www.w3.org/1999/xlink">
3 <xs:import namespace="http://www.w3.org/1999/xlink"
   schemaLocation="http://www.w3.org/1999/xlink.xsd"/>
4 <!--Importing Common XMLSchema-->
5 <xs:include schemaLocation="D:\MEng.Docs\MEng.Docs\Project\
   Common.Schema\XmlSchema.Common.xsd"/>

```



```

6
7 <xs:complexType name="allowedTracksType">
8   <xs:sequence>
9     <xs:element name="track" type="xs:string" maxOccurs="
      unbounded"></xs:element>
10   </xs:sequence>
11 </xs:complexType>
12 <xs:complexType name="termsType">
13   <xs:sequence>
14     <xs:element name="term" type="xs:string" maxOccurs="
      unbounded"></xs:element>
15   </xs:sequence>
16 </xs:complexType>
17 <xs:complexType name="preRequisiteType">
18   <xs:sequence>
19     <xs:element name="description" type="xs:string"></xs:
      element>
20     <xs:element name="preRequisiteLinkDescription"
      minOccurs="0" maxOccurs="unbounded">
21       <xs:complexType>
22         <xs:simpleContent>
23           <xs:extension base="xs:string">
24             <xs:anyAttribute namespace="http://www.w3.org
              /1999/xlink"/>
25           </xs:extension>
26         </xs:simpleContent>
27       </xs:complexType>
28     </xs:element>
29   </xs:sequence>
30 </xs:complexType>
31 <xs:complexType name="preRequisitesType">
32   <xs:sequence>
33     <xs:element name="preRequisite" type="preRequisiteType"
      maxOccurs="unbounded"></xs:element>
34   </xs:sequence>
35 </xs:complexType>
36 <xs:complexType name="aimType">
37   <xs:sequence>
38     <xs:element name="description" type="xs:string"></xs:
      element>
39     <xs:element name="aimLinkDescription" minOccurs="0"
      maxOccurs="unbounded">
40       <xs:complexType>
41         <xs:simpleContent>
42           <xs:extension base="xs:string">

```

```

43         <xs:anyAttribute namespace="http://www.w3.org
           /1999/xlink"/>
44     </xs:extension>
45 </xs:simpleContent>
46 </xs:complexType>
47 </xs:element>
48 </xs:sequence>
49 </xs:complexType>
50 <xs:complexType name="aimsType">
51     <xs:sequence>
52         <xs:element name="aim" type="aimType" maxOccurs="
           unbounded"></xs:element>
53     </xs:sequence>
54 </xs:complexType>
55 <xs:complexType name="learningOutcomesType">
56     <xs:sequence>
57         <xs:element name="learningOutcome" type="xs:string"
           maxOccurs="unbounded"></xs:element>
58     </xs:sequence>
59 </xs:complexType>
60 <xs:complexType name="ruleDetailsType">
61     <xs:sequence>
62         <xs:element name="ruleCategory" type="xs:string"
           minOccurs="0"></xs:element>
63         <xs:element name="ruleDescription" type="xs:string"
           "></xs:element>
64         <xs:element name="ruleLinkDescription" minOccurs="0"
           maxOccurs="unbounded">
65             <xs:complexType>
66                 <xs:simpleContent>
67                     <xs:extension base="xs:string">
68                         <xs:anyAttribute namespace="http://www.
                           w3.org/1999/xlink"/>
69                     </xs:extension>
70                 </xs:simpleContent>
71             </xs:complexType>
72         </xs:element>
73     </xs:sequence>
74 </xs:complexType>
75 <xs:complexType name="ruleType">
76     <xs:sequence>
77         <xs:element name="rule" maxOccurs="unbounded" type="
           ruleDetailsType"></xs:element>
78     </xs:sequence>
79 </xs:complexType>

```

```

80 <xs:complexType name="workloadType">
81   <xs:sequence>
82     <xs:element name="totalLectureHours" type="xs:string"
      minOccurs="0"></xs:element>
83     <xs:element name="eachLectureDuration" type="xs:string"
      minOccurs="0"></xs:element>
84     <xs:element name="totalPracticalHours" type="xs:string"
      minOccurs="0"></xs:element>
85     <xs:element name="totalPrivateStudyTime" type="xs:
      string" minOccurs="0"></xs:element>
86     <xs:element name="assessmentTime" type="xs:string"
      minOccurs="0"></xs:element>
87     <xs:element name="description" type="xs:string"
      minOccurs="0"></xs:element>
88   </xs:sequence>
89 </xs:complexType>
90 <xs:complexType name="feedbackType">
91   <xs:sequence>
92     <xs:element name="feedbackDescription" type="xs:string"
      "></xs:element>
93     <xs:element name="feedbackLinkDescription" minOccurs
      ="0" maxOccurs="unbounded">
94       <xs:complexType>
95         <xs:simpleContent>
96           <xs:extension base="xs:string">
97             <xs:anyAttribute namespace="http://www.w3.org
              /1999/xlink"/>
98           </xs:extension>
99         </xs:simpleContent>
100       </xs:complexType>
101     </xs:element>
102   </xs:sequence>
103 </xs:complexType>
104 <xs:complexType name="topicCoveredType">
105   <xs:sequence>
106     <xs:element name="description" type="xs:string"></xs:
      element>
107     <xs:element name="LinkDescription" minOccurs="0"
      maxOccurs="unbounded">
108       <xs:complexType>
109         <xs:simpleContent>
110           <xs:extension base="xs:string">
111             <xs:anyAttribute namespace="http://www.w3.org
              /1999/xlink"/>
112           </xs:extension>

```

```

113         </xs:simpleContent>
114     </xs:complexType>
115 </xs:element>
116 </xs:sequence>
117 </xs:complexType>
118 <xs:complexType name="topicsCoveredType">
119     <xs:sequence>
120         <xs:element name="topic" type="topicCoveredType"
121             maxOccurs="unbounded"></xs:element>
122     </xs:sequence>
123 </xs:complexType>
124 <xs:complexType name="teachingMaterialType">
125     <xs:sequence>
126         <xs:element name="materialName" type="xs:string"></xs:
127             element>
128         <xs:element name="materialType" type="xs:string"
129             minOccurs="0"></xs:element>
130         <xs:element name="materialDescription" type="xs:string"
131             minOccurs="0"></xs:element>
132         <xs:element name="materialLinkDescription" minOccurs
133             ="0" maxOccurs="unbounded">
134             <xs:complexType>
135                 <xs:simpleContent>
136                     <xs:extension base="xs:string">
137                         <xs:anyAttribute namespace="http://www.w3.org
138                             /1999/xlink"/>
139                     </xs:extension>
140                 </xs:simpleContent>
141             </xs:complexType>
142         </xs:element>
143     </xs:sequence>
144 </xs:complexType>
145 <xs:complexType name="teachingMaterialsType">
146     <xs:sequence>
147         <xs:element name="material" type="teachingMaterialType"
148             maxOccurs="unbounded"></xs:element>
149     </xs:sequence>
150 </xs:complexType>
151 <xs:complexType name="bookType">
152     <xs:sequence>
153         <xs:element name="bookName" type="xs:string"></xs:
154             element>
155         <xs:element name="bookType" type="xs:string" minOccurs

```

```

    ="0"></xs:element>
150 <xs:element name="bookDescription" type="xs:string"
    minOccurs="0"></xs:element>
151 <xs:element name="bookAuthor" type="xs:string"
    minOccurs="0"></xs:element>
152 <xs:element name="bookPublisher" type="xs:string"
    minOccurs="0"></xs:element>
153 <xs:element name="bookYear" type="xs:string" minOccurs
    ="0"></xs:element>
154 <xs:element name="bookLinkDescription" minOccurs="0"
    maxOccurs="unbounded">
155 <xs:complexType>
156 <xs:simpleContent>
157 <xs:extension base="xs:string">
158 <xs:anyAttribute namespace="http://www.w3.org
    /1999/xlink"/>
159 </xs:extension>
160 </xs:simpleContent>
161 </xs:complexType>
162 </xs:element>
163 </xs:sequence>
164 </xs:complexType>
165 <xs:complexType name="booksType">
166 <xs:sequence>
167 <xs:element name="book" type="bookType" maxOccurs="
    unbounded"></xs:element>
168 </xs:sequence>
169 </xs:complexType>
170 <xs:complexType name="assignmentExamType">
171 <xs:sequence>
172 <xs:element name="description" type="xs:string"></xs:
    element>
173 <xs:element name="weight" type="xs:string"></xs:element
    >
174 <xs:element name="LinkDescription" minOccurs="0"
    maxOccurs="unbounded">
175 <xs:complexType>
176 <xs:simpleContent>
177 <xs:extension base="xs:string">
178 <xs:anyAttribute namespace="http://www.w3.org
    /1999/xlink"/>
179 </xs:extension>
180 </xs:simpleContent>
181 </xs:complexType>
182 </xs:element>

```

```

183     </xs:sequence>
184 </xs:complexType>
185 <xs:complexType name="assignmentsType">
186     <xs:sequence>
187         <xs:element name="assignment" type="
            assignmentExamType" maxOccurs="unbounded"></xs:
            element>
188     </xs:sequence>
189 </xs:complexType>
190 <xs:complexType name="examsType">
191     <xs:sequence>
192         <xs:element name="exam" type="assignmentExamType"
            maxOccurs="unbounded"></xs:element>
193     </xs:sequence>
194 </xs:complexType>
195 <xs:complexType name="assessmentType">
196     <xs:sequence>
197         <xs:element name="description" type="xs:string"
            minOccurs="0"></xs:element>
198         <xs:element name="assignments" type="assignmentsType"
            minOccurs="0"></xs:element>
199         <xs:element name="exams" type="examsType" minOccurs
            ="0"></xs:element>
200         <xs:element name="startDate" type="xs:dateTime"
            minOccurs="0"></xs:element>
201         <xs:element name="endDate" type="xs:dateTime" minOccurs
            ="0"></xs:element>
202     </xs:sequence>
203 </xs:complexType>
204 <xs:complexType name="contentType">
205     <xs:sequence>
206         <xs:element name="topicsCovered" type="
            topicsCoveredType" minOccurs="0"></xs:element>
207         <xs:element name="teachingMaterial" type="
            teachingMaterialsType" minOccurs="0"></xs:element>
208         <xs:element name="books" type="booksType" minOccurs
            ="0"></xs:element>
209         <xs:element name="startDate" type="xs:date" minOccurs
            ="0"></xs:element>
210         <xs:element name="endDate" type="xs:date" minOccurs
            ="0"></xs:element>
211         <xs:element name="assessment" type="assessmentType"
            minOccurs="0"></xs:element>
212         <xs:element name="allowedTools" type="toolsType"
            minOccurs="0"></xs:element>

```

```

213     </xs:sequence>
214 </xs:complexType>
215 <xs:element name="modules">
216     <xs:complexType>
217         <xs:sequence>
218             <xs:element name="module" maxOccurs="unbounded">
219                 <xs:complexType>
220                     <xs:sequence>
221                         <xs:element name="school" type="xs:string"
222                             minOccurs="0"></xs:element>
223                         <xs:element name="moduleName" type="xs:string"
224                             minOccurs="0"></xs:element>
225                         <xs:element name="moduleCode" type="xs:string"
226                             minOccurs="0"></xs:element>
227                         <xs:element name="year" type="xs:string"
228                             minOccurs="0"></xs:element>
229                         <xs:element name="professors" type="
230                             contactsType" minOccurs="0"></xs:element>
231                         <xs:element name="status" type="xs:string"
232                             minOccurs="0"></xs:element>
233                         <xs:element name="requiredFor" type="xs:string"
234                             minOccurs="0"></xs:element>
235                         <xs:element name="allowedTracks" type="
236                             allowedTracksType" minOccurs="0"></xs:
237                             element>
238                         <xs:element name="numberOfCredits" type="xs:
239                             string" minOccurs="0"></xs:element>
240                         <xs:element name="teachingTermsAllowed" type="
241                             termsType" minOccurs="0"></xs:element>
242                         <xs:element name="preRequisites" type="
243                             preRequisitesType" minOccurs="0"></xs:
244                             element>
245                         <xs:element name="courseDescription" type="
246                             descriptionType" minOccurs="0"></xs:element>
247                         <xs:element name="aims" type="aimsType"
248                             minOccurs="0"></xs:element>
249                         <xs:element name="learningOutcomes" type="
250                             learningOutcomesType" minOccurs="0"></xs:
251                             element>
252                         <xs:element name="rules" type="ruleType"
253                             minOccurs="0"></xs:element>
254                         <xs:element name="workload" type="workloadType"
255                             minOccurs="0"></xs:element>
256                         <xs:element name="feedback" type="feedbackType"
257                             minOccurs="0"></xs:element>

```

```

238         <xs:element name="content" type="contentType"
           minOccurs="0"></xs:element>
239         <xs:element name="location" type="xs:string"
           minOccurs="0"></xs:element>
240         <xs:element name="results" type="resultsType"
           minOccurs="0"></xs:element>
241         <xs:element name="requiredLinks" type="
           relatedLinksType" minOccurs="0"></xs:element
           >
242     </xs:sequence>
243 </xs:complexType>
244 </xs:element>
245 </xs:sequence>
246 </xs:complexType>
247 </xs:element>
248 </xs:schema>

```

4.3 Viewing Data in Browser

Similar to challenge problems XMLs, course module XMLs also need a XSL to transform XML code into HTML code. This makes it able to be viewed in the browser. The XSL is implemented according to the course module schema given above. It checks if the information is present in the XML or not and then converts it into appropriate HTML code. The HTML code displays information in an eye pleasing and organized format. An online tool is used to transform XMLs using this XSL [Brib]. As discussed earlier, the tool accepts both XML and XSL, then transforms the given XML according to the XSL finally providing an HTML code as an output. The Listing 4.3 shows some sample of XSL code for course modules.

Listing 4.3: Sample of XSL for Course Module

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org
  /1999/XSL/Transform"
3     xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-
      result-prefixes="msxsl"
4     xmlns:xlink="http://www.w3.org/1999/xlink"
5 >
6 <xsl:output method="xml" indent="yes"/>
7 <xsl:template match="/">
8 <html >
9 <style type="text/css">

```



```

10 .alignmentCenter{
11   text-align: center;
12 }
13 .setWidth{
14   width: 50%;
15 }
16 }
17 </style>
18 <body>
19 <xsl:for-each select="modules/module">
20   <h3 class="alignmentCenter">
21     <xsl:value-of select="moduleName"/>
22   </h3>
23   <h3 class="alignmentCenter">
24     <xsl:text>Module code: </xsl:text>
25     <xsl:value-of select="moduleCode"/>
26   </h3>
27   <xsl:if test="school">
28     <b><xsl:text>School:- </xsl:text></b>
29     <xsl:value-of select="school"/>
30   <br/>
31   </xsl:if>
32   <xsl:if test="preRequisites">
33     <h3><xsl:text>Pre-requisites:-</xsl:text></h3>
34     <xsl:for-each select="preRequisites/preRequisite">
35       <li><xsl:value-of select="description"/></li>
36       <xsl:if test="preRequisiteLinkDescription">
37         <xsl:for-each select="preRequisiteLinkDescription">
38           <xsl:element name="a">
39             <xsl:attribute name="href">
40               <xsl:value-of select="./@xlink:href" />
41             </xsl:attribute>
42             <xsl:value-of select="." />
43           </xsl:element>
44         <br/>
45       </xsl:for-each>
46     </xsl:if>
47   </xsl:for-each>
48   </xsl:if>
49   <xsl:if test="aims">
50     <h3><xsl:text>Aims-</xsl:text></h3>
51     <xsl:for-each select="aims/aim">
52       <li><xsl:value-of select="description"/></li>
53       <xsl:if test="aimLinkDescription">
54         <xsl:for-each select="aimLinkDescription">

```

```

55 <xsl:element name="a">
56 <xsl:attribute name="href">
57 <xsl:value-of select="./@xlink:href" />
58 </xsl:attribute>
59 <xsl:value-of select="." />
60 </xsl:element>
61 <br/>
62 </xsl:for-each>
63 </xsl:if>
64 </xsl:for-each>
65 </xsl:if>
66 <xsl:if test="content">
67 <h3>Contents:-</h3>
68 <xsl:if test="content/topicsCovered">
69 <b><xsl:text>Topics Covered</xsl:text></b><br/>
70 <xsl:for-each select="content/topicsCovered/topic">
71 <li><xsl:value-of select="description"/></li>
72 <xsl:if test="LinkDescription">
73 <xsl:for-each select="LinkDescription">
74 <xsl:element name="a">
75 <xsl:attribute name="href">
76 <xsl:value-of select="./@xlink:href" />
77 </xsl:attribute>
78 <xsl:value-of select="." />
79 </xsl:element>
80 <br/>
81 </xsl:for-each>
82 </xsl:if>
83 </xsl:for-each>
84 </xsl:if>
85 </xsl:for-each>
86 </body>
87 </html>
88 </xsl:template>
89 </xsl:stylesheet>

```

The testing of the XSL with some sample XMLs is done and is later discussed in the report.

Chapter 5

Libraries and Tools

The third component of the repository saves the data related to libraries and tools. The repository requires a schema to store some certified software libraries as well as some verification software tools used in the area of software certification. Libraries and tools are related to each other. Tools are referred as a certification mechanism for libraries. The report presents two separate schemas for libraries and tools respectively.

5.1 Schema Design

The design process again begins with some sample libraries and tools [Ale13, DFC, MCH, COQ, CADa]. They are analyzed and attributes for the schemas are figured out. The schema for the library should be able to tag all the data related to a library such as its current versions, contents, downloading, tools used for compilation or verification, examples, references and dependencies if any. Along with these elements, some additional elements like execution environment, functionalities, getting the tool are added to the tools' schema. However, even though the tools and the libraries schemas contain some common elements they have their own attributes too and needs to be stored separately. Thus, they have their own attribute listings and implementations.

5.1.1 Attributes Listing: Libraries

The analysis of some libraries such as Microchip certified libraries, a COQ libraries and a PRL math library, some attributes regarding the libraries are noted down in hierarchical format. The Listing 5.1 shows the elements for

the libraries' schema.

Listing 5.1: Attributes for Libraries' Schema

- 1 A. Libraries
- 2 i. Library
- 3 1. Name
- 4 2. Overview
- 5 a. Description
- 6 b. Link
- 7 3. Current Version
- 8 4. Versions (Current/Previous descriptions)
- 9 a. Version name
- 10 b. Status (current/previous)
- 11 c. Description
- 12 d. Link
- 13 5. Experimental Library Contents (Similar to Extensions)
- 14 a. content
- 15 i. Description
- 16 ii. Link
- 17 6. Contributors
- 18 a. Contributor
- 19 i. Name
- 20 ii. Contribution Description
- 21 iii. Email
- 22 iv. Phone
- 23 v. Web page link
- 24 7. Content Files
- 25 a. Content File
- 26 i. Name
- 27 ii. Description
- 28 iii. Link
- 29 8. Downloads
- 30 a. Description
- 31 b. Format (OS)
- 32 c. Size
- 33 d. Link
- 34 9. Required Tools for compilation
- 35 a. Tool
- 36 i. Tool name
- 37 ii. Description
- 38 iii. Link
- 39 10. Documentation
- 40 a. Document
- 41 i. Name
- 42 ii. Description

```

43     iii. Link
44 11. Dependency details
45     a. Description
46     b. Link
47 12. Examples
48     a. Example
49         i. Name
50         ii. Description
51         iii. Link
52 13. Related Links
53     a. Link
54 14. References
55     a. Description
56     b. Link

```

5.1.2 Attributes Listing: Tools

After analyzing some tools like COQ, CADP, NuSMV etc [COQ, CAdA]., some elements which constitute the tool's schema are found out. The Listing 5.2 shows the elements for the tools' schema.

Listing 5.2: Attributes for Tools' Schema

```

1  Tools
2  o Tool
3    1. Name
4    2. Overview
5      a. Description
6      b. Link
7    3. Current Status
8      a. Latest available version
9      b. Description
10     c. Link
11   4. Functionalities/Features
12     a. Function/Feature
13       i. Description
14       ii. Link
15   5. Intended Users
16     a. User
17   6. Extensions
18     a. Extension
19       i. Description
20       ii. Link
21   7. Contacts
22     a. Contact

```

- 23 i. Name
- 24 ii. Email
- 25 iii. Phone
- 26 iv. Web page link
- 27 8. Contents
- 28 a. Content
- 29 i. Name
- 30 ii. Description
- 31 iii. Link
- 32 9. How ToObtain
- 33 a. Source
- 34 i. Size
- 35 ii. Description
- 36 iii. Link
- 37 b. Binaries
- 38 i. Binary
- 39 I. Description
- 40 II. Format (OS)
- 41 III. Size
- 42 IV. Link
- 43 c. Others
- 44 i. Other forms
- 45 I. Description
- 46 II. Link
- 47 10. Execution Environment
- 48 a. Languages Used
- 49 i. Language
- 50 b. Input Requirements
- 51 i. Input Requirement
- 52 c. Compatible Compilers
- 53 i. Compiler
- 54 I. Name
- 55 II. Description
- 56 d. Compatible Processors
- 57 i. Processor
- 58 I. Name
- 59 II. Memory
- 60 III. Description
- 61 e. Compatible OS
- 62 i. OS
- 63 I. Name
- 64 II. Version
- 65 III. Distribution
- 66 11. Related Tools
- 67 a. Tool

```

68         i. Tool name
69         ii. Description
70         iii. Link
71     12. Documentation
72         a. Document
73             i. Name
74             ii. Description
75             iii. Link
76     13. Related Links
77         a. Link

```

5.2 Implementation

As mentioned earlier, libraries and tools are related to each other. So, naturally they share some common elements which can be implemented separately. Moreover, both of these schemas also include the common schema discussed earlier in Listing 3.2. The latter chapter talks about the testing of these schemas in detail.

5.2.1 Common Schema for Libraries and Tools

The schema mainly contains implementation of a few common elements such as version details, contents' details and extensions. The Listing 5.3 shows the common schema code for libraries and tools.

Listing 5.3: Common Schema for Libraries and Tools

```

1  <?xml version="1.0" encoding="iso-8859-1"?>
2  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
3    xmlns:xlink="http://www.w3.org/1999/xlink">
4    <xs:import namespace="http://www.w3.org/1999/xlink"
5      schemaLocation="http://www.w3.org/1999/xlink.xsd"/>
6    <!--Importing Common XMLSchema-->
7    <xs:include schemaLocation="D:\MEng.Docs\MEng.Docs\Project\
8      Common.Schema\XmlSchema.Common.xsd"/>
9
10   <xs:complexType name="versionType">
11     <xs:sequence>
12       <xs:element name="versionName" type="xs:string"
13         minOccurs="0"></xs:element>
14       <xs:element name="year" type="xs:string" minOccurs
15         ="0"></xs:element>

```

```

11     <xs:element name="description" type="xs:string"
        minOccurs="0"></xs:element>
12     <xs:element name="linkDescription" minOccurs="0"
        maxOccurs="unbounded">
13         <xs:complexType>
14             <xs:simpleContent>
15                 <xs:extension base="xs:string">
16                     <xs:anyAttribute namespace="http://www.
                        w3.org/1999/xlink"/>
17                 </xs:extension>
18             </xs:simpleContent>
19         </xs:complexType>
20     </xs:element>
21 </xs:sequence>
22 <xs:attribute name="status" use="optional">
23     <xs:simpleType>
24         <xs:restriction base="xs:string">
25             <xs:enumeration value="Current"/>
26             <xs:enumeration value="Previous"/>
27         </xs:restriction>
28     </xs:simpleType>
29 </xs:attribute>
30 </xs:complexType>
31 <xs:complexType name="versionsType">
32     <xs:sequence>
33         <xs:element name="version" type="versionType" maxOccurs
            ="unbounded"></xs:element>
34     </xs:sequence>
35 </xs:complexType>
36 <xs:complexType name="exContentsType">
37     <xs:sequence>
38         <xs:element name="contentDescription" type="xs:string"
            minOccurs="0"></xs:element>
39         <xs:element name="content" type="descriptionType"
            maxOccurs="unbounded"></xs:element>
40     </xs:sequence>
41 </xs:complexType>
42 <xs:complexType name="contentFilesType">
43     <xs:sequence>
44         <xs:element name="contentDescription" type="xs:string"
            minOccurs="0"></xs:element>
45         <xs:element name="contentFile" type="descriptionType"
            maxOccurs="unbounded"></xs:element>
46     </xs:sequence>
47 </xs:complexType>

```


48 </xs:schema>

5.2.2 Schema for Libraries

The Listing 5.4 contains the implementation for libraries' schema.

Listing 5.4: Schema for Libraries

```
1 <?xml version="1.0" encoding="iso-8859-1"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xlink="http://www.w3.org/1999/xlink">
3 <xs:import namespace="http://www.w3.org/1999/xlink"
  schemaLocation="http://www.w3.org/1999/xlink.xsd"/>
4 <!--Importing Common XMLSchema-->
5 <xs:include schemaLocation="D:\MEng_Docs\MEng_Docs\Project\
  Common_Schema\XmlSchema_Common.xsd"/>
6 <xs:include schemaLocation="D:\MEng_Docs\MEng_Docs\Project\
  XmlSchema_Libraries\XMLSchema_Common_Libraries_Tools.xsd"
  />
7
8 <xs:complexType name="downloadType">
9   <xs:sequence>
10     <xs:element name="description" type="xs:string"
      minOccurs="0"></xs:element>
11     <xs:element name="compatibleOS" type="xs:string"
      minOccurs="0"></xs:element>
12     <xs:element name="size" type="xs:string" minOccurs
      ="0"></xs:element>
13     <xs:element name="linkDescription" minOccurs="0"
      maxOccurs="unbounded">
14       <xs:complexType>
15         <xs:simpleContent>
16           <xs:extension base="xs:string">
17             <xs:anyAttribute namespace="http://www.
              w3.org/1999/xlink"/>
18           </xs:extension>
19         </xs:simpleContent>
20       </xs:complexType>
21     </xs:element>
22   </xs:sequence>
23 </xs:complexType>
24 <xs:complexType name="downloadsType">
25   <xs:sequence>
26     <xs:element name="download" type="downloadType"
      maxOccurs="unbounded"></xs:element>
```

```

27     </xs:sequence>
28 </xs:complexType>
29 <xs:complexType name="dependencyDetailsType">
30     <xs:sequence>
31         <xs:element name="dependency" type="descriptionType"
32             maxOccurs="unbounded"></xs:element>
33     </xs:sequence>
34 </xs:complexType>
35 <xs:complexType name="examplesType">
36     <xs:sequence>
37         <xs:element name="example" type="descriptionType"
38             maxOccurs="unbounded"></xs:element>
39     </xs:sequence>
40 </xs:complexType>
41 <xs:complexType name="referencesType">
42     <xs:sequence>
43         <xs:element name="reference" type="descriptionType"
44             maxOccurs="unbounded"></xs:element>
45     </xs:sequence>
46 </xs:complexType>
47
48 <xs:element name="libraries">
49     <xs:complexType>
50         <xs:sequence>
51             <xs:element name="library" maxOccurs="unbounded">
52                 <xs:complexType>
53                     <xs:sequence>
54                         <xs:element name="libraryName" type="xs:string"
55                             "></xs:element>
56                         <xs:element name="libraryOverview" type="
57                             descriptionType" minOccurs="0"></xs:element>
58                         <xs:element name="availableVersions" type="
59                             versionsType" minOccurs="0"></xs:element>
60                         <xs:element name="
61                             experimentalContentsOrExtensions" type="
62                             exContentsType" minOccurs="0"></xs:element>
63                         <xs:element name="contributors" type="
64                             contactsType" minOccurs="0"></xs:element>
65                         <xs:element name="contentFiles" type="
66                             contentFilesType" minOccurs="0"></xs:element
67                             >
68                         <xs:element name="downloads" type="
69                             downloadsType" minOccurs="0"></xs:element>
70                         <xs:element name="requiredToolsForCompilation"
71                             type="toolsType" minOccurs="0"></xs:element>

```

```

59         <xs:element name="supportingDocuments" type="
            supportingDocumentsType" minOccurs="0"></xs:
            element>
60         <xs:element name="dependencyDetails" type="
            dependencyDetailsType" minOccurs="0"></xs:
            element>
61         <xs:element name="examples" type="examplesType"
            minOccurs="0"></xs:element>
62         <xs:element name="relatedLinks" type="
            relatedLinksType" minOccurs="0"></xs:element
            >
63         <xs:element name="references" type="
            referencesType" minOccurs="0"></xs:element>
64     </xs:sequence>
65 </xs:complexType>
66 </xs:element>
67 </xs:sequence>
68 </xs:complexType>
69 </xs:element>
70 </xs:schema>

```

5.2.3 Schema for Tools

The Listing 5.5 gives the implementation for tools' schema.

Listing 5.5: Schema for Libraries

```

1 <?xml version="1.0" encoding="iso-8859-1"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xlink="http://www.w3.org/1999/xlink">
3 <!--Importing Common XMLSchema-->
4 <xs:include schemaLocation="D:\MEng_Docs\MEng_Docs\Project\
    Common_Schema\XmlSchema_Common.xsd"/>
5 <xs:include schemaLocation="D:\MEng_Docs\MEng_Docs\Project\
    XmlSchema_Libraries\XMLSchema_Common_Libraries_Tools.xsd
    "/>
6 <xs:import namespace="http://www.w3.org/1999/xlink"
    schemaLocation="http://www.w3.org/1999/xlink.xsd"/>
7
8 <xs:complexType name="functionalitiesType">
9     <xs:sequence>
10         <xs:element name="functionality" type="descriptionType"
            maxOccurs="unbounded"></xs:element>
11     </xs:sequence>
12 </xs:complexType>

```

```

13 <xs:complexType name="intendedUsersType">
14   <xs:sequence>
15     <xs:element name="user" type="xs:string" maxOccurs="
      unbounded"></xs:element>
16   </xs:sequence>
17 </xs:complexType>
18 <xs:complexType name="downloadType">
19   <xs:sequence>
20     <xs:element name="description" type="xs:string"
      minOccurs="0"></xs:element>
21     <xs:element name="size" type="xs:string" minOccurs
      ="0"></xs:element>
22     <xs:element name="format" type="xs:string" minOccurs
      ="0"></xs:element>
23     <xs:element name="linkDescription" minOccurs="0"
      maxOccurs="unbounded">
24       <xs:complexType>
25         <xs:simpleContent>
26           <xs:extension base="xs:string">
27             <xs:anyAttribute namespace="http://www.
      w3.org/1999/xlink"/>
28           </xs:extension>
29         </xs:simpleContent>
30       </xs:complexType>
31     </xs:element>
32   </xs:sequence>
33   <xs:attribute name="status" use="required">
34     <xs:simpleType>
35       <xs:restriction base="xs:string">
36         <xs:enumeration value="Source"/>
37         <xs:enumeration value="Binary"/>
38         <xs:enumeration value="Other"/>
39       </xs:restriction>
40     </xs:simpleType>
41   </xs:attribute>
42 </xs:complexType>
43 <xs:complexType name="howToObtainType">
44   <xs:sequence>
45     <xs:element name="Download" type="downloadType"
      maxOccurs="unbounded"></xs:element>
46   </xs:sequence>
47 </xs:complexType>
48 <xs:complexType name="inputRequirementsType">
49   <xs:sequence>
50     <xs:element name="inputRequirement" type="xs:string"

```

```

maxOccurs="unbounded"></xs:element>
51 </xs:sequence>
52 </xs:complexType>
53 <xs:complexType name="compilersType">
54 <xs:sequence>
55 <xs:element name="compiler" maxOccurs="unbounded">
56 <xs:complexType>
57 <xs:sequence>
58 <xs:element name="compilerName" type="xs:string
"></xs:element>
59 <xs:element name="compilerDescription" type="xs:
:string" minOccurs="0"></xs:element>
60 </xs:sequence>
61 </xs:complexType>
62 </xs:element>
63 </xs:sequence>
64 </xs:complexType>
65 <xs:complexType name="processorsType">
66 <xs:sequence>
67 <xs:element name="processor" maxOccurs="unbounded">
68 <xs:complexType>
69 <xs:sequence>
70 <xs:element name="processorName" type="xs:
string"></xs:element>
71 <xs:element name="processorMemory" type="xs:
string"></xs:element>
72 <xs:element name="processorDescription" type="
xs:string" minOccurs="0"></xs:element>
73 </xs:sequence>
74 </xs:complexType>
75 </xs:element>
76 </xs:sequence>
77 </xs:complexType>
78 <xs:complexType name="osType">
79 <xs:sequence>
80 <xs:element name="OS" maxOccurs="unbounded">
81 <xs:complexType>
82 <xs:sequence>
83 <xs:element name="osName" type="xs:string"></xs
:element>
84 <xs:element name="osVersion" type="xs:string
"></xs:element>
85 </xs:sequence>
86 </xs:complexType>
87 </xs:element>

```

```

88     </xs:sequence>
89 </xs:complexType>
90 <xs:complexType name="languagesType">
91     <xs:sequence>
92         <xs:element name="language" type="xs:string" maxOccurs
            ="unbounded"></xs:element>
93     </xs:sequence>
94 </xs:complexType>
95 <xs:complexType name="executionEnvironmentType">
96     <xs:sequence>
97         <xs:element name="languagesUsed" type="languagesType"
            minOccurs="0"></xs:element>
98         <xs:element name="compilers" type="compilersType"
            minOccurs="0"></xs:element>
99         <xs:element name="processors" type="processorsType"
            minOccurs="0"></xs:element>
100        <xs:element name="OSUsed" type="osType" minOccurs
            ="0"></xs:element>
101        <xs:element name="description" type="xs:string"
            minOccurs="0"></xs:element>
102    </xs:sequence>
103 </xs:complexType>
104 <xs:element name="verificationTools">
105     <xs:complexType>
106         <xs:sequence>
107             <xs:element name="verificationTool" maxOccurs="
                unbounded">
108                 <xs:complexType>
109                     <xs:sequence>
110                         <xs:element name="toolName" type="xs:string"></
                            xs:element>
111                         <xs:element name="toolOverview" type="
                            descriptionType" minOccurs="0"></xs:element>
112                         <xs:element name="availableVersions" type="
                            versionsType" minOccurs="0"></xs:element>
113                         <xs:element name="Functionalities" type="
                            functionalitiesType" minOccurs="0"></xs:
                            element>
114                         <xs:element name="intendedUsers" type="
                            intendedUsersType" minOccurs="0"></xs:
                            element>
115                         <xs:element name="
                            experimentalContentsOrExtensions" type="
                            exContentsType" minOccurs="0"></xs:element>
116                         <xs:element name="contacts" type="contactsType"

```

```

117         minOccurs="0"></xs:element>
        <xs:element name="contentFiles" type="
            contentFilesType" minOccurs="0"></xs:element
118     >
        <xs:element name="supportingDocuments" type="
            supportingDocumentsType" minOccurs="0"></xs:
            element>
119     <xs:element name="relatedLinks" type="
            relatedLinksType" minOccurs="0"></xs:element
            >
120     <xs:element name="relatedTools" type="toolsType
            " minOccurs="0"></xs:element>
121     <xs:element name="howToObtain" type="
            howToObtainType" minOccurs="0"></xs:element>
122     <xs:element name="executionEnvironment" type="
            executionEnvironmentType" minOccurs="0"></xs
            :element>
123     </xs:sequence>
124 </xs:complexType>
125 </xs:element>
126 </xs:sequence>
127 </xs:complexType>
128 </xs:element>
129 </xs:schema>

```

5.3 View Data in Browser

Similar to challenge problems and course modules, libraries and tools XMLs are viewed in the browser with the help of XSLs.

5.3.1 XSL for Libraries

The Listing 5.6 shows some part of the XSL code for the libraries.

Listing 5.6: Sample of XSL for Course Module

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org
    /1999/XSL/Transform"
3     xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-
        result-prefixes="msxsl"
4     xmlns:xlink="http://www.w3.org/1999/xlink"
5 >
6 <xsl:output method="xml" indent="yes"/>

```

```

7 <xsl:template match="/">
8 <html>
9 <style type="text/css">
10   .alignmentCenter{
11     text-align: center;
12   }
13   .setWidth{
14     width: 50%;
15   }
16 }
17 </style>
18 <body>
19 <xsl:for-each select="libraries/library">
20   <h3 class="alignmentCenter">
21     <xsl:value-of select="libraryName"/>
22   </h3>
23   <xsl:if test="libraryOverview">
24     <h3><xsl:text>Overview:- </xsl:text></h3>
25     <xsl:value-of select="libraryOverview"/>
26     <br/>
27   </xsl:if>
28   <xsl:if test="contributors">
29     <h3>Contributors:-</h3>
30     <xsl:for-each select="contributors/contact">
31       <xsl:if test="name">
32         <li>
33           <xsl:value-of select="name"/>
34         </li>
35       </xsl:if>
36     <xsl:if test="contactsDescription">
37       <xsl:text>Description-</xsl:text>
38       <xsl:for-each select="contactsDescription">
39         <xsl:value-of select="."/>
40       <br/>
41     </xsl:for-each>
42   </xsl:if>
43   <xsl:if test="phone">
44     <xsl:text>Phone-</xsl:text>
45     <br/>
46     <xsl:for-each select="phone">
47       <xsl:value-of select="."/>
48     <br/>
49   </xsl:for-each>
50 </xsl:if>
51 <xsl:if test="email">

```



```

52     <xsl:text>Email-</xsl:text>
53     <br/>
54     <xsl:for-each select="email">
55         <xsl:value-of select="."/>
56         <br/>
57     </xsl:for-each>
58 </xsl:if>
59 <xsl:if test="linkDescription">
60     <xsl:for-each select="linkDescription">
61         <xsl:element name="a">
62             <xsl:attribute name="href">
63                 <xsl:value-of select="./@xlink:href" />
64             </xsl:attribute>
65             <xsl:value-of select="." />
66         </xsl:element>
67     <br/>
68 </xsl:for-each>
69 </xsl:if>
70 </xsl:for-each>
71 </xsl:if>
72 <xsl:if test="contentFiles">
73     <h3><xsl:text>Content Files:-</xsl:text></h3>
74     <xsl:if test="contentFiles/contentDescription">
75         <b>
76             <xsl:text>Description:- </xsl:text>
77             </b><xsl:value-of select="contentFiles/
              contentDescription"/><br/>
78         </xsl:if>
79         <b><xsl:text>Contents-</xsl:text></b>
80         <br/>
81         <xsl:for-each select="contentFiles/contentFile">
82             <xsl:if test="name">
83                 <b><xsl:value-of select="name"></xsl:value-of></b><br/>
84             </xsl:if>
85             <xsl:if test="description">
86                 <xsl:value-of select="description"></xsl:value-of><br/>
87             </xsl:if>
88         <xsl:if test="linkDescription">
89             <xsl:for-each select="linkDescription">
90                 <xsl:element name="a">
91                     <xsl:attribute name="href">
92                         <xsl:value-of select="./@xlink:href" />
93                     </xsl:attribute>
94                     <xsl:value-of select="." />
95                 </xsl:element>

```

```

96     <br/>
97     </xsl:for-each>
98     </xsl:if> <br/>
99     </xsl:for-each>
100  </xsl:if>
101 </xsl:for-each>
102 </body>
103 </html>
104 </xsl:template>
105 </xsl:stylesheet>

```

5.3.2 XSL for Tools

The Listing 5.7 shows some part of the XSL code for the tools.

Listing 5.7: Sample of XSL for Course Module

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org
   /1999/XSL/Transform"
3      xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-
      result-prefixes="msxsl"
4      xmlns:xlink="http://www.w3.org/1999/xlink"
5  >
6  <xsl:output method="xml" indent="yes"/>
7  <xsl:template match="/">
8  <html >
9  <style type="text/css">
10     .alignmentCenter{
11     text-align: center;
12     }
13     .setWidth{
14     width: 50%;
15     }
16  }
17 </style>
18 <body>
19   <xsl:for-each select="verificationTools/verificationTool">
20   <h3 class="alignmentCenter">
21   <xsl:value-of select="toolName"/>
22   </h3>
23   <xsl:if test="toolOverview">
24   <h3>
25   <xsl:text>Overview:– </xsl:text>
26   </h3>

```

```

27 <xsl:value-of select="toolOverview"/>
28 <br/>
29 </xsl:if>
30 <xsl:if test="howToObtain">
31 <h3>
32 <xsl:text>Getting the tool:-</xsl:text>
33 </h3>
34 <table border="1">
35 <tr bgcolor="green">
36 <th>Description </th>
37 <th>Size </th>
38 <th>Format</th>
39 <th>Link</th>
40 <th>Form</th>
41 </tr>
42 <xsl:for-each select="howToObtain/Download">
43 <tr>
44 <td>
45 <xsl:choose>
46 <xsl:when test="description">
47 <xsl:value-of select="description"/>
48 </xsl:when>
49 <xsl:otherwise>
50 <xsl:text>—</xsl:text>
51 </xsl:otherwise>
52 </xsl:choose>
53 </td>
54 <td>
55 <xsl:choose>
56 <xsl:when test="size">
57 <xsl:value-of select="size"/>
58 </xsl:when>
59 <xsl:otherwise>
60 <xsl:text>—</xsl:text>
61 </xsl:otherwise>
62 </xsl:choose>
63 </td>
64 <td>
65 <xsl:choose>
66 <xsl:when test="format">
67 <xsl:value-of select="format"/>
68 </xsl:when>
69 <xsl:otherwise>
70 <xsl:text>—</xsl:text>
71 </xsl:otherwise>

```

```

72     </xsl:choose>
73 </td>
74 <td>
75     <xsl:choose>
76     <xsl:when test="linkDescription">
77         <table>
78         <xsl:for-each select="linkDescription">
79             <tr>
80                 <td>
81                     <xsl:element name="a">
82                         <xsl:attribute name="href">
83                             <xsl:value-of select="./@xlink:href" />
84                         </xsl:attribute>
85                         <xsl:value-of select="." />
86                     </xsl:element>
87                 </td>
88             </tr>
89         </xsl:for-each>
90         </table>
91     </xsl:when>
92     <xsl:otherwise>
93         <xsl:text>—</xsl:text>
94     </xsl:otherwise>
95 </xsl:choose>
96 </td>
97 <td>
98     <xsl:choose>
99     <xsl:when test="@status">
100     <xsl:value-of select="@status"/>
101     </xsl:when>
102     <xsl:otherwise>
103     <xsl:text>—</xsl:text>
104     </xsl:otherwise>
105     </xsl:choose>
106 </td>
107 </tr>
108 </xsl:for-each>
109 </table>
110 </xsl:if>
111 </xsl:for-each>
112 </body>
113 </html>
114 </xsl:template>
115 </xsl:stylesheet>

```

Chapter 6

Testing

The testing begins with preparing some sample XMLs for all the areas discussed. The whole schema design, implementation, XSLs implementation and testing process follows an iterative approach. All the three areas viz. challenge problem, course modules and Libraries and tools are tested individually. However, they follow the same testing process.

6.1 Test Cases

After the schema design and implementation, a XML is prepared from a sample challenge problem, course module, library and a tool. This XML is validated against the first version of the schema. Some test cases described in table 6.1 are tested to check the strength, validity and effectiveness of the schema. According to the results, improvements are made to the schema. This process is repeated till the schema is tested against reasonable number of samples and passes all the test cases correctly without the further need of modification to the existing schema.

The set of XML samples is then transformed using the XSL designed according to the final tested schema. The HTML file produced as an output is tested against some test cases given in table 6.1. This again follows the iterative approach till all the test cases are satisfied.

The table 6.1 lists down the test cases used for both XSDs and XSLs of challenge problems, course modules, libraries and tools.

Sr.No.	Test Case Name	Description	Applicable For	Result
1.	Information coverage	check whether all the information is covered	XSD/XSL	Passed
2.	Appropriate attribute tagging	All the information should be tagged appropriately.	XSD	Passed
3.	Allowance of specific information	Specific but important information for the challenge should also be covered without affecting the general structure of the schema	XSD	Passed
4.	Flexibility	Not all the elements are present in all the samples. The schema should allow users to insert only available information. Checking the use of optional attributes.	XSD/XSL	Passed
5.	Attribute coverage	Testing on reasonable number of samples to verify all the attributes in the schema or XSL are utilized	XSD/XSL	Passed
6.	Common schema	Use of common schema to implement common elements	XSD	Passed
7.	Look and feel for XSL	Checking whether all the information when viewed in the browser is properly aligned and look and feel of HTML page	XSL	Passed

Table 6.1: Test Cases

6.2 Case Study

This section gives a sample XML and its transformed HTML code for a challenge problem. The XMLs illustrate how the actual information is stored in the repository using the newly designed schemas and HTML files depict how the XMLs are transformed and viewed in the browser using the newly XSLs implemented according to the corresponding schemas.

Note that the report just gives one sample XML and HTML for a challenge problem. However, the schemas and XSLs are tested against reasonable number of XMLs in order to verify their validity and effectiveness and similarly XMLs and HTMLs can be obtained for other categories and data samples.

Additionally, the XML and HTML has been truncated to fit inside the report. The actual files are longer than the ones included in the report. Thus, they act like representatives for all the XMLs and HTMLs required.

The Listing 6.1 gives XML for SAT challenge. The XML is prepared by extracting the information from SAT challenge website and then tagging it according to the challenge problem schema given in the Listing 3.4 [SAT]. The XML is then validated using the online tool against the challenge problem schema [Bria].

Listing 6.1: Sample XML for Challenge Problems: SAT Challenge

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <challenges xmlns:xlink="http://www.w3.org/1999/xlink">
3   <challenge>
4     <challengeName>SAT Challenge</challengeName>
5     <area>SAT Solving</area>
6     <challengeDescription>
7       <description>The SAT Challenge 2012 is a competitive
          event for solvers of the Boolean Satisfiability (
          SAT) problem.
8       It is organized as a satellite event to the Fifteenth
          International Conference on Theory and
          Applications of Satisfiability Testing (SAT 2012)
          and stands in the tradition of the SAT
          Competitions that have been held yearly from 2002
          to 2005 and biannually starting from 2007, and the
          SAT-Races held in 2006, 2008 and 2010.
9     </description>
10    </challengeDescription>
11    <rules>
12      <rule>
13        <ruleCategory>Participation</ruleCategory>
```

```

14      <ruleDescription>A solver may participate in
        multiple tracks. The organization committee
        reserves the right to restrict participation of
        a solver to certain tracks,
15    and to allow only a limited number of solvers
        submitted by the same person
16    </ruleDescription>
17    <inputRequirements>
18      <inputRequirement>Solvers must provide a
        satisfying truth assignment for satisfiable
        instances</inputRequirement>
19      <inputRequirement>An entrant to the SAT Challenge
        2012 is a SAT solver submitted in either
        source code or binary format using the WWW
        registration form</inputRequirement>
20    </inputRequirements>
21    <outputRequirements>
22      <outputRequirement>Solvers must provide a
        satisfying truth assignment for satisfiable
        instances</outputRequirement>
23    </outputRequirements>
24    <ruleLinkDescription xlink:type="simple" xlink:href
        ="satcompetition.org/2011/rules.pdf" xlink:show
        ="new">Rules Information</ruleLinkDescription>
25  </rule>
26 </rules>
27 <supportingDocuments>
28   <document>
29     <documentName>Google Groups</documentName>
30     <documentLinkDescription xlink:type="simple" xlink:
        href="https://groups.google.com/forum/#!forum/
        sat-challenge-2012" xlink:show="new">
31       SAT challenge 2012
32     </documentLinkDescription>
33     <documentLinkDescription xlink:type="simple" xlink:
        href="https://groups.google.com/forum/#!forum/
        edacc" xlink:show="new">
34       EDACC
35     </documentLinkDescription>
36   </document>
37 </supportingDocuments>
38 <year>2012</year>
39 <assessmentDescription>
40   <description>
41     Assessment of solvers will be based on the number

```



```

42         of successfully handled instances and the time
           needed to solve them. Solvers will be assessed
           based on the number of instances solved within
           the run-time limit. If several solvers
           successfully process the same number of
           instances, as a second criterion, the cumulated
           run-time (CPU time for sequential solvers,
43         wall-clock time for parallel solvers) of all solved
           instances is used to rank the solvers
44     </description>
45 </assessmentDescription>
46 <benchmarks>
47     <benchmark>
48         <categories>Random Challenge</categories>
49         <description>
           The benchmark pool will include instances from
           previous SAT-Races and SAT Competitions, but
           may also contain additional instances.
50         Note that you will not receive an automated
           acknowledgement of your submission,
51         as the benchmarks are added to the EDACC system
           manually. Your benchmark set will appear in
           the "List of submitted benchmarks" when it is
           added to the system.
52     </description>
53     <format>Submission of archive with your benchmarks
           is through the web form.
54     Please make sure to select the correct category,
           and to include the PDF file with the description
           of the benchmarks in the archive</format>
55     <timeline>PT15M</timeline>
56     <benchmarkLinkDescription xlink:type="simple" xlink
           :href="http://baldur.iti.kit.edu/SAT-Challenge
           -2012/submission.html" xlink:show="new">
57         Submit new benchmark problems
58     </benchmarkLinkDescription>
59 </benchmark>
60 </benchmarks>
61 <expectedSolution>
62     <AllowedForms>Source Code</AllowedForms>
63     <AllowedForms>Binary</AllowedForms>
64     <inputRequirements>
65         <inputRequirement>
66             In order to obtain reproducible results, SAT
           solvers should refrain from using non-

```

```

        deterministic
67     program constructs as far as possible.
68     It is allowed to set a random seed value on the
        command line , however.
69     </inputRequirement>
70 </inputRequirements>
71 <outputRequirements>
72     <outputRequirement>Each SAT Challenge 2012 entrant
        submitted as source code must include a
73     README file explaining how to compile the solver</
        outputRequirement>
74 </outputRequirements>
75 <executionEnvironment>
76     <environmentDescription>Solvers submitted to the
        SAT Challenge 2012 will be run on
77     the bwGRiD cluster of the State of Baden-
        Wrttemberg , Germany</environmentDescription>
78 <compilers>
79     <compiler>
80         <compilerName>GCC 4.1.2</compilerName>
81     </compiler>
82     <compiler>
83         <compilerName>javac 1.6.0</compilerName>
84     </compiler>
85 </compilers>
86 <processors>
87     <processor>
88         <processorName>2x Quad-Core Intel Xeon E5440 ,
            2.83 GHz</processorName>
89         <processorMemory>16 GB per node</
            processorMemory>
90         <processorDescription>Cache-12 MB L2 per Quad-
            Core CPU</processorDescription>
91     </processor>
92 </processors>
93 <OSUsed>
94     <OS>
95         <osName>Scientific Linux</osName>
96         <osVersion>kernel 2.6.18 , glibc 2.5 , both 32-
            bit and 64-bit</osVersion>
97     </OS>
98 </OSUsed>
99 </executionEnvironment>
100 <deadlines>
101     <deadline>

```

```

102         <deadlineName>
103             Webpage for submitting solvers and benchmarks (
104                 via the EDACC system) will open and
105                 the start of testing phase. During that phase
106                 submitters can test their solvers within the
107                 EDACC system.
108         </deadlineName>
109         <submissionDeadline>2012-03-11T00:00:00 </
110             submissionDeadline>
111     </deadline>
112 </deadlines>
113 <allowedSubmissions>The organization committee
114     reserves the right to restrict participation of a
115     solver to certain tracks ,
116     and to allow only a limited number of solvers
117     submitted by the same person.</allowedSubmissions>
118 </expectedSolution>
119 <contactDetails>
120     <contact>
121         <name>Organizing Committee</name>
122         <linkDescription xlink:type="simple" xlink:href="
123             http://www.uni-ulm.de/en/in/institute-of-
124             theoretical-computer-science/m/balint.html"
125             xlink:show="new">
126             Adrian Balint Web Link
127         </linkDescription>
128         <linkDescription xlink:type="simple" xlink:href="
129             http://anton.belov-mcdowell.com/baker/" xlink:
130             show="new">
131             Dr. Anton Belov Web Link
132         </linkDescription>
133     </contact>
134     <contact>
135         <name>Help with the EDACC system</name>
136         <email>edacc.contact@gmail.com</email>
137     </contact>
138 </contactDetails>
139 </challenge>
140 </challenges>

```

The Listing 6.2 gives the HTML code for above XML after transforming it using the corresponding XSL with the help of a tool [Brib].

Listing 6.2: Sample HTML for Challenge Problems: SAT Challenge

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <html xmlns:xlink="http://www.w3.org/1999/xlink">
3      <style type="text/css">
4          .alignmentCenter{
5              text-align: center;
6          }
7          .setWidth{
8              width: 50%;
9          }
10     }
11 </style>
12 <body>
13     <h1 class="alignmentCenter">SAT Challenge</h1>
14     <h2>SAT Solving</h2>
15     <h3>The SAT Challenge 2012 is a competitive event
        for solvers of the Boolean Satisfiability (SAT)
        problem.
16     It is organized as a satellite event to the Fifteenth
        International Conference on Theory and
        Applications of Satisfiability Testing (SAT 2012)
        and stands in the tradition of the SAT
        Competitions that have been held yearly from 2002
        to 2005 and biannually starting from 2007, and the
        SAT-Races held in 2006, 2008 and 2010.
17 </h3>
18     <h3>Rules</h3>
19     <b>Participation: </b>
20     <br/>A solver may participate in multiple tracks.
        The organization committee reserves the right to
        restrict participation of a solver to certain
        tracks ,
21     and to allow only a limited number of solvers
        submitted by the same person
22     <br/>
23     <b>Input Requirements-</b>
24     <br/>
25     <li>Solvers must provide a satisfying truth
        assignment for satisfiable instances</li>
26     <li>An entrant to the SAT Challenge 2012 is a SAT
        solver submitted in either source code or binary
        format using the WWW registration form</li>
27     <b>Output Requirements-</b>
28     <br/>
29     <li>Solvers must provide a satisfying truth
        assignment for satisfiable instances</li>

```

```

30      <b>Links-</b>
31      <br/>
32      <li>
33          <a href="satcompetition.org/2011/rules.pdf">
34              Rules Information</a>
35      </li>
36      <br/>
37      <h3>Supporting Documents</h3>
38      <table border="1">
39          <tr bgcolor="green">
40              <th>Document Name</th>
41              <th>Description</th>
42              <th>Link</th>
43          </tr>
44          <tr>
45              <td class="setWidth">Google Groups</td>
46              <td>
47                  <table>
48                      <tr>
49                          <td>
50                              <a href="https://groups.
51                                  google.com/forum/#!forum
52                                  /sat-challenge-2012">
53                                  SAT challenge 2012
54                                  </a>
55                              </td>
56                              </tr>
57                              <tr>
58                                  <td>
59                                      <table>
60                                          <tr>
61                                              <td>
62                                                  <table>
63                                                      <tr>
64                                                          <td>
65                                                              EDACC
66                                                              </td>
67                                                              </tr>

```

based on the number of instances solved within
 the run-time limit. If several solvers
 successfully process the same number of
 instances, as a second criterion, the cumulated
 run-time (CPU time for sequential solvers,
 wall-clock time for parallel solvers) of all solved
 instances is used to rank the solvers

 <h3>Benchmarks</h3>
 Random Challenge

Description:–
 The benchmark pool will include instances from
 previous SAT-Races and SAT Competitions, but
 may also contain additional instances.
 Note that you will not receive an automated
 acknowledgement of your submission,
 as the benchmarks are added to the EDACC system
 manually. Your benchmark set will appear in
 the "List of submitted benchmarks" when it is
 added to the system.

Format:– Submission of archive with your
 benchmarks is through the web form.
 Please make sure to select the correct category,
 and to include the PDF file with the description
 of the benchmarks in the archive
Timeline:–
 PT15M
Links:–
 <a href="http://baldur.itl.kit.edu/SAT-
 Challenge-2012/submission.html">
 Submit new benchmark problems

 <h3>Expected Solution</h3>Allowed Forms
Source
 Code
Binary

 Input Requirements–

 In order to obtain reproducible results, SAT
 solvers should refrain from using non-
 deterministic
 program constructs as far as possible.
 It is allowed to set a random seed value on the
 command line, however.

 Output Requirements–

```

92     <br/>
93     <li>Each SAT Challenge 2012 entrant submitted as
          source code must include a
94     README file explaining how to compile the solver</li>
95     <b>Execution Enviornment</b>
96     <br/>Description:– Solvers submitted to the SAT
          Challenge 2012 will be run on
97     the bwGRiD cluster of the State of Baden–
          Wrttemberg , Germany<br/>Compiler Name: – GCC
          4.1.2<br/>Compiler Name: – javac 1.6.0<br/>
          Processor Name: – 2x Quad–Core Intel Xeon E5440 ,
          2.83 GHz<br/>Processor Memory: – 16 GB per node
          <br/>Processor Description: – Cache–12 MB L2 per
          Quad–Core CPU<br/>OS Name: – Scientific Linux<br/>
          OS Version: – kernel 2.6.18 , glibc 2.5 , both
          32–bit and 64–bit<br/>
98     <b>Deadlines </b>
99     <table border="1">
100         <tr bgcolor="green">
101             <th>Deadline For</th>
102             <th>Date</th>
103         </tr>
104         <tr>
105             <td>
106                 Webpage for submitting solvers and benchmarks (
107                 via the EDACC system) will open and
108                 the start of testing phase. During that phase
109                 submitters can test their solvers within the
110                 EDACC system.
111             </td>
112             <td>2012–03–11T00:00:00</td>
113         </tr>
114     </table>
115     <b>Allowed Submissions: – </b>The organization
          committee reserves the right to restrict
          participation of a solver to certain tracks ,
116     and to allow only a limited number of solvers
          submitted by the same person.<br/>
117     <h3>Contacts</h3>
118     <li>
          <b>Organizing Committee</b>
          <br/>
          <a href="http://www.uni-ulm.de/en/in/institute-of-
          theoretical-computer-science/m/balint.html">

```

```
119         Adrian Balint Web Link
120     </a>
121     <br/>
122     <a href="http://anton.belov-mcdowell.com/baker/">
123         Dr. Anton Belov Web Link
124     </a>
125     <br/>
126 </body>
127 </html>
```

Chapter 7

Conclusion

Software certification refers to the process of certification of software part in the system. The report talked about some approaches to accomplish this. Component reusability is now getting popular in software certification. However, the component to be reused needs to have the appropriate certification and certification management system. One of the challenges in this area is the management of useful information regarding the certification. Therefore, the report proposed the overall design of the software repository which would manage information related to various areas around software certification, mainly focussing on the three areas viz. challenge problems, course modules and libraries and tools.

Future tasks for the repository can be categorized into two streams. First one would be to find more samples for all the three components discussed here. The major challenge for the work presented in the report was data gathering. It is a challenging task to find the relevant data samples. This is true for all the areas such as challenge problems, course modules, libraries and tools. Hence, the next task for the three areas in the report is to find some more relevant challenge problems, course modules, libraries and tools. The second part of the future task is to design and implement schema for the rest of the two areas of the repository. The schemas can be designed and implemented by following the same methods illustrated in the report. This would complete the implementation of the proposed repository.

Finally, to conclude, the report explained the process of schema designing and presented implementations followed by their XSL implementations for the three areas such as challenge problems, course modules, libraries and tools related to software certification and therefore provided an effective mechanism to index some available materials around these areas.

Bibliography

- [Ale13] Alea: a library for reasoning on randomized algorithms in Coq . <https://www.lri.fr/~paulin/ALEA/>, 2013.
- [AW10] Mark Lawford Alan Wassyng, Tom Maibaum. On software certification: We need product-focused approaches. Number LNCS 6028. Springer-Verlag Berlin Heidelberg, 2010. C. Choppy and O. Sokolsky (Eds.): Monterey Workshop 2008.
- [Ben11] Marc Bender. What is software certification? McMaster Centre for Software Certification (McSCert), April 13 2011. McSCert Seminar 2.
- [Bria] Domenico Briganti. W3C XML Schema (XSD) Validation online. http://www.utilities-online.info/xsdvalidation/#.U0M3I_ldXwg.
- [Brib] Domenico Briganti. XSLT (eXtensible Stylesheet Language Transformations) online transformations. <http://www.utilities-online.info/xslttransformation/#.U0M6RfldXwg>.
- [CAdA] Construction and Analysis of Distributed Processes. <http://cadp.inria.fr/>. Toolbox for the design of asynchronous concurrent systems.
- [CAdB] The CADE ATP System Competition. <http://www.cs.miami.edu/~tptp/CASC/>. The World Championship for Automated Theorem Proving.
- [COQ] The Coq Proof Assistant. <http://coq.inria.fr/>.
- [CW94] Per Runeson Claes Wohlin. Certification of software components. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, 20, June 1994.

- [DF05] Ewen Denney and Bernd Fischer. Software certification and software certificate management systems. In *2005 Automated Software Engineering Workshop on Software Certificate Management*, Long Beach, California, USA, November 8 2005. Association for Computing Machinery.
- [DFC] A Coq Library on Floating-Point Arithmetic . <http://lipforge.ens-lyon.fr/www/pff/>.
- [MaC] MECHTRON/SFWR ENG 4AA4/6GA3 Real-Time Systems . <http://www.cas.mcmaster.ca/~downd/courses/4aa4/4aaout.htm>. McMaster University.
- [MCH] Microchip Certified Class B Safety Software Library for 16 bit and PIC32 MCUs . http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2680&dDocName=en548988.
- [Pac07] Pacemaker Formal Methods Challenge. <http://sqr1.mcmaster.ca/pacemaker.htm>, 2007.
- [SAT] SAT Challenge 2012. <http://baldur.iti.kit.edu/SAT-Challenge-2012/>.
- [SMT] SMT-COMP 2012. <http://smtcomp.sourceforge.net/2012/>.
- [SV13] Competition on Software Verification (SV-COMP). <http://sv-comp.sosy-lab.org/2013/index.php>, 2013.
- [Vic] Embedded and Networked Systems . <http://www.vu.edu.au/units/ENE3202>. Victoria University.
- [Voa14] Jeffrey Voas. Developing a usage-based software certification process. *Reliable Software Technologies*, 03 2014.
- [Wat] ECE423: Embedded Computer Systems . <https://ece.uwaterloo.ca/~rpellizz/ECE423.php>. University of Waterloo.
- [Yor] Course Modules Offered at York University . <https://www.cs.york.ac.uk/postgraduate/taught-courses/msc-scse/#tab-2>. MSc in Safety Critical Systems Engineering.