1. ATL Transformation Example: Table to TabularHTML

The Table to TabularHTML example describes a transformation from a Table model to an HTML file containing HTML tables.

1.1. Transformation Overview

The aim of this transformation is to generate an HTML file from the input data contained in a Table model. This file can next be read with an HTML viewer or Internet browser.

The generation of the output HTML file is realised by a first transformation from Table to HTML. Next, an extraction to an HTML file is necessary. This is done by applying a transformation from HTML to XML and the use of the XML extractor to obtain an XML file, which will be renamed into an .html file.

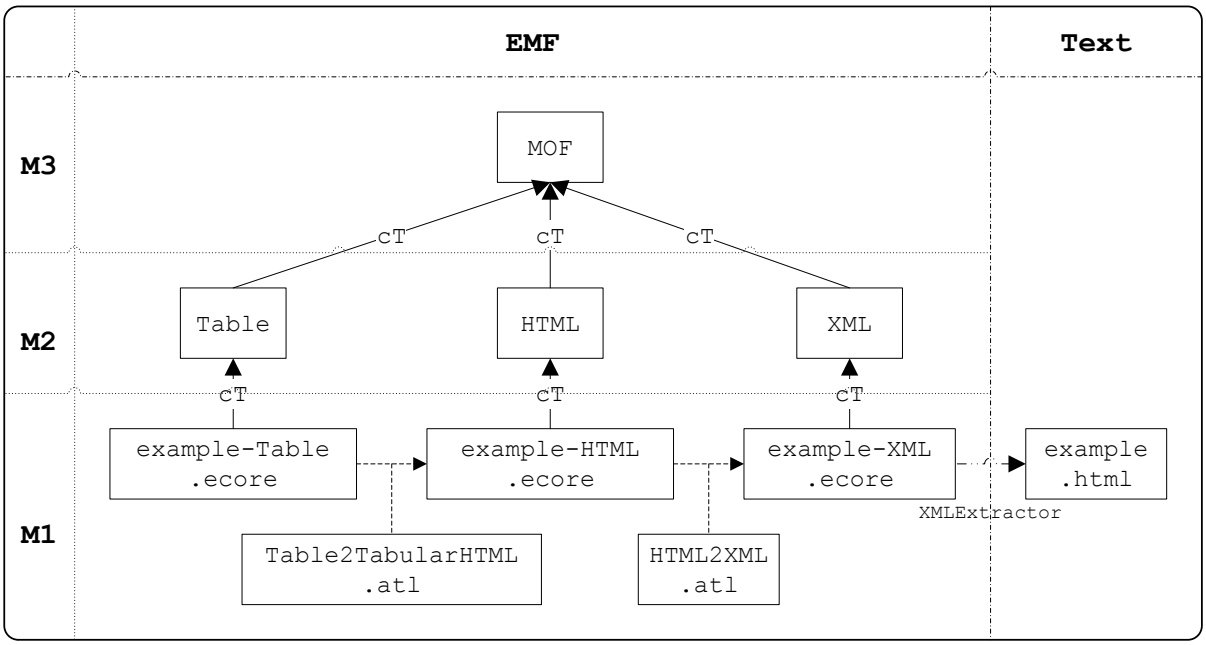
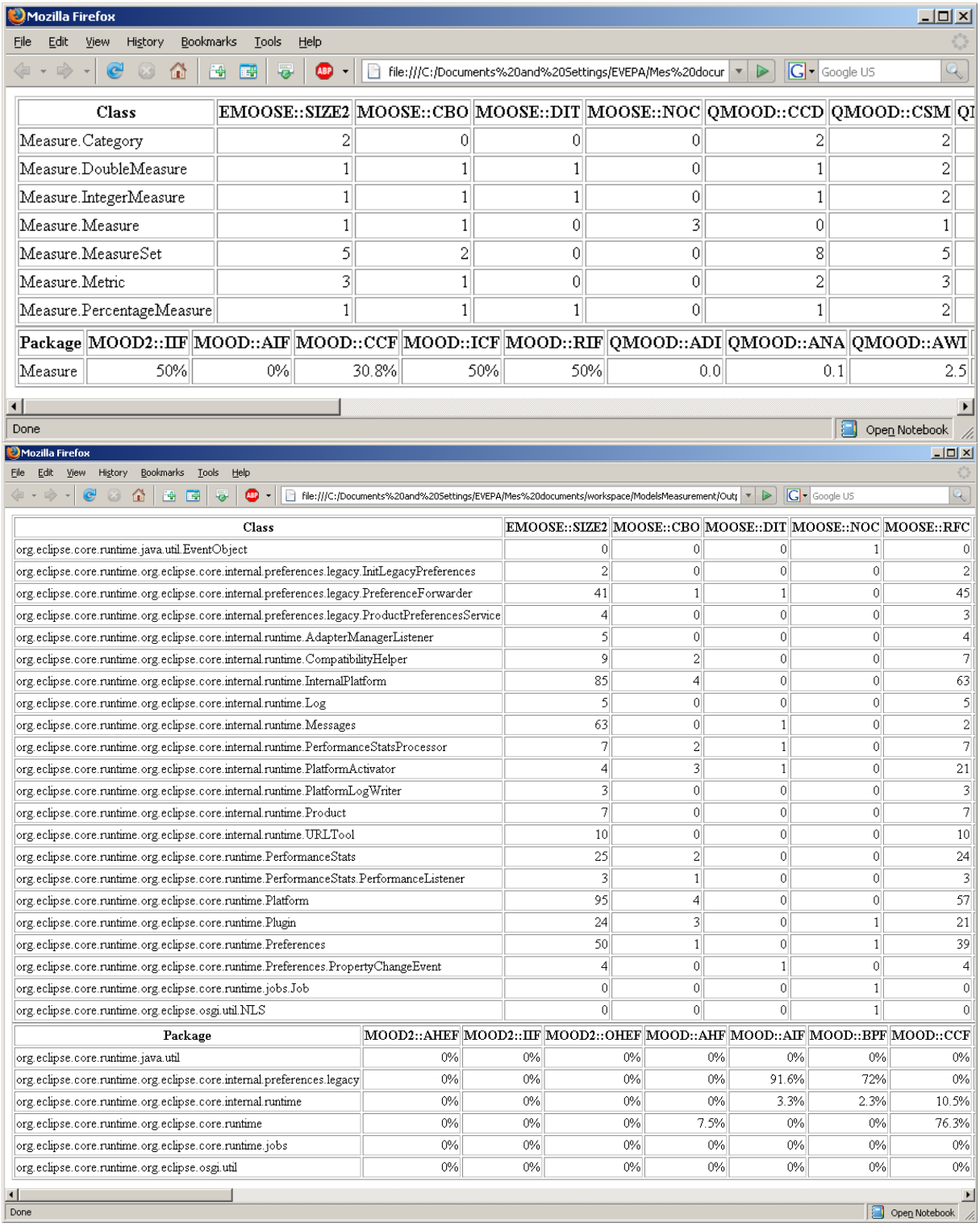


Figure 1: Overview of the transformation



Mozilla Firefox

File Edit View History Bookmarks Tools Help

file:///C:/Documents%20and%20Settings/EVEPA/Mes%20docur

Google US

Class	EMOOSE::SIZE2	MOOSE::CBO	MOOSE::DIT	MOOSE::NOC	QMOOD::CCD	QMOOD::CSM	QI
Measure.Category	2	0	0	0	2	2	
Measure.DoubleMeasure	1	1	1	0	1	2	
Measure.IntegerMeasure	1	1	1	0	1	2	
Measure.Measure	1	1	0	3	0	1	
Measure.MeasureSet	5	2	0	0	8	5	
Measure.Metric	3	1	0	0	2	3	
Measure.PercentageMeasure	1	1	1	0	1	2	

Package	MOOD2::IIF	MOOD::AIF	MOOD::CCF	MOOD::ICF	MOOD::RIF	QMOOD::ADI	QMOOD::ANA	QMOOD::AWI
Measure	50%	0%	30.8%	50%	50%	0.0	0.1	2.5

Done

Open Notebook

Mozilla Firefox

File Edit View History Bookmarks Tools Help

file:///C:/Documents%20and%20Settings/EVEPA/Mes%20documents/workspace/ModelsMeasurement/Outp

Google US

Class	EMOOSE::SIZE2	MOOSE::CBO	MOOSE::DIT	MOOSE::NOC	MOOSE::RFC
org.eclipse.core.runtime.java.util.EventObject	0	0	0	1	0
org.eclipse.core.runtime.org.eclipse.core.internal.preferences.legacy.InitLegacyPreferences	2	0	0	0	2
org.eclipse.core.runtime.org.eclipse.core.internal.preferences.legacy.PreferenceForwarder	41	1	1	0	45
org.eclipse.core.runtime.org.eclipse.core.internal.preferences.legacy.ProductPreferencesService	4	0	0	0	3
org.eclipse.core.runtime.org.eclipse.core.internal.runtime.AdapterManagerListener	5	0	0	0	4
org.eclipse.core.runtime.org.eclipse.core.internal.runtime.CompatibilityHelper	9	2	0	0	7
org.eclipse.core.runtime.org.eclipse.core.internal.runtime.InternalPlatform	85	4	0	0	63
org.eclipse.core.runtime.org.eclipse.core.internal.runtime.Log	5	0	0	0	5
org.eclipse.core.runtime.org.eclipse.core.internal.runtime.Messages	63	0	1	0	2
org.eclipse.core.runtime.org.eclipse.core.internal.runtime.PerformanceStatsProcessor	7	2	1	0	7
org.eclipse.core.runtime.org.eclipse.core.internal.runtime.PlatformActivator	4	3	1	0	21
org.eclipse.core.runtime.org.eclipse.core.internal.runtime.PlatformLogWriter	3	0	0	0	3
org.eclipse.core.runtime.org.eclipse.core.internal.runtime.Product	7	0	0	0	7
org.eclipse.core.runtime.org.eclipse.core.internal.runtime.URLTool	10	0	0	0	10
org.eclipse.core.runtime.org.eclipse.core.runtime.PerformanceStats	25	2	0	0	24
org.eclipse.core.runtime.org.eclipse.core.runtime.PerformanceStats.PerformanceListener	3	1	0	0	3
org.eclipse.core.runtime.org.eclipse.core.runtime.Platform	95	4	0	0	57
org.eclipse.core.runtime.org.eclipse.core.runtime.Plugin	24	3	0	1	21
org.eclipse.core.runtime.org.eclipse.core.runtime.Preferences	50	1	0	1	39
org.eclipse.core.runtime.org.eclipse.core.runtime.Preferences.PropertyChangeEvent	4	0	1	0	4
org.eclipse.core.runtime.org.eclipse.core.runtime.jobs.Job	0	0	0	1	0
org.eclipse.core.runtime.org.eclipse.osgi.util.NLS	0	0	0	1	0

Package	MOOD2::AHEF	MOOD2::IIF	MOOD2::OHEF	MOOD::AHF	MOOD::AIF	MOOD::BPF	MOOD::CCF
org.eclipse.core.runtime.java.util	0%	0%	0%	0%	0%	0%	0%
org.eclipse.core.runtime.org.eclipse.core.internal.preferences.legacy	0%	0%	0%	0%	91.6%	72%	0%
org.eclipse.core.runtime.org.eclipse.core.internal.runtime	0%	0%	0%	0%	3.3%	2.3%	10.5%
org.eclipse.core.runtime.org.eclipse.core.runtime	0%	0%	0%	7.5%	0%	0%	76.3%
org.eclipse.core.runtime.org.eclipse.core.runtime.jobs	0%	0%	0%	0%	0%	0%	0%
org.eclipse.core.runtime.org.eclipse.osgi.util	0%	0%	0%	0%	0%	0%	0%

Done

Open Notebook

Figure 2: Samples of output HTML file

2. Metamodels

2.1. Table

The source metamodel of Table is described in Figure 3 and can be found in the Atlantic Zoo [5].

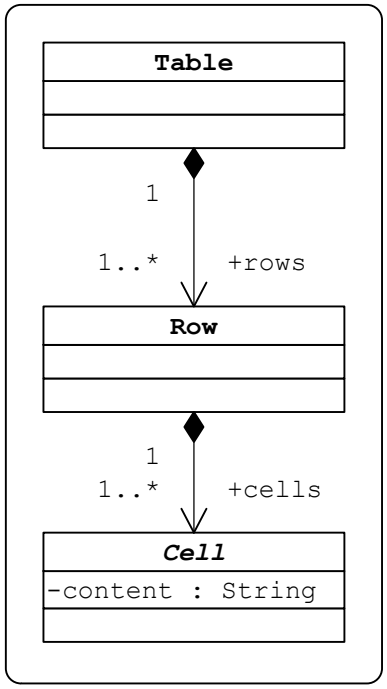


Figure 3: Table Metamodel

Within this metamodel, a Table is associated with a Table element. Such an element is composed of several Rows that, in their turn, are composed of several Cells.

2.2. HTML

This transformation uses only a subset of the HTML metamodel which represents the HTML language. The HTML metamodel can be found in the Atlantic Zoo [5].

3. Transformation from Table to TabularHTML

3.1. Rules specification

These are the rules to transform a Table model to a HTML model.

- For the whole model, the following elements are created:
  - An HTML element composed of a HEAD element and a BODY element.
  - A HEAD element, linked to the HTML element, composed of a TITLE element.
  - A TITLE element, linked to the HEAD element. The value of the title is set to an empty String because the HTML specifications say that "Every HTML document must have a TITLE element in the HEAD section.".
  - A BODY element, linked to the HTML element.
- For each Table element, the following elements are created:
  - A TABLE element, linked to the unique BODY element, with border attribute set to "1" and composed of several TR elements.
  - A TR element, linked to the TABLE element, for the first Row element and composed of several TH elements.
  - A TH element, linked to the TR element, for each Cell element of the first Row element. The value is set to the content of the Cell element.
- For each Row element, the following elements are created:
  - A TR element, linked to the TABLE element, composed of several TD elements.
- For each Cell element, the following elements are created:
  - A TD element, linked to the TR element. The value is set to the content of the Cell element.

3.2. ATL code

This ATL code for the Table2TabularHTML transformation consists in 1 helper and 7 rules.

The attribute helper *html* is used to store the HTML tag for the whole document.

The entrypoint rule HTML() allocates the structure of the HTML file. The rule creates an HTML element (“html”) which is composed of a HEAD element (“head”) and an empty BODY element (“body”). A TITLE element (“title”), with an empty String value, is also created and associated to the HEAD element.

The called rule Table2TABLE allocates a TABLE for each Table element. The rule creates a TABLE element (“table”) and calls a rule for the first row and collects the result for other rows.

The lazy rule Row2TRWithTH allocates a TR for the first Row element. The rule creates a TR element (“tr”) which is composed of the elements allocated for each of his Cell elements.

The lazy rule Cell2TH allocates a TH for a cell. The rule creates a TH element (“th”) with the content of the cell.

The lazy rule Row2TRWithTD allocates a TR for the each other Row element. The rule creates a TR element (“tr”) which is composed of the elements allocated for each of his Cell elements.

The lazy rule Cell2TD allocates a TD for a cell. The rule creates a TD element (“td”) with the content of the cell.

The lazy rule Cell2TDRightAlign allocates a TD for a cell. The rule creates a TD element (“td”) with the content of the cell and an attribute align set to *right*.

4. ATL Library TableHelpers

4.1. ATL code

This ATL code for the TableHelpers library consists in 9 helpers.

The helpers *isInteger*, *isReal* and *isPercentage* format a raw value.

The helpers *value* are used to format the value of a cell (adding a unit for a percentage value, truncating a too long real, etc…).

The helper *realValue* is used to convert a percentage value into a real (remove the '%' unit and a real between 0 and 1).

The helper *seqWithoutFirst* returns a sequence without the first element.

The helper *allValidTables* is not used in this transformation.

The helper *valueNotNull* is used to check if the content of a cell (converted as a real) is null or not.

5. XML Extractor

The HTML language is a XML-like language, so the model elements are mapped to XML Element with the same name as this of the HTML model element.

The XML Root element is the HTML element.

The attributes of any HTML element are mapped to XML Attribute or Text elements.

The relations between HTML model elements are represented using the *children* relation between XML Nodes.

6. References

[1] ATL "ATLAS Transformation Language" official site (Eclipse/M2M subproject).
http://www.eclipse.org/m2m/atl/

[2] The Kernel MetaMetaModel (KM3) Manual.
http://www.eclipse.org/gmt/am3/km3/doc/KernelMetaMetaModel%5Bv00.06%5D.pdf

[3] KM3 Wiki.
http://wiki.eclipse.org/index.php/KM3

[4] The Atlantic Zoo.

[5] http://www.eclipse.org/gmt/am3/zoos/atlanticZoo/