

31. Documentation as Synchronized Dependent Model in a Macromodel

Documentation Generation as App for RAG

Prof. Dr. rer. nat. Uwe Aßmann
Institut für Software- und
Multimediatechnik
Lehrstuhl Softwaretechnologie
Fakultät für Informatik
TU Dresden
<http://st.inf.tu-dresden.de>
Version 21-0.3, 08.01.22

- 1) Tasks
- 2) Template-Driven Documentation Tools
- 3) Literate Programming
- 4) Elucidative Modeling and Documentation Tools
- 5) Web-based API Documentation Generators

References

- ▶ D. E. Knuth, Literate Programming, *The Computer Journal*, Volume 27, Issue 2, 1984, Pages 97–111, <https://doi.org/10.1093/comjnl/27.2.97>
- ▶ D. Cordes and M. Brown, "The literate-programming paradigm," in *Computer*, vol. 24, no. 6, pp. 52-61, June 1991, doi: 10.1109/2.86838.
- ▶ Kurt Nørmark. Elucidative programming. *Nordic Journal of Computing*, 2000. Citeseer: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.408.2506&rep=rep1&type=pdf>
- ▶ C. Wilke, A. Bartho, J. Schroeter, S. Karol, U. Aßmann. Elucidative Development for Model-Based Documentation and Language Specification (Extended Version). Technische Universität Dresden. Institut für Software- und Multimediatechnik. Technical Reports TUD-FI12-01-Januar 2012, ISSN 1430-211X.
 - <http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-83442>
- ▶ Andreas Bartho. Elucidative Modeling. PhD thesis, Technische Universität Dresden, Fakultät Informatik, May 2014.
 - <http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-208060>
 - <https://www.linkedin.com/p/in/andreas-bartho/ba/922/8a4?trk=pub-pbmap>



Interesting

3 Model-Driven Software Development in Technical Spaces (MOST)

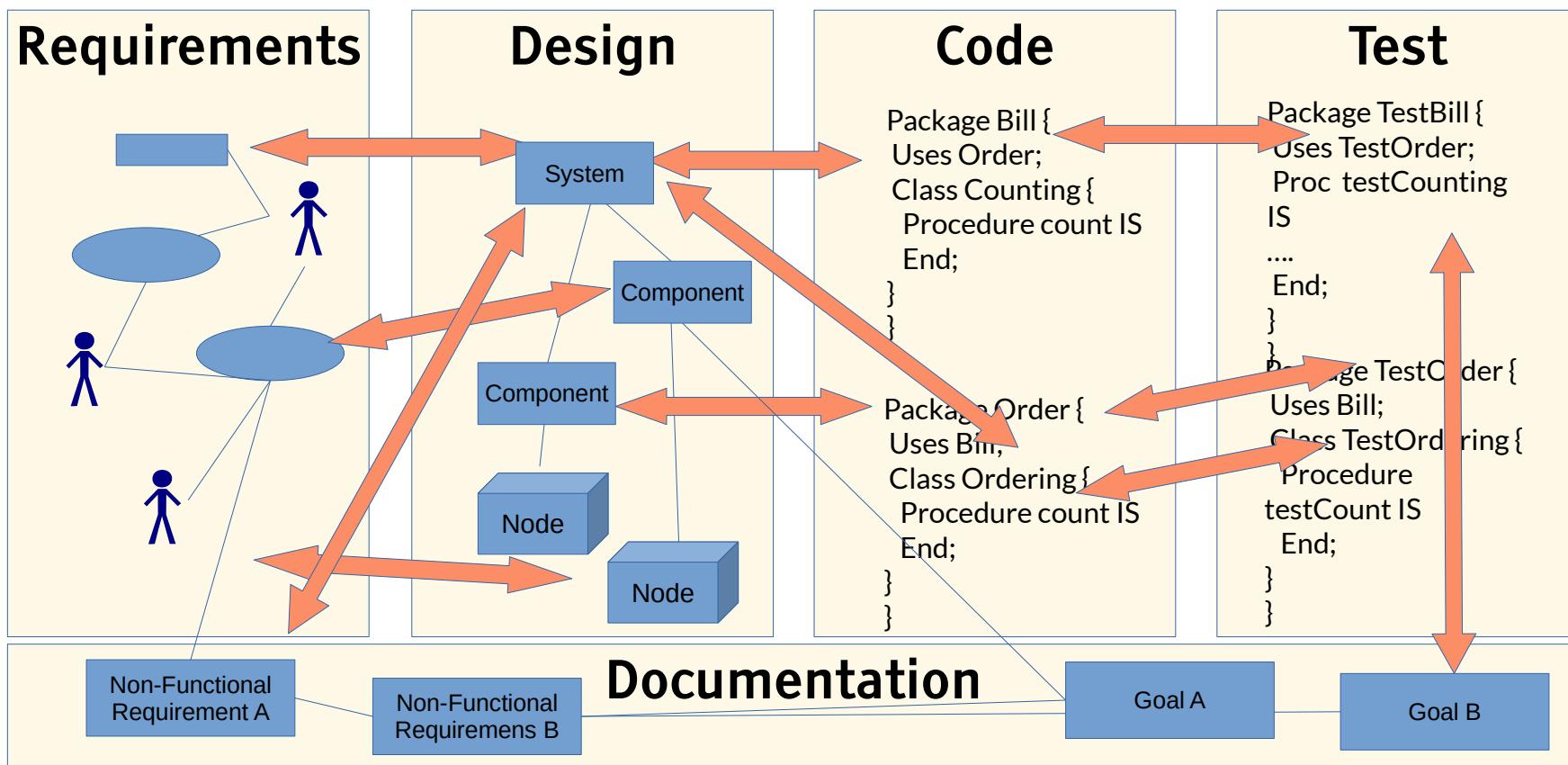
- ▶ <https://www.writethedocs.org/> is a conference for documentation practitioners
- ▶ <https://waset.org/software-implementation-and-software-documentation-conference>

31.1 Tasks of Documentation Tools

http://en.wikipedia.org/wiki/Software_documentation

Q12: The ReDoDeCT Problem and its Macromodel

- ▶ The **ReDoDeCT problem** is the problem how requirements, documentation, design, code, and tests are related (→ V model)
- ▶ Mappings between the Requirements model, Documentation files, Design model, Code, Test cases
- ▶ A **ReDoDeCT macromodel** has maintained mappings between all 5 models



Basics of Software Documentation

6 Model-Driven Software Development in Technical Spaces (MOST)

- ▶ Documentation is a means of **communication** to keep software alive
 - between developers and future developers
 - between coders and testers
 - between developers and managers (for reviews and audits)
- ▶ Problems:
 - Documentation *ages* because code is modified and evolved (**documentation aging**)
 - Good documentation costs time and money
- ▶ Different kinds of documentation:
 - **Generated documentation** is derived from code and models
 - **Integrated Documentation** is derived from the code (e.g., in comments), e.g., JavaDoc
 - **Elucidative Documentation**, derives both from another and keeps it consistent (generative or round-trip engineering)
- ▶ Standards:
 - national DIN 66230, 66231, 66232, 66270(1998)
 - international ISO/IEC 6592(2000), ISO/IEC 18019(2004)

Without documentation, a program is not software

Taxonomy of Documentation Documents

- ▶ **User documentation** (Benutzerdokumentation) explains the program to end users
 - Tutorials, user handbook, online documentation
- ▶ **System documentation** for installation, test cases, code documentation, maintenance, operations
 - **API documentation** documents interfaces of the system or framework, to let programmers use them for writing apps
 - **Architecture documentation** to highlight the architectural structure of the software, e.t., with arc42 (<https://www.arc42.de/>)
- ▶ **Project documentation**
 - Developer documentation
 - Project documentation (project plan, requirements specification, status reports, after study)
- ▶ **Quality documentation**
 - Test-, review, audit documentation
- ▶ **Process documentation**
 - Standards, processes

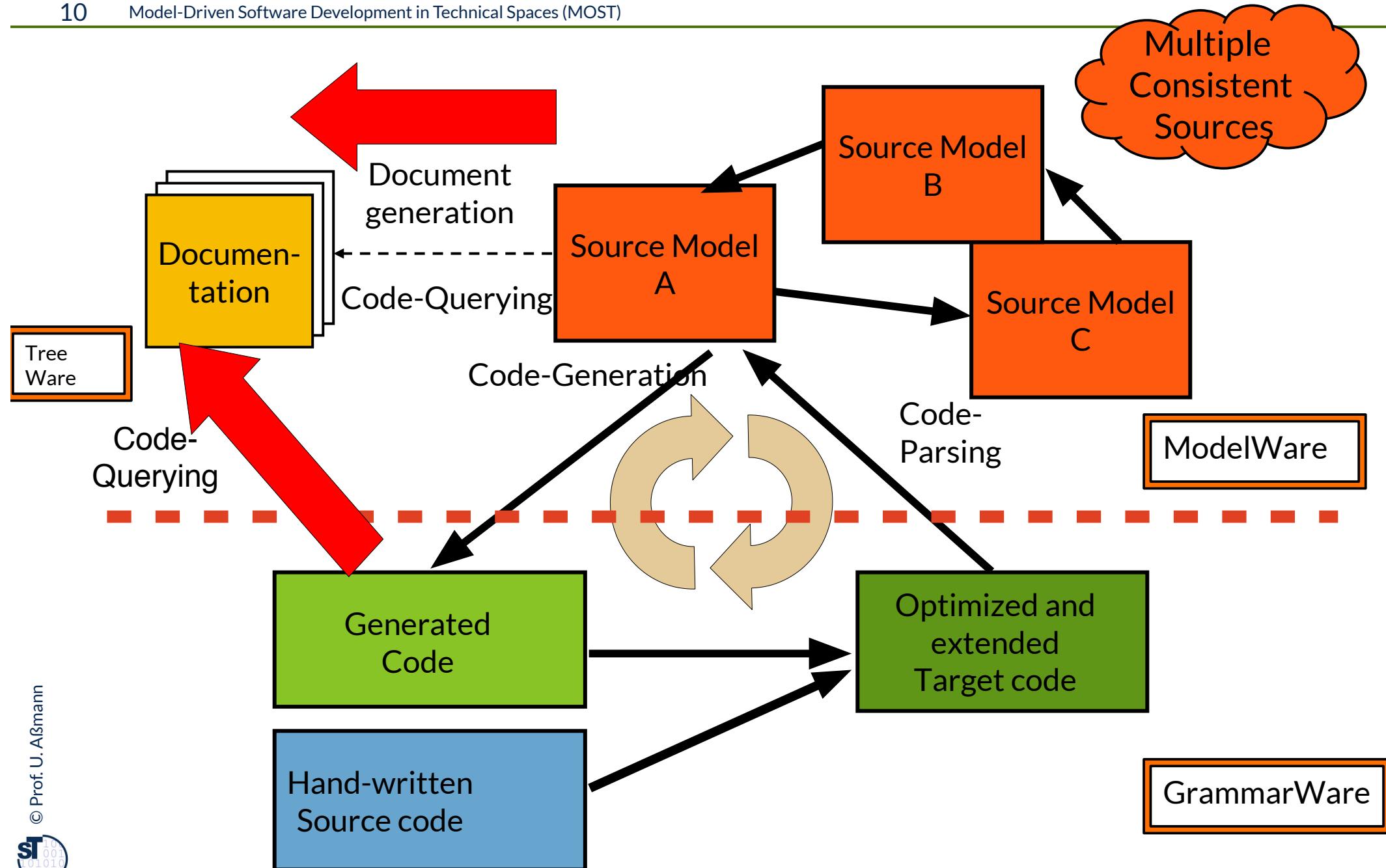
Tasks of Documentation Tools

- ▶ Basically, documentation generation is similar to code generation. Documentation is created in higher-order attributes on a link tree by a RAG
- ▶ **Documentation generation is an application areas for RAG**
- ▶ **Generation** of derived documents from code and models
 - Generation of Word (docx), LibreOffice (odt), rtf, xml, html formats
 - Generation of figures (svg, png, pdf)
 - Generation of snippets and generic snippets
 - Back-linking to originals
- ▶ **Filling** of documentation templates (with the hedge-principle)
- ▶ **Parameterization** with layouts
 - via css-style sheets

31.2 Generative, Template-Driven Documentation Tools

.. Documentation derived from code and models, based on template-based code generation

Macromodel Principle and Round-Trip Engineering



Documentation Tool JavaDoc is a Template Expander

- ▶ JavaDoc reads Java source code and extracts html from the code comments, based on **html templates**
 - Typical hedge-based code generation with generic snippets
- ▶ Generation of additional contents and indices
- ▶ Controlled by Java metadata attributes
 - @author, @date, @param
- ▶ Layouting via plugin classes called *doclets*
- ▶ JavaDoc has been realized for all programming languages

JavaDoc is a Typical HRAG Application

- ▶ The html documentation is computed in a higher-order synthesized attribute `htmldoc : HTML`

```
// schematic, synthesis from bottom to top
Interpretation javaDoc(Tree → Tree) {
    Attributions of Root(classes[]) {
        this.htmldoc := map + classes.htmldoc;
        <println(„Result is %S“, this.htmldoc)>
    }
    Attributions of Class(superclass:Class,methods{}) {
        this.htmldoc := <superclass.Name + methods.htmldoc;
    }
    Attributions of Method(name,comment) {
        this.htmldoc := „<h1>“+name+“</h2>“+comment.htmldoc;
    }
    Attributions of Comment(text) {
        this.htmldoc := text;
    }
}
```

Composition of Separated Hand-Written and Generated Documentation Snippets

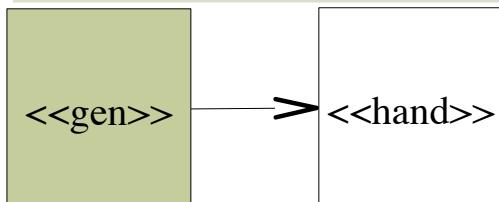
In separate files: Coupling by “include”

- ▶ Only possible if document format supports subdocument inclusion
 - e.g., TeX or Framemaker

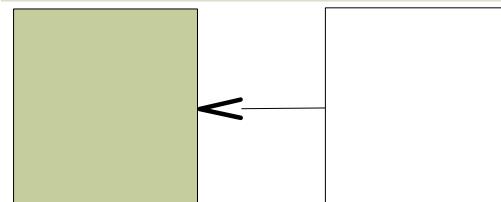
In one file:

Coupling with `hegdes` (Trennmarkierung)

Generated Delegator



Generated Delegatee



Generated Wrapper

```
/** Generated documentation  
**/
```

/* *** Hedge *** /

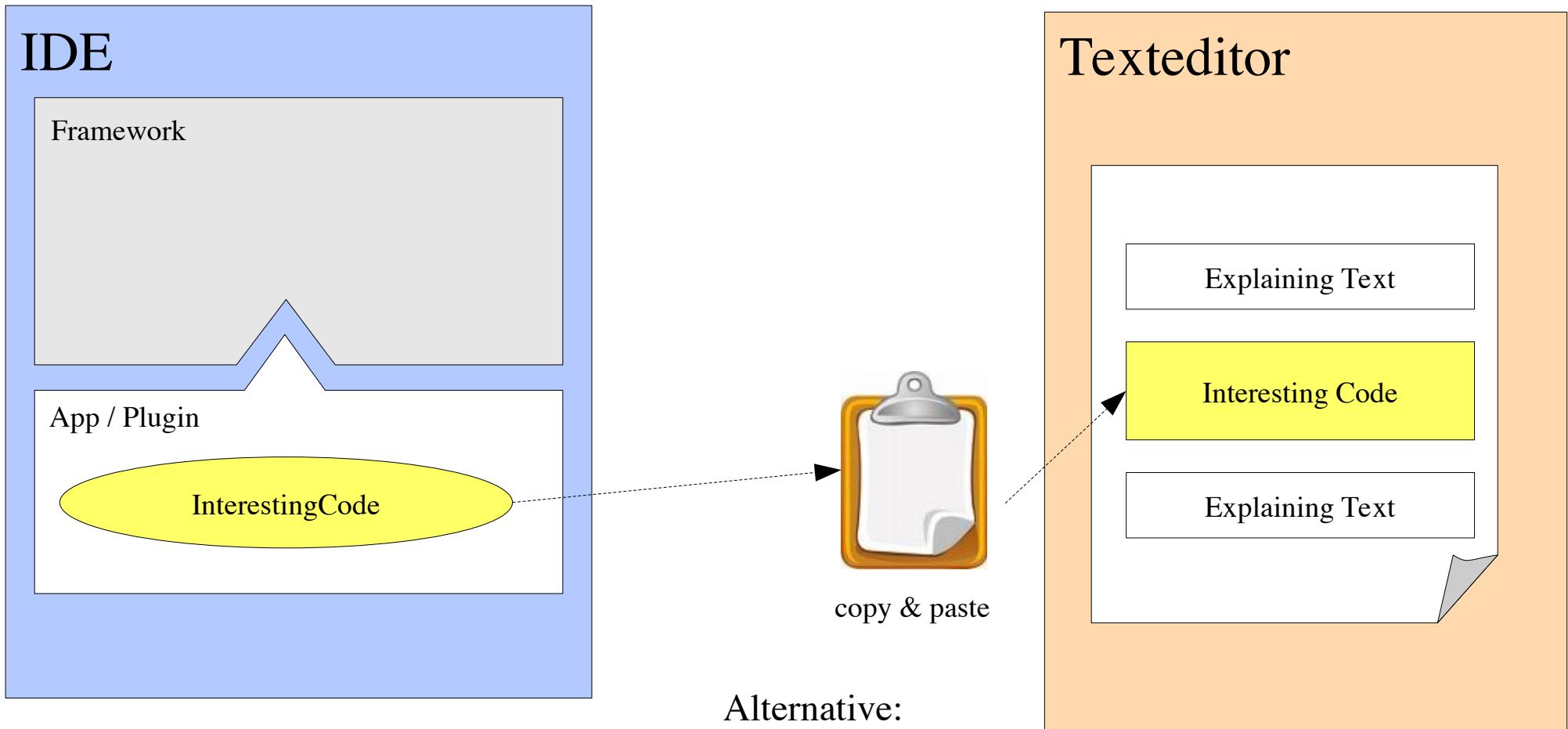
... Hand-written Documentation

/* *** Hedge *** /

31.3 Literate Programming

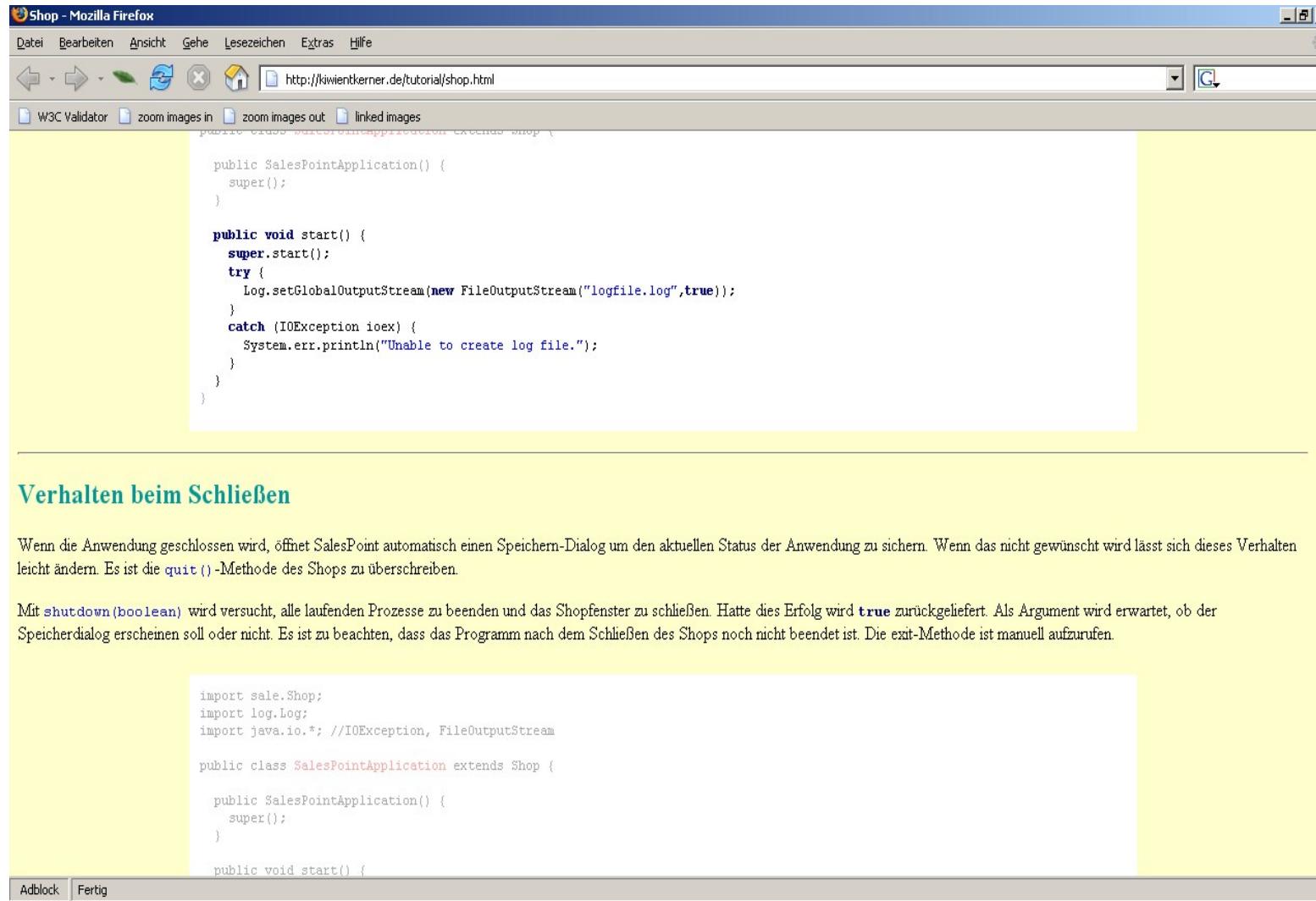
- They integrate code, models and documentation by **separating code from documentation**

Classic: Manual Writing of Tutorials



Alternative:
Code query
e.g., with
Xcerpt or QL

How to Write Integrated Documentation and Tutorials?



The screenshot shows a Mozilla Firefox browser window with the title "Shop - Mozilla Firefox". The address bar displays the URL <http://kiwientkerner.de/tutorial/shop.html>. Below the address bar, there are several developer tools icons: W3C Validator, zoom images in, zoom images out, and linked images. The main content area of the browser shows a Java code snippet:

```
public class SalesPointApplication extends Shop {  
    public SalesPointApplication() {  
        super();  
    }  
  
    public void start() {  
        super.start();  
        try {  
            Log.setGlobalOutputStream(new FileOutputStream("logfile.log",true));  
        }  
        catch (IOException ioex) {  
            System.err.println("Unable to create log file.");  
        }  
    }  
}
```

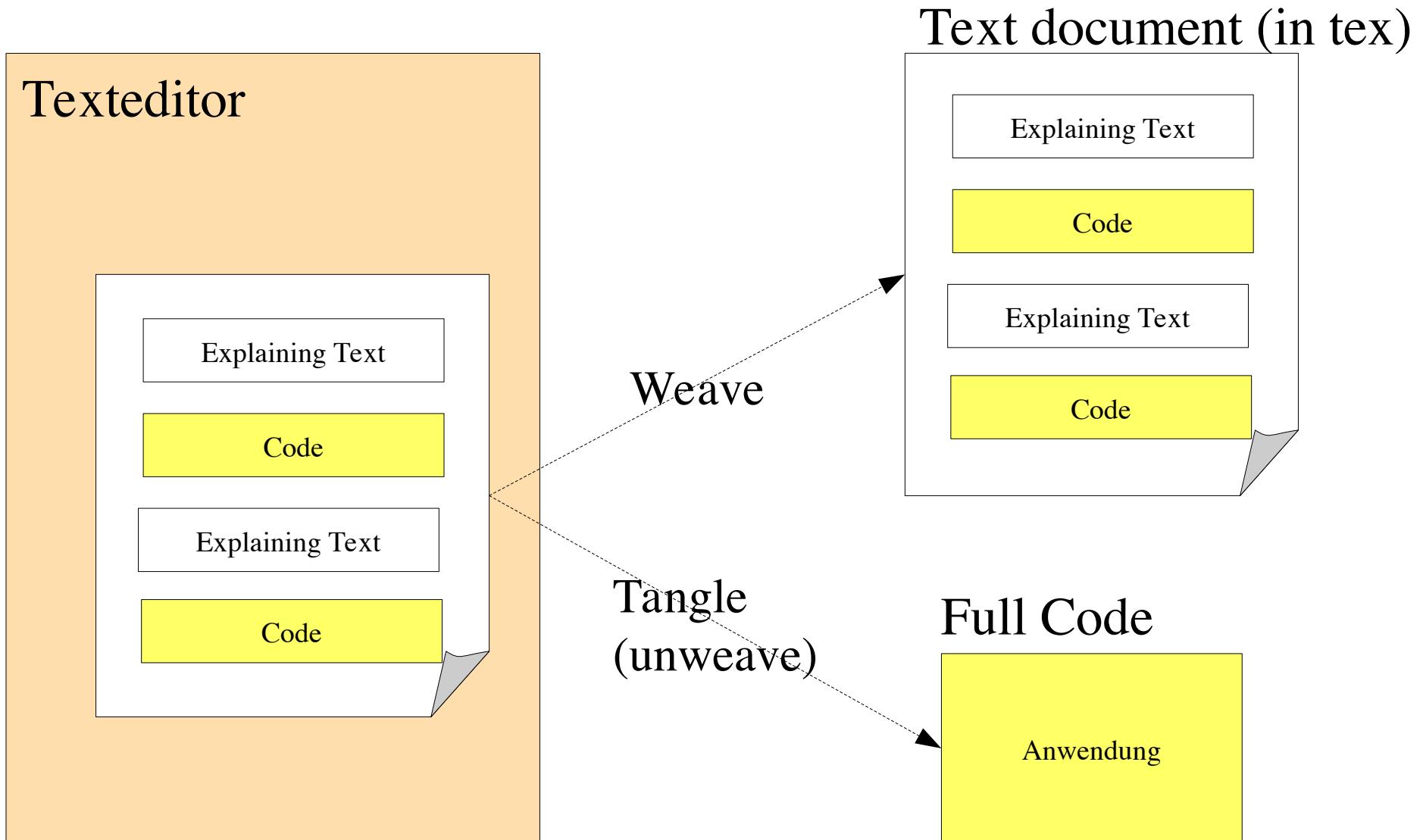
Verhalten beim Schließen

Wenn die Anwendung geschlossen wird, öffnet SalesPoint automatisch einen Speichern-Dialog um den aktuellen Status der Anwendung zu sichern. Wenn das nicht gewünscht wird lässt sich dieses Verhalten leicht ändern. Es ist die `quit()`-Methode des Shops zu überschreiben.

Mit `shutdown(boolean)` wird versucht, alle laufenden Prozesse zu beenden und das Shopfenster zu schließen. Hatte dies Erfolg wird `true` zurückgeliefert. Als Argument wird erwartet, ob der Speicherdialog erscheinen soll oder nicht. Es ist zu beachten, dass das Programm nach dem Schließen des Shops noch nicht beendet ist. Die `exit`-Methode ist manuell aufzurufen.

```
import sale.Shop;  
import log.Log;  
import java.io.*; //IOException, FileOutputStream  
  
public class SalesPointApplication extends Shop {  
  
    public SalesPointApplication() {  
        super();  
    }  
  
    public void start() {  
    }
```

[Knuth] Literate Programming by Code Unweaving



Literate Programming

[The program text below specifies the “expanded meaning” of ‘⟨Program to print . . . numbers 2⟩’; notice that it involves the top-level descriptions of three other sections. When those top-level descriptions are replaced by their expanded meanings, a syntactically correct PASCAL program will be obtained.]

⟨Program to print the first thousand prime
numbers 2⟩ ≡
program *print_primes*(*output*);
 const *m* = 1000;
 ⟨Other constants of the program 5⟩
 var ⟨Variables of the program 4⟩
 begin ⟨Print the first *m* prime numbers 3⟩;
 end.

[Literate Programming
von Donald E. Knuth]

- ▶ The TeX engine is programmed literately
- ▶ Overview: <http://www.literateprogramming.com/>
- ▶ OMNotebook/DrModelica: <http://www.modelica.org/tools>

OMNotebook – Literate Programming with DrModelica

The screenshot shows the OMNotebook application window titled "OMNotebook: Exercise2-classes.onb". The main content area displays the following text:

Exercise 2 - Instances

1 Question
What is an instance?

1.1 Answer

2 Creating Instances

```
class Dog
  constant Real legs = 4;
  parameter String name = "Dummy";
end dog;
```

Ok

Create an instance of the class Dog.

Create another instance and give the dog the name "Tim".

2.1 Answer

Ready Ready Ln 1, Col 1

- ▶ Linked documents with interactive exercises
- ▶ Inspired by DrScheme und DrJava, learning tools for Scheme resp. Java
- ▶ www.openmodelica.org

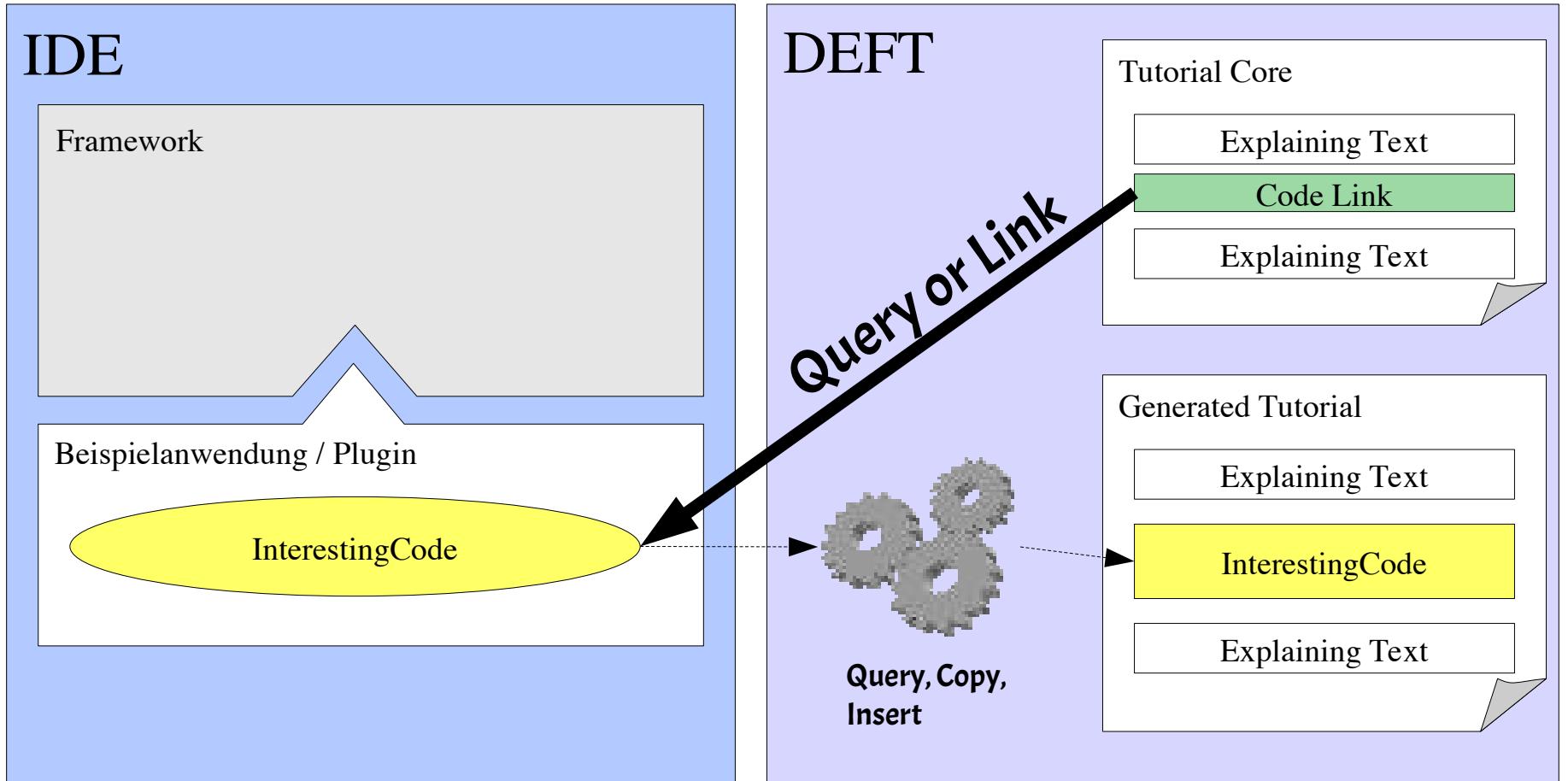
31.4 Elucidative Documentation Tools

- They link code, models and documentation by **model and code mapping**
- and renew the documentation by ***hot updates***

Elucidative Programming Links Documentation with Queries to Code

21

Model-Driven Software Development in Technical Spaces (MOST)

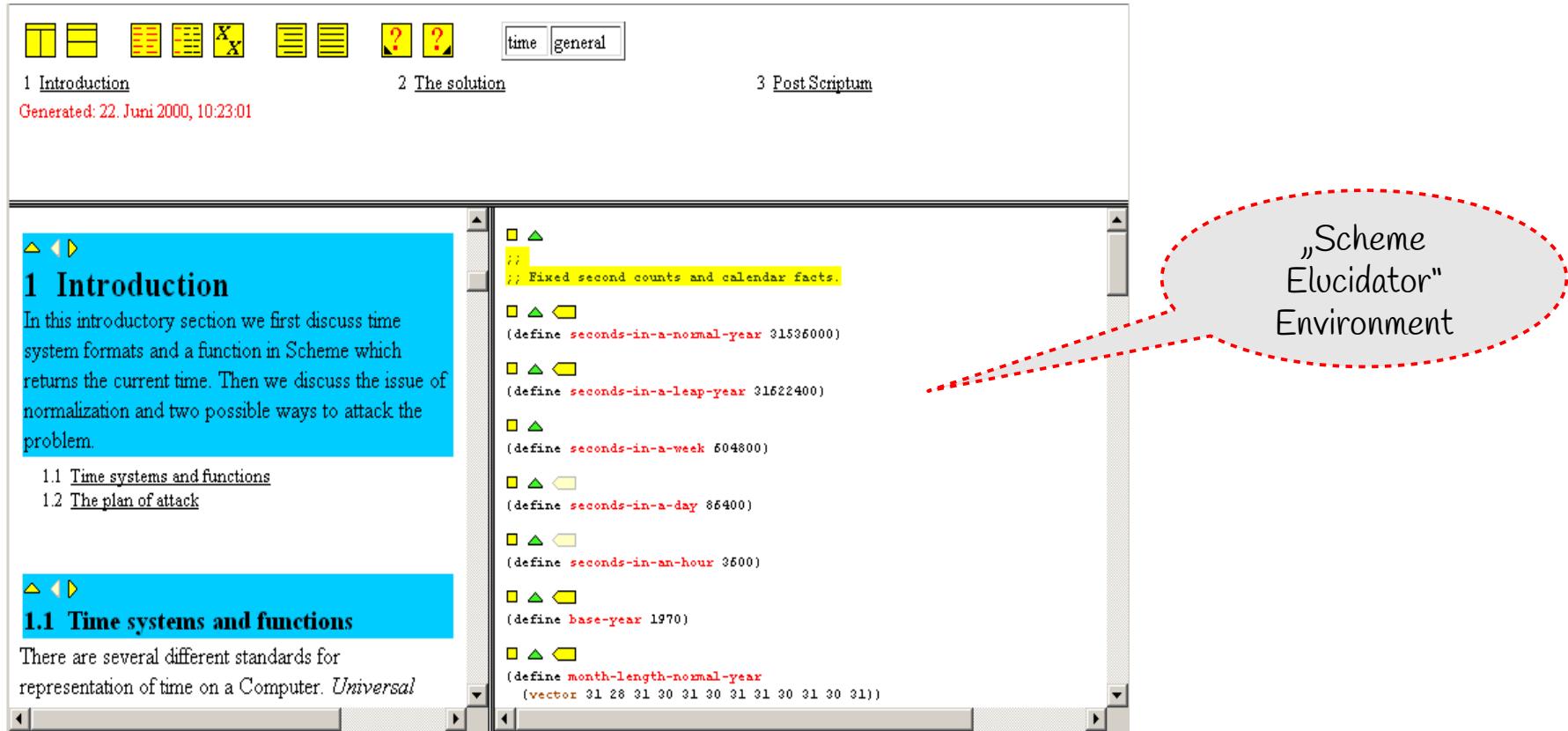


hot update
(hot synchronisation):
round-trip engineering

Elucidative Programming

22

Model-Driven Software Development in Technical Spaces (MOST)



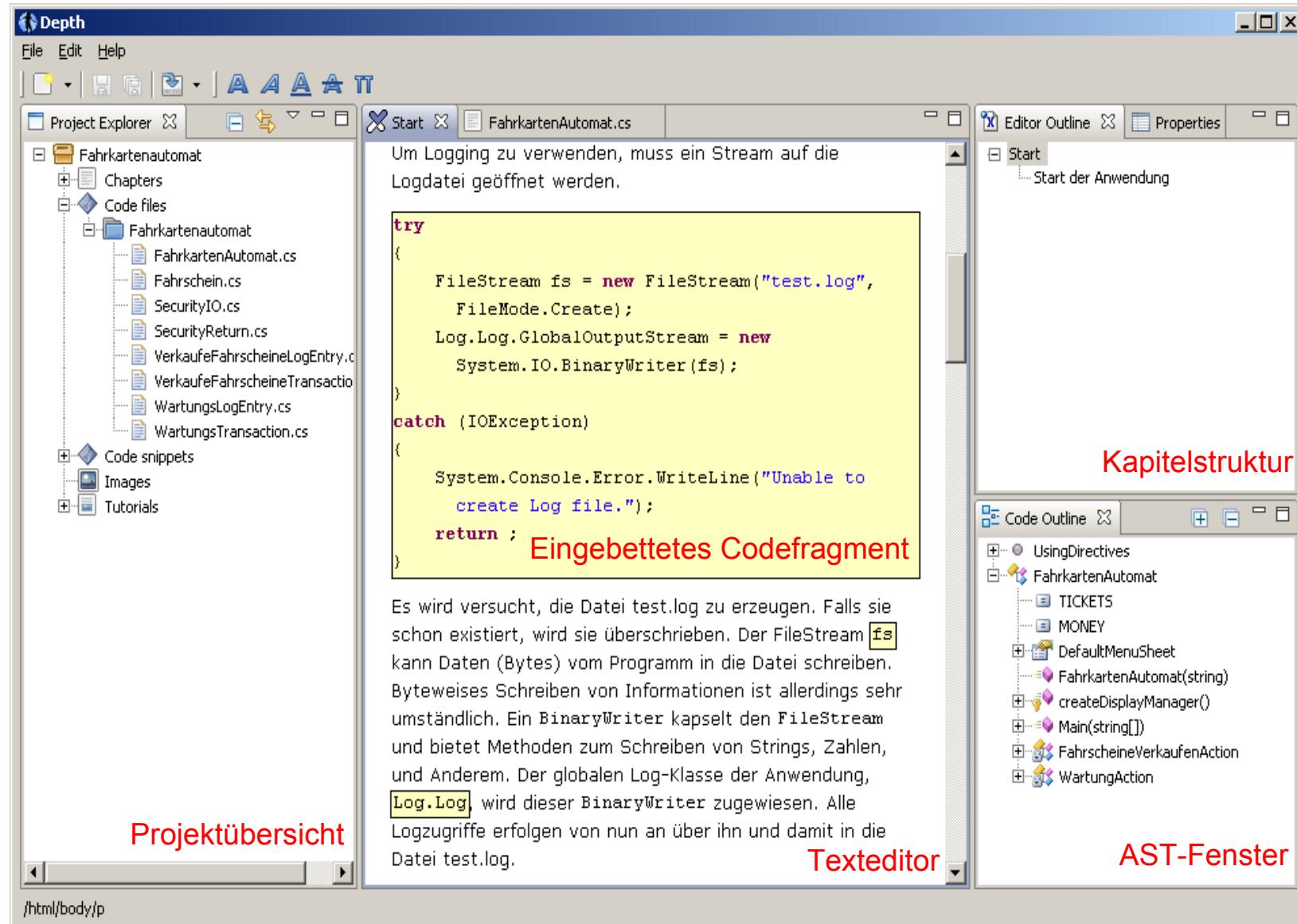
- ▶ Elucidative Programming shows documentation and code in parallel
- ▶ <http://www.cs.aau.dk/~normark/elucidative-programming>
- ▶ <http://deftproject.org>

hot update
(hot synchronisation):
round-trip engineering

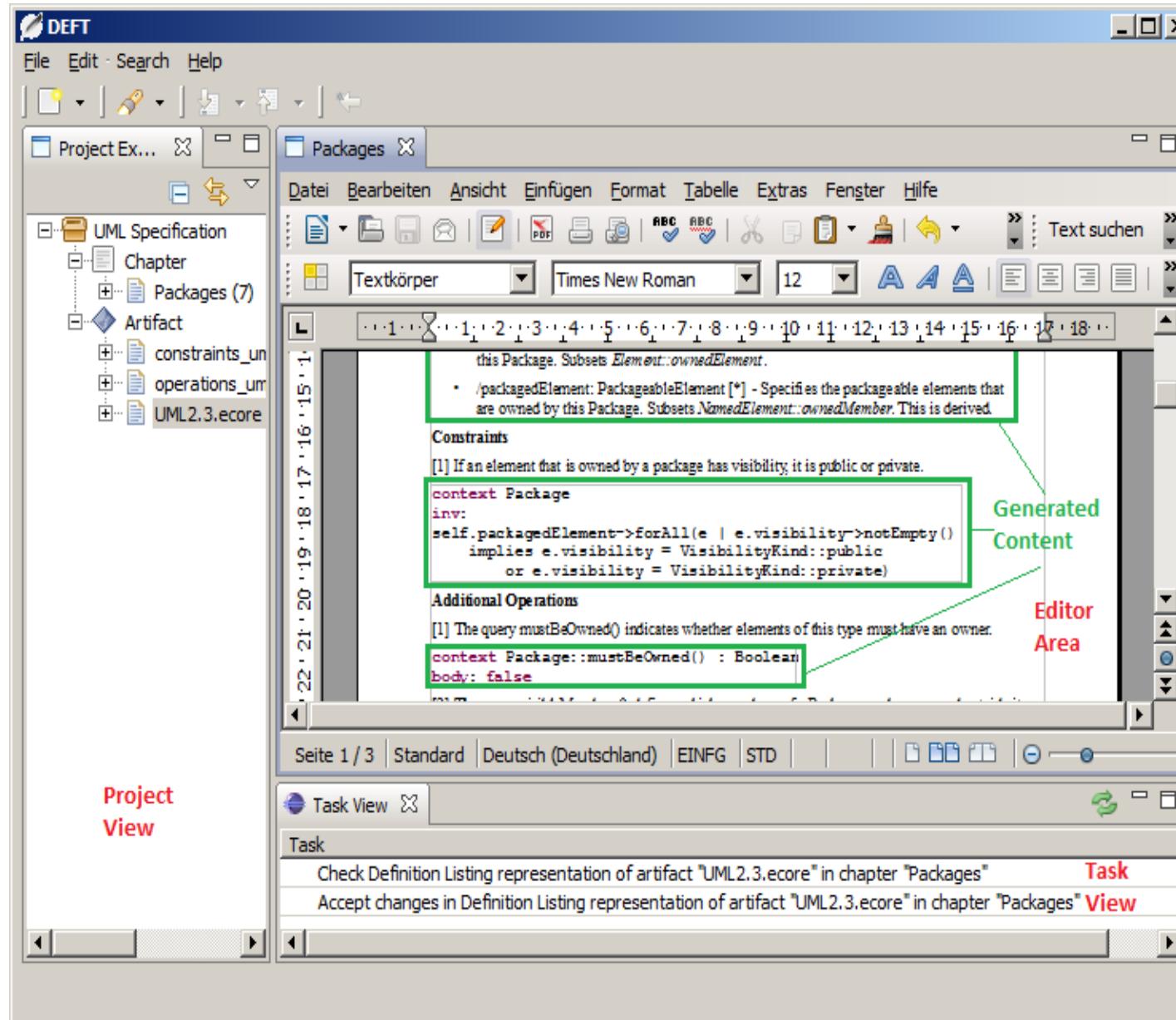
Development Environment For Tutorials (DEFT) www.deftproject.org)

23

Model-Driven Software Development in Technical Spaces (MOST)



Embedding UML Constraints for UML Models into Documentation



Development Environment For Tutorials (DEFT)

- ▶ Eclipse RCP application, language independent
- ▶ Management of code, models and text
- ▶ Prettyprinting of code fragments from code templates
- ▶ Hot update of generated documentation
 - Automatic update of embedded code fragments
 - Notification if code fragments have changed

Generated HTML Tutorial

26

Model-Driven Software Development in Technical Spaces (MOST)

Start der Anwendung

In der Klasse `Fahrkartenautomat` befindet sich die `Main`-Methode, mit der sich das Programm starten lässt. Dort werden Daten initialisiert und der Fahrkartenautomat instantiiert.

Logging

Der erste Schritt ist die Konfiguration des Loggings. Das SalesPoint-Framework bietet Funktionen und Datentypen an, mit denen Aktionen geloggt werden können. Es gibt GUI-Komponenten, mit denen die Inhalte des Logs wieder nutzerfreundlich angezeigt werden können. Eine Anzeige des Logs ist derzeit nicht im Fahrkartenautomaten implementiert, geloggt wird aber trotzdem schon.

Um Logging zu verwenden, muss ein Stream auf die Logdatei geöffnet werden.

```
try
{
    FileStream fs = new FileStream("test.log", FileMode.Create);
    Log.Log.GlobalOutputStream = new System.IO.BinaryWriter(fs);
}
catch (IOException)
{
    System.Console.Error.WriteLine("Unable to create Log file.");
    return;
}
```

Es wird versucht, die Datei test.log zu erzeugen. Falls sie schon existiert, wird sie überschrieben. Der `FileStream` kann Daten (Bytes) vom Programm in die Datei schreiben. Byteweises Schreiben von Informationen ist allerdings sehr umständlich. Ein `BinaryWriter` kapselt den `FileStream` und bietet Methoden zum Schreiben von Strings, Wahlen, und Anderem. Der globale Log-Klasse der Anwendung, `Log.Log`, wird dieser `BinaryWriter` zugewiesen. Alle

```

(
)

protected override DisplayManager createDisplayManager()
{
    Size d = System.Windows.Forms.Screen.PrimaryScreen.Bounds.Size;
    Point tempAux = new Point((d.Width - 100) / 2, (d.Height - 80) / 2);
    Point tempAux2 = new Point(5, 5);
    return new AWTDisplayManager(this, ref templux, ref templux2);
}

[STAThread]
public static void Main(string[] args)
{
    //System initialisieren
    try
    {
        FileStream fs = new FileStream("test.log", FileMode.Create);
        Log.Log.GlobalOutputStream = new System.IO.BinaryWriter(fs);
    }
    catch (IOException)
    {
        System.Console.Error.WriteLine("Unable to create Log file.");
        return;
    }

    // Kataloge anlegen

    // Fahrscheinkatalog
    Catalog cTickets = Catalog.forName(TICKETS);

    cTickets.addItem(new Fahrschein("Einzelfahrt", 300));
    cTickets.addItem(new Fahrschein("Sammelfahrschein", 1500));
    cTickets.addItem(new Fahrschein("ermäßigte Einzelfahrt", 150));
}
```

31.5 Web-based Documentation Generators based on Markdown

Sphinx and the Documentation Cloud readthedocs.org

- ▶ **readthedocs** is a cloud for documentation projects
- ▶ supporting two documentation generators **sphinx** and **mkdocs**

The screenshot shows the homepage of the [Read the Docs](https://readthedocs.org) website. The header features the logo "Read the Docs" with the tagline "Dokumentation erstellen, hosten und durchsuchen." Below the header, a large banner states "Technical documentation lives here" and explains that Read the Docs simplifies software documentation by automating building, versioning, and hosting. The page is divided into several sections: "Free docs hosting for open source" (describing free hosting for over 100,000 projects), "Always up to date" (describing automatic builds from version control), "Downloadable formats" (describing support for PDFs, HTML, and eReaders), and "Multiple versions" (describing the ability to have multiple versions of documentation). At the bottom, there are links to various PDF files and a "Alle anzeigen" button.

Sphinx

- ▶ Architecture documentation
- ▶ User documentation
- ▶ Files in formats restructuredText and Markdown are transformed to HTML
- ▶ Treats entire directories
- ▶ many output formats (e.g., Latex)
- ▶ Can be coupled with Javadoc or similar API doc generators

The screenshot shows a web browser window displaying the Sphinx documentation on the Read the Docs platform. The URL in the address bar is docs.readthedocs.io/en/stable/intro/getting-started-with-sphinx.html. The page title is "Getting Started with Sphinx". On the left, there is a sidebar with a dark background containing a list of links related to Sphinx and the Read the Docs platform. The main content area features a large heading "Getting Started with Sphinx" followed by a paragraph of text and a bulleted list of features. Below this, there is a "Quick start" section with a "See also" link and some instructions. At the bottom, there is a terminal-like interface showing the command `$ pip install sphinx` and a note about creating a directory for documentation.

Example Sphinx Project

► Petrinet compiler
Reconfnet

<https://petrinets.pages.st.inf.tu-dresden.de/adaptive-petrinets/index.html>

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Reconfnet" and displays the content of the URL <https://petrinets.pages.st.inf.tu-dresden.de/adaptive-petrinets/index.html>. The page content includes a sidebar with navigation links such as Installation, Getting started, Examples, Meta-models, Stages, Getting started, Used libraries, Roadmap, Authors, and API Documentation. The main content area features a heading "Reconfnet" and a sub-section "We are:" containing a diagram of a Petri net. Below this, a paragraph describes Reconfnet as a compiler for SCROLL code, context-role Scala library, adaptive petrinets (APN), dynamic petrinets (DPN), and dynamic context-adaptive petrinets (DAPN) to synthesize VHDL machine code using model-checkable (dynamic) Petri nets as intermediate meta models. Another paragraph states that Reconfnet is based on a graph rewriting transformation chain in GrGen.net (www.grgen.net). A "Last changes" section lists three commits from 2020-2021, and a "Usage" section lists several links to documentation pages.

Reconfnet

We are:

Reconfnet is a compiler compiling SCROLL code, the context-role Scala library, and adaptive petrinets (APN), dynamic petrinets (DPN) and dynamic context-adaptive petrinets (DAPN) to synthesizable VHDL machine code, using model-checkable (dynamic) petri nets as intermediate meta models.

Reconfnet is based on a graph rewriting transformation chain in [GrGen.net](www.grgen.net).

Last changes

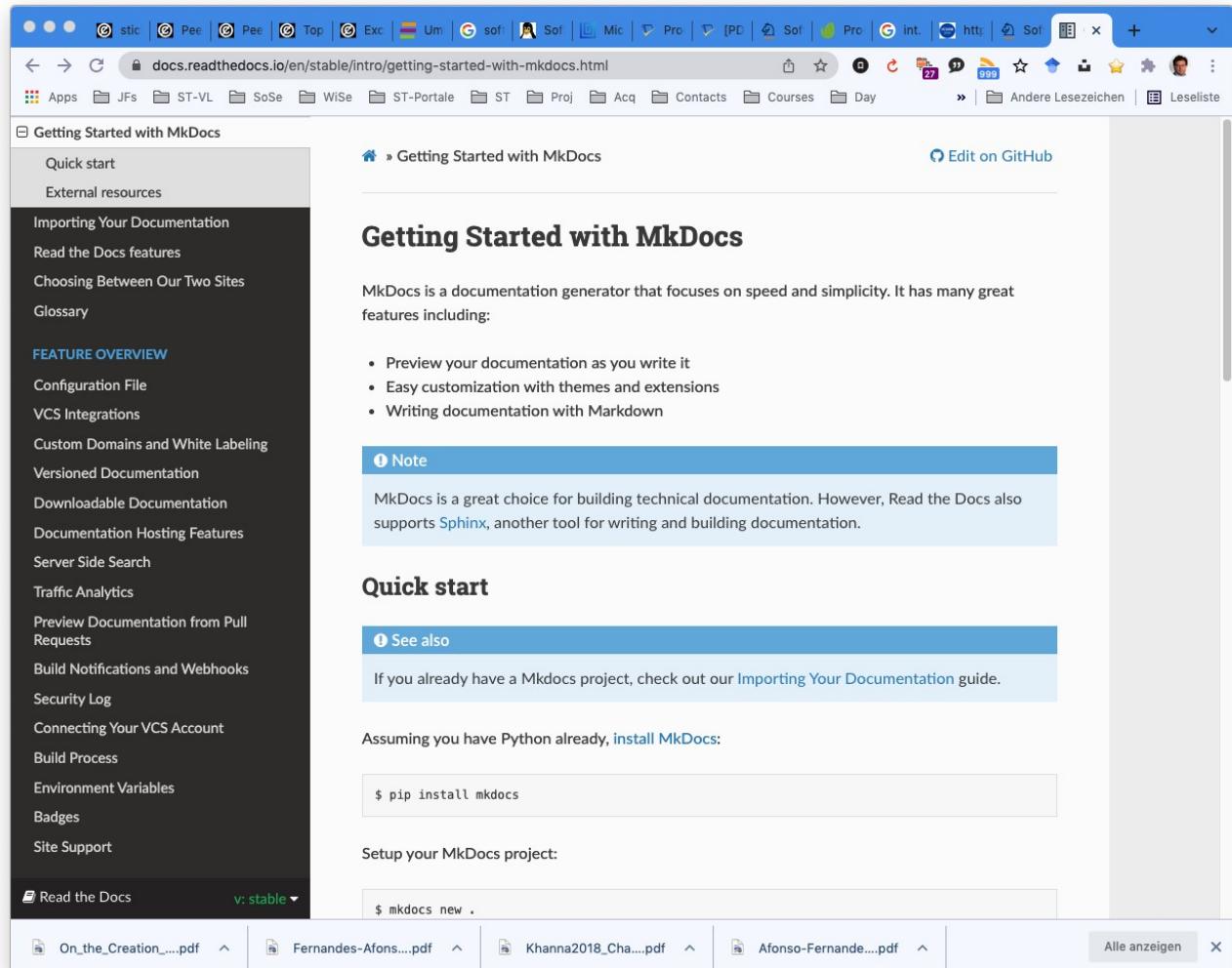
- ua | 2021-01-15 | getting more text into docu.
- ua | 2020-12-28| getting more experience with sphinx.
- ua | 2020-12-30| copying API documentation here.

Usage

- Installation
 - Docker
- Getting started
- Examples
- Meta-models
- Stages
- Getting started



- ▶ Markdown files to HTML files
- ▶ several output formats



Alot – a Mail User Agent Documented on readthedocs

- ▶ <https://alot.readthedocs.io/>

The screenshot shows a web browser window with multiple tabs open at the top, including 'hyb', 'mer', 'Me...', 'Qu...', 'mu...', 'Ho...', 'Git...', 'Alo...', and others. The main content area displays the Alot documentation page on readthedocs.io. The page has a dark blue header with the 'alot' logo and 'latest' tag. A search bar is present. On the left, a sidebar lists navigation links: 1. Installation, 2. Usage, 3. Configuration, 4. API and Development, and 5. Frequently Asked Questions. Below these are links for 'Manpage' and a Tidelift badge. The Tidelift badge features a dark background with orange and white text, stating 'TIDELIFT' and '59% of maintainers have quit or considered quitting'. A 'READ THE REPORT' button is also visible. The main content area starts with the heading 'Alot' and a paragraph describing it as a terminal-based mail user agent for the notmuch mail system. It highlights its modular and command prompt driven interface. To the right of the main content, there is a sidebar with a list of topics under each main category: 1. Installation (1.1. Manual installation, 1.2. Generating the Docs), 2. Usage (2.1. Command-Line Invocation, 2.2. UNIX Signals, 2.3. First Steps in the UI, 2.4. Commands, 2.5. Cryptography), 3. Configuration (3.1. Configuration Options). At the bottom of the sidebar, there are 'Read the Docs' and 'v: latest' buttons.

NaturalDocs Generic API Documentation Generator

33

Model-Driven Software Development in Technical Spaces (MOST)

- ▶ Similar to JavaDoc, but more than 20 languages
- ▶ own keywords can be defined
- ▶ Example gitlab project from which API documentation for GrGen can be generated
 - https://git-st.inf.tu-dresden.de/adaptive_petrinets/reconfnet/-/tree/master/doc

The screenshot shows a GitLab interface for a project named 'reconfnet' in the 'Adaptive Petrinets' namespace. The repository is at the 'master' branch. A commit by Uwe Aßmann is visible, dated 9 months ago, with the commit ID 06295069. Below the commit, a table lists files and their details:

| Name | Last commit | Last update |
|-----------------------|--|---------------|
| .. | | |
| api | new readme in doc/api. 6 Modules are documented now. | 11 months ago |
| Comments.txt | naturaldocs: new comment type Question, Algorithm. | 11 months ago |
| Languages.txt | The language of GRSHELL was added in Languages.txt, to incl... | 11 months ago |
| Makefile | naturaldocs: new comment type Question, Algorithm. | 11 months ago |
| Project.txt | documented petri2dynamic nets | 10 months ago |
| Readme.md | new readme in doc/api. 6 Modules are documented now. | 11 months ago |
| protocol-reconfnet.md | more files from codiMD | 9 months ago |

Below the table, a file named 'Readme.md' is shown with the title 'Generation of API documentation for Reconfnet'. It contains instructions for generating API documentation:

- Install naturaldocs (e.g., with brew).
- make sure that in your file system the projects **reconfnet** and its documentation project **reconfnet.pages.st.tu-dresden.de** are located as sister directories.
 - Naturaldocs needs to find the **relative paths** it is looking for.
 - At the moment naturaldocs is located in the **reconfnet/doc** directory; which might change to **reconfnet.pages.st.tu-dresden.de**, but this works only if relative paths are consistent.
- Then run:

```
cd doc
naturaldocs .
open api/index.html
```

At the bottom, a note states: "make sure that the subdirectory **api** is not deleted, naturaldocs relies on it".

Example NaturalDocs API generated for GrGen

- ▶ GrGen.net is a generator for graph rewrite specifications (see Part IV)
- ▶ There is no specific API doc generator for GrGen, but NaturalDocs can be tailored to it

The screenshot shows a web browser window displaying a NaturalDocs-generated API documentation page. The title bar of the browser reads "petrinets.pages.st.inf.tu-dresden.de/adaptive-petrinets/apiindex.html#File4:ContextDepen...". The main content area is titled "Reconfignet Compiler" and "for Dynamic Adaptive Petrinets, Version 0.6.3". On the left, there is a sidebar with tabs for "Files" and "Classes". Under "Files", the "context" file is selected, showing its contents and metadata. The "Properties" section indicates it was authored by CM and commented by UA on 2020-12-29. The "GraphTypes" section lists "Weak Inclusion", "Strong Inclusion", "requirement", and "exclusion" with their respective symbols. The right side of the page contains detailed descriptions for each of these four types, including examples and activation/deactivation triggers.

context/ContextDependencyModel.gm

This defines the types for contexts in Dynamic Adaptive Petrinets.

PROPERTIES

Author: CM. Commented by UA (2020-12-29)

We need two approaches, vertical and horizontal extension/reconfiguration.

GRAPHTYPES

Weak Inclusion

empty triangle on activation/deactivation trigger this with target act(source) -> act(target) deact(source) -> deact(target) act(target) -> deact(target) ->

Strong Inclusion

when target gets deactivated source also empty triangle act(source) -> act(target) deact(source) -> deact(target) act(target) -> deact(target) -> deact(source) -> deact(target)

requirement

can only be activated when target is already empty triangle act(source) -> only if already: act(target) deact(source) -> act(target) -> deact(target) -> deact(source)

exclusion

empty boxes both can not be active at same time act(source) -> deact(target) act(target) -> deact(source)

arrow

The End

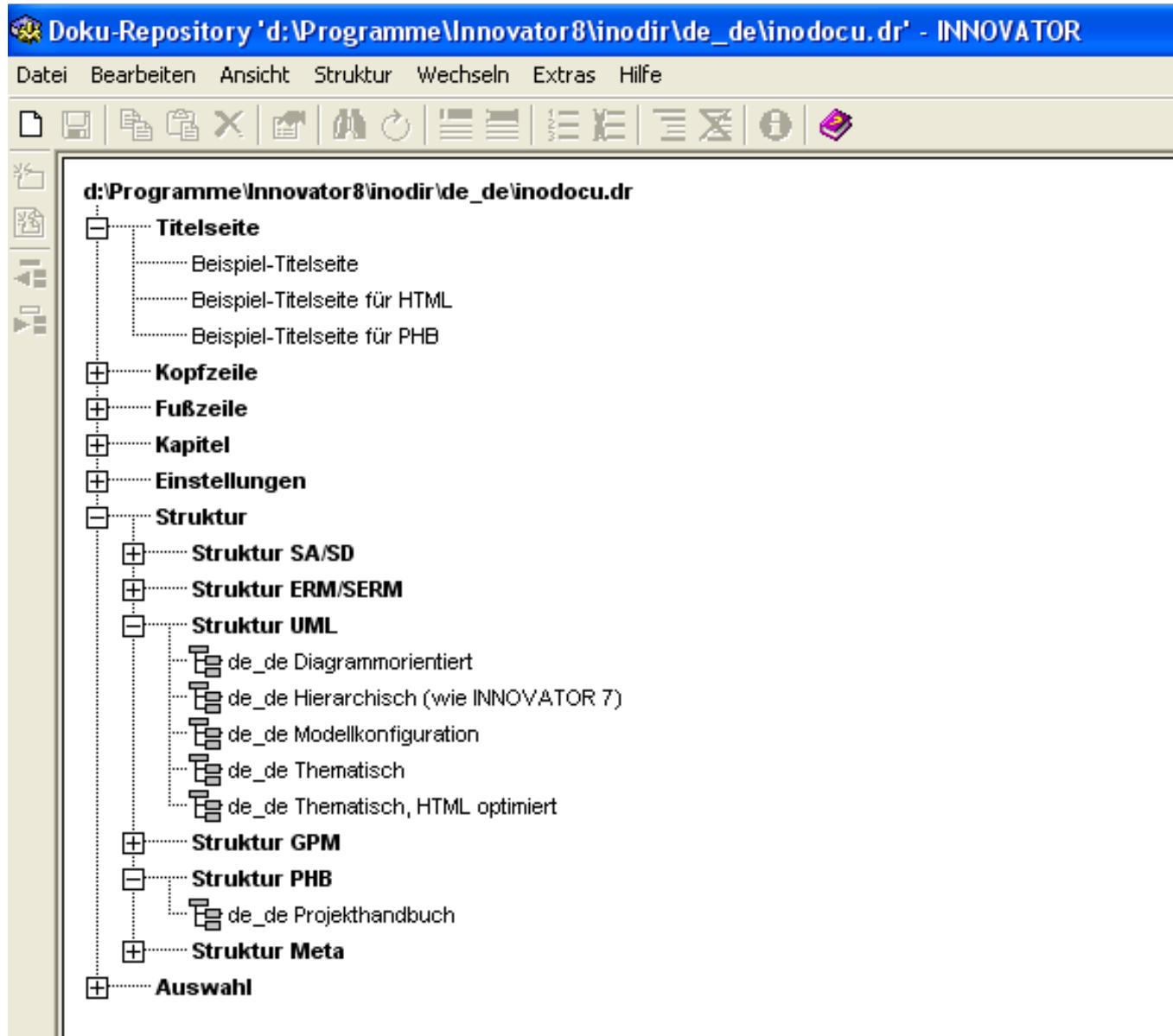
- ▶ Why is generation of documentation similar to code generation?
- ▶ Explain why a higher-order RAG is useful for documentation generation
- ▶ Which role does a pattern-matching language such as Xcerpt play in documentation generation?
- ▶ Why is the generation of documentation part of a macromodel?
- ▶ Why is a documentation a *derived model*?
- ▶ What happens if text from the API documentation flows back into the code as comments?

A.1 Other Template Expanders for Documentation Generation

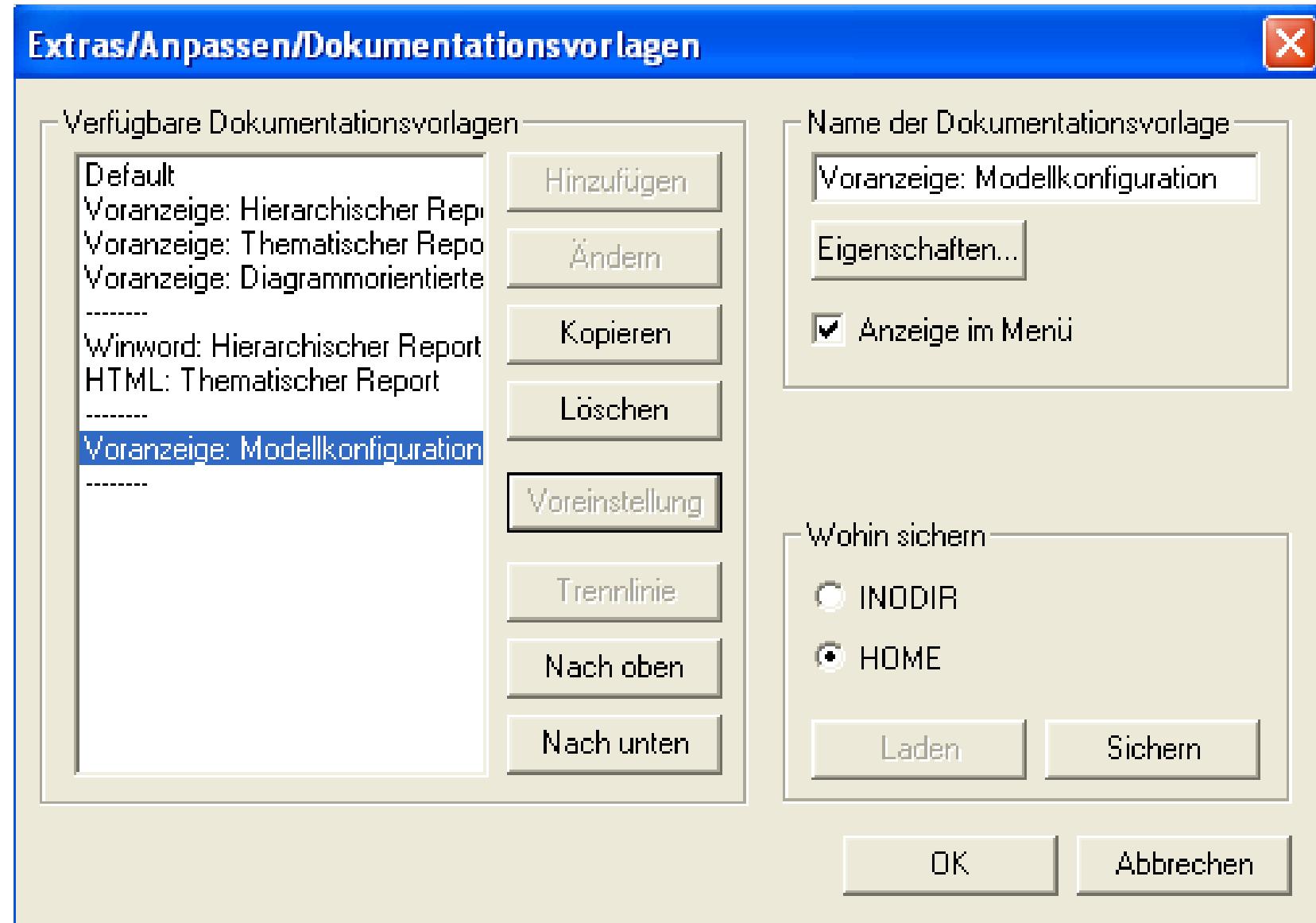
Documentation Tools of MID Innovator

- ▶ Innovator provides documentation templates, into which diagrams, models, code can be embedded
- ▶ Several formats:
 - pdf
 - Word
 - ASCII
 - XML

Ex.: Innovator Documentation Template (Dokumentationsvorlage)



Ex.: Innovator Documentation Template (Dokumentationsvorlage): Adaptation



Innovator - Generated Example Word Document

41 Model-Driven Software Development in Technical Spaces (MOST)

The screenshot shows a Microsoft Word document window titled "Voranzeige c:\temp\lidl21912". The menu bar includes "Datei", "Wechseln", "Optionen", and "Hilfe". The title bar has standard window controls. The main content area displays a table of contents (Inhaltsverzeichnis) and a search results window.

Inhaltsverzeichnis

| | |
|---|---|
| 1. externes Kapitel | 1 |
| 1.1. Unterkapitel1 | 1 |
| 1.2. Unterkapitel2 | 1 |
| 1.3. Unterkapitel3 | 1 |
| 2. Doku | 2 |
| 2.1. systemModel | 3 |
| 2.1.1. use case system | 3 |
| 2.1.1.1. Anwendungsfalldiagramm UseCaseDiagram | 3 |
| 2.1.1.2. Knetdiagramm Create Defaults for Use Cases | 4 |
| 3. Index | 5 |

A mouse cursor is hovering over the link "2.1.1.1. Anwendungsfalldiagramm UseCaseDiagram" in the table of contents. A black diagonal line is drawn from this link to a second Microsoft Word window titled "Voranzeige c:\temp\lidl21914". This second window displays the content of the selected table of contents item:

2.1. systemModel

2.1.1. use case system

2.1.1.1. Anwendungsfalldiagramm Us...

Table of Contents

Integration of a Use Case Diagram (section 2.1.1.1.)

Index is generated

