25. 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 17-0.6, 24.11.17

1) Tasks
2) Template-Driven Documentation Tools
3) Elucidative Modeling and Documentation Tools

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Literature

- 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 fur 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/pub/andreas-bartho/ba/922/8a4?trk=pub-pbmap

© Prof. U. Aßmann

25.1 Tasks of Documentation Tools

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

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Basics of Software Documentation

- Documentation is a means of **communication**
  - 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
  - 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

Quelle: [24 S. 241 ff.]

© Prof. U. Aßmann

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
- **Project documentation**
  - Developer documentation
  - Project documentation (project plan, requriements specification, status reports, after study)
- **Quality documentation**
  - Test-, review, audit documentation
- **Process documentation**
  - Standards, processes

Quelle: [24 S. 245 ff.]

© Prof. U. Aßmann

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 app of a RAG**
- **Generation** of derived documents from code and models
  - Generation of Word, 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 (hedge-principle)
- **Parameterization** with layouts
  - via css-style sheets

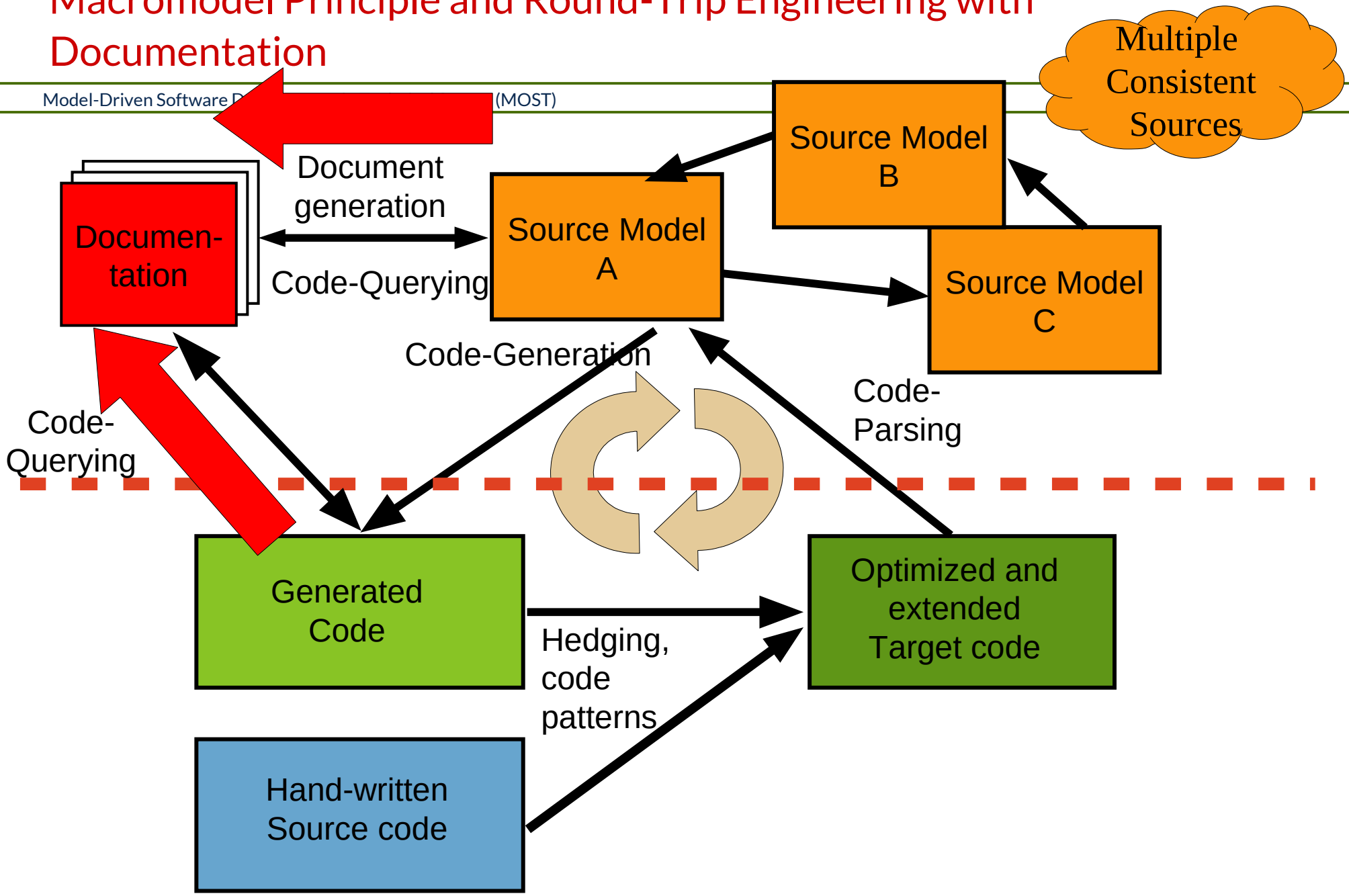
© Prof. U. Aßmann

25.2 Generative, Template-Driven Documentation Tools

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

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Macromodel Principle and Round-Trip Engineering with Documentation



© Prof. U. Aßmann

Documentation Tool JavaDoc

- 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; Wikipedia: http://de.wikipedia.org/wiki/JavaDoc]

© Prof. U. Aßmann

JavaDoc is a Typical HRAG Application

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

```
// scematic, 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 + st2.htmldoc;
  }
  Attributions of Method(name,comment) {
     this.htmldoc := „<h1>“+name+“</h2>“+comment.htmldoc;
  }
  Attributions of Comment(text) {
     this.htmldoc := text;
  }
}
```

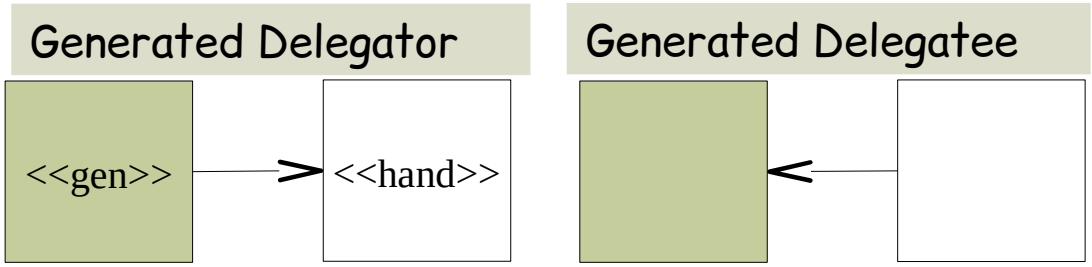
© Prof. U. Aßmann

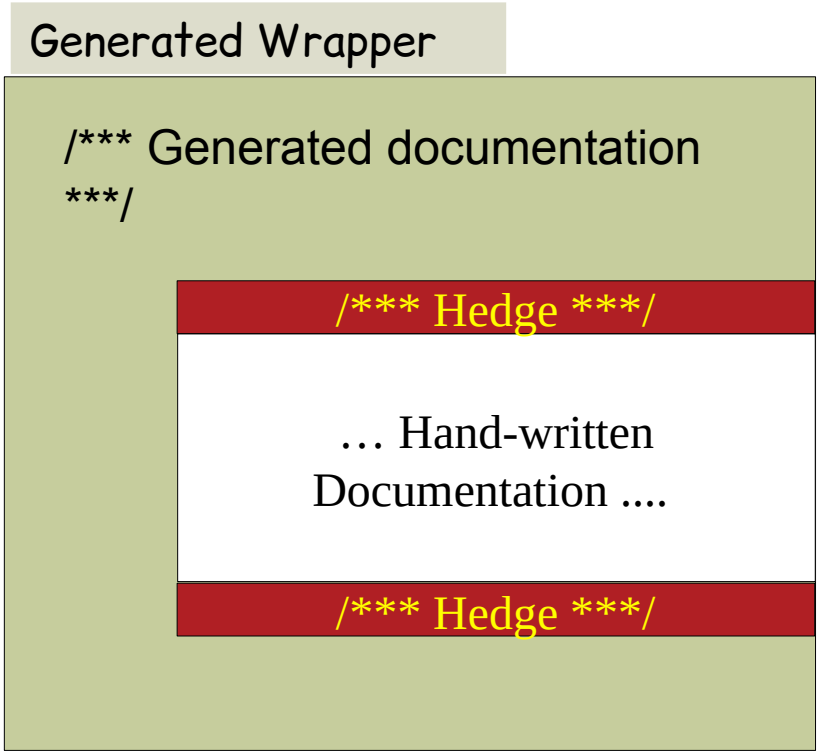
Composition of Separated Hand-Written and Generated Documentation Snippets

- **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)**





© Prof. U. Aßmann

Document Generator *SELECT*

- Zusammenarbeit mit MS Word-Templates
  - Hardcopy von Diagrammen mit OLE in ein Word-Dokument
  - Der SELECT Document Generator kann ein Dokumentationsgerüst in MS Word generieren
- Inhalt der generierten Dokumentation:
  - Diagramme des SELECT-Modells, genaue Objektbeschreibungen, leere Abschnitte für Analysen, Berichte oder Zusammenfassungen
  - Auswahl aus vorgefertigten Dokumentvorlagen
    - Auswahl der einzufügenden Abschnitte aus der Vorlage
  - Nachträgliches Einfügen und Entfernen von Abschnitten möglich
  - Aktualisierung des Inhalts durch selektive Synchronisierung mit dem SELECT-Modell
- Das Layout lässt sich anpassen durch Bearbeiten der Dokumentvorlagen oder des generierten Dokumentes
- Navigation in den generierten Abschnitten der Dokumentation direkt aus dem Document Generator heraus

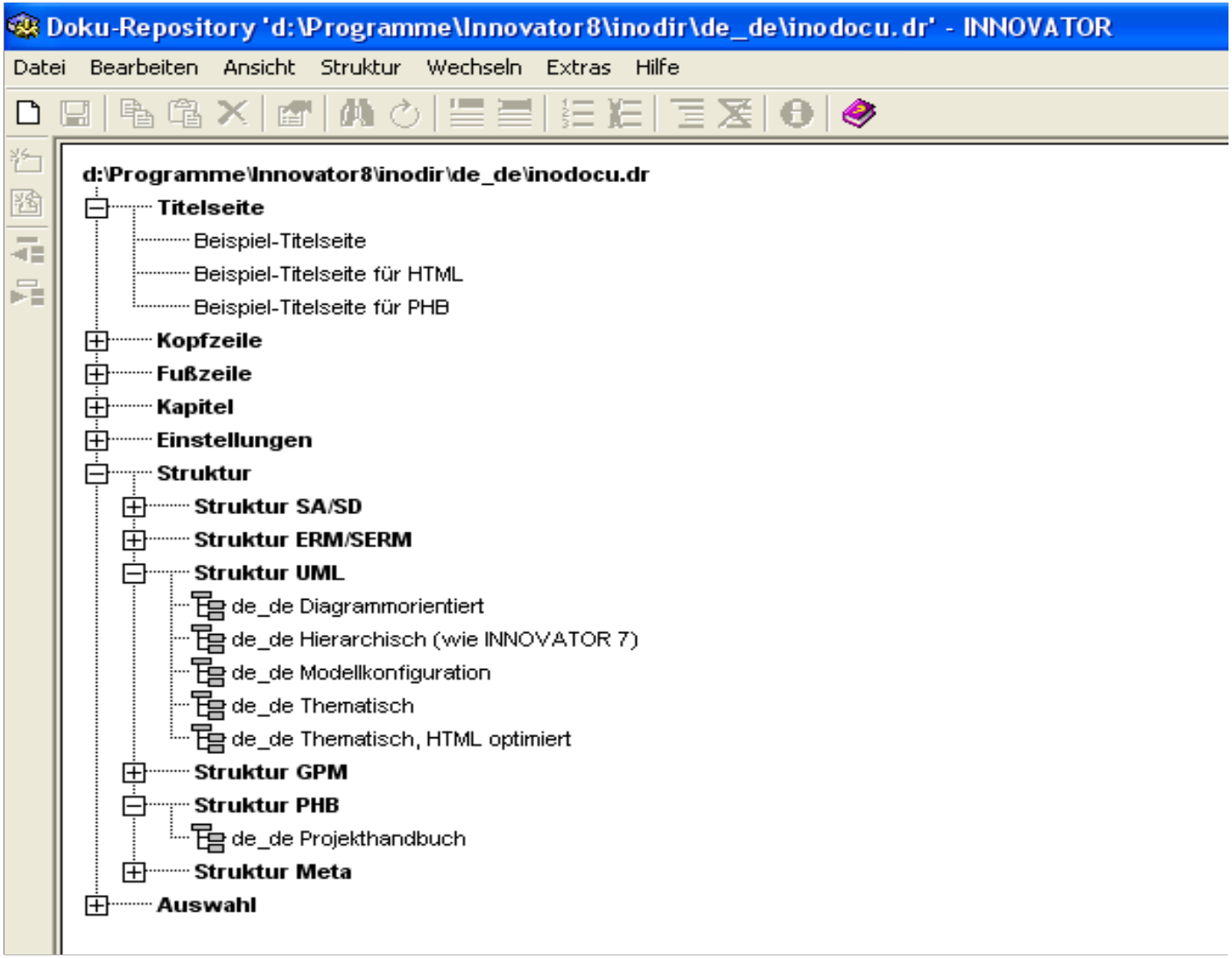
© Prof. U. Aßmann

Documentation Tools of MID Innovator

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

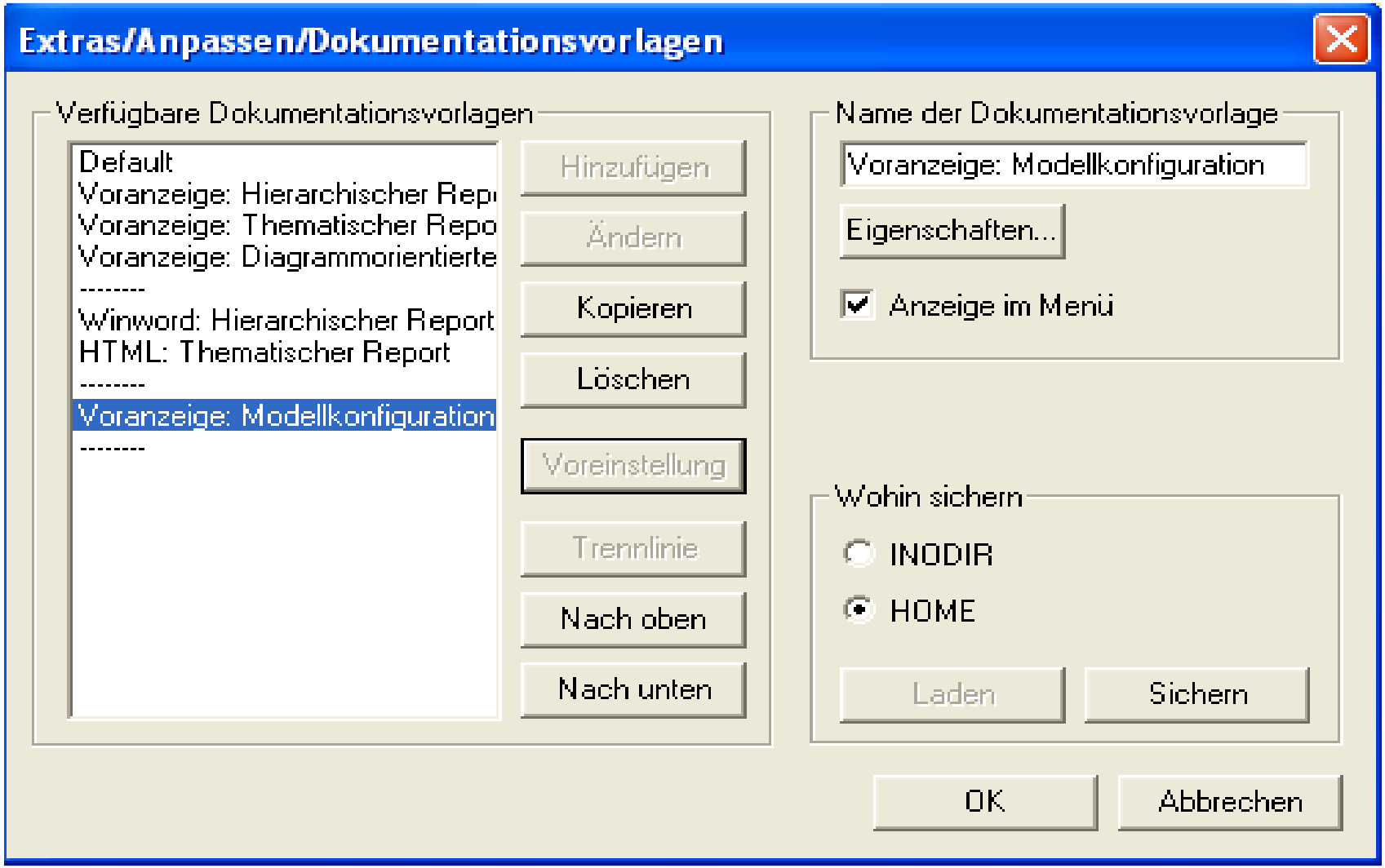
© Prof. U. Aßmann

Ex.: Innovator Documentation Template (Dokumentationsvorlage)



© Prof. U. Aßmann

Ex.: Innovator Documentation Template (Dokumentationsvorlage): Adaptation



© Prof. U. Aßmann

Innovator - Generated Example Word Document

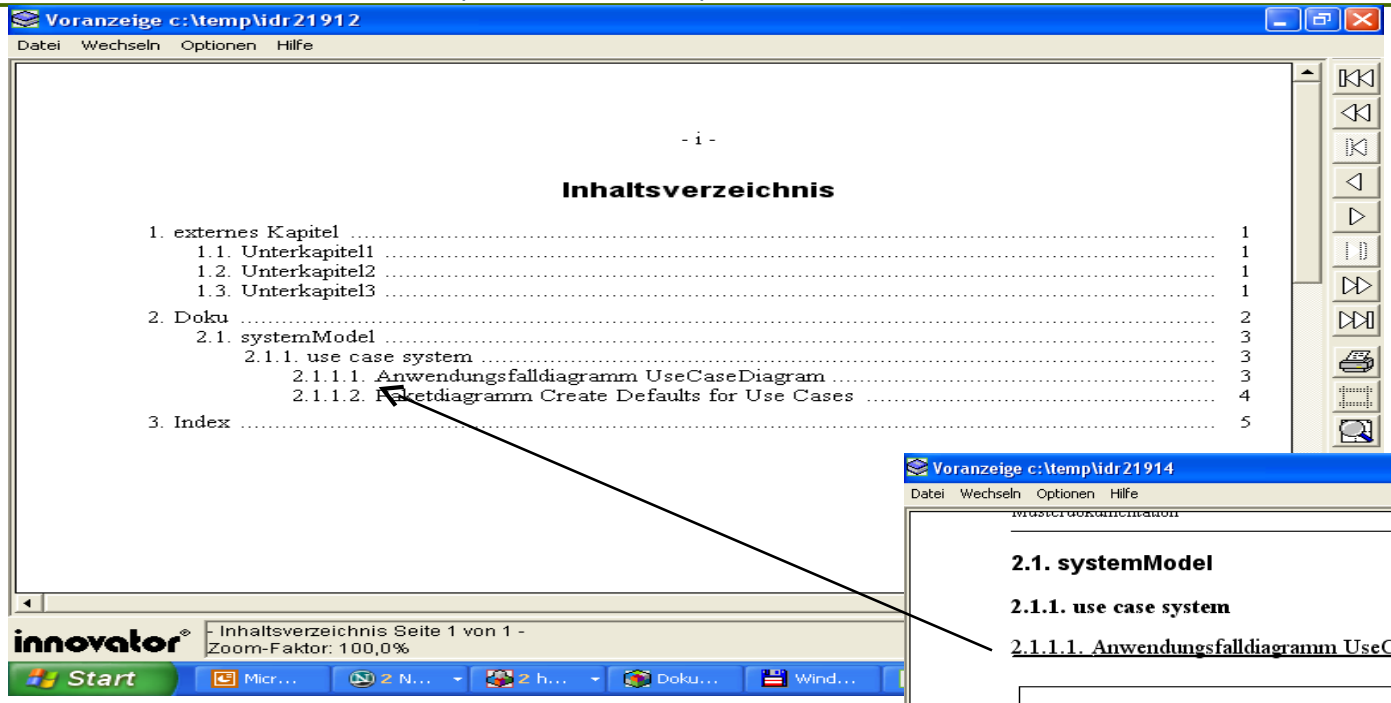
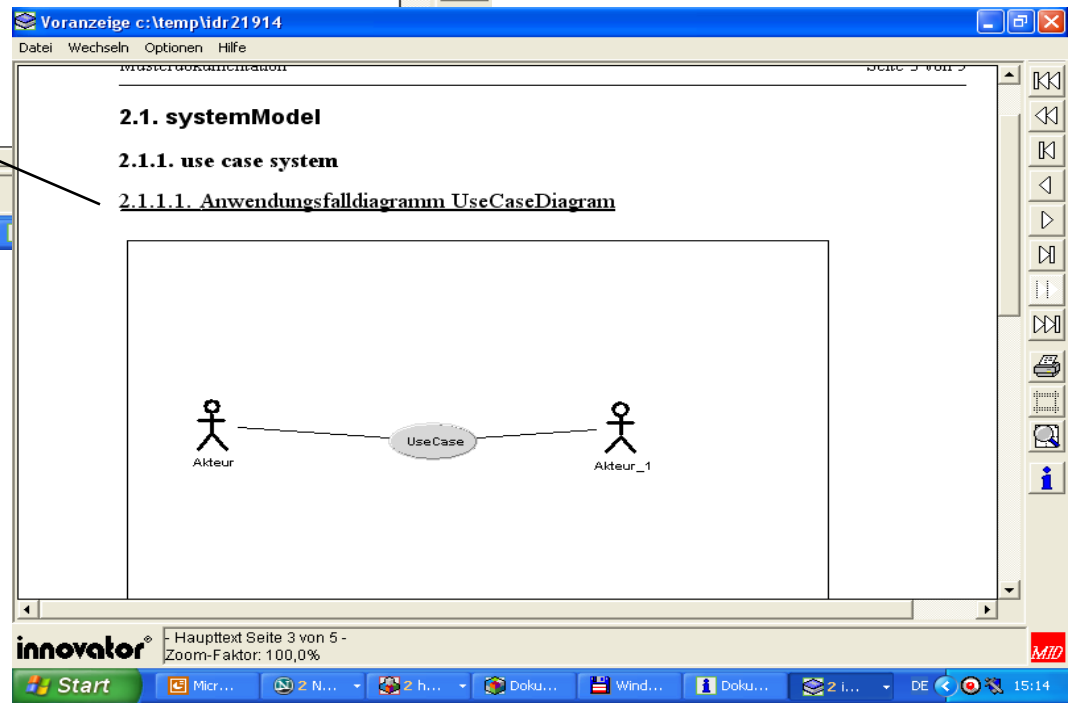


Table of Contents



Integration of a Use Case Diagram (section 2.1.1.1.)

Index is generated

© Prof. U. Aßmann

25.3 Elucidative Documentation Tools

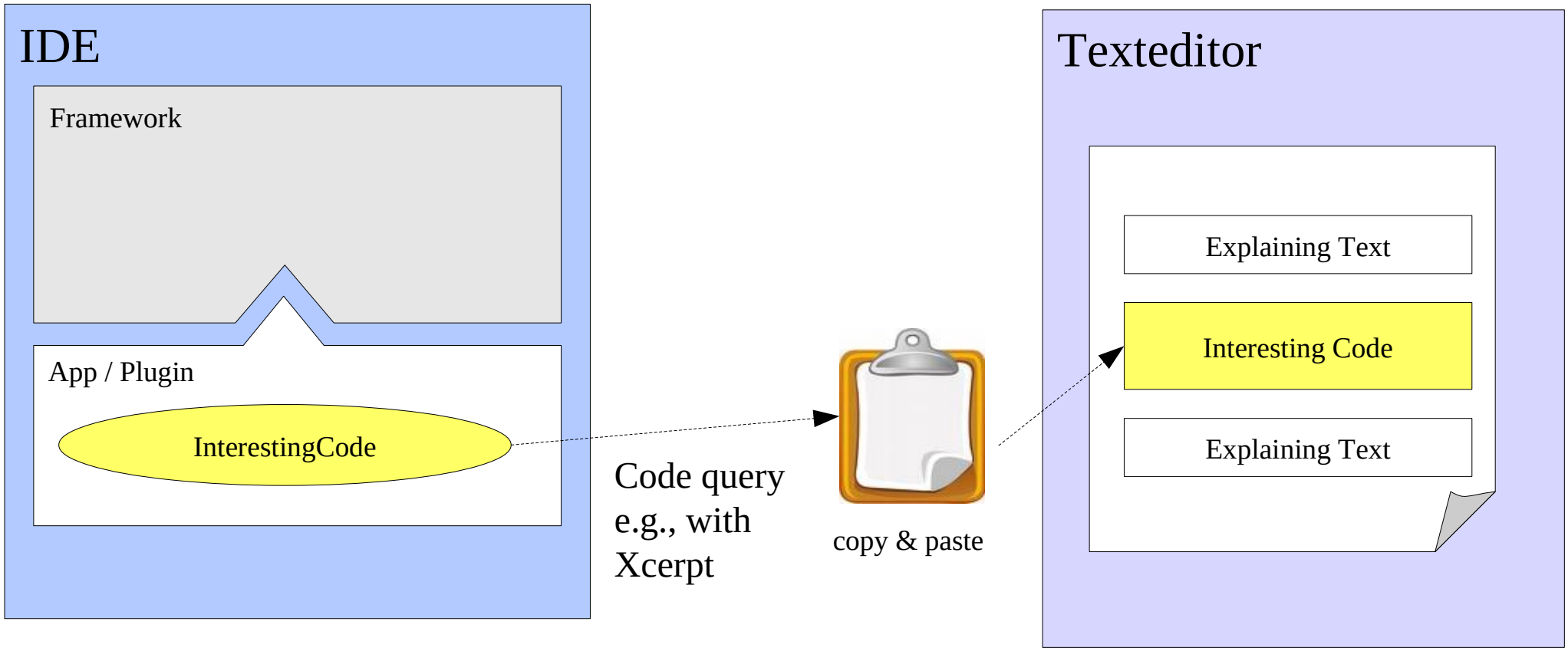
- They link code, models and documentation by **model mapping**

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Tutorial: Example

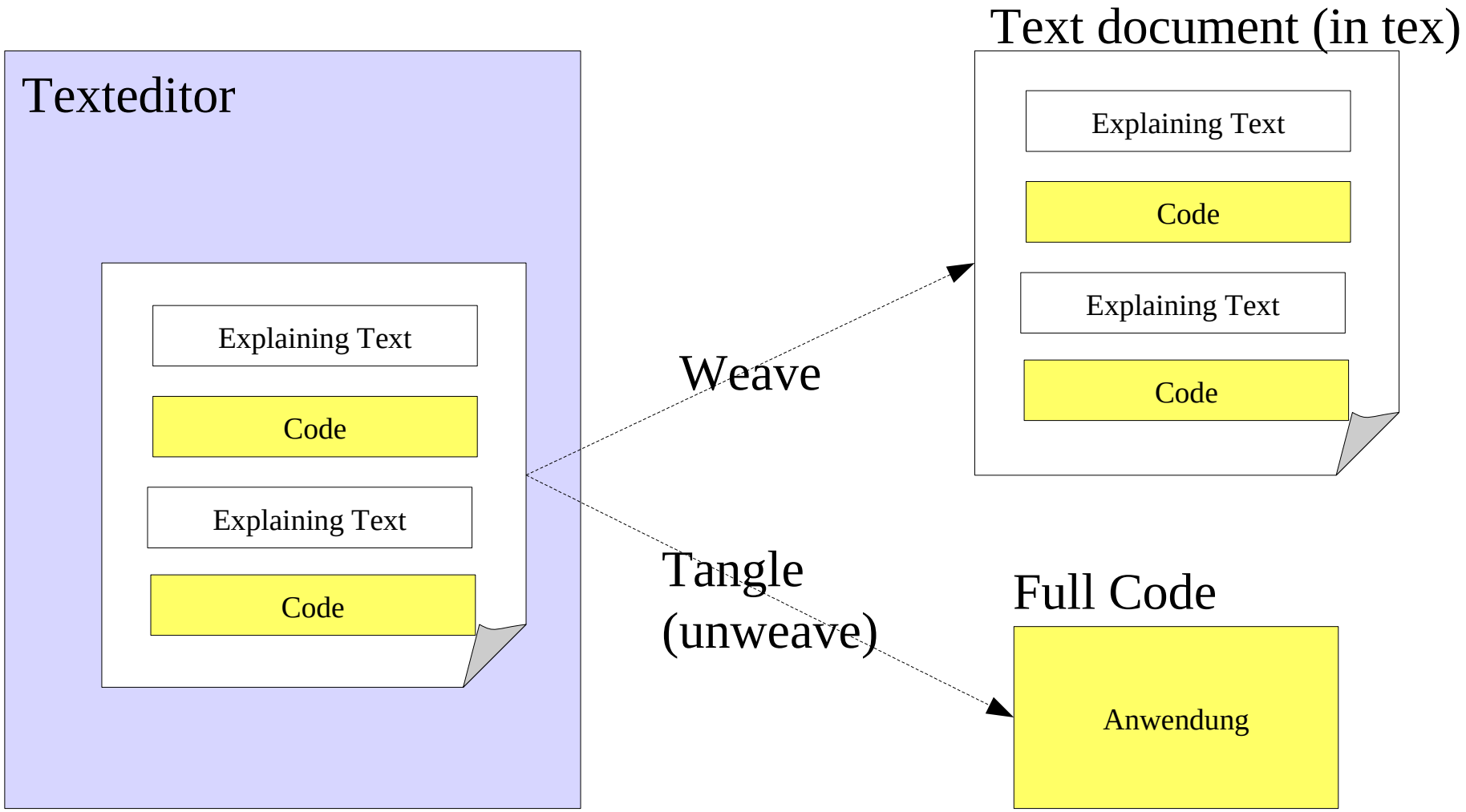
© Prof. U. Aßmann

Manual Writing of Tutorials



© Prof. U. Aßmann

Literate Programming by Code Unweaving



© Prof. U. Aßmann

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.⟧

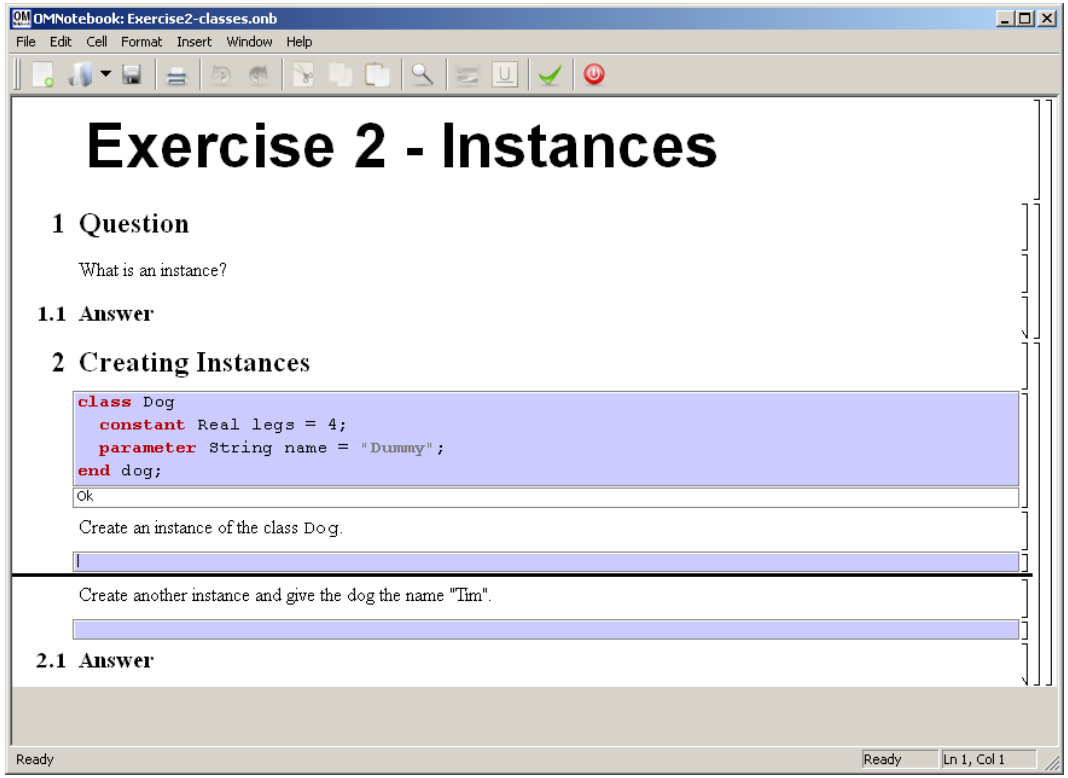
```
⟨ Program to print the first thousand prime
      numbers 2 ⟩ ≡
program print_primes(output);
  const m = 1000;
    ⟨ Other constants of the program 5 ⟩
  var ⟨ Variables of the program 4 ⟩
    begin ⟨ Print the first m prime numbers 3 ⟩;
    end.
```

aus Literate Programming
von Donald E. Knuth

- Overview: http://www.literateprogramming.com/
- OMNotebook/DrModelica: http://www.modelica.org/tools

© Prof. U. Aßmann

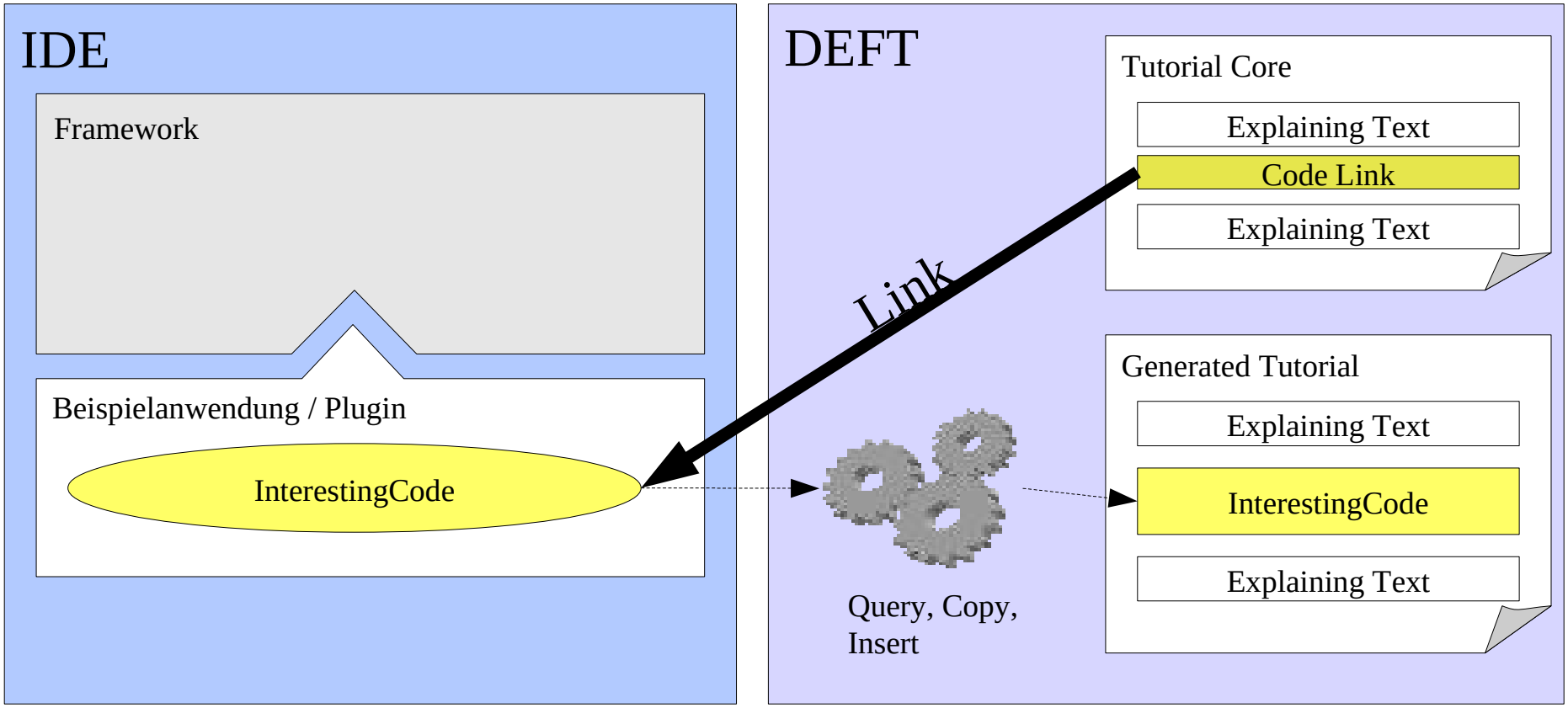
OMNotebook - DrModelica



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

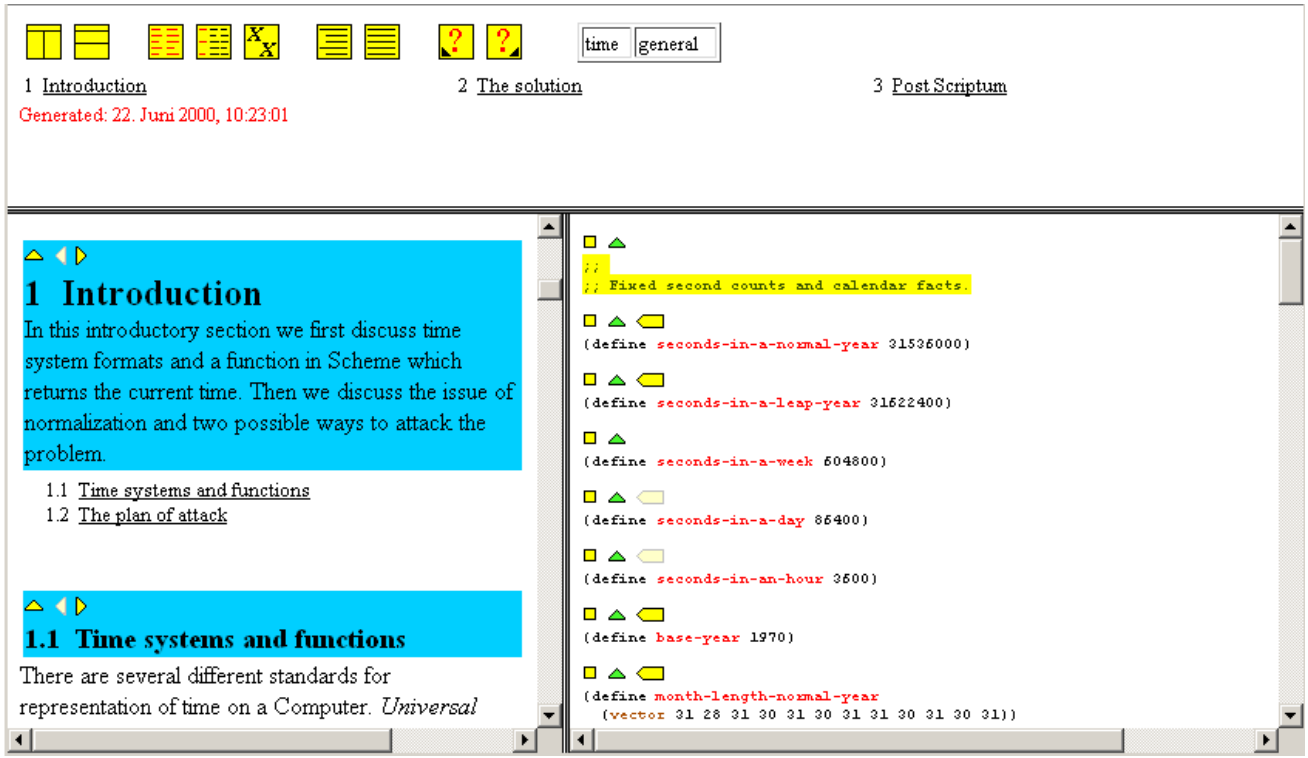
© Prof. U. Aßmann

Elucidative Programming Links Documentation with Queries to Code



© Prof. U. Aßmann

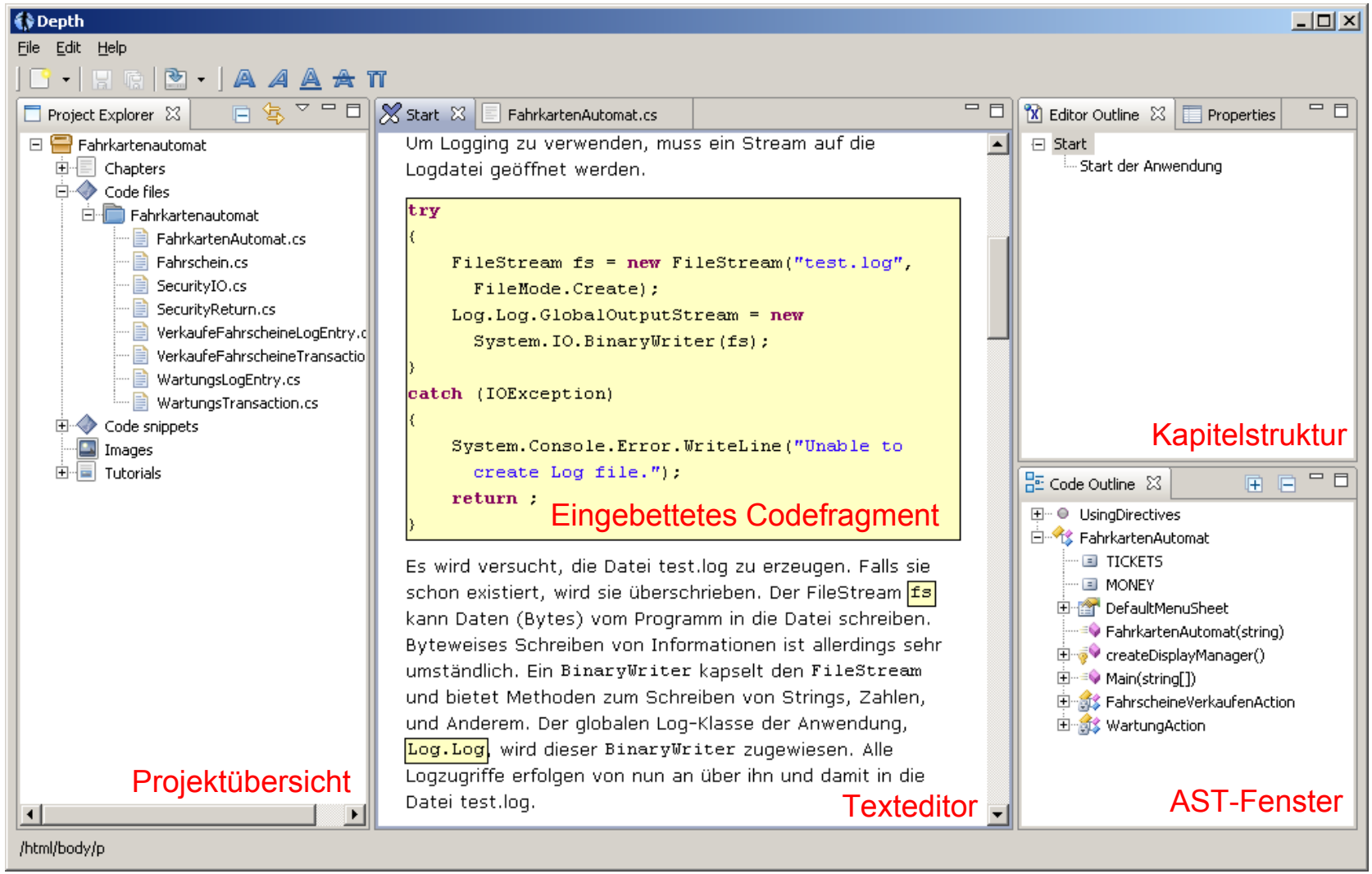
Elucidative Programming



- http://www.cs.aau.dk/~normark/elucidative-programming/
- http://deftproject.org

© Prof. U. Aßmann

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



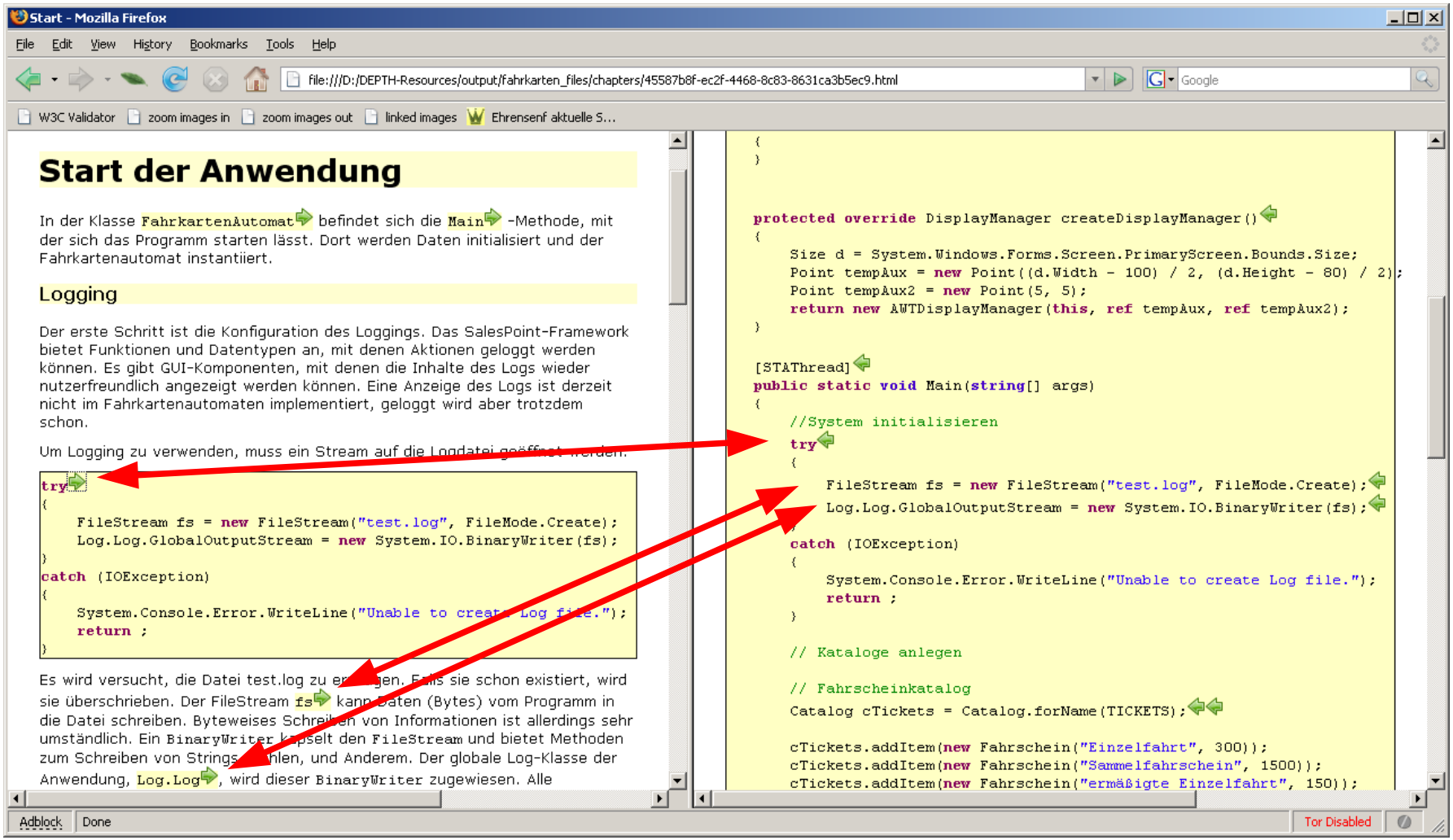
© Prof. U. Aßmann

Development Environment For Tutorials

- 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
  - Notiviation if code fragments have changed

© Prof. U. Aßmann

Generated HTML Tutorial



© Prof. U. Aßmann

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?

© Prof. U. Aßmann