AM3 Tool Management

Goal: cartography of tools & formats

Way: chains of (ATL) transformations

Use case: Business Process Modeling for BNP/Paribas → cartography of BPM tools

Current state :

- input: an Excel table about some BPM tools
- first version of Tool cartography metamodel (AM3)
- 1st chain: from Excel table to Tool model
- 2nd chain: Tool model visualization in AM3 generic viewer
- (*interoperability with SIV visualization tool*)

Use case: BPM tools & formats

Tools	BPMN (Business Process Modeling Notation)	BPEL (Business Process Execution Language)	XPDL (XML Process Definition Language)	WSDL (Web Services Description Language)	AML (ARIS Markup Language)	UML (Unified Modeling Language)	JPDL (jBPM Process Definition Language)	JWT (Java Workflow Tooling)	Visio
ARIS Platform (IDS Scheer)	I/E	I/E	E	I	B	I/E			
oFlow Toolbox					I				I
Bonita Open Solution	B		I/E						
BPEL Process Manager (Oracle)		B		I					
BPEL Project (Eclipse.org)		B		I					
BPMN Project (Eclipse.org)	B								
BPM-Xchange	I/E	I/E	I/E		I/E	I/E			I/E
Business Process Visual Architect (Visual Paradigm)	B	B	E	B			E		
Enterprise Architect 7.0 (Sparx Systems)	B			B		B			
JWT (Eclipse.org)	I/E		E				E	B	
MagicDraw Architect or Enterprise (No Magic)	I/E	E		I/E		B			
MEGA Process	B	E	E	I					
Rational Software Architect (IBM)	B			E		B			
Tibco Business Studio	B		B	I/E	I/E	B			I/E
UModel Enterprise Edition (Altova)	B			B (XMLSpy)		B			

Legend:

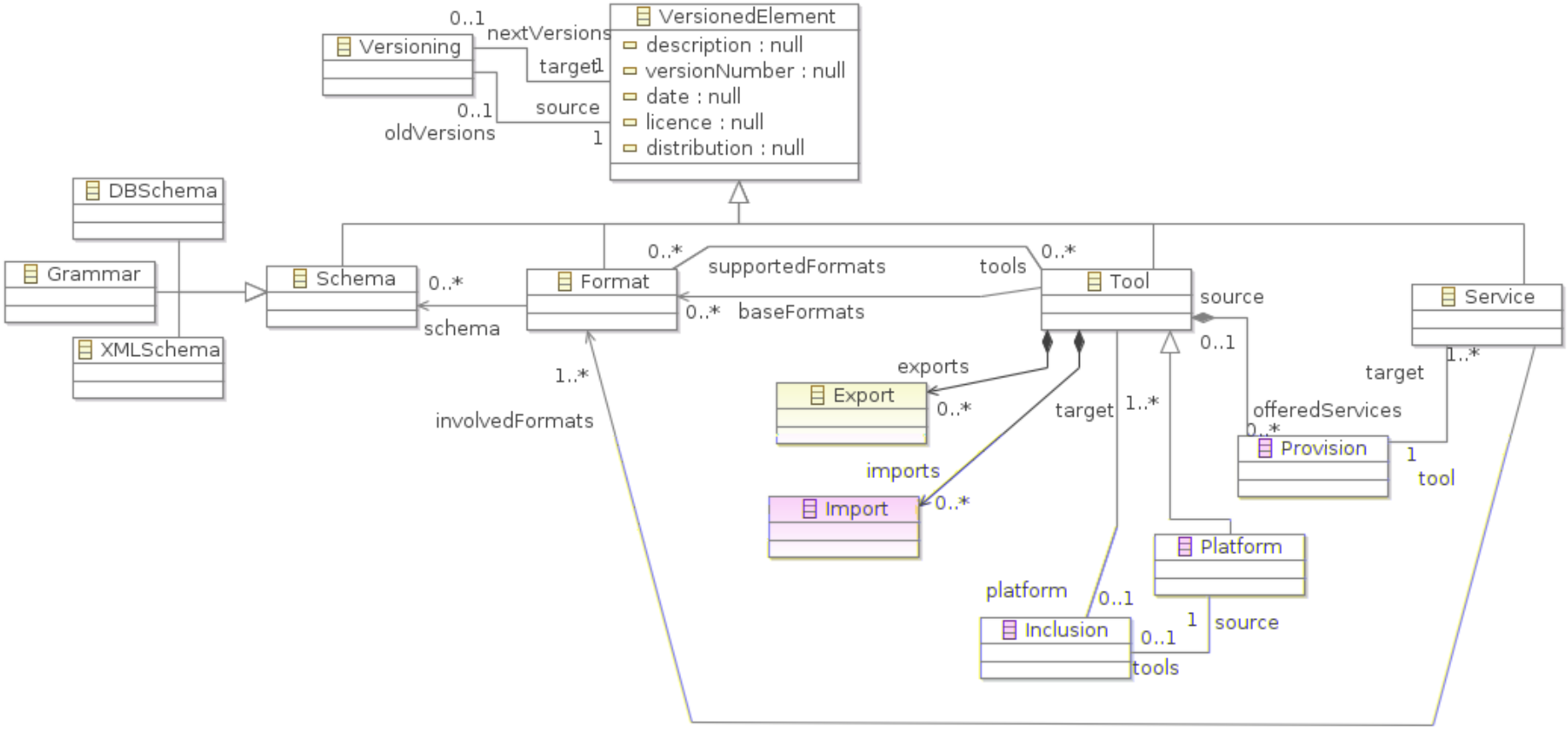
B for Base format
I for available Import format
E for available Export format

ToolManagement metamodel

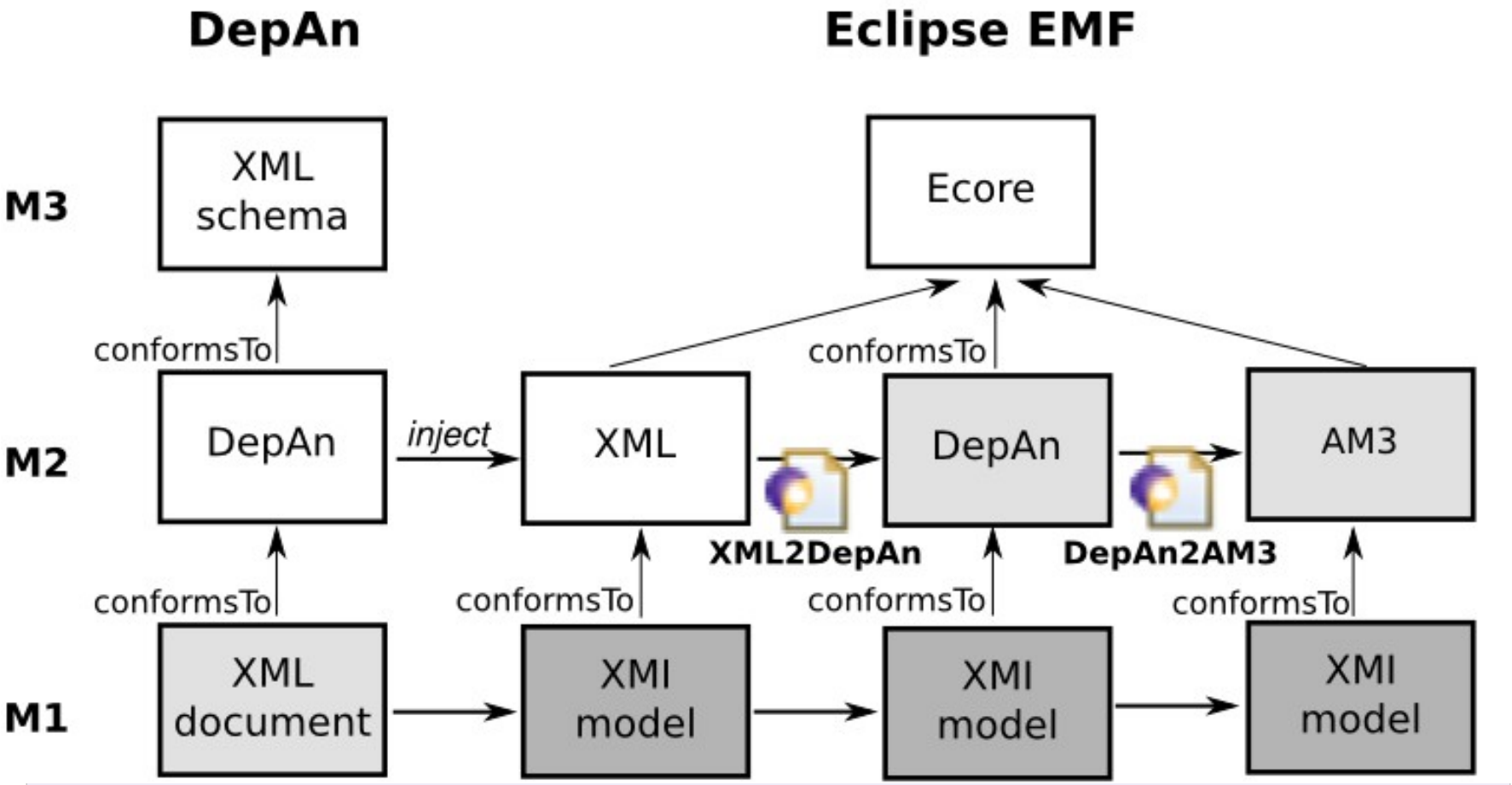
Specializes main AM3 types

Tool & **Format**: *inherit* AM3::Entity

Export & **Import**: *inherit* AM3::DirectedRelationship



1st chain: From **Excel** to **Tool**



1st chain: From **Excel** to **Tool**

1a: Excel table → Excel 2003 XML

- We collect data in Excel/OpenOffice
- We save data into Excel 2003 XML format

Tools	BPMN (Business Process Modeling Notation)	BPEL (Business Process Execution Language)	XPDL (XML Process Definition Language)	WSDL (Web Services Description Language)	AML (ARIS Markup Language	UML (Unified Modeling Language)	JPDL (jBPM Process Definition Language)	JWT (Java Workflow Tooling)	Visio
ARIS Platform (IDS Scheer)	I/E	I/E	E	I	B	I/E			
oFlow Toolbox					I				I
Bonita Open Solution	B		I/E						
BPEL Process Manager (Oracle)		B		I					
BPEL Project (Eclipse.org)		B		I					
BPMN Project (Eclipse.org)	B								
BPM-Xchange	I/E	I/E	I/E		I/E	I/E			I/E
Business Process Visual Architect (Visual Paradigm)	B	B	E	B			E		
Enterprise Architect 7.0 (Sparx Systems)	B			B		B			
JWT (Eclipse.org)	I/E		E				E	B	
MagicDraw Architect or Enterprise (No Magic)	I/E	E		I/E		B			
MEGA Process	B	E	E	I					
Rational Software Architect (IBM)	B			E		B			
Tibco Business Studio	B		B	I/E	I/E	B			I/E
UModel Enterprise Edition (Altova)	B			B (XMLSpy)		B			

Legend:

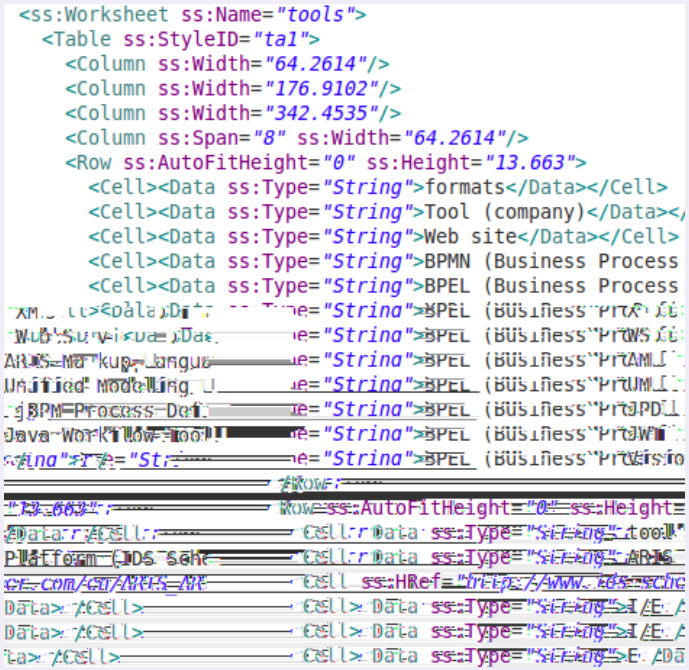
B for Base format
I for available Import format
E for available Export format

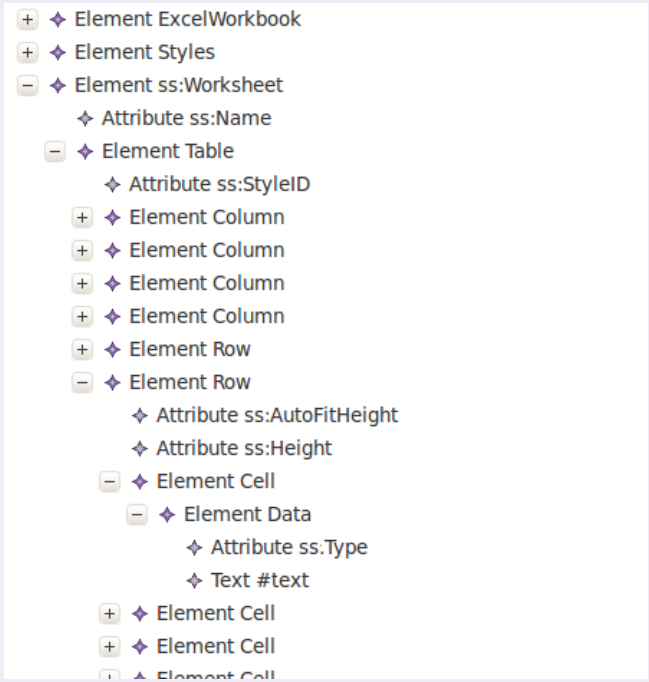
```
<ss:Worksheet ss:Name="tools">
  <Table ss:StyleID="ta1">
    <Column ss:Width="64.2614"/>
    <Column ss:Width="176.9102"/>
    <Column ss:Width="342.4535"/>
    <Column ss:Span="8" ss:Width="64.2614"/>
    <Row ss:AutoFitHeight="0" ss:Height="13.663">
      <Cell><Data ss:Type="String">formats</Data></Cell>
      <Cell><Data ss:Type="String">Tool (company)</Data></
      <Cell><Data ss:Type="String">Web site</Data></Cell>
      <Cell><Data ss:Type="String">BPMN (Business Process
      <Cell><Data ss:Type="String">BPEL (Business Process
```

1st chain: From **Excel** to **Tool**

1b: Excel XML → XML model

- The ATL XML injector gives the EMF model corresponding to the XML file

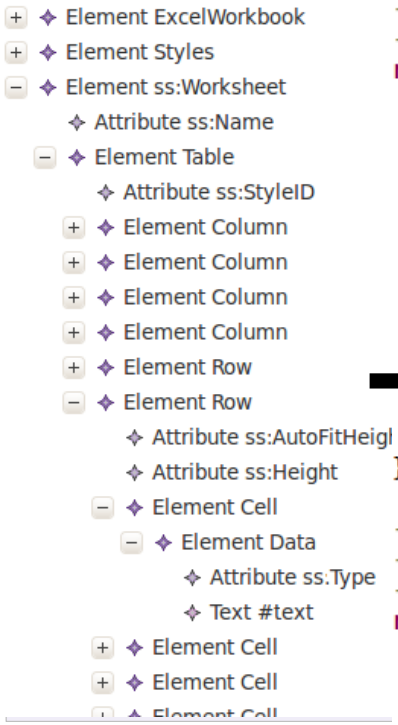




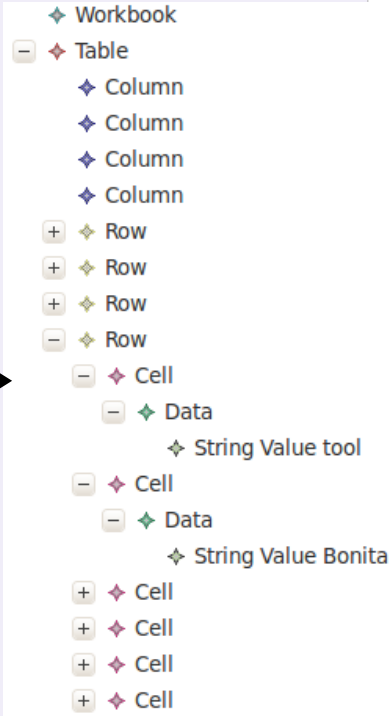
1st chain: From **Excel** to **Tool**

1c: XML model → Excel model

generic **XML2Excel** (independent from content)



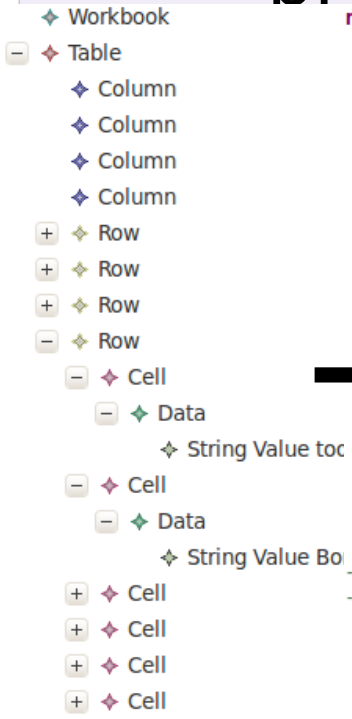

```
-- Rule 'Table'
-- This rule generates the table for a worksheet.
-- It's the table which contains the columns and rows.
rule Table {
    from
        et : XML!Element (
            et.name = 'Table'
        )

    to
        tab : SpreadsheetMLSimplified!Table (
            t_cols <- Sequence{et.getChildrenByName('Column')
                ->collect(e | thisModule.resolveTemp(e, 'col'))},
            t_rows <- Sequence{et.getChildrenByName('Row')
                    ->collect(e | thisModule.resolveTemp(e, 'row'))}
        )
}

-- Rule 'Column'
-- This rule generates the columns contained in a table.
-- They don't store the data but they give some specific information
rule Column {
    from
        ec : XML!Element (
            ec.name = 'Column'
        )
```

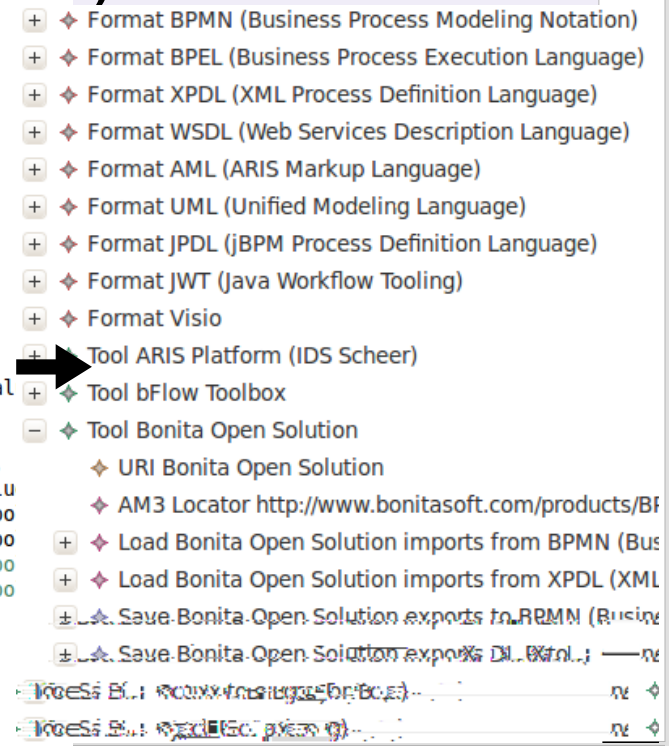


1st chain: From **Excel** to **Tool**

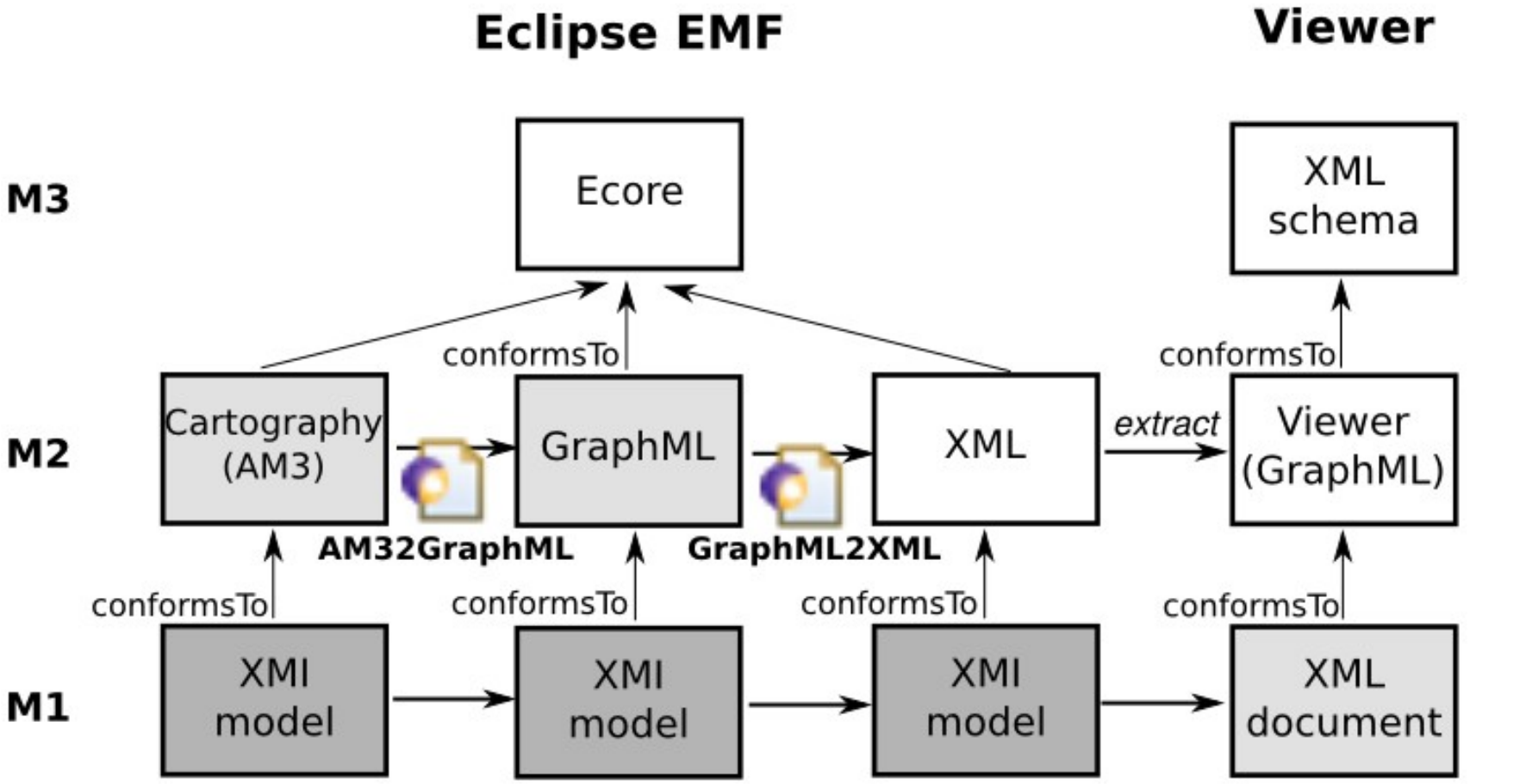
1d: Excel model → Tool model

- ATL transformation specific to our problem (tool cartography)





2nd chain: AM3 visualization



2nd chain: AM3 visualization

Tool types inherit from **AM3** types

- so AM3 transformations apply on them

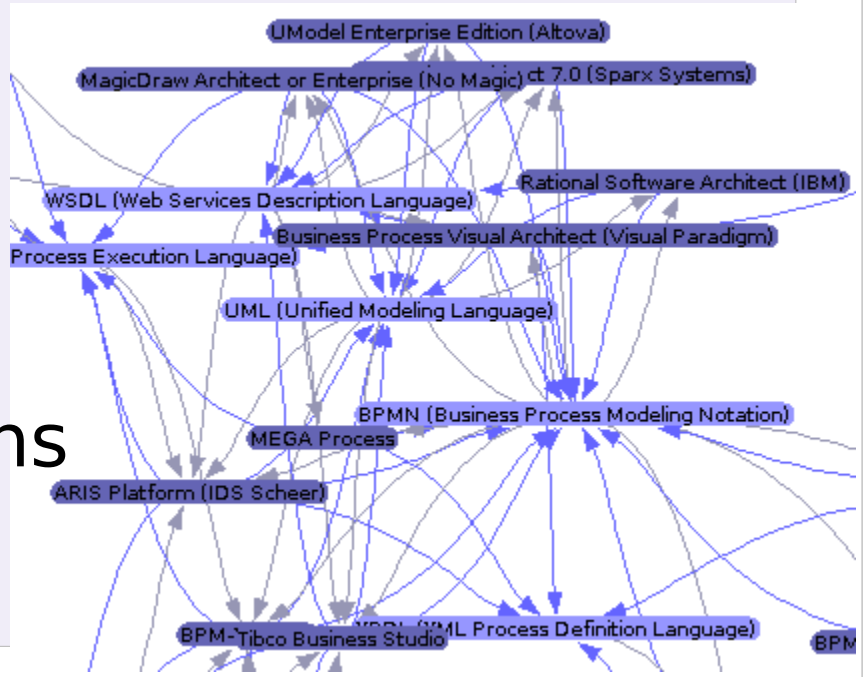
We apply the AM3→visualization chain:

2a : **AM3toGraphML**

2b: **GraphML2XML**

2c: **XML** extraction

2d: run Prefuse-based AM3 visualizations



*Interoperability: **SIV** export*

Works as the AM3 generic chain

- 3a: **AM3toSIVGraphML**
  - a GraphML model with SIV specific tags
- 3b: **GraphML2XML**
- 3c: **XML** extraction
- 3d: loading in **SIV**