

# Model Based Document Generation

Michele Zamparelli

Robert Karban

# The Problem

- Document is culturally the primary artifact
- Models of system live a separate life
- Diagrams are copy/pasted into documents
- Description of model elements is partially kept separate from model (description added to diagrams in document)
- Two sources of information (Model and Document) have to be maintained – get out of sync
- Document generation engines (velocity template) have typically hard-coded references to model elements and diagrams

# Suggested solution

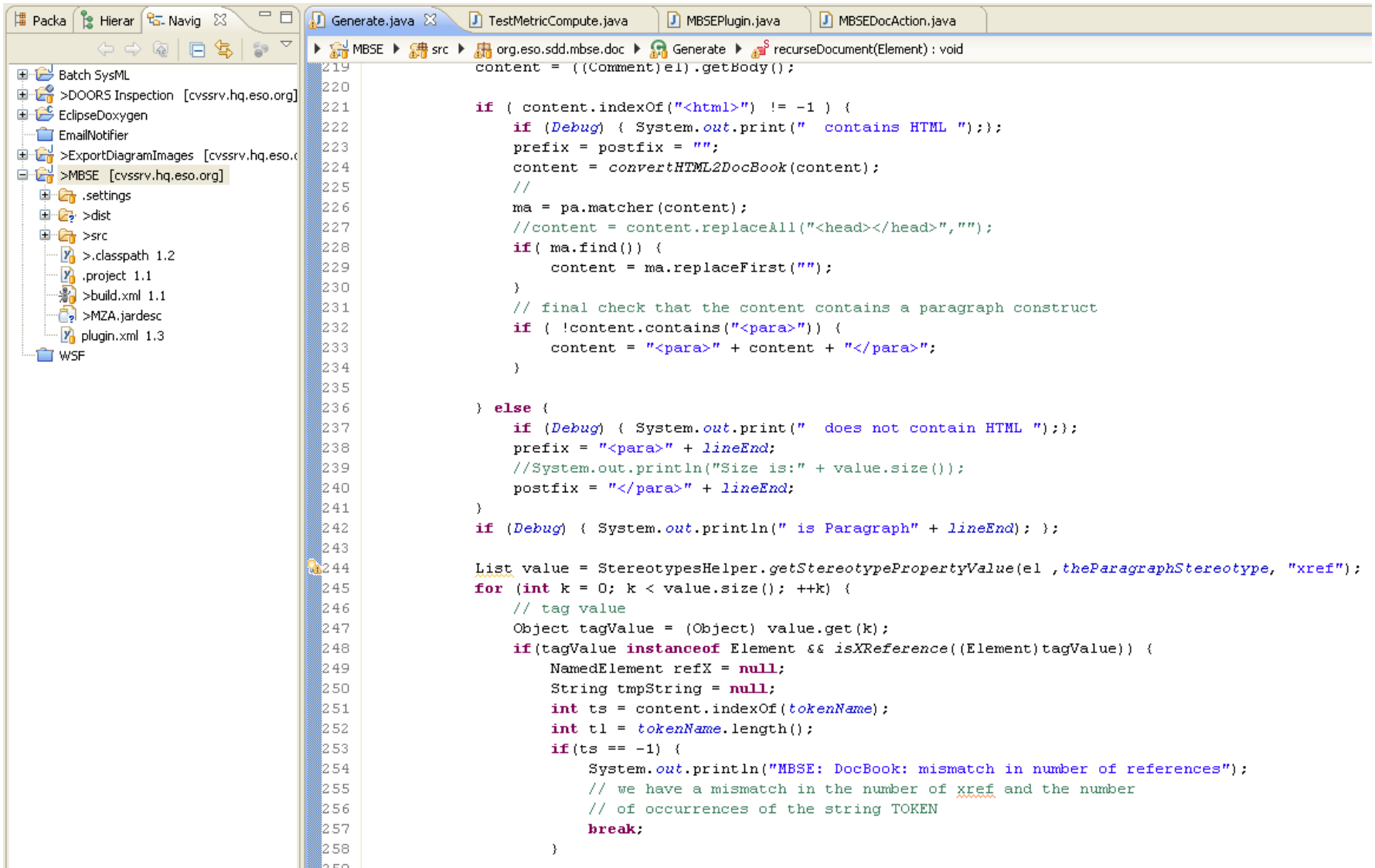
- Create a model of the document
- Reference existing system model elements by unique identifiers
- Document model and system model may be aggregated in same project model
- A subset of the DocBook ontology is mapped to model elements using a dedicated SysML profile



# Implementation

- Use standard documentation format:
  - DocBook, maintained by OASIS, <http://www.oasis-open.org/home/index.php>
- Use SysML profiles to introduce DocBook elements in SysML models, in a tool independent fashion
- Use customization facilities provided by SysML Tool vendor to write code for export

# Software



The image shows a screenshot of an IDE with a project explorer on the left and a code editor on the right. The project explorer shows a project named 'MBSE' with various sub-projects and files. The code editor displays a Java file named 'Generate.java' with the following code:

```
219 content = ((Comment)e1).getBody();
220
221
222 if ( content.indexOf("<html>") != -1 ) {
223     if (Debug) { System.out.print(" contains HTML ");};
224     prefix = postfix = "";
225     content = convertHTML2DocBook(content);
226     //
227     ma = pa.matcher(content);
228     //content = content.replaceAll("<head></head>","");
229     if( ma.find() ) {
230         content = ma.replaceFirst("");
231     }
232     // final check that the content contains a paragraph construct
233     if ( !content.contains("<para>") ) {
234         content = "<para>" + content + "</para>";
235     }
236
237 } else {
238     if (Debug) { System.out.print(" does not contain HTML ");};
239     prefix = "<para>" + lineEnd;
240     //System.out.println("Size is:" + value.size());
241     postfix = "</para>" + lineEnd;
242 }
243
244 if (Debug) { System.out.println(" is Paragraph" + lineEnd); };
245
246 List value = StereotypesHelper.getStereotypePropertyValue(e1 ,theParagraphStereotype, "xref");
247 for (int k = 0; k < value.size(); ++k) {
248     // tag value
249     Object tagValue = (Object) value.get(k);
250     if(tagValue instanceof Element && isXReference((Element)tagValue)) {
251         NamedElement refX = null;
252         String tmpString = null;
253         int ts = content.indexOf(tokenName);
254         int tl = tokenName.length();
255         if(ts == -1) {
256             System.out.println("MBSE: DocBook: mismatch in number of references");
257             // we have a mismatch in the number of xref and the number
258             // of occurrences of the string TOKEN
259             break;
260         }
261     }
262 }
```

# Usage

- Stereotype tags are used to reference to system element models
- Text is entered in comments (which can be visualized in diagrams)
- References to model element or diagrams are selected using the tool navigation facilities

# Usage Example: adding a chapter

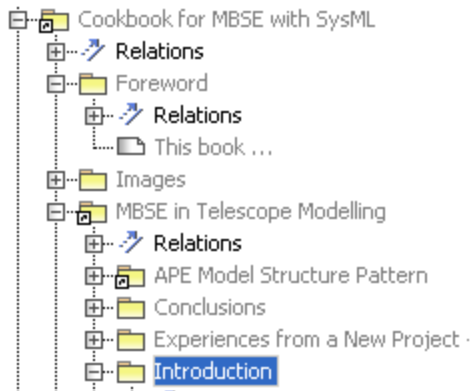
Adding a chapter B to a book A

1. Create a package B for the chapter, give it a name
2. Stereotype package B by «chapter»
3. Modify properties of «book» element A, by adding a reference to B, .e.g. modifying the tagged value to include B in its list



# Internal References

- Unique elements ID are generated following containment tree hierarchy

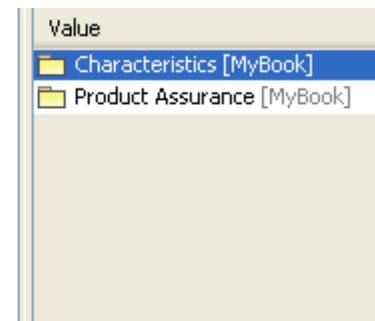
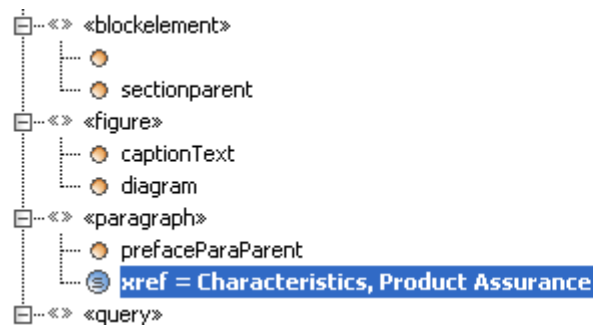
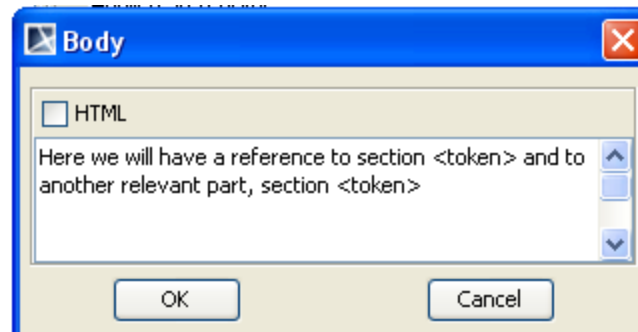


<chapter

```
xml:id="APE_Project_Views__Cookbook__CookbookforMBSE  
withSysML__MBSEinTelescopeModelling__Introduction" >
```

# Usage: adding internal reference

Paragraphs may contain references to book elements



# Shortcomings

- Missing validation rules (but could be added)
- Difficult deployment
- Dependency on API version (potentially)
- The insertion of document hierarchies is tedious without a WYSIWIG editor or similar

# Shortcomings 2

- which subset of DocBook should be supported?
- creating cross-references (also internal ones) is possible but a bit cumbersome
- comments are elements without a name and therefore a bit difficult to find in the model browser
- [MagicDraw's](#) text editor supports only HTML which is not compliant with DocBook and the text has to be transformed (which is done automatically when generating)

# Advantages

- It adds advantages for configuration control since different document parts may be modified concurrently by different authors (assumes a version control system for underlying model from vendor's side)

# Using comments

- Pros and Cons of using the *comment* element:
  1. Comments can be visualized directly in a SysML diagram
  2. Comments do not have a name, which makes it difficult to identify them.

# Document Example

APE-DocBook.xml

book info

## Cookbook for MBSE with SysML

### Foreword

This book contains guidelines, recipes, and best practices for Model Based Systems Engineering (MBSE) with the Systems Modeling Language (SysML) on the basis of a interdisciplinary model of a complex real-world project, a high-tech opto-mechatrical system called the Active Phasing Experiment (APE)

The book addresses:

- Model structure and overview
- Objectives and Requirements
- Context, System Structure
- Behavior and Data
- Verification
- Model library and domain specific profiles

## Part: MBSE in Telescope Modelling

### Chapter 1: Introduction

In the framework of INCOSE's strategic initiative, the Systems Engineering Vision 2020, one of the main areas of focus is model-based systems engineering. In keeping with this emphasis, the European Southern Observatory (ESO; <http://www.eso.org/>) is collaborating with the German Chapter of INCOSE (<http://www.gfse.de/>) in the form of an "MBSE Challenge" team. The team's task is to demonstrate solutions to challenging problems using MBSE. The Active Phasing Experiment (APE; see Gonte et al. 2004), a European Union Framework Program 6 project, was chosen as the subject of the SE<sup>2</sup> Challenge Team (<http://mbse.gfse.de/>). Many technical products in the telescope domain show an increasing integration of mechanics with electronics,

# Advantages

- consistent integration of system model and system documentation
- direct linking to model elements (also diagrams) from the document
- changes of diagram names is automatically reflected in the document
- using proper definition of the stereotype associations only compatible elements can be selected to compose the document, e.g.
- a figure references diagrams, a chapter references paragraphs.
- the documentation is at the same time navigable in the model and printable
- documents are modeled in a tool independent way. Only a small plugin is needed to generate DocBook XML.