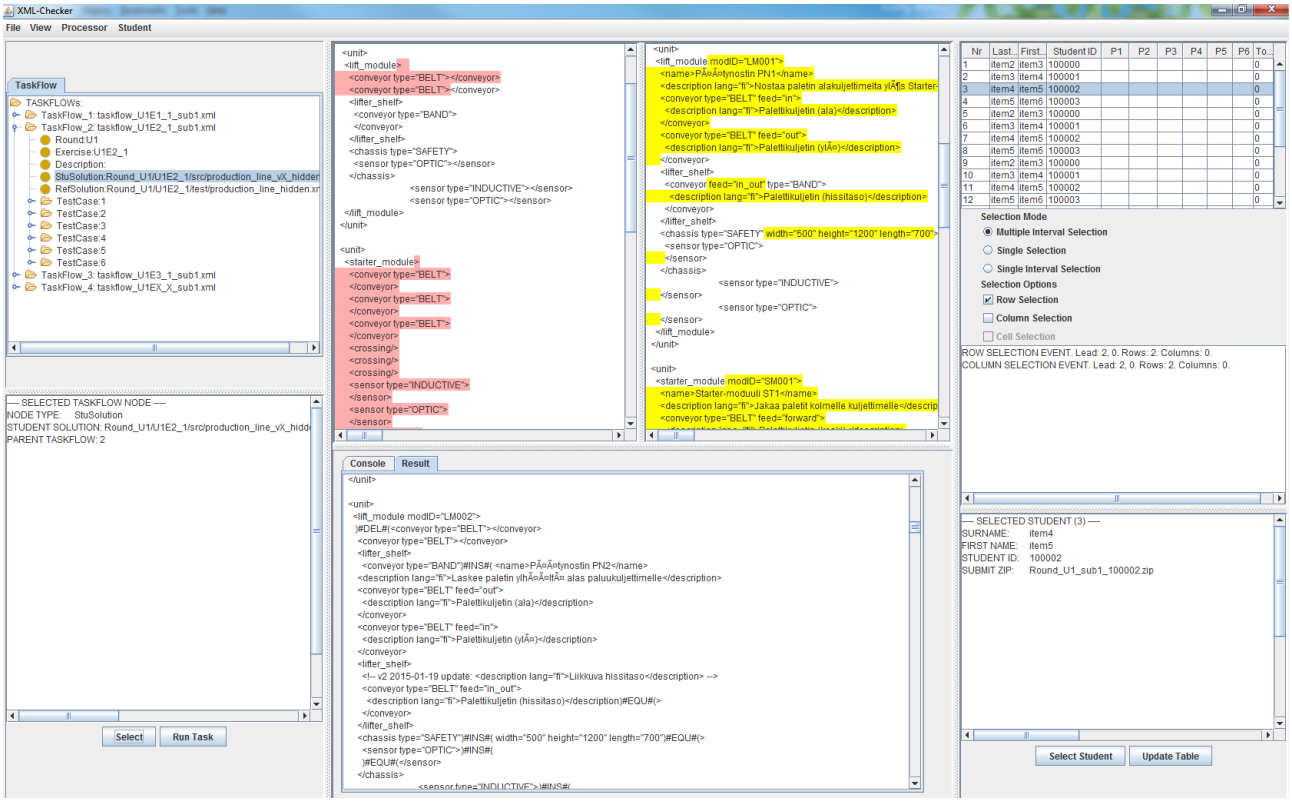
XChecker User Guide 1.0

2018-03-02

P. Aarnio



Contents

XChecker User Guide

Introduction

XChecker application is an automatic testing and grading tool for XML exercises. The basic idea is that there exists a correct reference solution for each exercise against which the student solution is compared. In most cases the comparison is not made directly, instead the same specified operations are targeted to both solutions and the results of these operations are compared. They should be similar for a correct student solution.

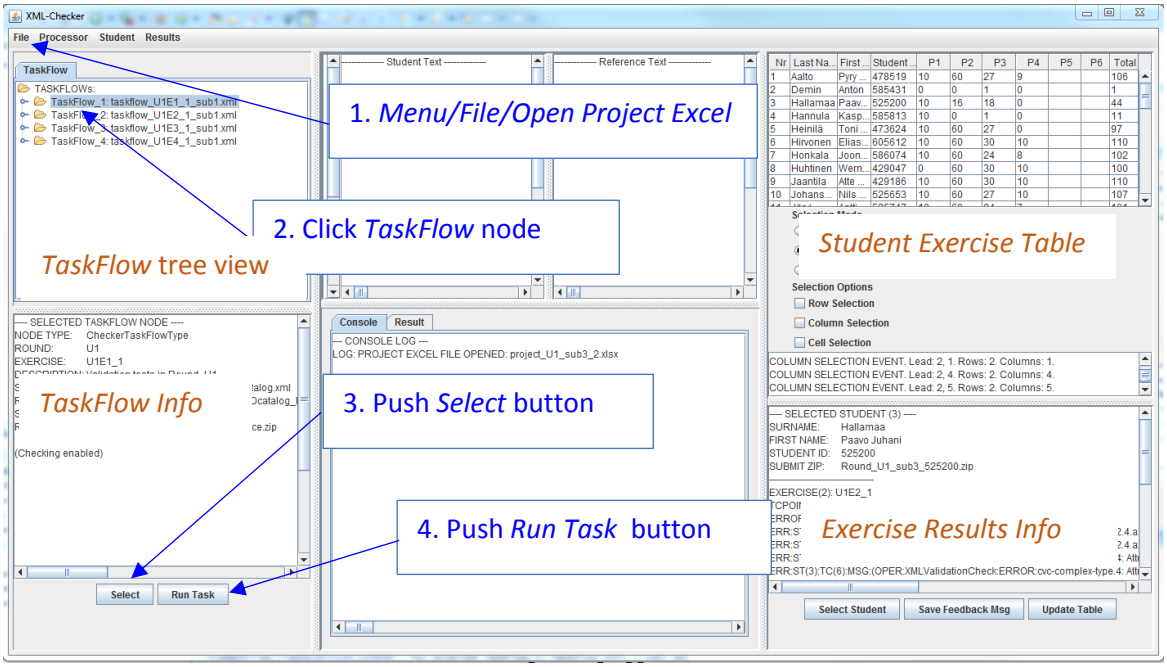


Figure 1.

Main Steps

Application user can start the checking process, when all prerequisite source files have been specified and deployed into the current project folder. First, the user of the tool opens XChecker project file, which causes an automatic loading of all student data and exercise testing rules into the application memory. Second, the user selects one *Taskflow* (tree view) containing the checking rules for some exercise and pushes '*Run Task*' button to start exercise solution correctness checking/testing process. Several test cases related to that exercise are executed for all student submitted solutions. Next, when this checking process is finished, the grading results of each student can be displayed in *Student Info* table view by pushing '*Update*' button. Furthermore, detailed checking results can be displayed in a text area under the Student Info table by selecting the marking/points of a single student from an exercise field column of *Student Info* table.

Exercise organization

In an exercise description student is asked to solve an xml task by writing or editing some type of xml document. All the solutions are textual files containing, for instance XML, XML transformation or XML Schema code.

The exercises are organized as rounds (U1-U3) each containing several exercises (U1E1_1-U1E4_1). Students solve the exercises of one round at the time and pack them in a zip package and submit (upload) it

to MyCourses (course information content management system) course pages. The teacher of the course downloads all the student submits into the submit folder of the current XChecker project. Each round should have its own project folder (if a round contains several submit turns, each of them should have a separate project folder)

Each project folder should contain a project spreadsheet file (excel) providing the main meta-data related to that round and submit turn.

XCProject Folder Structure

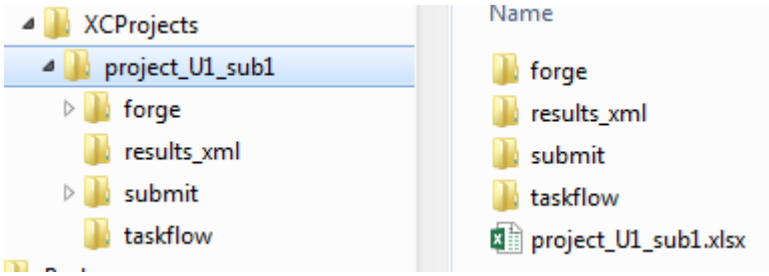


Fig. Project Folder containing the project file *project_U1_sub1.xlsx* and sub folders

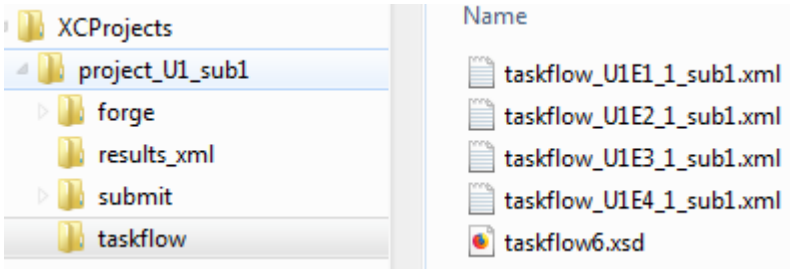


Fig. TaskFlow Folder contains a taskflow xml file for each exercise.
C:\Users\paarnio\Documents\Opetus\XML_2018\XCProjects\project_U1_sub1

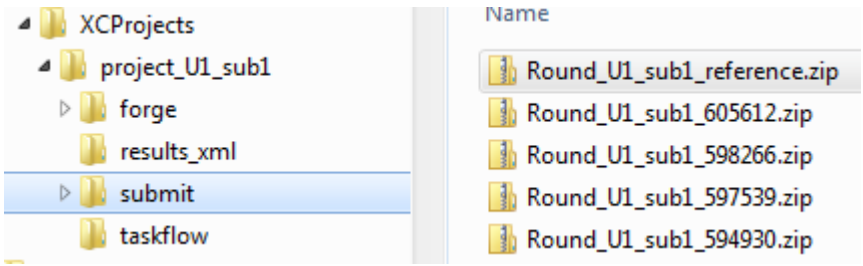
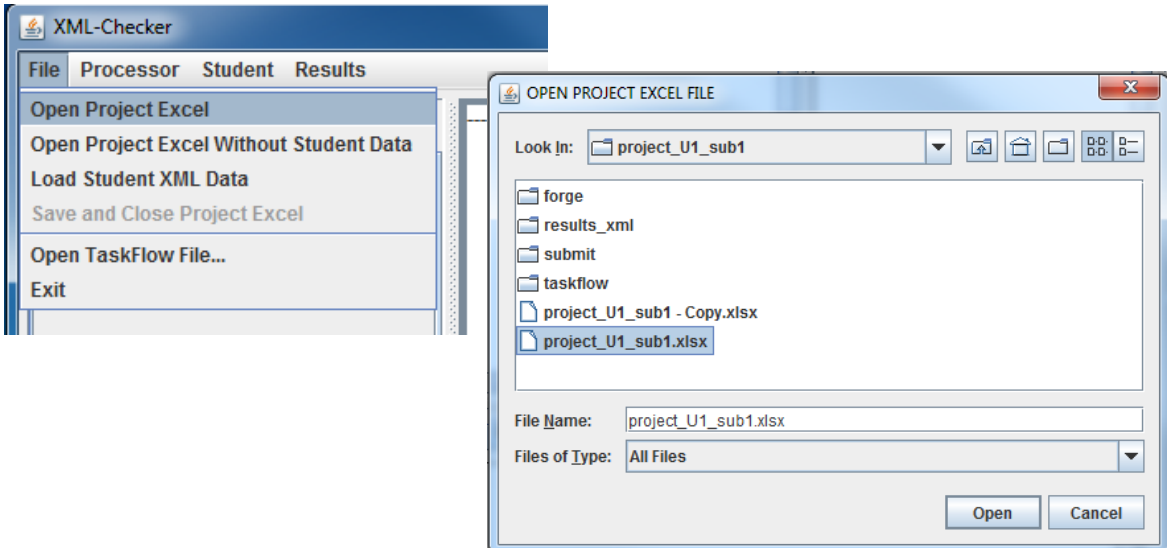
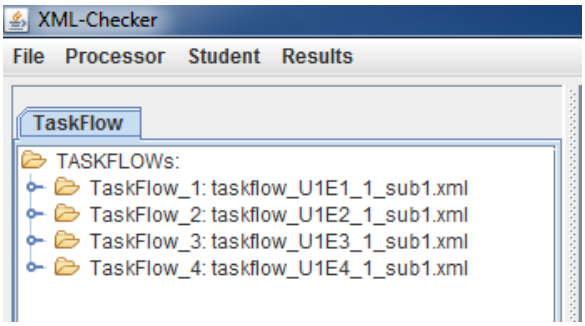


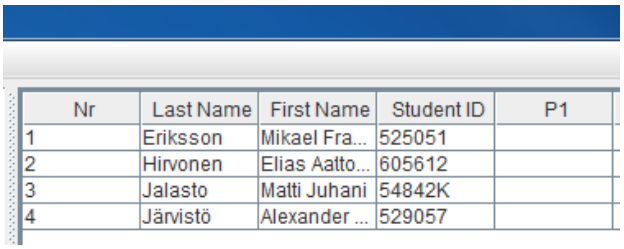
Fig. Submit Folder contains packed student solutions in zip files. This zip contains solutions of all the exercises of the specified round.

Step1: Opening Project

Application user can start the checking process, when all prerequisite source files have been specified and deployed into the current project folder. First, the user of the tool opens XChecker project file, which causes an automatic loading of all student data and exercise testing rules into the application memory (NOTE: Remember to Close Project Excel before Running XChecker)







Nr	Last Name	First Name	Student ID	P1
1	Eriksson	Mikael Fra...	525051	
2	Hirvonen	Elias Aatto...	605612	
3	Jalasto	Matti Juhani	54842K	
4	Järvistö	Alexander ...	529057	

Checking Existence and Structure of Student Submits

It is important to first check that all student submit packages exist and are zipped correctly. Otherwise the checking process stops, when it tries to access missing student solution file.

Click *StuSolution* node of the first TaskFlow and push Select so that *StuSolution* info is shown in bottom text area. Next click Menu/Student/Check Existence of Solutions

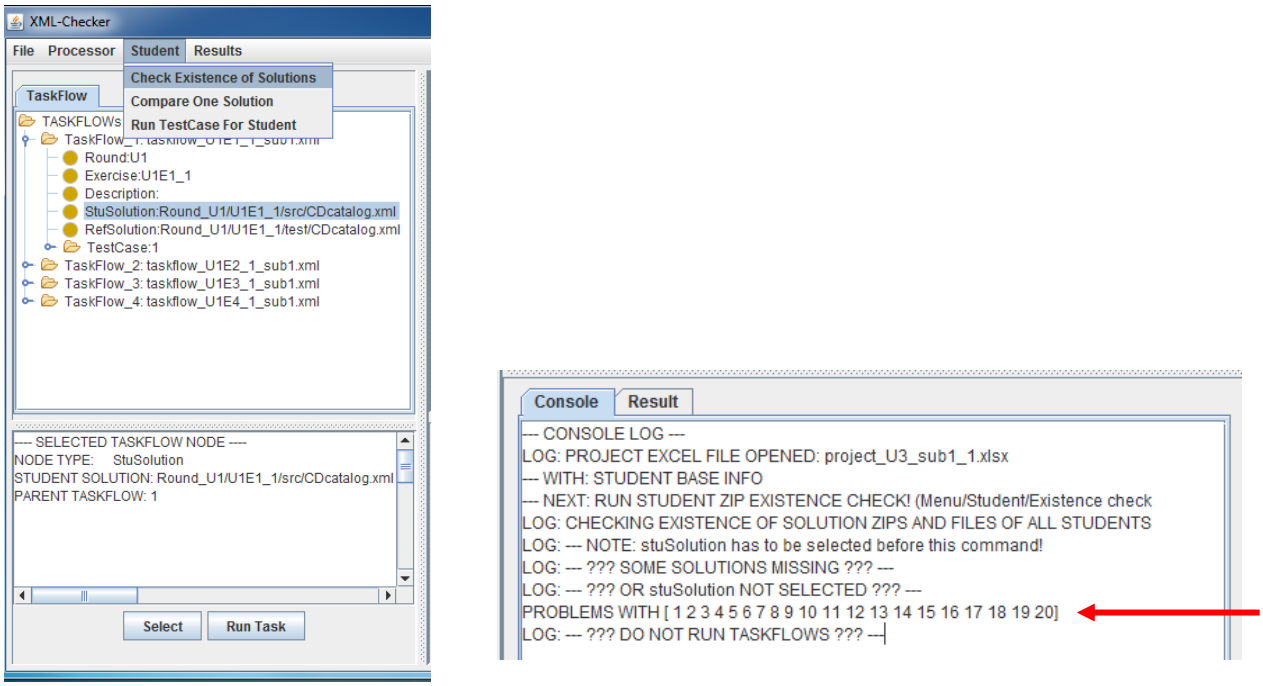


Figure. A) Checking Existence of Solutions. B) Forgot to move submit zip files 1-20 to project/submit folder

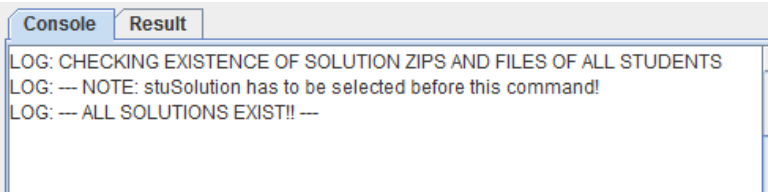


Figure. All solutions files exist and can be accessed

Step2: Selecting TaskFlow and running the checking process

Second, the user selects one *Taskflow* (tree view) containing the checking rules for some exercise. (By clicking one of the taskflows and pushing *Select* button below the taskFlow info text area). Next the user pushes *'Run Task'* button to start exercise solution correctness checking/testing process. Several test cases related to that exercise are executed for all student submitted solutions.

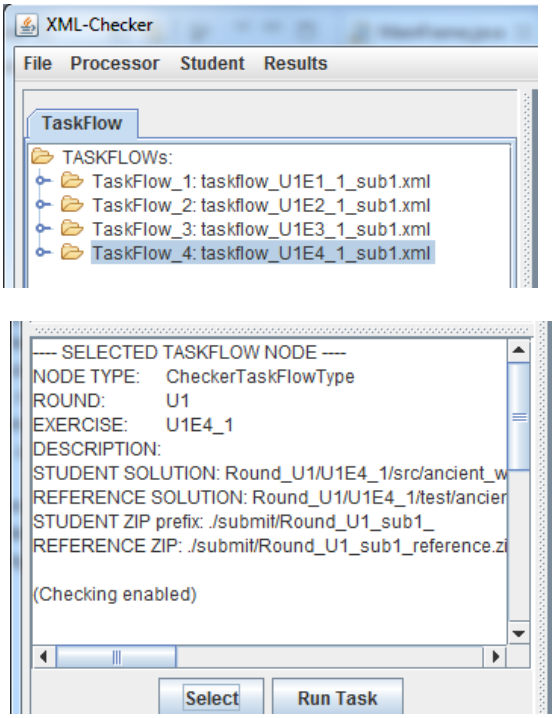
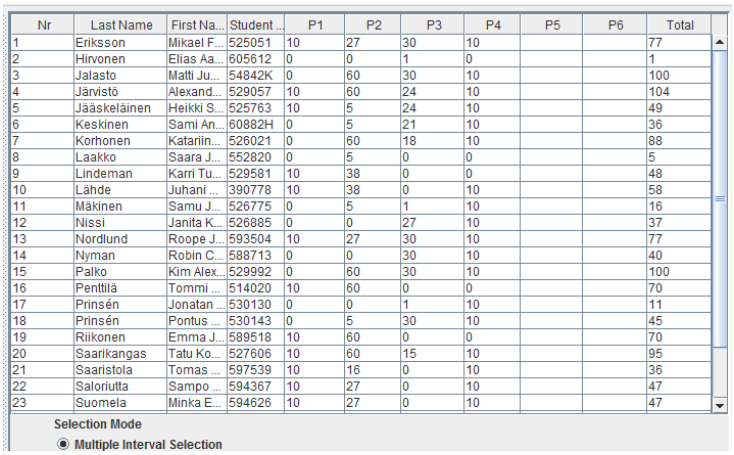


Figure.

Next, when this checking process is finished, the grading results of each student can be displayed in Student Info table view by pushing 'Update' button.



Nr	Last Name	First Na...	Student ...	P1	P2	P3	P4	P5	P6	Total
1	Eriksson	Mikael F...	525051	10	27	30	10			77
2	Hirvonen	Elias Aa...	605612	0	0	1	0			1
3	Jalasto	Matti Ju...	54842K	0	60	30	10			100
4	Järvistö	Alexand...	529057	10	60	24	10			104
5	Jääskeläinen	Heikki S...	525763	10	5	24	10			49
6	Keskinen	Sami An...	60882H	0	5	21	10			36
7	Korhonen	Katariin...	526021	0	60	18	10			88
8	Laakko	Saara J...	552820	0	5	0	0			5
9	Lindeman	Karri Tu...	529581	10	38	0	0			48
10	Lähde	Juhani ...	390778	10	38	0	10			58
11	Mäkinen	Samu J...	526775	0	5	1	10			16
12	Nissi	Janita K...	526885	0	0	27	10			37
13	Nordlund	Roope J...	593504	10	27	30	10			77
14	Nyman	Robin C...	588713	0	0	30	10			40
15	Palko	Kim Alex...	529992	0	60	30	10			100
16	Penttilä	Tommi ...	514020	10	60	0	0			70
17	Prinsén	Jonatan ...	530130	0	0	1	10			11
18	Prinsén	Pontus ...	530143	0	5	30	10			45
19	Riikonen	Emma J...	589518	10	60	0	0			70
20	Saarikangas	Tatu Ko...	527606	10	60	15	10			95
21	Saaristola	Tomas ...	597539	10	16	0	10			36
22	Saloriutta	Sampo ...	594367	10	27	0	10			47
23	Suomela	Minka E...	594626	10	27	0	10			47

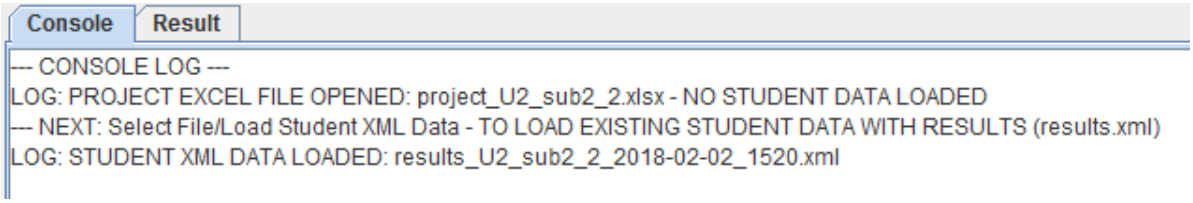
Figure. Results of taskflow execution are displayed on Student Results table after pushing *Update* button.

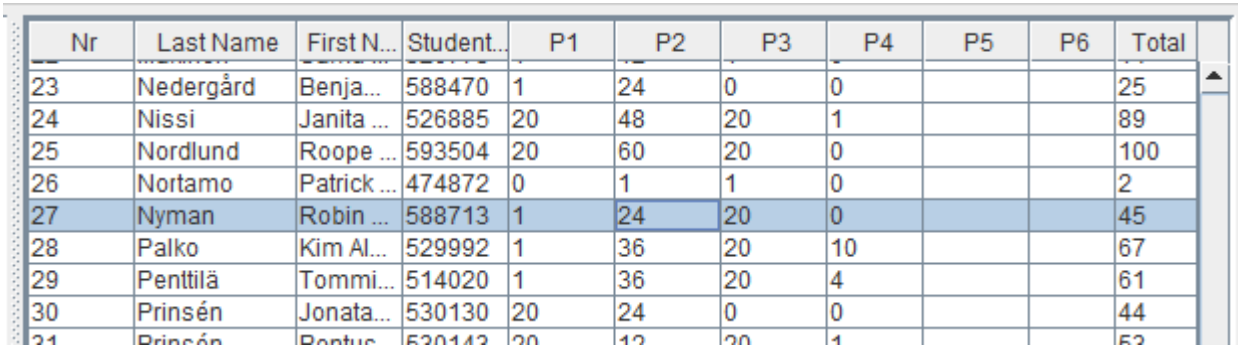
Post analysis of a student solution

When the results of exercises have been published, student can access his results and error messages from the results xml-file (e.g. using the provided XPath query expression and some online tool e.g. *XPathtester.com/xpath*) If the student does not fully understand the provided error messages, he can send a message to the teacher about the problem.

The teacher can load the previously generated results by first opening the corresponding project excel without student data and separately loading the existing student data xml related to that project:

- Menu: File/Open Project Excel without student data
- Menu: File/Load Student XML data





Nr	Last Name	First N...	Student...	P1	P2	P3	P4	P5	P6	Total
23	Nedergård	Benja...	588470	1	24	0	0			25
24	Nissi	Janita ...	526885	20	48	20	1			89
25	Nordlund	Roope ...	593504	20	60	20	0			100
26	Nortamo	Patrick ...	474872	0	1	1	0			2
27	Nyman	Robin ...	588713	1	24	20	0			45
28	Palko	Kim Al...	529992	1	36	20	10			67
29	Penttilä	Tommi...	514020	1	36	20	4			61
30	Prinsén	Jonata...	530130	20	24	0	0			44

Fig. Selecting the exercise P2 of student nro 27 for analysis. The student has got 24/60 points from this exercise.

Push '*Select Student*' button

```
ROW SELECTION EVENT. Lead: 26, 5. Rows: 26. Columns: 5.
COLUMN SELECTION EVENT. Lead: 26, 5. Rows: 26. Columns: 5.

---- SELECTED STUDENT (27) ----
SURNAME:      Nyman
FIRST NAME:   Robin Christian
STUDENT ID:   588713
SUBMIT ZIP:   Round_U2_sub2_588713.zip
--------------------------------
EXERCISE(2): U2E2_1
TCPOINTS: 1 11 0 12 0 0
ERRORS:
ERR:ST(27):TC(3):MSG:(EQU#(Name attributes of InternalElements:
chassis_)#DEL#(5)#INS#(1)#EQU#(
chassis_)#DEL#(5)#INS#(1)#EQU#(
conveyor_)#DEL#(3)#INS#(1)#EQU#(
conveyor_)#DEL#(3)#INS#(1)#EQU#(
conveyor_)#DEL#(3)#INS#(1)#EQU#(
conveyor_)#DEL#(3)#INS#(1)#EQU#(
conveyor_)#DEL#(4)#INS#(2)#EQU#(
```

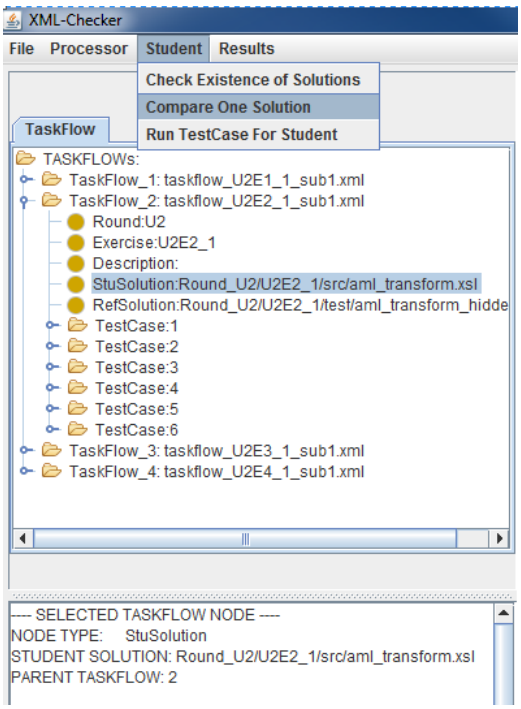
Fig. Testcases 3,5 and 6 have zero points. The reason for this can be read from the corresponding error messages. For instance, the Error message ERR:ST(1):TC(3):MSG: reveals that 'Name' attribute values are incorrect: the number suffix of the name is incorrect: *chassis_5* should be *chassis_1* etc.

When the cause of the error has been understood from the error messages (See Fig) description of the cause can be written as a Feedback message to the student after the FEEDBACK(2): line in student exercise data window. This feedback is saved in Student Info by pushing 'Save Feedback Msg' button (See fig.).

Comparing Student Solution with the reference solution

Student Solution can be compared with the reference solution by first selecting the student (Student information displayed in Student Info window) and then selecting the *StuSolution* node of an exercise in the TaskFlow tree window. Then select:

Menu: Student/Compare One Solution



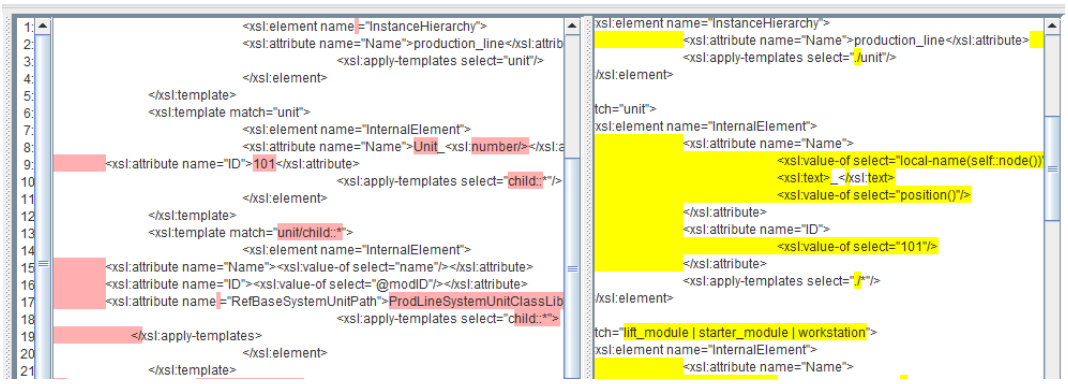
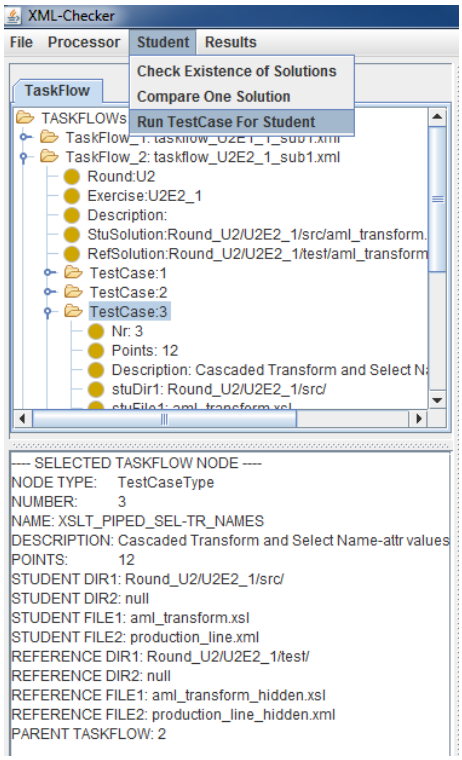


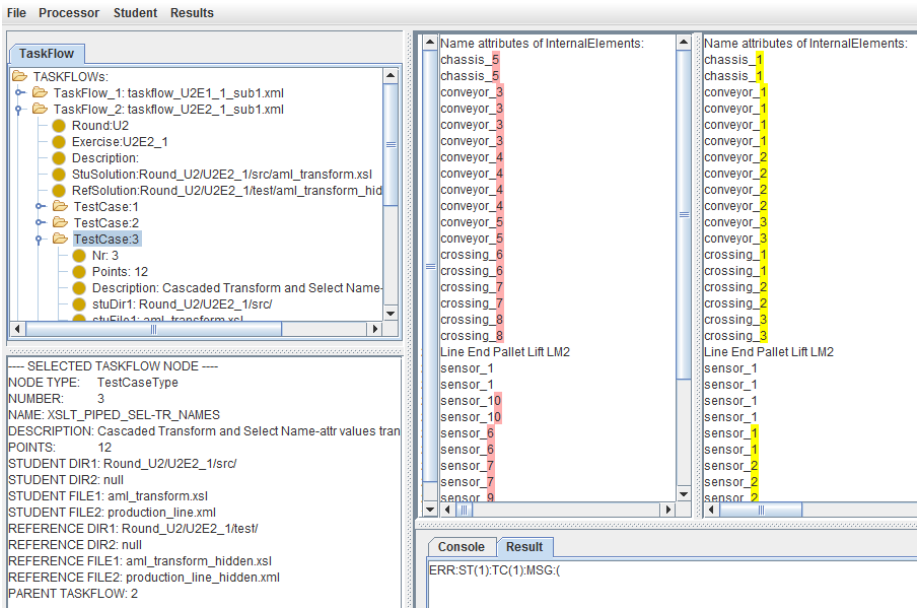
Fig. Comparing Student Solution with the reference solution. The text fraqments with RED background in the student's text do not exist in the reference text posibly indicating some sort of an error (DELETE). The text fraqments with YELLOW background in the reference text are missing from the student text (INSERT)

Rerunning a specific testcase for one student

Select the specific exercise of the student whose solution you want to recheck so that it's info is displayed in Student Info window. Then select the specific testcase node to be rerun from the Taskflow tree window. And finally select:

Menu: Student/Run TestCase for Student

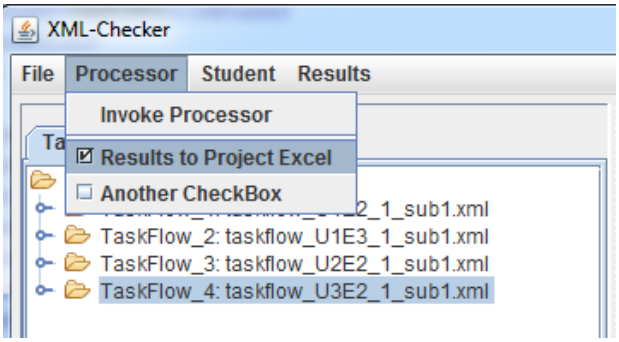




Saving Checking Results

Writing results directly into project excel

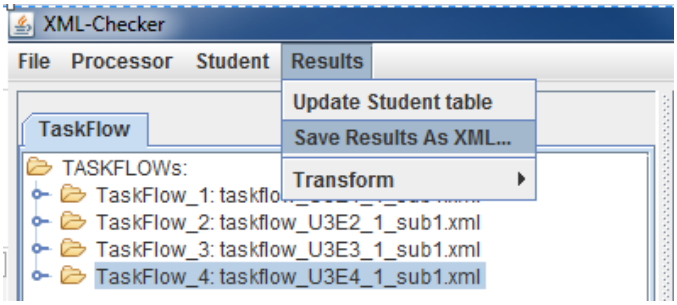
The results of the checking process can be written directly into the project excel by checking that option from the menu before task flow execution (*Menu/Processor/[x] Results to Project Excel*). The main info sheet of the project excel should have a reference to the corresponding workbook sheet for each exercise/taskflow to be tested in the project. WARNING: This should be done only for a limited set of student submits, because the error messages will be written into excel cells. The length of the error messages should also be limited by defining a filter option (-F DELETE) for the merge operation.

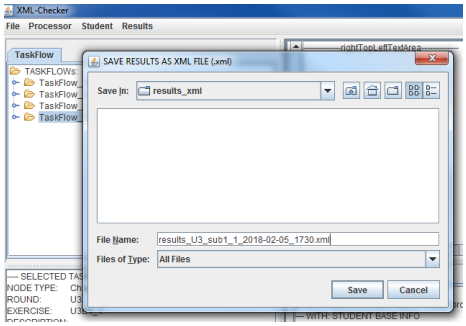


Figure

Saving Checking Results in XML format

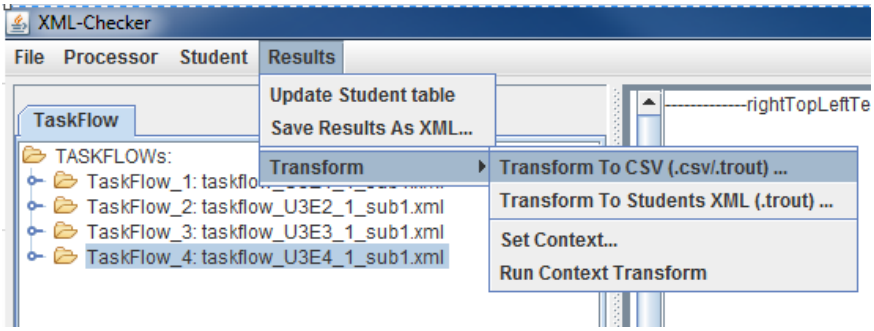
After running the checking process, results can be saved in a specific XML format defined by *student.xsd* schema. *StudentSubmits* JAXB object is filled with the student information including the results of all the exercises during taskflow execution. This object can be marshalled as xml instance document into projects *results_xml* folder by invoking *Menu:Results/Save Results As XML…* (C:\Users\paarnio\Documents\Opetus\XML_2018\XCProjects\project_U3_sub1\results_xml)

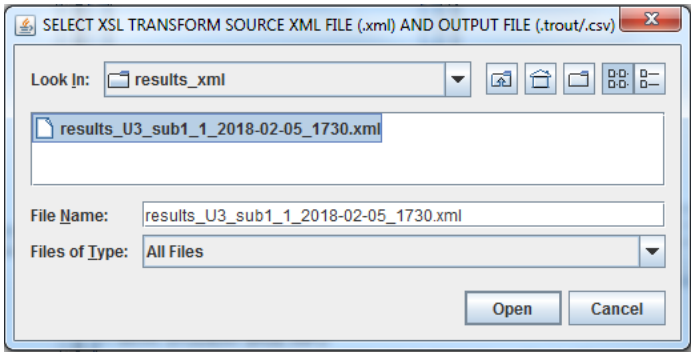


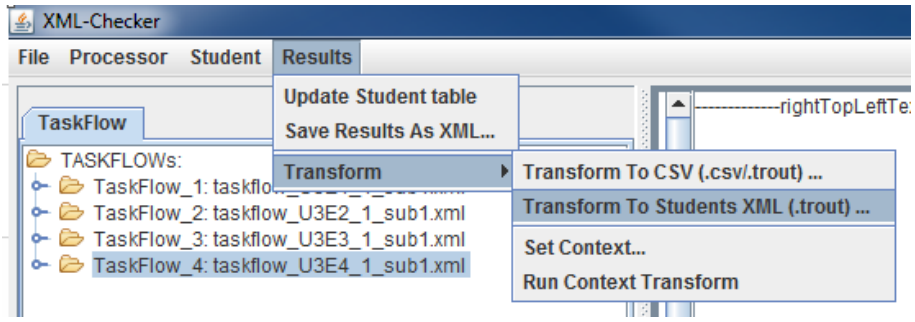


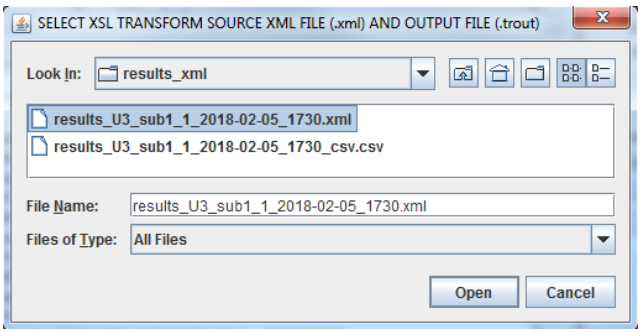
Figure

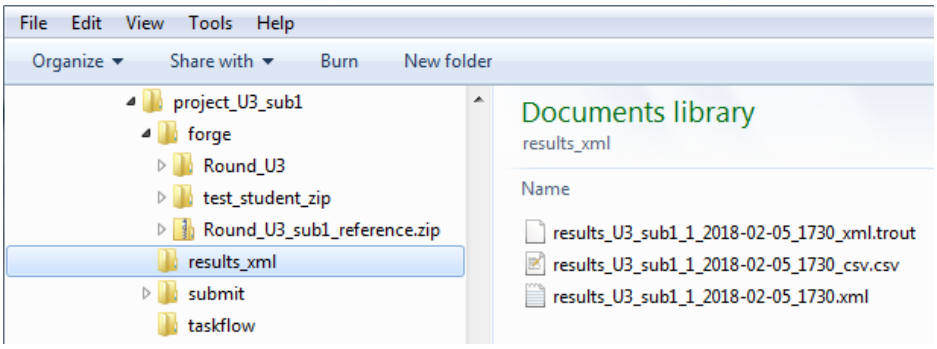
Transforming XML Results in CSV Format and Anonymous XML











Opening Result CSV file with Excel and Copy Paste to MyCourses Grading Page

results_U3_sub1_1_2018-02-05_1730_csv.csv

	A	B	C	D	E
7					
8			FORCE MAX 100		
9					
10		StudentId	Points_U3	Feedback_U3	
11		605612	94	EXID:U3E1_1 POINTS:20 TCPS: +1 +19 EXID:U3E2_1	
12		529057	110	EXID:U3E1_1 POINTS:20 TCPS: +1 +19 EXID:U3E2_1	
13		592071	101	EXID:U3E1_1 POINTS:20 TCPS: +1 +19 EXID:U3E2_1	
14		602945	71	EXID:U3E1_1 POINTS:20 TCPS: +1 +19 EXID:U3E2_1	
15		526021	94	EXID:U3E1_1 POINTS:20 TCPS: +1 +19 EXID:U3E2_1	
16		605793	101	EXID:U3E1_1 POINTS:20 TCPS: +1 +19 EXID:U3E2_1	

XChecker Project Excel File

Downloading Student Submits and Filling Student Data Excel

The submitted student solution zip files are downloaded from MyCourses into the submit sub-folder of the current project folder. At the same time, student rows are marked in student data excel for record keeping purposes (C:\Users\...\Documents\Opetus\XML_2018\opiskelijat\XML_Student_data_2018-01-04.xlsx)

Copying Student Data and Generated Submit File Names to Project Excel

The information of those marked students are copied from the student data excel into the project excel ZipFiles sheet. (C:\Users\...\Documents\Opetus\XML_2018\XCProjects\project_U3_sub1\ project_U3_sub1_1.xlsx: Sheet: ZipFiles_U3_sub1)

J	K	L	M	N	O
SET					
Accept				PO: TEXT	
	0	Surname	First N	StudentId	Submit Zip
1	1	Hirvonen	Elias A	605612	Round_U3_sub1_605612.zip
1	2	Järvistö	Alexai	529057	Round_U3_sub1_529057.zip
1	3	Kaisanlahti	Anna I	592071	Round_U3_sub1_592071.zip
1	4	Karvinen	Samul	602945	Round_U3_sub1_602945.zip
1	5	Korhonen	Katari	526021	Round_U3_sub1_526021.zip
1	6	Leinonen	Mikae	605793	Round_U3_sub1_605793.zip
1	7	Lindeman	Karri T	529581	Round_U3_sub1_529581.zip
1	8	Lähde	Juhan	390778	Round_U3_sub1_390778.zip
1	9	Nedergård	Benjai	588470	Round_U3_sub1_588470.zip
1	10	Nissi	Janita	526885	Round_U3_sub1_526885.zip
1	11	Nordlund	Roope	593504	Round_U3_sub1_593504.zip
1	12	Nyman	Robin	588713	Round_U3_sub1_588713.zip
1	13	Penttilä	Tomm	514020	Round_U3_sub1_514020.zip
1	14	Prinsén	Jonata	530130	Round_U3_sub1_530130.zip
1	15	Prinsén	Pontu	530143	Round_U3_sub1_530143.zip
1	16	Riikonen	Emma	589518	Round_U3_sub1_589518.zip
1	17	Tarvo	Noora	521372	Round_U3_sub1_521372.zip
1	18	Vaarnamo	Juha P	594930	Round_U3_sub1_594930.zip
1	19	Westerholm	Lauri \	530868	Round_U3_sub1_530868.zip
1	20	Öz	Emreh	598266	Round_U3_sub1_598266.zip
0	20				
0	20				

Figure. ZipFiles_U3_sub1 sheet of project excel project_U3_sub1_1.xlsx

	A	B	C
1	MainInfo	StudentSheet	ZipFiles_U3_sub1
2		ZipFileCount	20
3			
4		TASKFLOW ROWS	
5		KeyFirstRow	9
6		KeyLastRow	18
7			
8			
9		Key	Value
10		TASKFLOW	U3E1_1
11		TaskFlowXmlFile	taskflow/taskflow_U3E1_1_sub1.xml
12		ZipFilesSheet	ZipFiles_U3_sub1
13		ZipFileCount	20
14		StudentZipFileFolder	submit/
15		ReferenceZipFileFolder	submit/
16		ReferenceZipFile	Round_U3_sub1_reference.zip
17		ResultsSheet	Results_U3E1_1
18			

Figure. MainInfo sheet contains important metadata for the XChecker project

Exercise Solution Checking Process

The testing process of each exercise is described in a so called *TaskFlow* xml file. This XML instance data is marshalled as a set of JAXB objects for the application during the runtime providing control information for the testing process execution. Each exercise is tested in a task cycle process (*TaskCycleProcessor*) that loops over all the submitted student solution xml/text files following the rules defined in the *CheckerTaskFlowType* object specific to that exercise. A taskflow consists of one or more test cases (*TestCaseType*) each of which consists of three flows (*FlowType*): *student flow, reference flow* and *merge flow*. Flows consist of *operations* that can be cascaded with an *interim data flow pipe*.

Operations of the student flow are targeted to student's solution text file and reference flow operations to the reference solution text file. The results of these two flows are compared with each other in the merge flow phase. If the results of the operations of a test case are equal to those of the reference solution, the student solution is considered as correct with respect of the current test case and the specified test case points are allocated to this student (*StudentType*). If they are not equal, error messages of all the applied operations are provided as feedback information to the student.

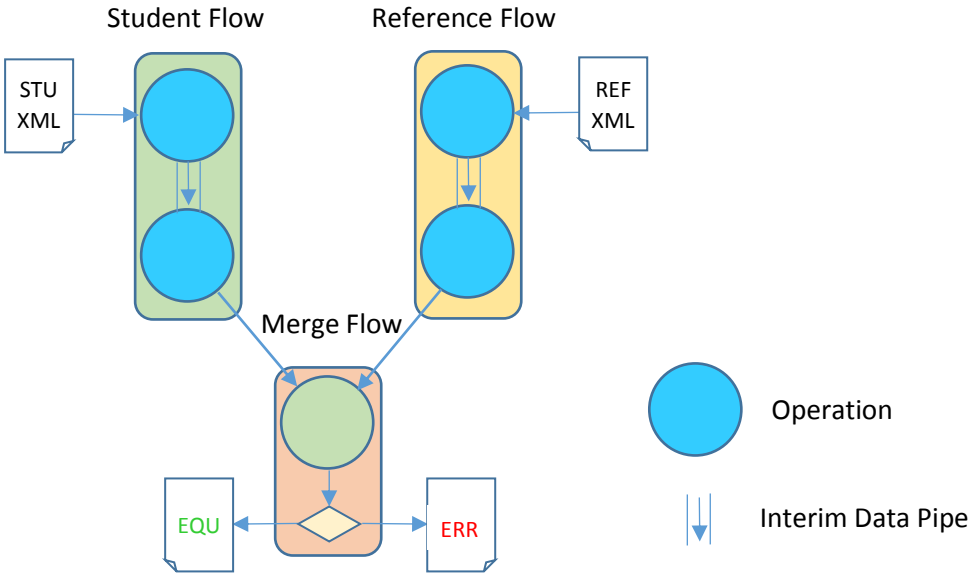


Figure. The structure of testing process (task flow) of one test case.

The following Student and Reference flow operation types have been implemented:

1. *XSLTransform*
2. *XSDValidation*
3. *XMLWellFormed*
4. *ReadTxtContent*
5. *DirectStringOutOper*
6. *(XPathQuery)*

Merge flow operation types are *StringCompare* and *DirectStringOutOper*

TaskFlow XML File

For each exercise there must be a *Taskflow.xml* file (e.g. taskflow_U2E2_1_sub1.xml) in *taskflow* sub folder, which contains the specifications of all the test cases to be executed during the testing task cycle by *TaskCycleProcessor.* The structure of this document is specified in *taskflow.xsd* schema. This xml instance data is marshalled as JAXB object (*CheckerTaskFlowType*) during the application runtime.

CheckerTaskFlow root element

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <!--  2018-01-26 taskflow_U2E2_1_sub1.xml validated by taskflow8.xsd -->
3   <checkerTaskFlow>
4     <round>U2</round>
5     <exercise>U2E2_1</exercise>
6     <description></description>
7     <stuSolution>Round_Ux/U2E2_1/src/aml_transform.xsl</stuSolution>
8     <refSolution>Round_Ux/U2E2_1/test/aml_transform_hidden.xsl</refSolution>
9     <stuZip>./submit/Round_Ux_sub1_</stuZip>
10    <refZip>./submit/Round_Ux_sub1_reference.zip</refZip>
11    <testCase number="1">
12      <name>WELLFORMED CHECK</name>
13      <description>The Well Formed testcase</description>
```

Fig. CHECKERTASKFLOW: A fragment of *taskflow_U2E2_1_sub1.xml* specifying how the solutions of exercise U2E2_1 is processed. The element <*stuSolution*> defines the path to the student's solution file in student's solution package zip file. <*refSolution*> contains the corresponding information for the reference solution. (stu = student; ref=reference).

TestCase element and studentFlow flow type

```
11    <testCase number="1">
12      <name>WELLFORMED CHECK</name>
13      <description>The Well Formed testcase</description>
14      <points>1</points>
15      <stuDir1>Round_Ux/U2E2_1/src/</stuDir1>
16      <stuFile1>aml_transform.xsl</stuFile1>
17      <refDir1>Round_Ux/U2E2_1/test/</refDir1>
18      <refFile1>aml_transform_hidden.xsl</refFile1>
19      <flow>
20        <type>studentFlow</type>
21        <name>studentFlow1</name>
22        <description>The student flow</description>
23        <operation>
24          <type>XMLWellFormed</type>
25          <name>XMLWellFormed1</name>
26          <par1>stuDir1/stuFile1</par1>
27          <return>stuC001</return>
28        </operation>
29      </flow>
30      <flow>
31        <type>referenceFlow</type>
```

Fig. TESTCASE: The first test case is xml *well-formedness* check. One XCP point is given for a correct solution as defined in <*points*> element. <stuDir1> element contains the path to a student directory in student's solution package zip file that contains the source file defined in <stuFile1> element. <refDir1> and <refFile1> elements define the corresponding information for the reference solution in reference zip file. The first flow of this test case is studentFlow, in which XMLWellFormed operation is targeted to the student

file referenced by the operation parameter <par1>stuDir1/stuFile1</par1>. The result of the operation is returned in student channel *stuC001*.

ReferenceFlow flow type

```
29      </flow>
30      <flow>
31        <type>referenceFlow</type>
32        <name>referenceFlow1</name>
33        <description>Direct Answer for XMLWellFormed question</description>
34        <operation>
35          <type>DirectStringOutOper</type>
36          <name>DirectStringOutOper1</name>
37          <par1>WELLFORMED</par1>
38          <return>refC001</return>
39        </operation>
40      </flow>
41      <flow>
42        <type>mergeFlow</type>
```

Fig. REFERENCEFLOW: The next flow type is *referenceFlow* which normally specifies the same operation as the previous *studentFlow*, but now targeting the corresponding file in reference solution. However, in this case the well-formedness of the reference solution is not checked, but instead the direct expected result of this operation is given as an argument for the parameter *<par1>WELLFORMED</par1>*. This operation type *DirectStringOutOper* is used for performance reasons; because we know that the reference solution is well-formed xml, the operation is unnecessary and the expected known result can be directly written into the result channel (*refC001* in this case).

MergeFlow flow type

```
40      </flow>
41      <flow>
42        <type>mergeFlow</type>
43        <name>mergeFlow2</name>
44        <description>Merging student and reference flow</description>
45        <operation>
46          <type>StringCompare</type>
47          <name>StringCompare1</name>
48          <par1>stuC001</par1>
49          <par2>refC001</par2>
50          <return>merC002</return>
51        </operation>
52      </flow>
53    </testCase>
```

Fig. MERGEFLOW: The last flow type is always *mergeFlow*, in which the results of student and reference flows are compared. The basic operation is *StringCompare* that calculates the difference between the text strings in channels specified in *<par1>* and *<par2>* elements i.e. the strings in student *stuC001* and reference *refC001* channels

Cascaded operations and Interim Pipe

```
53  </testCase>
54  <testCase number="2">
55    <name>XSLT_PIPED_SEL-TR_COUNT</name>
56    <description>Cascaded Transform and Select count of elements transform.</description>
57    <points>11</points>
58    <stuDir1>Round_Ux/U2E2_1/src/</stuDir1>
59    <stuFile1>aml_transform.xsl</stuFile1>
60    <stuFile2>production_line.xml</stuFile2><!-- not used -->
61    <refDir1>Round_Ux/U2E2_1/test/</refDir1>
62    <refFile1>aml_transform_hidden.xsl</refFile1>
63    <refFile2>production_line_hidden.xml</refFile2>
64    <flow>
65      <type>studentFlow</type>
66      <name>studentFlow1</name>
67      <operation>
68        <type>XSLTransform</type>
69        <name>XSLTransform1</name>
70        <par1>stuDir1/stuFile1</par1>
71        <par2>refDir1/refFile2</par2>
72        <return>TO_INTERIM_PIPE</return>
73      </operation>
74       <operation>
75        <type>XSLTransform</type>
76        <name>Pipeline XSLTransform1</name>
77        <par1>refDir1:pipe_tr_sel_count_hidden.xsl</par1>
78        <par2>FROM_INTERIM_PIPE</par2>
79        <return>stuC001</return>
80      </operation>
81    </flow>
82    <flow>
```

File reference as parameter

1. Transform result to data pipe

File name as parameter

2. Transform target from data pipe

Fig. INTERIM_PIPE: Two or more operations can be cascaded using a special data pipe called INTERIM_PIPE. The result of the first operation can be piped by defining the return channel to be *TO_INTERIM_PIPE*. In this case one of the parameters of the next operation must be defined as *FROM_INTERIM_PIPE*. In this example, an XSLTransform defined in student's solution file aml_transform.xsl is first targeted to reference file production_line_hidden.xml. Next, the correctness of the result is checked by counting the number of some specific elements using another XSLTransform given as an argument: *<par1>refDir1:pipe_tr_sel_count_hidden.xsl</par1>* (NOTE: file name can be given directly in <par> element.) (See project_U_example: taskflow_U2E2_1_sub1.xml: testcase 2) (NOTE: INTERIM_PIPE implemented for XSLTransform operation (v.1.0))

Operation Option element

Operations can have <opt> elements as children.

- Option –W (write) defines the base filename for a file where the results or error messages of the current operation are written. The actual generated file name is concatenated with student, exercise and test case indexes for name uniqueness e.g. *ST4_ExU3E2_1_TC3_xslt_transform_1.txt*. These operation text files are saved in *results_xml* sub folder of the current project.
- Option –F (filter) is used for filtering the text difference results of StringCompare operation in mergeFlow. Possible values are 'ALL', 'DELETE_INSERT', 'DELETE' and 'INSERT'

(See option usage examples in *project_U_example: taskflow_U3E2_1_sub1.xml: testcase 3*).

```
109    <flow>
110      <type>studentFlow</type>
111      <name>studentFlow1</name>
112      <operation>
113        <type>XSLTransform</type>
114        <name>XSLTransform1</name>
115        <par1>refDir1/refFile1</par1>
116        <par2>stuDir1/stuFile2</par2>
117        <opt>-W xslt_transform_1.txt</opt>
118        <return>stuC001</return>
119      </operation>
120    </flow>
```

Fig. OPTION: <opt>-W xslt_transform_1.txt</opt> generates a XSLTransform result file for each student. For example *ST4_ExU3E2_1_TC3_xslt_transform_1.txt*, where ST4 = 4. student; TC3=3. test case.

```
131    <flow>
132      <type>mergeFlow</type>
133      <name>mergeFlow1</name>
134      <operation>
135        <type>StringCompare</type>
136        <name>StringCompare1</name>
137        <par1>stuC001</par1>
138        <par2>refC001</par2>
139        <opt>-F DELETE_INSERT</opt>
140        <opt>-W differences.txt</opt>
141        <return>merC001</return>
142      </operation>
143    </flow>
144  </testCase>
```

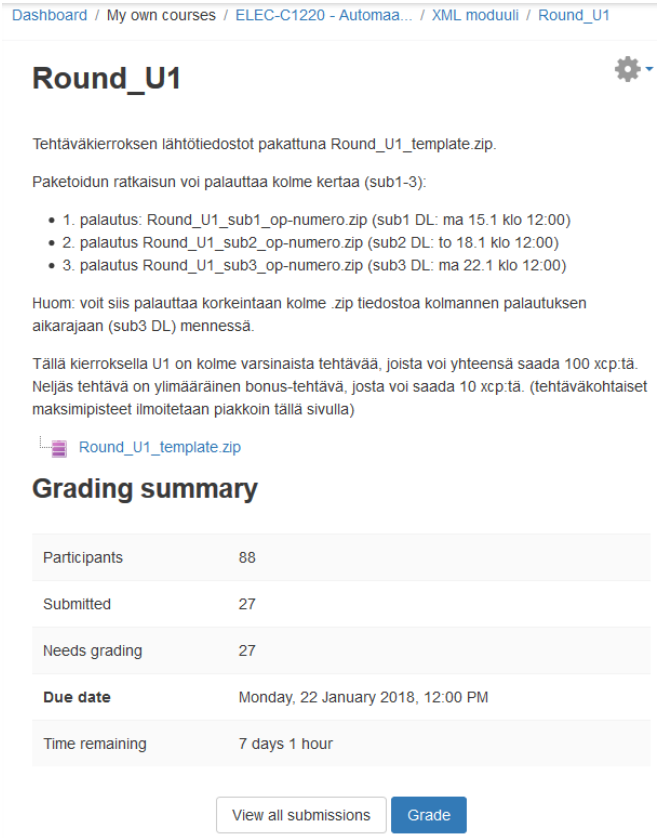
Fig. OPTION: *<opt>-F DELETE_INSERT</opt>* limiting the text difference results to DELETE and INSERT differences, i.e. EQUAL string fragments are not included into difference messages that will be written also to differences.txt files, because of the option element <opt>-W differences.txt</opt>. For example: e.g. *ST3_ExU3E2_1_TC3_differences.txt.*

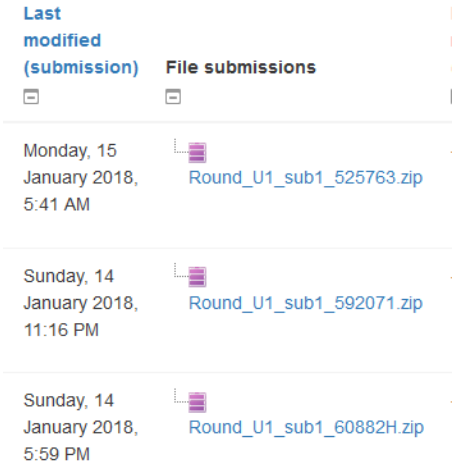
ANNEX A

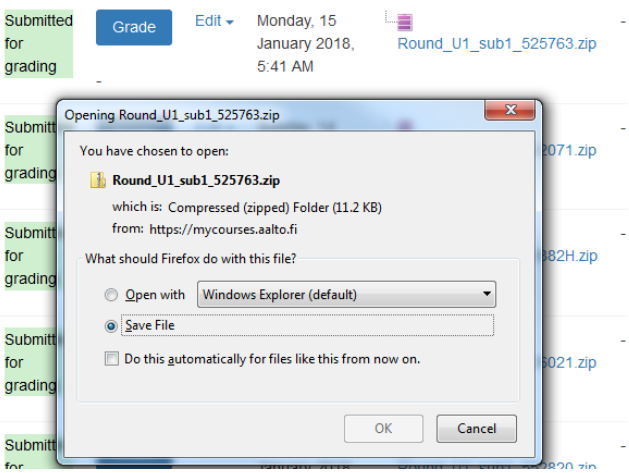
ANNEX A: Working with MyCourses CMS

Preparing source information for XChecker checking process

MyCourses Assignment Page Round_U1







Downloading Student Submits and Filling Student Data Excel

C:\Users\paarnio\Documents\Opetus\XML_2018\opiskelijat XML_Student_data_2018-01-04.xlsx

	A	B	C	D	E
1					
2		Round_U3		5.2.2018	
3					
4		CURRENT PICK			
5		SUB1			
6		Current Loading			
7		5.2.2018			
8		klo 16:50			
9		Submit	New Order		
10		EXISTS	0	StudentID	Lastname
11		0	0	478519	Aalto
12		0	0	428307	Aito
13		0	0	585224	Aumasto
14		0	0	61480T	Blomberg
15		0	0	585431	Demin
16		0	0	525051	Eriksson
17		0	0	525200	Hallamaa
18		0	0	585813	Hannula
19		0	0	473624	Heinilä
20		1	1	605612	Hirvonen
21		0	1	586074	Honkala

	R	S	T	U
		PICKING AUTOMATION		
		For Project Excel		
		Lastname	Firstname	StudentID
	1	Hirvonen	Elias Aatto	605612
	2	Järvistö	Alexander	529057
	3	Kaisanlahti	Anna Hele	592071
	4	Karvinen	Samuli Vil	602945
	5	Korhonen	Katariina I	526021
	6	Leinonen	Mikael Al	605793
	7	Lindeman	Karri Tuuk	529581
	8	Lähde	Juhani Mil	390778
	9	Nedergård	Benjamin	588470
	10	Nissi	Janita Kris	526885
	11	Nordlund	Roope Juh	593504

X	Y	Z	AA	AB
SUBMIT ZIP NAME GENERATION				
				Submit zip
Round_U3_sub1_	605612	.zip		Round_U3_sub1_605612.zip
Round_U3_sub1_	529057	.zip		Round_U3_sub1_529057.zip
Round_U3_sub1_	592071	.zip		Round_U3_sub1_592071.zip
Round_U3_sub1_	602945	.zip		Round_U3_sub1_602945.zip
Round_U3_sub1_	526021	.zip		Round_U3_sub1_526021.zip
Round_U3_sub1_	605793	.zip		Round_U3_sub1_605793.zip
Round_U3_sub1_	529581	.zip		Round_U3_sub1_529581.zip
Round_U3_sub1_	390778	.zip		Round_U3_sub1_390778.zip
Round_U3_sub1_	588470	.zip		Round_U3_sub1_588470.zip
Round_U3_sub1_	526885	.zip		Round_U3_sub1_526885.zip
Round_U3_sub1_	593504	.zip		Round_U3_sub1_593504.zip

Copying Student Data and Generated Submit File Names to Project Excel

C:\Users\paarnio\Documents\Opetus\XML_2018\XCProjects\project_U3_sub1
project_U3_sub1_1.xlsx
Sheet: ZipFiles_U3_sub1

J	K	L	M	N	O
SET					
Accept				PO: TEXT	
	0	Surname	First N	StudentId	Submit Zip
1	1	Hirvonen	Elias A	605612	Round_U3_sub1_605612.zip
1	2	Järvistö	Alexa	529057	Round_U3_sub1_529057.zip
1	3	Kaisanlahti	Anna I	592071	Round_U3_sub1_592071.zip
1	4	Karvinen	Samul	602945	Round_U3_sub1_602945.zip
1	5	Korhonen	Katari	526021	Round_U3_sub1_526021.zip
1	6	Leinonen	Mikae	605793	Round_U3_sub1_605793.zip
1	7	Lindeman	Karri T	529581	Round_U3_sub1_529581.zip
1	8	Lähde	Juhan	390778	Round_U3_sub1_390778.zip
1	9	Nedergård	Benja	588470	Round_U3_sub1_588470.zip
1	10	Nissi	Janita	526885	Round_U3_sub1_526885.zip
1	11	Nordlund	Roope	593504	Round_U3_sub1_593504.zip
1	12	Nyman	Robin	588713	Round_U3_sub1_588713.zip
1	13	Penttilä	Tomm	514020	Round_U3_sub1_514020.zip
1	14	Prinsén	Jonat	530130	Round_U3_sub1_530130.zip
1	15	Prinsén	Pontu	530143	Round_U3_sub1_530143.zip
1	16	Riikonen	Emma	589518	Round_U3_sub1_589518.zip
1	17	Tarvo	Noora	521372	Round_U3_sub1_521372.zip
1	18	Vaarnamo	Juha P	594930	Round_U3_sub1_594930.zip
1	19	Westerholm	Lauri \	530868	Round_U3_sub1_530868.zip
1	20	Öz	Emreh	598266	Round_U3_sub1_598266.zip
0	20				
0	20				

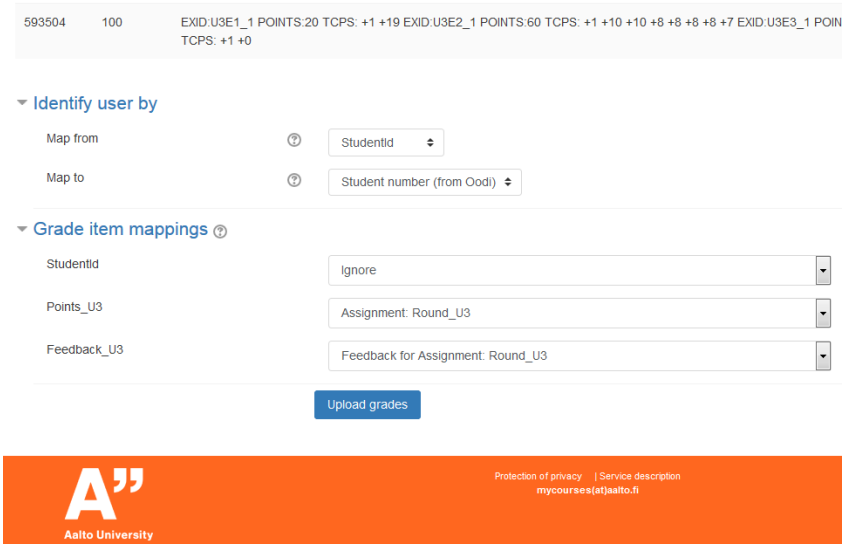
	A	B	C
1	MainInfo	StudentSheet	ZipFiles_U3_sub1
2		ZipFileCount	20
3			
4		TASKFLOW ROWS	
5		KeyFirstRow	9
6		KeyLastRow	18
7			
8			
9		Key	Value
10		TASKFLOW	U3E1_1
11		TaskFlowXmlFile	taskflow/taskflow_U3E1_1_sub1.xml
12		ZipFilesSheet	ZipFiles_U3_sub1
13		ZipFileCount	20
14		StudentZipFileFolder	submit/
15		ReferenceZipFileFolder	submit/
16		ReferenceZipFile	Round_U3_sub1_reference.zip
17		ResultsSheet	Results_U3E1_1
18			

Fig. Sheet MainInfo

ANNEX B

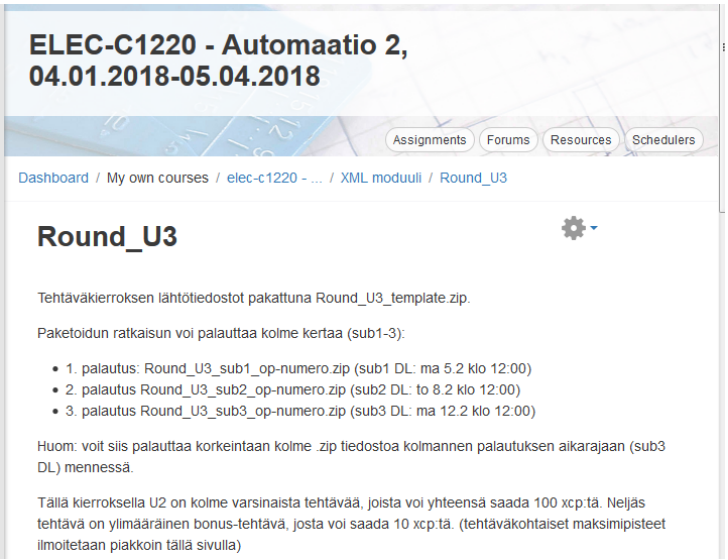
ANNEX B: Publishing XChecker results in MyCourses CMS

Checking Results to MyCourses CMS



Uploading Anonymous Student Results to MyCourses Round_U3 Assignment Page

Rename results_U3_sub1_1_2018-02-05_1730_xml.trout as results_U3_sub1_2018-02-05_d1.xml and upload it to Round_U3 assignment page.



XChecker tarkistimen tulokset

Tarkistimen opiskelijakohtaiset väliaikatulokset ovat luettavissa oheisista xml-dokumenteista:
sub1: results_U3_sub1_2018-02-05_d1.xml.

Lataa dokumentti (tai copy-paste) esim. online xpath testeriin: http://www.xpathtester.com/xpath ja kohdista siihen seuraava xpath kysely saadaksesi omat tuloksesi näkyviin:

1. Kaikki tulokset XPATH: //student[@studentId='op.numero']
2. Tehtävän 'U3E2_1' tulokset XPATH: //student[@studentId= 'op.numero ']/exercise[@exerciseId='U3E2_1']
3. Tehtävän 'U3E2_1' kokonaispisteet XPATH: sum(//student[@studentId= 'op.numero ']/exercise[@exerciseId='U3E2_1']/pointsOfTestCases)
4. Virheilmoitukset XPATH: //student[@studentId= 'op.numero ']/exercise[@exerciseId='U3E2_1'] /errorsOfTestCases

XChecker-tarkistimen antamien tulosten ja virheilmoitusten rakenne ja koodien tulkinta on selitetty *Toimintaohjeet*-kansion *XChecker_palaute.pdf* dokumentissa. Se sisältää myös esimerkkejä tyypillisistä virheistä.

Tehtävien maksimipisteet

Round	Exercise	TC points	Total
U3	U3E1_1	1+19	20
U3	U3E2_1	1+10+10+8+8+8+8+7	60
U3	U3E3_1	1+7+6+6	20
U3	U3E4_1	1+9	10

results_U3_sub1_2018-02-05_d1.xml
Round_U3_template.zip

Grading summary

Participants	88
Submitted	20
Needs grading	20
Due date	Monday, 12 February 2018, 12:00 PM
Time remaining	6 days 17 hours

View all submissions | Grade

ANNEX C

ANNEX C: CheckerTaskFlow XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!-- taskflow8.xsd version=8 2018-02-04 -->
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
     2018-02-04 taskflow8.xsd : XChecker taskflow schema for siima.model.jaxb.checker.taskflow
    </xsd:documentation>
  </xsd:annotation>

  <xsd:element name="checkerTaskFlow" type="checkerTaskFlowType"/>

  <xsd:complexType name="checkerTaskFlowType">
    <xsd:sequence>
      <xsd:element name="round" type="xsd:string"/>
      <xsd:element name="exercise" type="xsd:string"/>
      <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="stuSolution" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="refSolution" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="stuZip" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="refZip" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="testCase" type="testCaseType" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="testCaseType">
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="points" type="xsd:string"/>
      <xsd:element name="stuDir1" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="stuDir2" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="stuFile1" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="stuFile2" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="refDir1" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="refDir2" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="refFile1" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="refFile2" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="flow" type="flowType" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="number" type="xsd:int"/>
  </xsd:complexType>
```

Figure. First fragment of schema file taskflow8.xsd (continues in next page..)

```
  <xsd:complexType name="flowType">
    <xsd:sequence>
      <xsd:element name="type" type="xsd:string"/>
      <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="inChannel" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="outChannel" type="xsd:string"  minOccurs="0" maxOccurs="1"/>
      <xsd:element name="operation" type="operationType" minOccurs="0" maxOccurs="unbounded"/>
     </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="operationType">
    <xsd:sequence>
      <xsd:element name="type" type="xsd:string"/>
      <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="par1" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="par2" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="paramValueList" type="paramValueListType" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="opt" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="return" type="xsd:string"/>
     </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="paramValueListType">
    <xsd:sequence>
      <xsd:element name="paramList" type="paramlistType" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="valueList" type="valuelistType" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>

   <xsd:simpleType name="paramlistType">
    <xsd:list itemType="xsd:string"/>
  </xsd:simpleType>

  <xsd:simpleType name="valuelistType">
    <xsd:list itemType="xsd:string"/>
  </xsd:simpleType>

</xsd:schema>
```

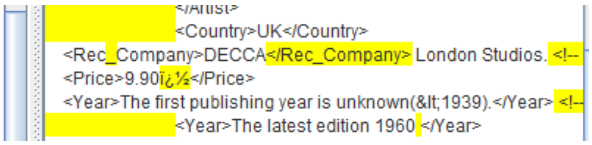
Figure. (continues..) Second fragment of schema file taskflow8.xsd

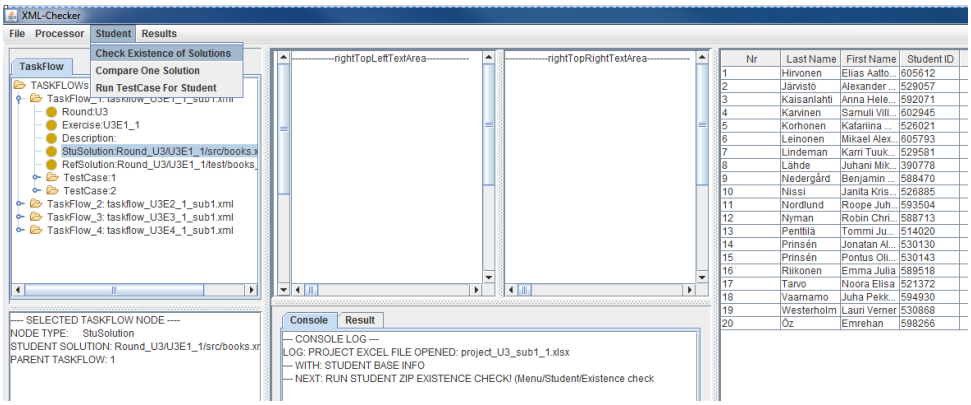
ANNEX D

ANNEX D: StudentSubmits XML Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!-- students3.xsd version=3 2018-01-08 feedback added -->
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
    students3.xsd : XML checker student info schema for siima.model.jaxb.checker.student
    </xsd:documentation>
  </xsd:annotation>

  <xsd:element name="studentSubmits">
    <xsd:annotation>
        <xsd:documentation>Root-element of the student schema. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
        <xsd:sequence>
        <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="referenceZip" type="xsd:string" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="student" type="studentType" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="submitId" type="xsd:string"/><!-- e.g. U1_sub2 -->
    </xsd:complexType>
  </xsd:element>

   <xsd:complexType name="studentType">
    <xsd:sequence>
      <xsd:element name="surname" type="xsd:string"/>
      <xsd:element name="firstname" type="xsd:string"/>
      <xsd:element name="round" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="submitZip" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="exercise" type="exerciseType" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="studentId" type="xsd:string"/>
  </xsd:complexType>

  <xsd:complexType name="exerciseType">
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="round" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="pointsOfTestCases" type="xsd:int" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="resultsOfTestCases" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="errorsOfTestCases" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="feedback" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="exerciseId" type="xsd:string"/>
  </xsd:complexType>

</xsd:schema>
```

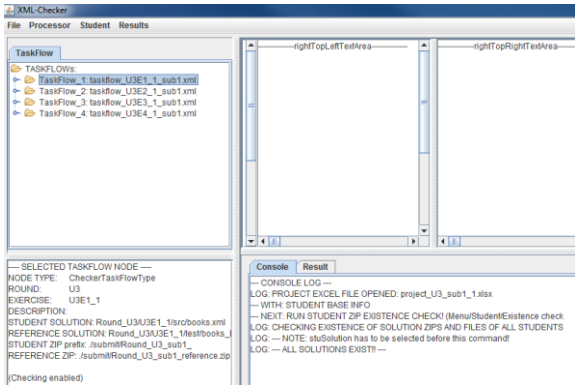
SCREENSHOTS:

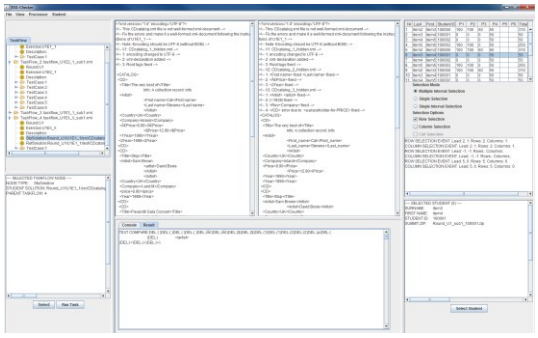




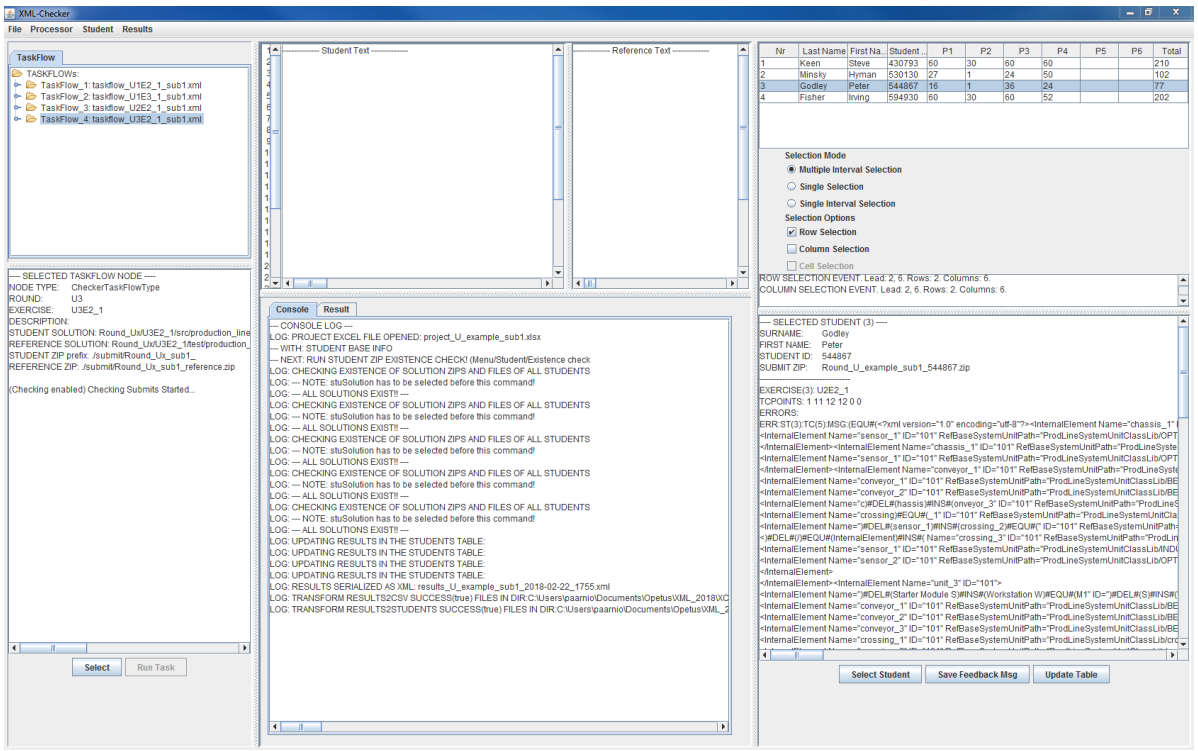
Nr	Last Name	First Name	Student ID
1	Hirvonen	Elias Aalto...	605612
2	Järvistö	Alexander ...	529057
3	Kaisanlahti	Anna Hele...	592071
4	Karvinen	Samuli Vill...	602945
5	Korhonen	Katariina ...	526021
6	Leinonen	Mikael Alex...	605793
7	Lindeman	Karri Tuuk...	529581
8	Lähde	Juhani Mik...	390778
9	Nedergård	Benjamin ...	588470
10	Nissi	Janita Kris...	526885
11	Nordlund	Roope Juh...	593504
12	Nyman	Robin Chri...	588713
13	Pentitilä	Tommi Ju...	514020
14	Prinsén	Jonatan Al...	530130
15	Prinsén	Pontus Oli...	530143
16	Riikonen	Emma Julia	589518
17	Tarvo	Noora Elisa	521372
18	Vaarnamo	Juha Pekk...	594930
19	Westerholm	Lauri Verner	530868
20	Öz	Emrehan	598266

Nr	Last Name	First ...	Student ID	P1	P2	P3	P4	P5	P6	Total
1	Hirvonen	Elias ...	605612	20	60	13	1			94
2	Järvistö	Alexa...	529057	20	60	20	10			110
3	Kaisanlahti	Anna ...	592071	20	60	20	1			101
4	Karvinen	Samu...	602945	20	37	13	1			71
5	Korhonen	Katari...	526021	20	60	13	1			94
6	Leinonen	Mikae...	605793	20	60	20	1			101
7	Lindeman	Karri ...	529581	1	60	12	1			74
8	Lähde	Juhan...	390778	20	50	13	1			84
9	Nedergård	Benja...	588470	20	60	13	1			94
10	Nissi	Janita...	526885	20	60	20	10			110
11	Nordlund	Roop...	593504	20	60	20	1			101
12	Nyman	Robin...	588713	20	24	13	1			58
13	Pentitilä	Tom...	514020	20	52	8	1			81
14	Prinsén	Jonat...	530130	20	50	13	1			84
15	Prinsén	Pontu...	530143	20	60	13	1			94
16	Riikonen	Emm...	589518	20	52	20	1			93
17	Tarvo	Noora...	521372	20	24	13	1			58





project_U_example_sub1



TASKFLOWs:
- TaskFlow_1: taskflow_U1E2_1_sub1.xml
- TaskFlow_2: taskflow_U1E3_1_sub1.xml
- TaskFlow_3: taskflow_U2E2_1_sub1.xml
- TaskFlow_4: taskflow_U3E2_1_sub1.xml

```
---- SELECTED TASKFLOW NODE ----
NODE TYPE:     CheckerTaskFlowType
ROUND:         U3
EXERCISE:      U3E2_1
DESCRIPTION:
STUDENT SOLUTION: Round_Ux/U3E2_1/src/production_line
REFERENCE SOLUTION: Round_Ux/U3E2_1/test/production_
STUDENT ZIP prefix: ./submit/Round_Ux_sub1_
REFERENCE ZIP: ./submit/Round_Ux_sub1_reference.zip

(Checking enabled) Checking Submits Started...
```

```
--- CONSOLE LOG ---
LOG: PROJECT EXCEL FILE OPENED: project_U_example_sub1.xlsx
--- WITH: STUDENT BASE INFO
--- NEXT: RUN STUDENT ZIP EXISTENCE CHECK! (Menu/Student/Existence check
LOG: CHECKING EXISTENCE OF SOLUTION ZIPS AND FILES OF ALL STUDENTS
LOG: --- NOTE: stuSolution has to be selected before this command!
LOG: --- ALL SOLUTIONS EXIST!! ---
LOG: CHECKING EXISTENCE OF SOLUTION ZIPS AND FILES OF ALL STUDENTS
LOG: --- NOTE: stuSolution has to be selected before this command!
LOG: --- ALL SOLUTIONS EXIST!! ---
LOG: CHECKING EXISTENCE OF SOLUTION ZIPS AND FILES OF ALL STUDENTS
LOG: --- NOTE: stuSolution has to be selected before this command!
LOG: --- ALL SOLUTIONS EXIST!! ---
LOG: CHECKING EXISTENCE OF SOLUTION ZIPS AND FILES OF ALL STUDENTS
LOG: --- NOTE: stuSolution has to be selected before this command!
LOG: --- ALL SOLUTIONS EXIST!! ---
LOG: CHECKING EXISTENCE OF SOLUTION ZIPS AND FILES OF ALL STUDENTS
LOG: --- NOTE: stuSolution has to be selected before this command!
LOG: --- ALL SOLUTIONS EXIST!! ---
LOG: UPDATING RESULTS IN THE STUDENTS TABLE:
LOG: UPDATING RESULTS IN THE STUDENTS TABLE:
LOG: UPDATING RESULTS IN THE STUDENTS TABLE:
LOG: UPDATING RESULTS IN THE STUDENTS TABLE:
LOG: RESULTS SERIALIZED AS XML: results_U_example_sub1_2018-02-22_1755.xml
LOG: TRANSFORM RESULTS2CSV SUCCESS(true) FILES IN DIR:C:\Users\paarnio\Documents\Opetus\XML_2018\XC
LOG: TRANSFORM RESULTS2STUDENTS SUCCESS(true) FILES IN DIR:C:\Users\paarnio\Documents\Opetus\XML_2
```

Nr	Last Name	First Na...	Student ...	P1	P2	P3	P4	P5	P6	Total
1	Keen	Steve	430793	60	30	60	60			210
2	Minsky	Hyman	530130	27	1	24	50			102
3	Godley	Peter	544867	16	1	36	24			77
4	Fisher	Irving	594930	60	30	60	52			202

Selection Mode

● Multiple Interval Selection

○ Single Selection

○ Single Interval Selection

Selection Options

☑ Row Selection

☐ Column Selection

☐ Cell Selection

```
ROW SELECTION EVENT. Lead: 2, 6. Rows: 2. Columns: 6.
COLUMN SELECTION EVENT. Lead: 2, 6. Rows: 2. Columns: 6.
```

```
---- SELECTED STUDENT (3) ----
SURNAME:      Godley
FIRST NAME:   Peter
STUDENT ID:   544867
SUBMIT ZIP:   Round_U_example_sub1_544867.zip
--------------------------------
EXERCISE(3): U2E2_1
TCPOINTS: 1 11 12 12 0 0
ERRORS:
ERR:ST(3):TC(5):MSG:(EQU#(<?xml version="1.0" encoding="utf-8"?><InternalElement Name="chassis_1" I
<InternalElement Name="sensor_1" ID="101" RefBaseSystemUnitPath="ProdLineSystemUnitClassLib/OPT
</InternalElement><InternalElement Name="chassis_1" ID="101" RefBaseSystemUnitPath="ProdLineSyste
<InternalElement Name="sensor_1" ID="101" RefBaseSystemUnitPath="ProdLineSystemUnitClassLib/OPT
</InternalElement><InternalElement Name="conveyor_1" ID="101" RefBaseSystemUnitPath="ProdLineSyste
<InternalElement Name="conveyor_1" ID="101" RefBaseSystemUnitPath="ProdLineSystemUnitClassLib/BE
<InternalElement Name="conveyor_2" ID="101" RefBaseSystemUnitPath="ProdLineSystemUnitClassLib/BE
<InternalElement Name="c)#DEL#(hassis)#INS#(onveyor_3" ID="101" RefBaseSystemUnitPath="ProdLineS
<InternalElement Name="crossing)#EQU#(_1" ID="101" RefBaseSystemUnitPath="ProdLineSystemUnitCla
<InternalElement Name=")#DEL#(sensor_1)#INS#(crossing_2)#EQU#(" ID="101" RefBaseSystemUnitPath=
<)#DEL#(/)#EQU#(InternalElement)#INS#( Name="crossing_3" ID="101" RefBaseSystemUnitPath="ProdLin
<InternalElement Name="sensor_1" ID="101" RefBaseSystemUnitPath="ProdLineSystemUnitClassLib/INDU
<InternalElement Name="sensor_2" ID="101" RefBaseSystemUnitPath="ProdLineSystemUnitClassLib/OPT
</InternalElement>
</InternalElement><InternalElement Name="unit_3" ID="101">
<InternalElement Name=")#DEL#(Starter Module S)#INS#(Workstation W)#EQU#(M1" ID=")#DEL#(S)#INS#(
<InternalElement Name="conveyor_1" ID="101" RefBaseSystemUnitPath="ProdLineSystemUnitClassLib/BE
<InternalElement Name="conveyor_2" ID="101" RefBaseSystemUnitPath="ProdLineSystemUnitClassLib/BE
<InternalElement Name="conveyor_3" ID="101" RefBaseSystemUnitPath="ProdLineSystemUnitClassLib/BE
<InternalElement Name="crossing_1" ID="101" RefBaseSystemUnitPath="ProdLineSystemUnitClassLib/cro
```