***Automatic generation of documentation from UML models.***

***Guidelines from the ISO/TC 211 ad hoc group***



# Advantages and disadvantages

### Advantages

The advantages of documenting the UML model elements automatically from the UML model are several:

* All the standards document the UML models according to a limited number of templates. This is far from the reality today.
* Definition of the UML concepts may (and should) be added during the model phase, and should be agreed upon at this stage. It ensures compatibility between the models and the text in the standards.
* If the same model element is documented several places in the standard, the documentation will be identical.
* Manual work that would otherwise be required to reproduce text from the model should be reduced or eliminated
* Given the above, the potential for human error should be reduced when generating documentation

### Disadvantages

The disadvantages of documenting the model elements automatically from the UML model may be:

* A revised version of the standard according to new documentation structure may look quite different from the existing ones, particularly in a transition period. This may confuse some readers.
* Text formatting is restricted to the formatting functions supported by Enterprise Architect’s text formatting options. Additional documentation requires manual editing.
* Some effort and tool-specific knowledge would be required to create and maintain the appropriate templates in the document generation tool(s).

# Underlying requirements and recommendations

## ISO/DIS 19103

The revised ISO introduces a new conformance class, ModelDocumentation, giving requirements on the documentation of the UML model elements:

Table 1 — Model documentation conformance class

|  |  |
| --- | --- |
| Conformance class identifier | ModelDocumentation |
| Standardization target type | Documentation of UML schemas for geographic information |
| Dependency | UML2 |
| Requirements | All requirements in Clause 6.16 |

### Requirements in Clause 6.16 Documentation of models

**Requirement 18**

All classifiers shall be documented in a "context diagram" where all attributes,

operations and all relationships that are navigable from the central classifier are

displayed.

**Requirement 19.**

Each classifier shall have a definition describing its intended meaning or semantics

**Requirement 20**

Each package, class, operation, attribute, association role, association and constraint shall have a textual description in the text near its context diagram

**Requirement 21**

Package dependency diagrams shall be included, including external package dependencies.

ISO/DIS 19103 makes no requirements for documentation of models beyond Requirements 18 to 21.

### Recommendations

**Recommendation 15**

For the diagrams, readability concerns should be taken into account. Font size should be no smaller than 8 points after diagram scale is taken into account (a 12 pt font at 90% is 12\*0.9 = 10.8 pt).

**Recommendation 16**

For each package it is recommended to have a class diagram for the entire package (without any associations, operations or attributes shown)

# Documentation of UML model concepts in Enterprise Architect

## Documentation fields

In Enterprise Architect all the properties have a "Notes" field. This field should be populated with the definitions for the UML model concepts in the UML model.

Definitions of classes, attributes and associations should be documented in the "Notes" field according to Figure 1.



Figure 1 - Definition of classes, attributes and associations

Definitions of source role and target role should be documented in the "Role Notes" field according to Figure 2.



Figure 2 – Definitions of roles

Constraints should be defined according to the example in Figure 3, (in OCL according to ISO 19103)



Figure 3 – Definitions of constraints

## Text formatting

Figure 4 below shows the formatting options in Enterprise Architect.



Figure 4 – Text formatting options in Enterprise Architect

Other formatting options need to be done manually in the generated output.

# Tools for automatic documentation

## Introduction

During the work in the ad hoc group two candidates were provided, Enterprise Architect with an RTF document template and a revised version of ShapeChange. Both candidates gave a demonstration during the workshop in Redlands, US. None of the candidates produces both the UML figures and the textual documentation during the workshop. ShapeChange provides two alternatives.

## Enterprise Architect

A document template "ISOTC211\_EA\_report\_template.xml" is made available. It can be imported into an EA project via the Main Menu.

The document template will be available at ISO/TC 211 Best practice at https://github.com/ISO-TC211/UML-Best-Practices/wiki/Documentation.

If this generated output shall be used to document a UML model in an ISO standard (or any other document containing UML models), the generated output needs to be copied into that document together with the UML figures.

**Step one** –import the report template into an EA project via the Main Menu: Project > Data Management > Import Reference Data > [select file] > RTF Document Templates



Figure 5 – Importing the documentation template

**Step two** – Select the template file “isotc211-ea-templates-20150603.xml”, click on "RTF document Templates" and "Import".



Figure 6 – Selecting the correct document template

**Step 3** – Select the UML package that you want to document, “right click”, choose "Documentation" and "Generate Documentation".



Figure 7 – Start generating the documentation

**Step 4** – Chose the output directory ("Output to file") and name of the documentation file, choose the following configuration:

Table 2 - Configuration

|  |  |
| --- | --- |
| Package  | The package in the model that should be documented (Chose a package in the project view before starting the documentation) |
| Output to File | The file structure where the document should be generated |
| Template | ISOTC211\_Document Body (This template is available after importing the xml file) |
| Output Format | Microsoft Document Format (DOCX) |
| Cover | <none>(There is an example of an ISO Cover page (ISO 19135 cover page) but this needs to be adopted to ISO document template anyhow) |
| Table of | <none>  |
| Stylesheet | ISOTC211\_Object Type Styles |
| Diagram Theme | Monochrome for printing(This will ensure that all diagrams are documented without colours) |



Figure 8 – Configuration for documentation of UML models

## ShapeChange

### Introduction

ShapeChange is a tool to generate different artifacts from UML model for geographic information. There are several input options, one of them is an Enterprise Architect project file. Among the output targets is the generation of a feature catalogue. This report will only describe the functionality for the documentation of feature catalogues.

ShapeChange is designed for application schemas according to ISO 19109, but works with other UML models if they use a compatible UML profile.

There are several output formats for feature catalogues, among them is DOCX. Using an ISO template file it can be configured (xslt) to generate the documentation directly into the an ISO standard document.

The standard feature catalogue output layout is the style used in the ISO 19135-1 standard. This style is generated using the standard stylesheet docx.xsl. By adapting the stylesheet other layouts can be created as well.

For the workshop in Redlands, a more compact "dictionary" layout like in ISO 19115-1, dictionary has been created for demonstration purposes (docx-compact.xsl). This stylesheet has been created to illustrate the process, but it is not complete. For example, it does not include the tables for enumerants.

Requirements:

* Configuration file
* DOCX template

The software is available at <http://shapechange.net/>. New versions of the software should continue to work with the configurations.

### Configuration

The configuration files states which model (EAP) that should be applied, input and output directory, and a set of other configuration issues.

These configuration parameters and their explanation is maintained by Interactive instruments GmbH and The MITRE Corporation at <http://www.shapechange.net/>.

Table 3 — General configuration for ShapeChange

|  |
| --- |
| **General configuration** |
| Parameter | Default value | Description |
| inputModelType | EA7 | EnterpriseArchitect (version) or xmi. |
| inputFile" |  | Name of the EA project or name of XMI file. |
|  |  |  |
| publicOnly |  | ?? |
| checkingConstraints |  | Enable or disable.  |
| sortedSchemaOutput |  | True or False |
| tmpDirectory |  | This is the directory where all the temoprary files are stored, like images. |
| loadDiagrams | False | Boolean. Set to ‘true’ to load diagram images.. |
| packageDiagramRegex | NAME(the reserved keyword that will be replaced by the package name) | Regular expression to control which package diagrams get assigned to a package. If the name of a package diagram contained in a package matches the expression (with the occurrence(s) of ‘NAME’ having been replaced with the name of that package) then the diagram will be assigned to the package. |
| classDiagramRegex | NAME(the reserved keyword that will be replaced by the class name) | Regular expression to control which logical (i.e. class) diagrams get assigned to a class. For each logical diagram contained in a package, the expression is evaluated for each class that is also contained in that package. If the name of the logical diagram matches the expression (with the occurrence(s) of ‘NAME’ having been replaced with the name of a given class) then the diagram will be assigned to that class. |
| **Logging configuration** |
| reportLevel |  | 'info' both warnings and errors'error' – only errors. |
| logFile |  | Catalog and name of the logfile |
| **Feature catalogue configuration** |
| outputFilename | FeatureCatalogue | The name of the output file without extension. The appropriate extension will be added depending on the output format. |
| outputFormat | **Required** | (Required) A string representing the desired output formats.  The possible values are:* HTML
* FRAMEHTML
* DOCX

Case is unimportant.Multiple values may be provided separated with commas. |
| name | unknown | The name to be used for this feature catalogue. |
| scope | unknown | Information on the scope of this feature catalogue. This may contain HTML markup and “[NEWLINE]” to separate paragraphs. |
| versionNumber | unknown | The version number for this feature catalogue. |
| versionDate | unknown | The version date for this feature catalogue. |
| producer | unknown | The producer name. Note: The producer is a mandatory property of a feature catalogue according to ISO 19110. |
| xsltPfad | http://shapechange.net/resources/xslt | The path (without a trailing “/”) where the required XSLT files are located. |
| xsldocxFile | Docx.xsl | Name of the XSLT script for converting to DOCX. |
| featureTerm | Feature | A string indicating the term to be used for a Feature (a different term can be substituted if necessary).Usually this parameter will not be needed, it has been added to support INSPIRE, which does not use the term “feature” and uses “spatial object” instead.For ISO standards we use Object. |
| includeVoidable | False | Not used for ISO standards. Voidable is a INSPIRE concept. |
| includeTitle | False | Not used for ISO standards. |
| includeDiagrams | Ture | Allows diagrams to be generated in the output file. |
| xslTransformerFactory | net.sf.saxon.TransformerFactoryImpl | It is important to use a tool that can handle xslt. Saxon has been tested for this project. |
| docxTemplateFilePath | Template\_neu.docx | Name of the template used for the output.  |

Documentation of the configuration files is available (and will be updated) at <http://shapechange.net/>.

### Example of configuration fil

<?xml version="1.0" encoding="UTF-8"?>

<ShapeChangeConfiguration xmlns:xi="http://www.w3.org/2001/XInclude" xmlns="http://www.interactive-instruments.de/ShapeChange/Configuration/1.1" xmlns:sc="http://www.interactive-instruments.de/ShapeChange/Configuration/1.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.interactive-instruments.de/ShapeChange/Configuration/1.1 http://shapechange.net/resources/schema/ShapeChangeConfiguration.xsd">

 <input>

 <parameter name="inputModelType" value="EA7"/>

 <parameter name="inputFile" value="ISO19135-1\_Full19135\_extension.eap"/>

 <!--parameter name="appSchema" value="ISO 19135-1 Procedures for Registration"/-->

 <parameter name="publicOnly" value="true"/>

 <parameter name="checkingConstraints" value="disabled"/>

 <parameter name="sortedSchemaOutput" value="true"/>

 <parameter name="tmpDirectory" value="examples/iso/tmp"/>

 <parameter name="loadDiagrams" value="true"/>

 <parameter name="packageDiagramRegex" value="^(.\*[\W]+)?Overview([\W]+.\*)?$"/>

 <parameter name="classDiagramRegex" value="^(.\*[\W]+)?NAME([\W]+.\*)?$"/>

 <xi:include href="http://shapechange.net/resources/config/StandardAliases.xml"/>

 </input>

 <log>

 <parameter name="reportLevel" value="INFO"/>

 <parameter name="logFile" value="output1/log\_19135.xml"/>

 </log>

 <targets>

<Target mode="enabled"class="de.interactive\_instruments.ShapeChange.Target.FeatureCatalogue.FeatureCatalogue" >

 <targetParameter name="outputDirectory" value="output1"/>

 <targetParameter name="outputFilename" value="19135"/>

 <targetParameter name="outputFormat" value="DOCX"/>

 <targetParameter name="name" value="ISO 19135-1 Procedures for Registration"/>

<targetParameter name="scope" value="This part of ISO 19135 specifies procedures to be followed in establishing, maintaining and publishing registers of unique, unambiguous and permanent identifiers and meanings that are assigned to items of geographic information. In order to accomplish this purpose, this part of ISO 19135 specifies elements that are necessary to manage the registration of these items."/>

 <targetParameter name="versionNumber" value="ISO/DIS 19135-1:2013"/>

 <targetParameter name="versionDate" value="2013"/>

<targetParameter name="producer" value="International Organisation for Standardization"/>

 <targetParameter name="xsltPfad" value="http://shapechange.net/resources/xslt"/>

 <!--targetParameter name="xsltPfad" value="src/main/resources/xslt"/-->

 <targetParameter name="xsldocxFile" value="docx.xsl"/>

 <targetParameter name="featureTerm" value="Object"/>

 <targetParameter name="includeVoidable" value="false"/>

 <targetParameter name="includeTitle" value="false"/>

 <targetParameter name="includeDiagrams" value="true"/>

<targetParameter name="xslTransformerFactory" value= "net.sf.saxon.TransformerFactoryImpl"/>

 <targetParameter name="docxTemplateFilePath" value="template\_ISO.docx"/>

 </Target>

 </targets>

</ShapeChangeConfiguration>

### ISO template configuration options

Shape Change has two options for the output. This is controled by the parameter "xsldocFile".

#### Raw based (like ISO 19135)

 <targetParameter name="xsldocxFile" value="docx.xsl"/>

DOCX allow the generation of UML figures together with the textual description if properly configured.

#### Coloumn based (like ISO 19115-1)

 <targetParameter name="xsldocxFile" value="docx-compact.xsl"/>

DOCX-COMPACT is a more compact way of documenting the UML models, more like a dictionary. It does not allow the generation of the figures itself.

Generating according to ISO template requires to use the template\_ISO,docx file.

Note: This stylesheet has been created to illustrate the process, but it is not complete. For example, it does not include the tables for enumerants

### Automatically inclusion in ISO standard document.

ShapeChange may generate the documentation directly into a draft international standard that is compliant with the ISO template.

The keyword "ShapeChangeFeatureCatalogue" in the document will be replaced with the generated output form ShapeChange.

The following configuration parameter must be set in the configuration file:

<targetParameter name="docxTemplateFilePath" value="template.docx"

Template.doc must be replaced with the name of the document where documentation should be added. This document must have the keyword: "ShapeChangeFeatureCatalogue"

### Sorting of diagrams and relevant textual descriptions.

ShapeChange has two configuration parameters for sorting diagrams and text:

<parameter name="packageDiagramRegex" value="^(.\*[\W]+)?Overview([\W]+.\*)?$"/>

 <parameter name="classDiagramRegex" value="^(.\*[\W]+)?NAME([\W]+.\*)?$"/>

For further explanations, see clause 6.3.2.

### Running ShapeChange in batch mode

The figure below shows the file and catalogue structure that will be available at ISO/TC 211 Best practice at https://github.com/ISO-TC211/UML-Best-Practices/wiki/Documentation.



Figure 9 – catalogue structure for ShapeChange

The file "convert\_ISOtemplate.bat" is a batch file starting the java program according to the configuration stated in "19135\_doc.xml" and "19135\_doc\_compact.xml" (Generates two output documents).

### Running ShapeChange as a plug-in to Enterprise Architect

ShapeChange is implemented as a plug-in for EA, with a user interface for the configuration. This plug-in is developed by Arkitektum on behalf of the Norwegian mapping Authority.

A test version is available at <http://www.arkitektum.no/files/sosi/shapechange_setup.exe>

This plug-in allows you to start ShapeChange directly from EA.

In EA this will look like:



Figure 10 – Running ShapeChange in batch mode

Running the transform command, a new pop-up window will appear:



Figure 11 – Configuration

This gives the possibility to set some of the configuration parameters.

**Remark 1**:

For this test version, the template filename cannot be set. This will be fixed.

**Remark 2:**

The plug-in is still a test version. Not all the possible configuration parameters can be configured. There is a discussion going on regarding a graphical user interface for ShapeChange provided by Interactive Instruments.