	ATL Transformation Example	Author Éric Vépa evepa@sodius.com
	Table to TabularHTML	August 30th , 2007

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.

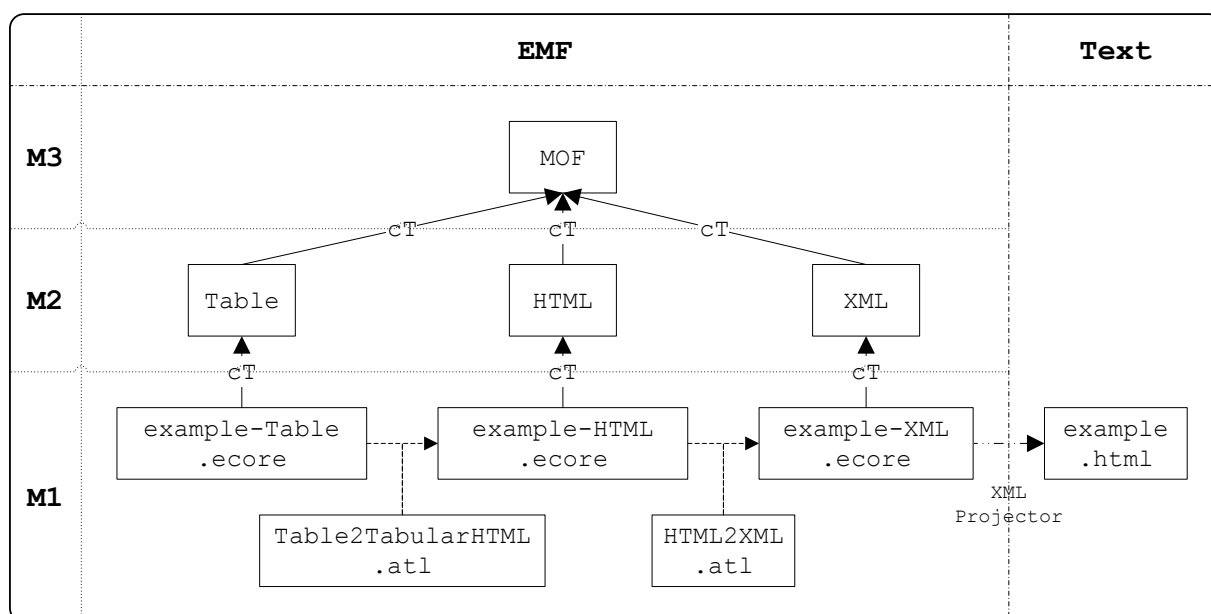
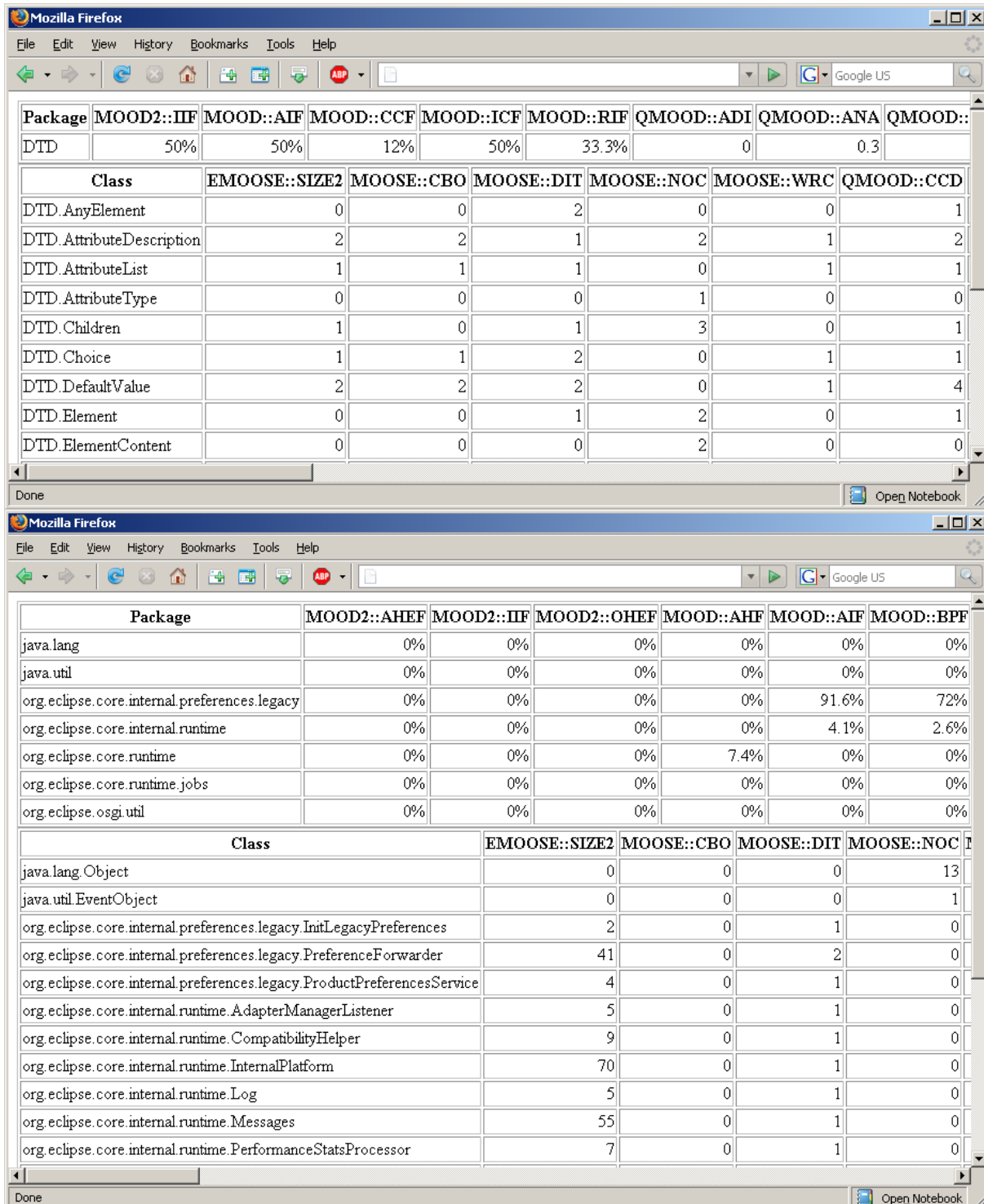


Figure 1: Overview of the transformation



The generation of the output HTML file is realized by a first transformation from Table to HTML, followed by the usage of a projector. The projector consists in a transformation from HTML to XML and the predefined XML extractor (HTML is a XML-like language). The output .html file contains HTML tables for each kind of model element measured.



Package	MOOD2::IIF	MOOD2::AIF	MOOD2::CCF	MOOD2::ICF	MOOD2::RIF	QMOOD2::ADI	QMOOD2::ANA	QMOOD2::...
DTD	50%	50%	12%	50%	33.3%	0	0.3	
Class	EMOOSE::SIZE2	MOOSE::CBO	MOOSE::DIT	MOOSE::NOC	MOOSE::WRC	QMOOD2::CCD		
DTD.AnyElement	0	0	2	0	0	1		
DTD.AttributeDescription	2	2	1	2	1	2		
DTD.AttributeList	1	1	1	0	1	1		
DTD.AttributeType	0	0	0	1	0	0		
DTD.Children	1	0	1	3	0	1		
DTD.Choice	1	1	2	0	1	1		
DTD.DefaultValue	2	2	2	0	1	4		
DTD.Element	0	0	1	2	0	1		
DTD.ElementContent	0	0	0	2	0	0		

Package	MOOD2::AHEF	MOOD2::IIF	MOOD2::OHEF	MOOD2::AHF	MOOD2::AIF	MOOD2::BPF
java.lang	0%	0%	0%	0%	0%	0%
java.util	0%	0%	0%	0%	0%	0%
org.eclipse.core.internal.preferences.legacy	0%	0%	0%	0%	91.6%	72%
org.eclipse.core.internal.runtime	0%	0%	0%	0%	4.1%	2.6%
org.eclipse.core.runtime	0%	0%	0%	7.4%	0%	0%
org.eclipse.core.runtime.jobs	0%	0%	0%	0%	0%	0%
org.eclipse.osgi.util	0%	0%	0%	0%	0%	0%
Class	EMOOSE::SIZE2	MOOSE::CBO	MOOSE::DIT	MOOSE::NOC	MOOSE::WRC	QMOOD2::CCD
java.lang.Object	0	0	0	13		
java.util.EventObject	0	0	0	1		
org.eclipse.core.internal.preferences.legacy.InitLegacyPreferences	2	0	1	0		
org.eclipse.core.internal.preferences.legacy.PreferenceForwarder	41	0	2	0		
org.eclipse.core.internal.preferences.legacy.ProductPreferencesService	4	0	1	0		
org.eclipse.core.internal.runtime.AdapterManagerListener	5	0	1	0		
org.eclipse.core.internal.runtime.CompatibilityHelper	9	0	1	0		
org.eclipse.core.internal.runtime.InternalPlatform	70	0	1	0		
org.eclipse.core.internal.runtime.Log	5	0	1	0		
org.eclipse.core.internal.runtime.Messages	55	0	1	0		
org.eclipse.core.internal.runtime.PerformanceStatsProcessor	7	0	1	0		

Figure 2: Samples of output HTML file

 	ATL Transformation Example	Author Éric Vépa evepa@sodius.com
	Table to TabularHTML	August 30th , 2007

2. Metamodels

2.1. Table

The source metamodel of Table is described in Figure 3 and can be found in the Atlantic Zoo **Error! Reference source not found.**

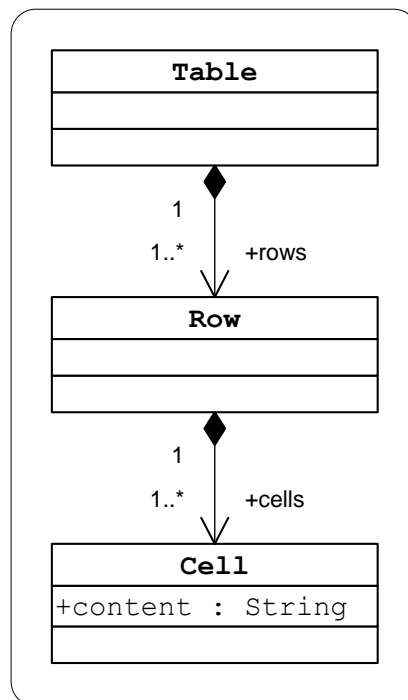




Figure 3: Table Metamodel

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 [2].

 	ATL Transformation Example	Author Éric Vépa evepa@sodius.com
	Table to TabularHTML	August 30th , 2007

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.

 	ATL Transformation Example	Author Éric Vépa evepa@sodius.com
	Table to TabularHTML	August 30th , 2007

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.

 	ATL Transformation Example	Author Éric Vépa evepa@sodius.com
	Table to TabularHTML	August 30th , 2007

5. HTML Projector

The HTML projector is a transformation from HTML to XML followed by the predefined XML extractor.

This can be done in this way, because HTML is a XML-like language.

The HTML element is mapped to the XML Root element.

Other HTML mark-up are mapped to XML Element element.

Each attribute of a HTML mark-up is mapped as a XML Attribute element.

The HTML CDATA or PCDATA are mapped to XML Text element.

6. References

- [1] ATLAS (ATLantic dAta Systems) Official Webpage: <http://www.sciences.univ-nantes.fr/lina/ATLAS/>
- [2] The Atlantic Zoo: <http://www.eclipse.org/gmt/am3/zoos/atlanticZoo/>