

Multi-Media Instruction in Safe and Secure Systems

Bernd Krieg-Brückner, Dieter Hutter, Arne Lindow,
Christoph Lüth, Achim Mahnke, Erica Melis,
Arnd Poetzsch-Heffter, Markus Roggenbach, George Russell,
Jan-Georg Smaus, Martin Wirsing et al.

18 July 2003

MMiSS

**Multi-Media Instruction in
Safe and Secure Systems**

MMiSS Project Members

- **Universität Bremen**
 - Bernd Krieg-Brückner, Arne Lindow, Christoph Lüth, Achim Mahnke, George Russell, Hui Shi, et al.
- **FernUniversität Hagen, Universität Kaiserslautern**
 - Arnd Poetzsch-Heffter, et al.
- **Universität Freiburg**
 - David Basin, Jan-Georg Smaus, et al.
- **Universität des Saarlandes**
 - Serge Autexier, Dieter Hutter, Carsten Ullrich, Erica Melis, et al.
- **Ludwig-Maximilians-Universität München**
 - Rolf Hennicker, Martin Wirsing et al.

Goals of the MMiSS Project

- develop **slides**, lecture notes, **courses**
- work on the **board**, with slides, interactively with **tools**
- embed mathematical formulae, **programs**, etc.
- manage **English** and **German** documents in parallel
- publish **complete** and **consistent** “packages”
- (partially) **re-use** the slides of a colleague
- be made aware of **revisions** by your colleague
- agree with your colleague on a **uniform terminology**

Document Structuring for Formal Methods

What Could Be Your Interest in This Talk?

- What have we learned from
 - Formal Methods, **Cas1**, **development** support systems

What Could Be Your Interest in This Talk?

- What have we learned from
 - Formal Methods, **Cas1**, **development** support systems
- Your future favorite way to
 - prepare, **reuse** and **manage** high-quality **L^AT_EX** slides
 - manage language / formalism **variants**
 - **re-use** and **adapt** course material from colleagues
 - **coordinate** slides with other documents
 - **integrate** Cas1 specifications with documentation
 - manage and **configure** results of research project

This Talk and Demo

- semantic interrelation via **ontology**
- **slides**, layout, animation
- **refinement** to/**abstraction** from article or book, **variants**
- “Literate Specification” with **Cas1**, CATS tools

This Talk and Demo

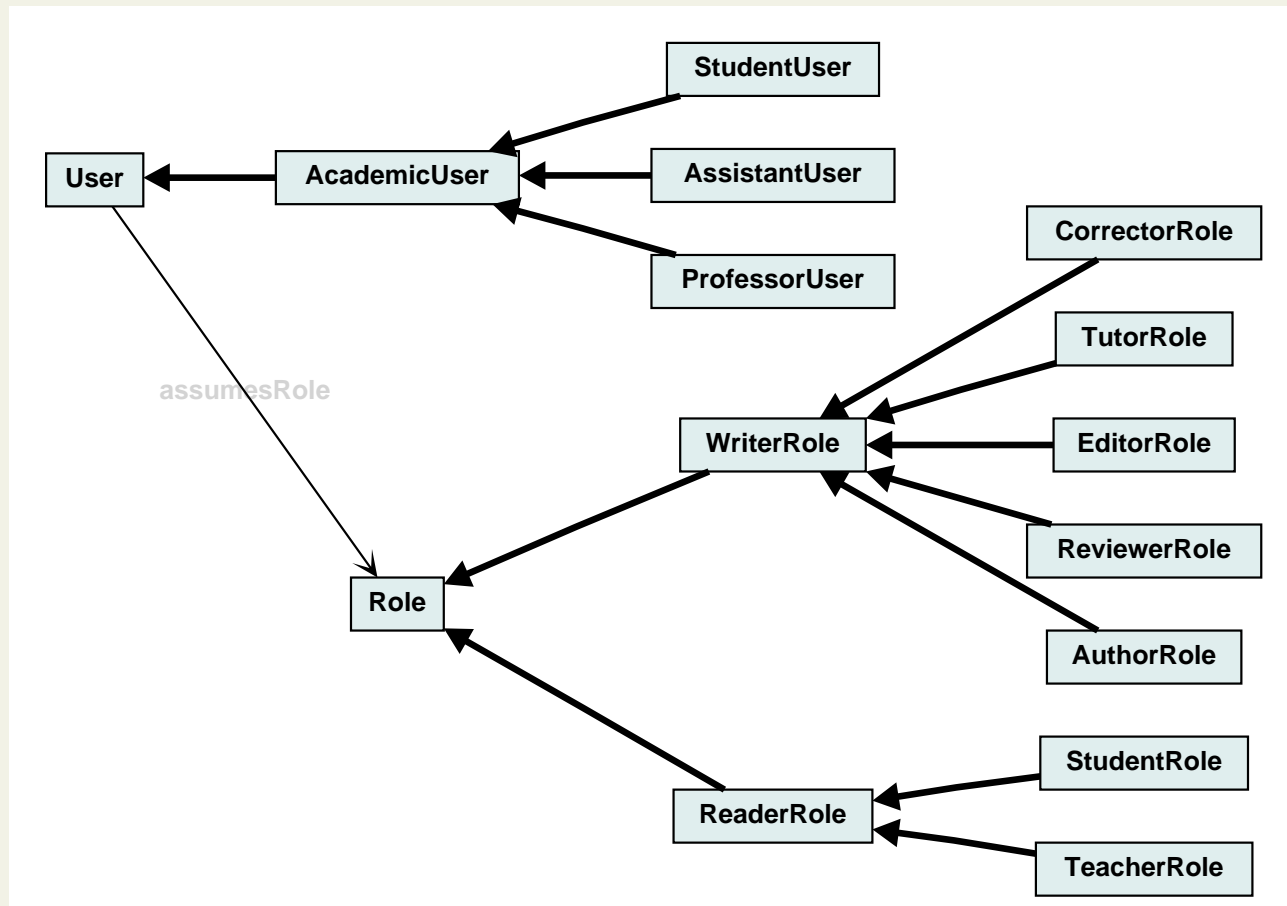
- semantic interrelation via **ontology**
- **slides**, layout, animation
- **refinement** to/**abstraction** from article or book, **variants**
- “Literate Specification” with **Cas1**, CATS tools
- **structuring “in-the-large”** via Packages – ontology
- **structure graph**, CATS development graph
- authoring tools, version control, sessions in the **Repository**
- **re-use**, configuration management, change management

Why an Ontology?

Ontology

- Classes
- Relations
 - “typed” by source and target Classes
- subClasses, subRelations
 - inheritance of formal properties
- Objects
 - of a Class, related by Relations
- Axioms
 - Description Logic: OWL
 - FOL: **Cas1**

Ontology of Users and Roles



Kinds of Users, Roles

- *Author* , Reviewer, Editor
 - Prepares initial Lecture: slides, board sketches, tool demonstrations
 - Reacts to student feedback
 - Extends to LectureNotes by annotations and explanatory comments
 - Expands to self-paced hypermedial Course
 - Extends the ontology, interrelates material in the Repository
 - Selects material from the Repository, adapts by revision and extension
- **Teacher**
 - Selects material from the Repository, teaches class

- *Student*
 - Attends class (!), uses material in Repository for review after class
 - Studies background material or self-paced Courses
 - Works on assignment Exercises, using tools interactively
 - Interacts with other Students
- Tutor, Corrector
 - Answers questions, compiles frequently asked questions and answers
 - Prepares and corrects Exercises, develops model solutions
- ToolDeveloper, SystemDeveloper
- Administrator
 - Manages user groups, moderates editions and distributes them

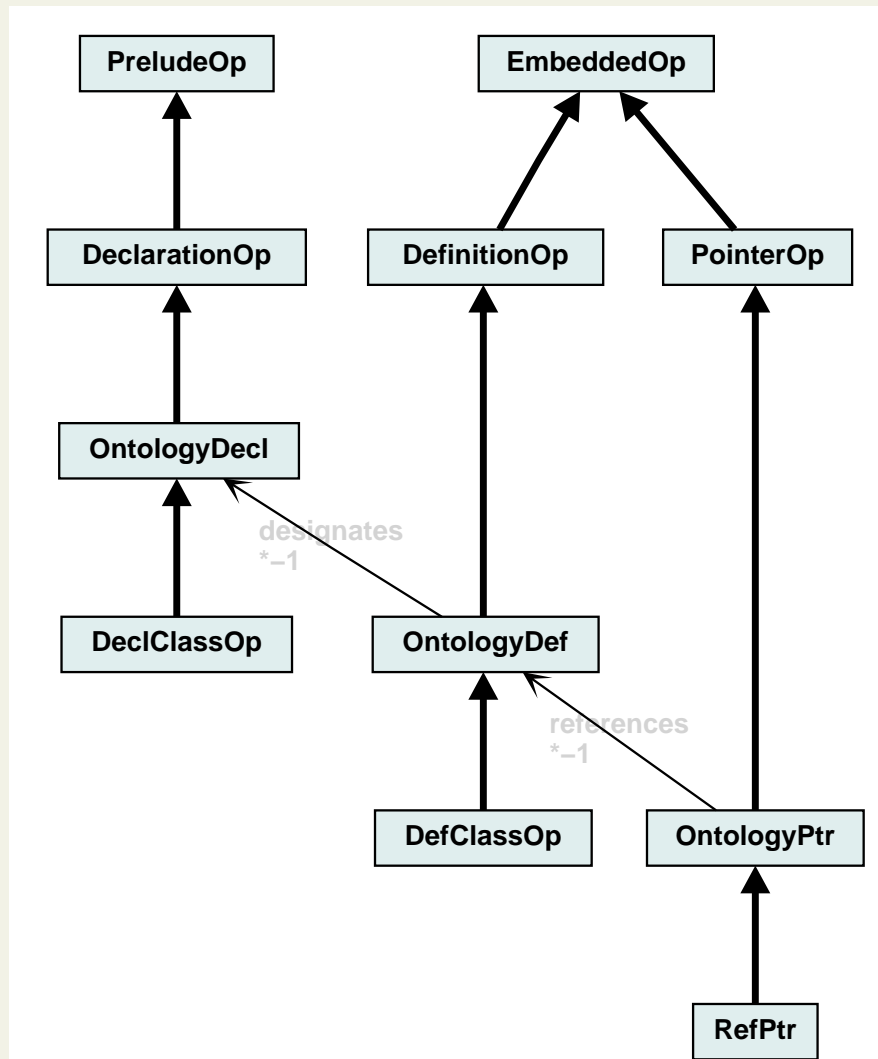
Ontology: Resolution of Ambiguity

a **Student** may assume the role of a **Student**

```
\DeclClass{User}{User}{}  
  \DeclClass{AcademicUser}{Academic User}{User}  
    \DeclClass{StudentUser}{Student}{AcademicUser}  
\DeclClass{Role}{Role}{}  
  \DeclClass{ReaderRole}{Reader}{Role}  
    \DeclClass{StudentRole}{Student}{ReaderRole}  
  
  \DeclRel{*-*}{assumesRole}{assumesRole}{}  
  \RelType{assumesRole}{User}{Role}
```

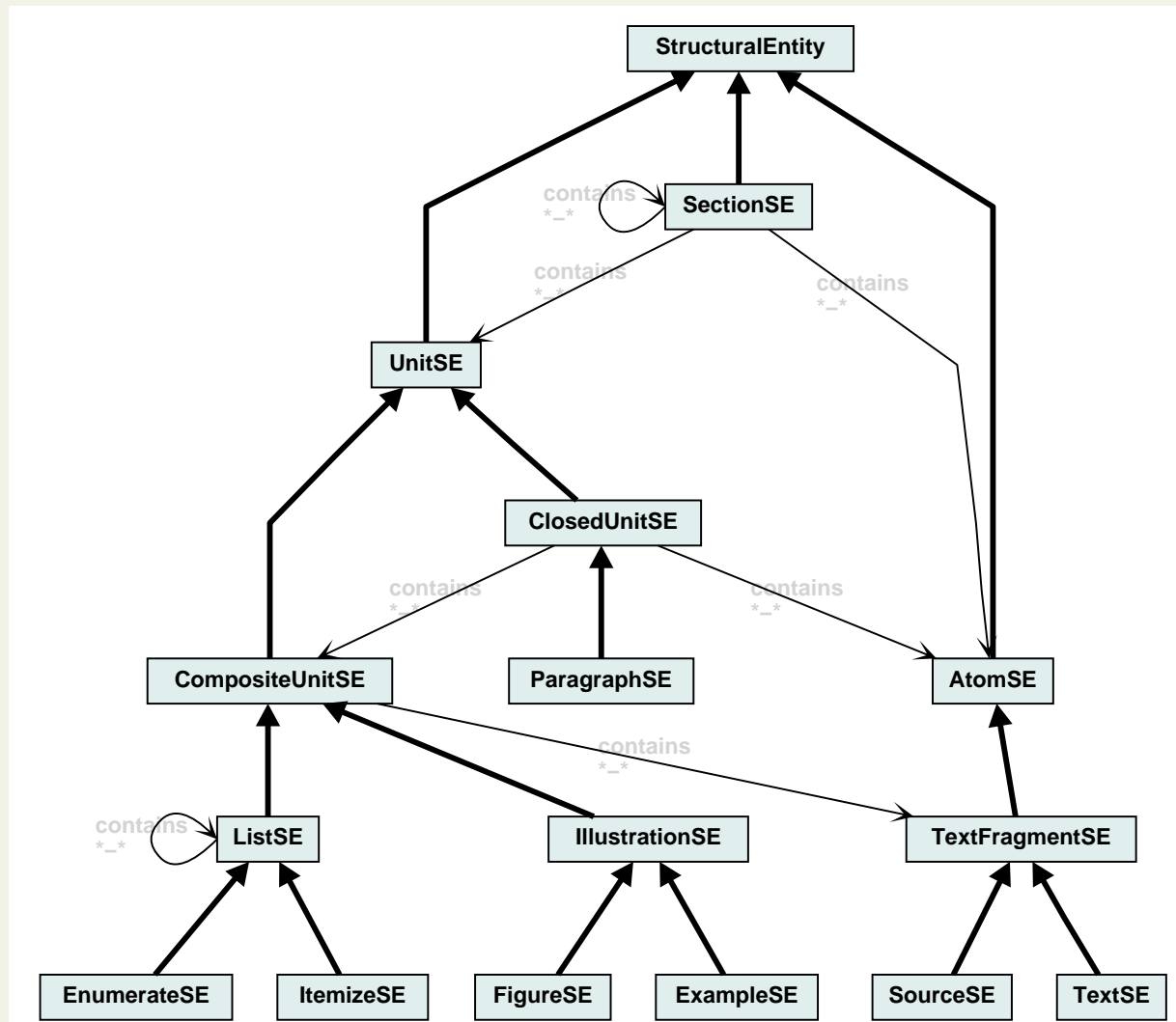
Ontology Declaration, Definition, Reference

- **Declare** in OntologyPrelude (“signature” of a Package)
`\DeclClass{StudentRole}{Student}{ReaderRole}`
- **Define** embedded in text at defining occurrence
`\DefClass{StudentRole} \rightsquigarrow Student`
- **Reference** embedded in text
`\StudentRole{} \rightsquigarrow Student,`
`\Ref{StudentRole} \rightsquigarrow Student`
`\Ref[student author]{AuthorRole}`
 `\rightsquigarrow student author`

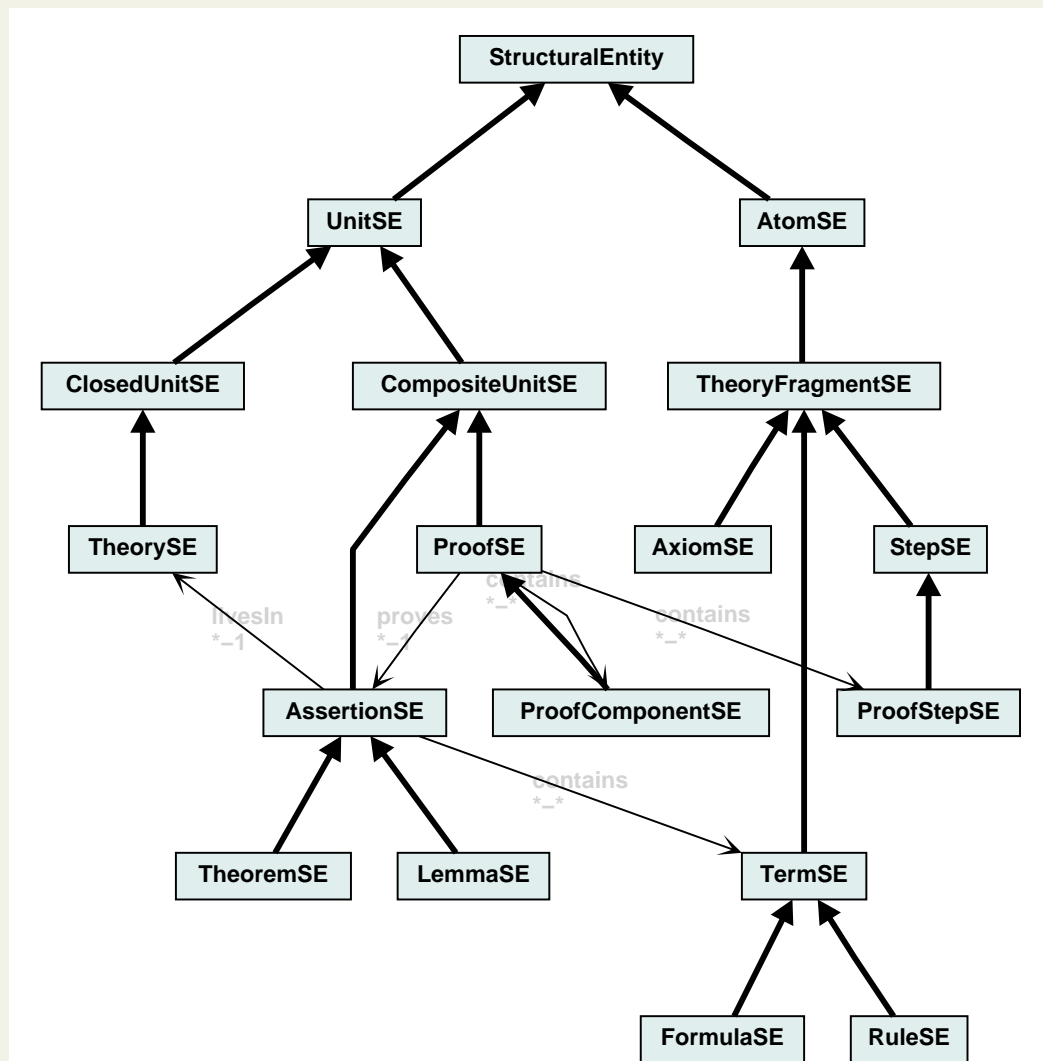


- **Declare**
in `OntologyPrelude`
 - structured via `Packages`,
language variants
- **Define**
at defining occurrence
- **Reference**
in text
- **pointsTo** relations
 - `Def` **designates** `Decl`
 - `Ref` **references** `Def`

Conceptual and Formal Structural Entities



- **Package** contains
 - Preludes
 - Sections
- **Sections**
 - nested
- **Unit**
 - smallest editing context
 - should fit on one page
- **Atom**
 - no structure
 - tools



- reliesOn relations
 - Package imports Package
 - Section after Section
 - Theorem livesIn Theory
 - Proof proves Theorem

Extensions

Macros

- parameterless macros may be used as abbreviations

```
\Macro{\Conc}{++}
```

```
\SourceText{x\Conc{ }y}  $\rightsquigarrow$  x++y
```

- parameterised macros **as in L^AT_EX**

```
\Macro{\SourceText}[1]{\Typewriter{#1}}
```

```
\Macro{\Concat}[2]{$#1\Join#2$}
```

```
\SourceText{\Concat{x}{y}}  $\rightsquigarrow$  x  $\bowtie$  y
```

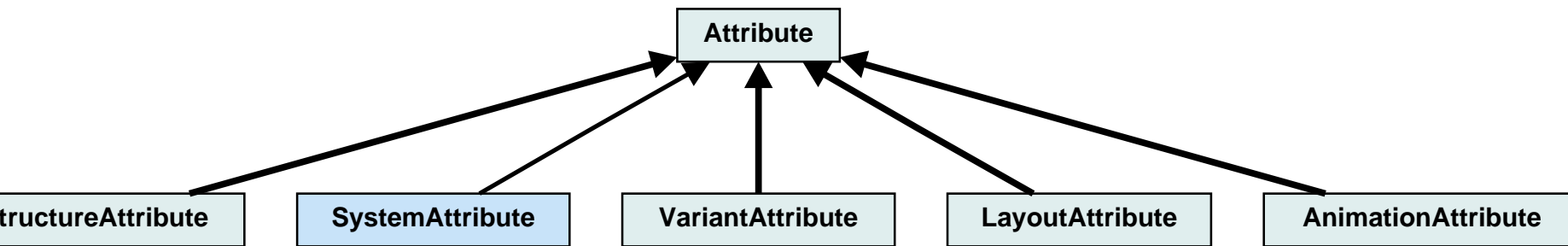
```
\Macro{\Future}[1]{\NoBold{#1}}
```

```
\Future{Future} Extensions  $\rightsquigarrow$  Future Extensions
```

Future Extensions

- **System Ontology** extended for new application domain
 - new Structural Entity
 - new Embedded Operation
 - new Relation
- **Generics**

Attributes



Structure Attributes

```
\begin{Section} [Label=Attributes, Title=Attributes,  
    Authors={Bernd~Krieg-Br{"u}ckner}]  
    \begin{Paragraph} [Title=Layout,  
        Authors={Jan-Georg~Smaus, Markus~Roggenbach}]  
    \end{Paragraph}  
    \begin{Paragraph} [Title={Variant Attributes}]
```

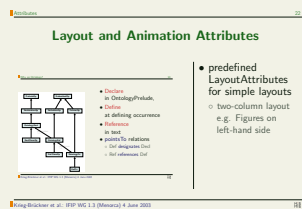
- most recent AuthorsAttribute Attribute, stack of PriorAuthors Attributes
- Attributes are **inherited** downwards, may be overwritten

Layout and Animation

Layout and Animation Attributes

Layout and Animation Attributes

Layout and Animation Attributes



- predefined LayoutAttributes for simple layouts
 - two-column layout e.g. Figures on left-hand side

- predefined LayoutAttributes for simple layouts
 - two-column layout e.g. Figures on left-hand side

Layout and Animation Attributes

Variant Attributes

33

	<i>Hyper</i>	<i>Replay</i>	<i>Interactive</i>
Contents skeleton	Hyper-Medium	Replay in Tool	Interaction
Outline abstracts			
Lecture presentation in class	laptop browsing during lecture	demonstrate use of tool	experiment in tool
Lecture Notes annotated after presentation	offline browsing personal annotation	demonstrate examples	experiment with examples
Course self-contained for self-study	personal navigation dynamic?	model solutions	personal solutions

Krieg-Brückner et al.: IFIP WG 1.3 (Menorca) 4 June 2003

MiS

- predefined
LayoutAttributes
for simple layouts
 - n by m matrix
(as list of lists)

Layout and Animation Attributes

- predefined `LayoutAttributes` for simple layouts
 - two-column layout, e.g. for Figures on left-hand side
 - n by m matrix (as list of lists)
- predefined `AnimationAttributes` for simple animations
 - rollout of list items
 - traversal of matrix: left-to right or top-to-bottom
 - sequential build-up of Figures
 - pointers via arrows
- avoid tedious \LaTeX commands or pause levels

Animation of Proof Trees

- \LaTeX proof-tree style
- pause levels
 - based on pause command provided by PPower4
 - text fragment (“subtree”) appears at specified pause level range
 - original pause levels are restored afterwards

(Black)Board Presentation, Notes

- overhead screen + script/annotations on laptop

Layout and Animation Attributes

Thumbnail of slide 23 content:

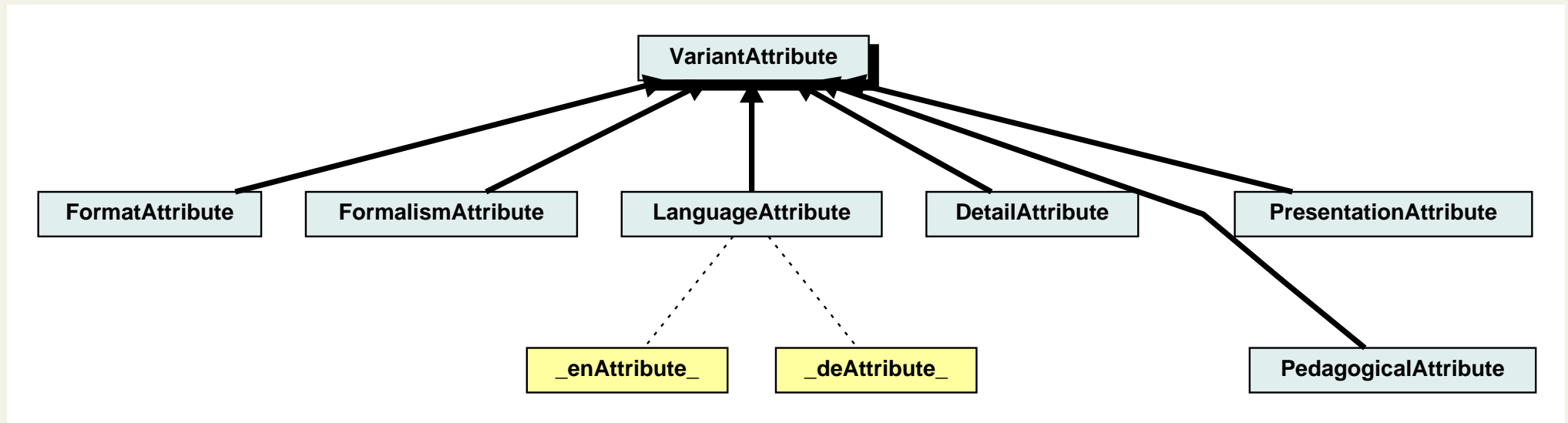
- Layout and Animation Attributes
- predefined LayoutAttributes for simple layouts
 - two-column layout e.g. Figures on left-hand side

Layout and Animation Attributes

- predefined LayoutAttributes for simple layouts
 - two-column layout, e.g. for Figures on left-hand side
 - n by m matrix (as list of lists)
- predefined AnimationAttributes for simple animations
 - rollout of list items
 - traversal of matrix: left-to right or top-to-bottom
 - sequential build-up of Figures
 - pointers via arrows
- avoid tedious \LaTeX commands or pause levels

Variant Attributes

Variant Attributes Ontology



	<i>Paper</i>	<i>Board</i>	<i>Hyper</i>
Contents	<i>Text+Pictures</i>	Manual	<i>Hyper-Medium</i>
skeleton			
Outline			
abstracts			
Lecture			
presentation	handout	black/white board	laptop browsing
in class	before lecture	during lecture	during lecture
Lecture Notes			
annotated	handout	annotated	offline browsing
after presentation	after lecture	manuscript	personal annotation
Course			
self-contained	course script	integrated	personal navigation
for self-study	personalized?	(manu)script	dynamic?

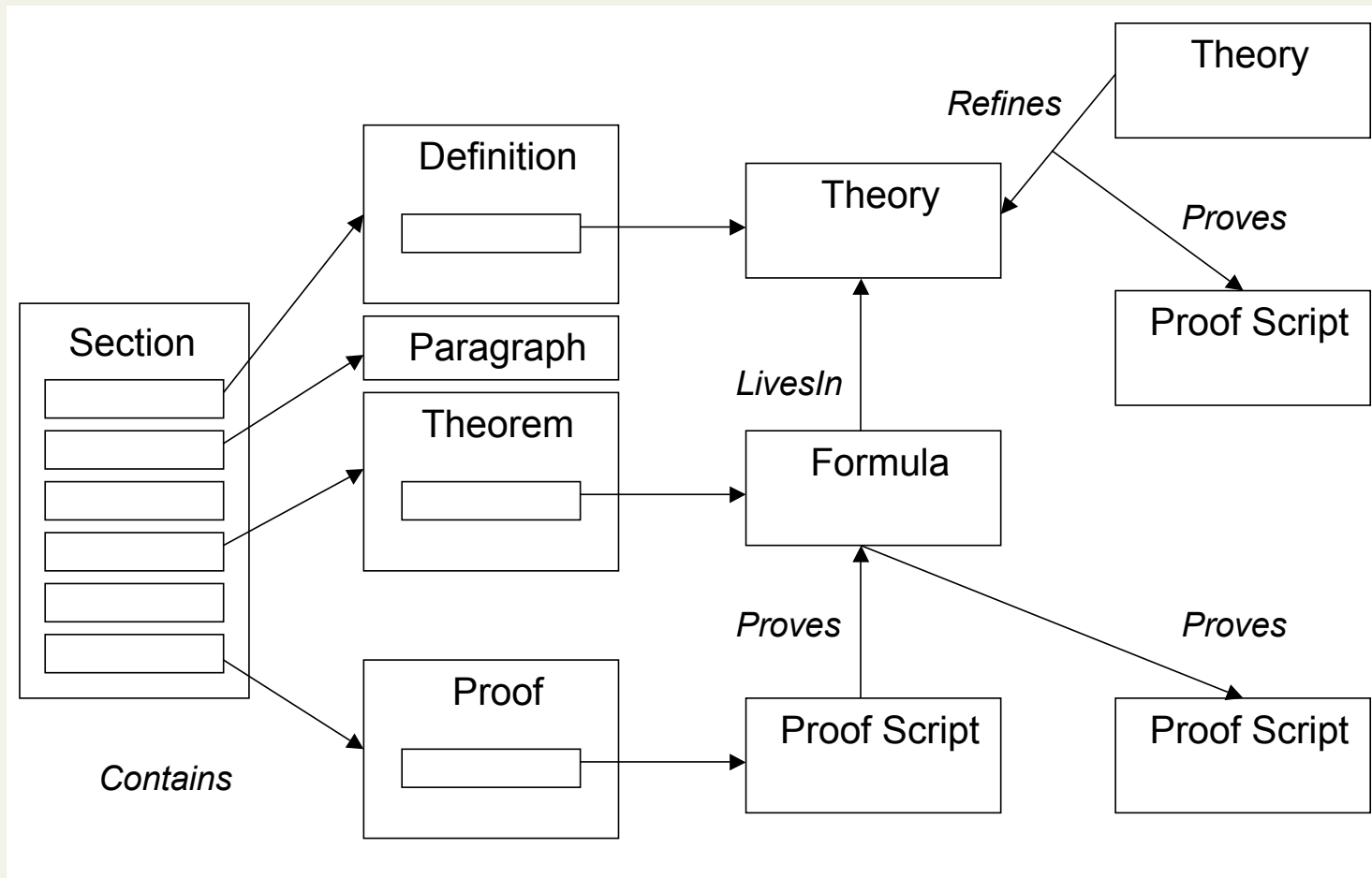
	<i>Hyper</i>	<i>Replay</i>	<i>Interactive</i>
Contents	<i>Hyper-Medium</i>	<i>Replay in Tool</i>	<i>Interaction</i>
skeleton			
Outline			
abstracts			
Lecture			
presentation in class	laptop browsing during lecture	demonstrate use of tool	experiment in tool
Lecture Notes			
annotated after presentation	offline browsing personal annotation	demonstrate examples	experiment with examples
Course			
self-contained for self-study	personal navigation dynamic?	model solutions	personal solutions

Variants

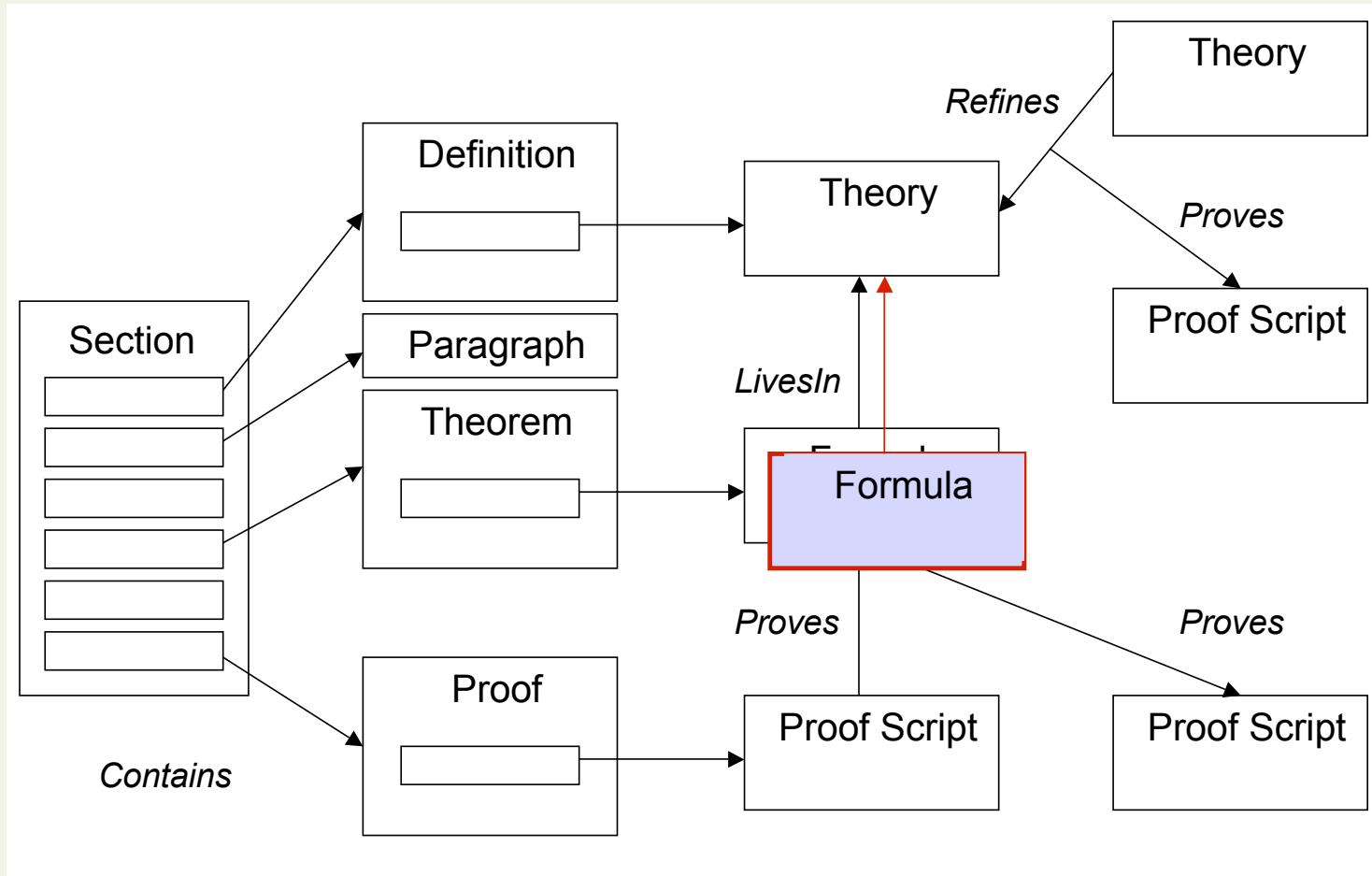
- **one structure graph** for all variants
- **different variants** for one node with the same Label
- can be edited **side-by-side**, or
- for a given **selection** of VariantAttributes
a **projection** of the structure graph
can be used for specialised editing
- **variants** are perhaps the **most important** novelty

Configuration Management

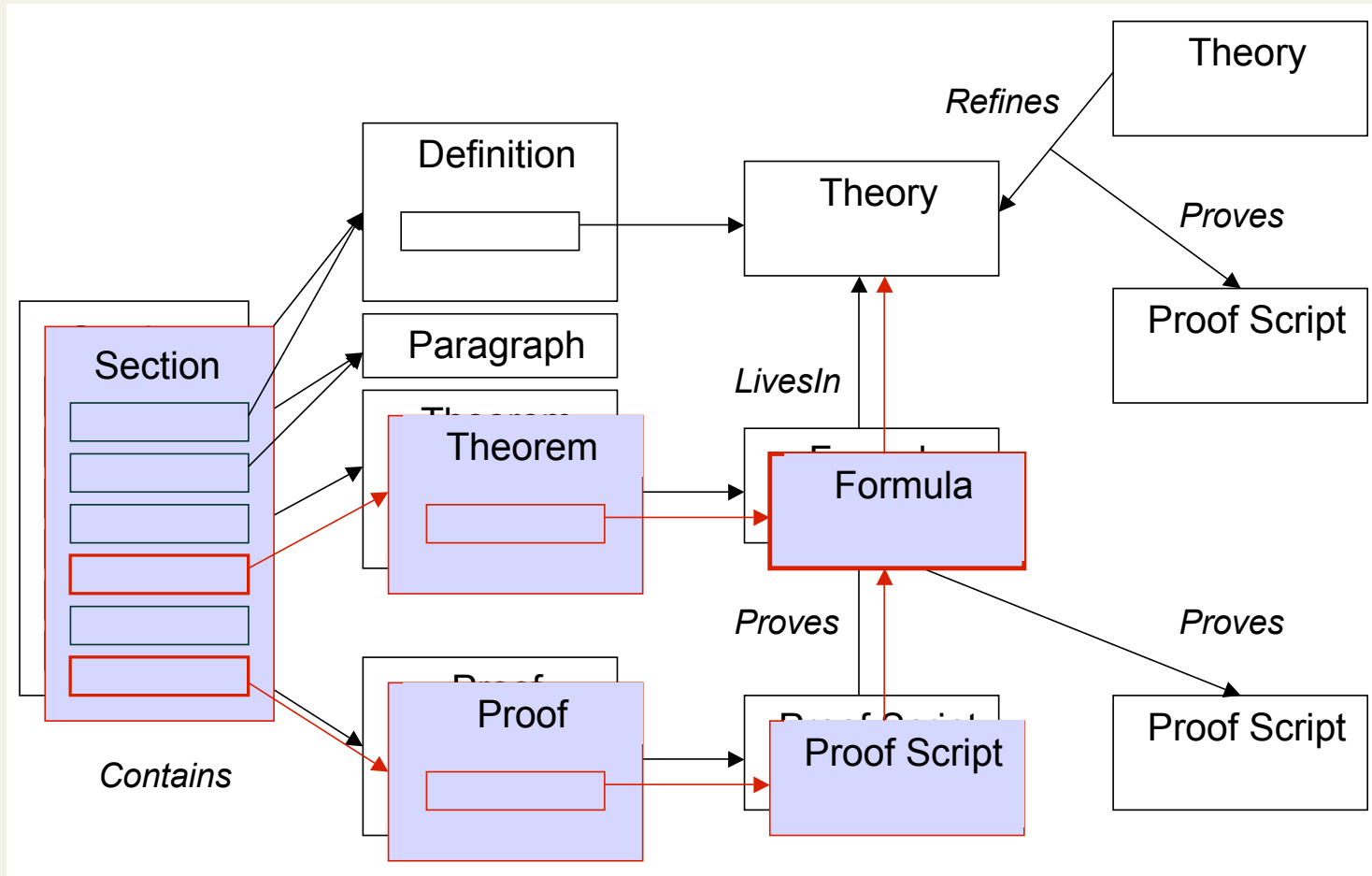
Conceptual and Formal Structure



Versions and Configurations



Versions and Configurations



Versions, Configurations, Views

- revision
 - merge of revisions
- (partial) configuration
 - Start: node in the development graph
- complete configuration, selected views
 - German slides for a particular Lecture
 - Exercises and model solutions, incomplete Proofs
- consistent configuration
 - compatible versions
 - change management

Change Management

Inheritance for Relations in Ontology

```
\DeclRel{*-*}{comprises}{comprises}{relatesDocConstructs}  
  \DeclRel{<-}{contains}{contains}{comprises}  
\DeclRel{>}{reliesOn}{reliesOn}{relatesDocConstructs}  
  \DeclRel{}{imports}{imports}{reliesOn}  
  \DeclRel{}{livesIn}{livesIn}{reliesOn}  
  \DeclRel{}{proves}{proves}{reliesOn}  
  \DeclRel{}{after}{after}{reliesOn}  
\DeclRel{->}{pointsTo}{pointsTo}{relatesDocConstructs}  
  \DeclRel{}{designates}{designates}{pointsTo}  
  \DeclRel{}{references}{references}{pointsTo}  
\DeclRel{->}{variantOf}{variantOf}{relatesDocConstructs}
```

Consistency and Completeness

- variantOf relations
 - **completeness**: at least one target
 - **consistency**: at most one target,
e.g. for A_{de} *languageVariantOf* A_{en}
 - **completeness**: for all sources and targets,
e.g. $\forall X_{en} . \exists X_{de} . X_{de}$ *languageVariantOf* X_{en}

Consistency and Completeness

- **pointsTo** relations
 - many-to-one (not transitive, “one step”)
 - **completeness**: at least one target,
e.g. no dangling **Ref** references **Def**
 - **consistency**: at most one target
 - weak **completeness**: at least one source,
e.g. for **Def** designates **Decl**

Consistency and Completeness

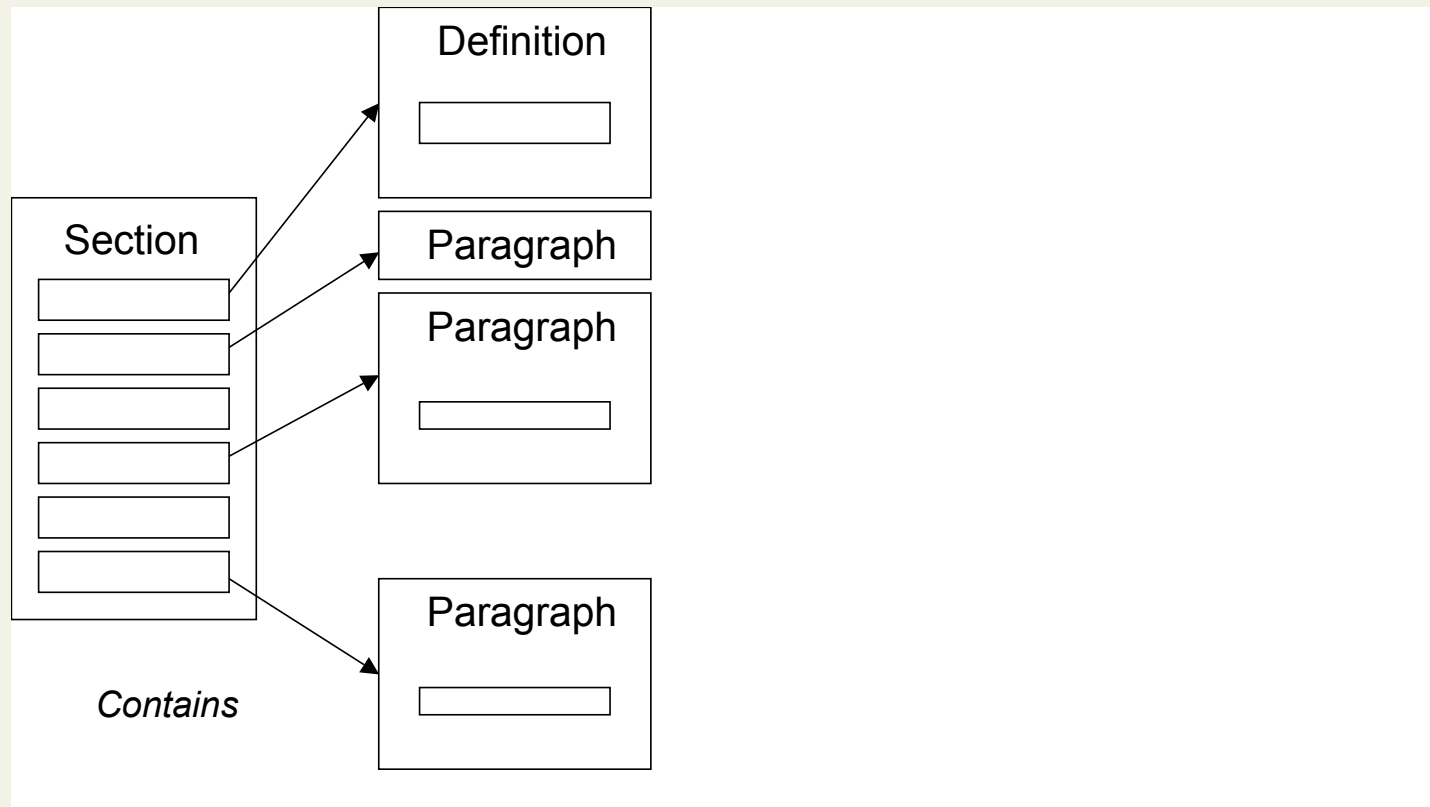
- **reliesOn** relations
 - strict order (irreflexive, transitive)
 - **completeness**: at least one target,
e.g. no dangling **Theorem livesIn Theory**
 - **consistency**: **reliesOn** relations internally consistent
e.g. **A after B** and **B after C** implies **A after C**
 - **consistency**: **reliesOn** relations consistent with each other
e.g. **A livesIn B** implies **A after B**

Consistency and Completeness

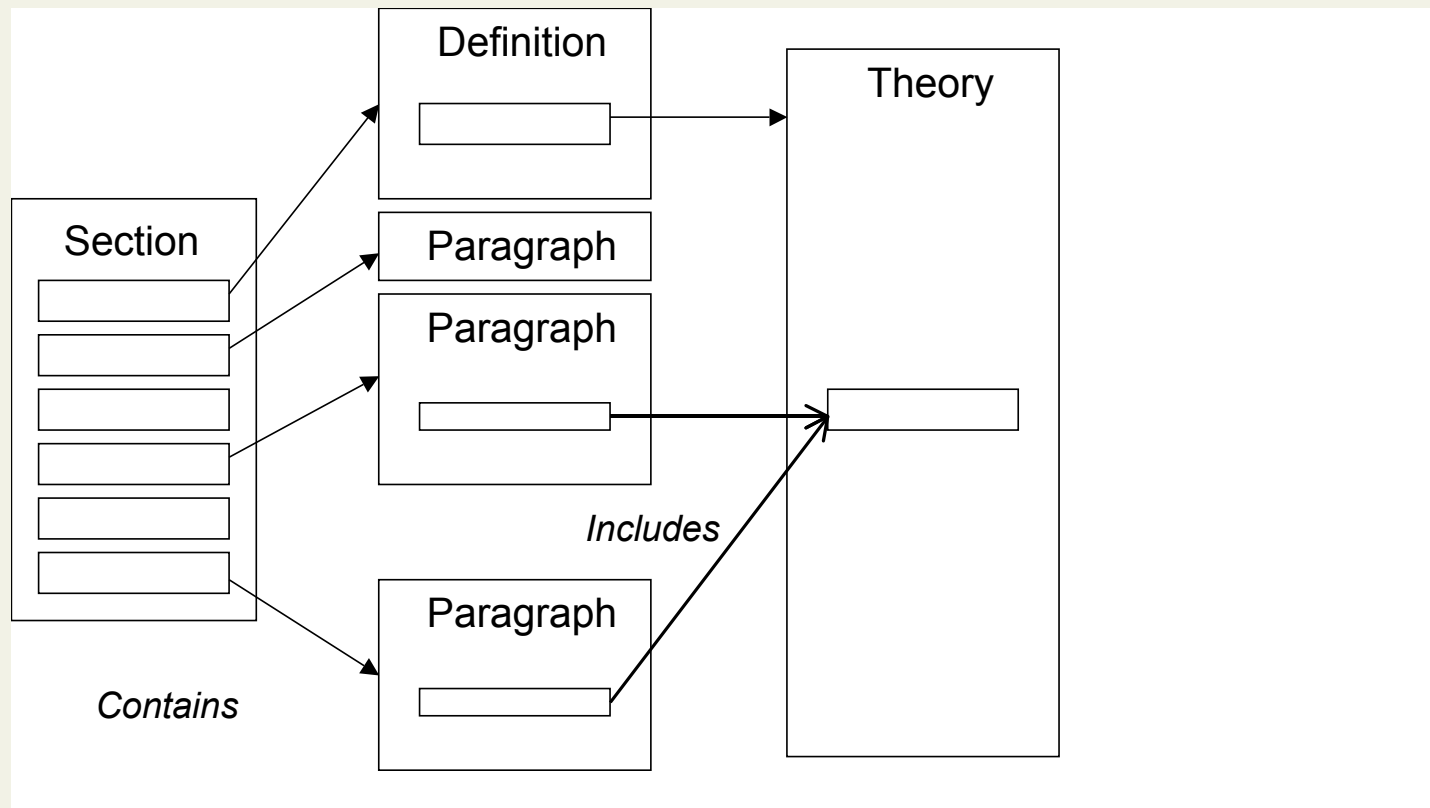
- **pointsTo** and **reliesOn** relations
 - **consistency** (possibly): warnings for forward references
- **comprises** and **reliesOn** relations
 - **completeness**: closure properties,
e.g. a Proof for each Theorem in a Package

Literate Programming

Sharing



Sharing



Literate Programming

- complete Program in extra Section (e.g. appendix)
 - complete Program with internal structure, down to Atoms
 - Labels allow sharing, e.g. IncludeAtom in document
- use of tools, e.g. for Haskell
 - internal structure converted to special annotations in Haskell
 - tools can be applied to ASCII format
 - original structure is recovered from special annotations
 - additional \LaTeX variants are generated, associated one-to-one to ASCII variants in Repository
 - revisions are only made in source

Conclusion

Sustainable Development

- versions, configurations, views
 - development graph
 - complete configurations, consistent configurations
 - merge of revisions
 - change management
- Attributes, variants
 - FormalismAttributes
 - DetailAttribute, refinement, abstraction
 - PresentationAttribute (with tools)

Sustainable Development

- semantic interrelation
 - structured ontology, classes, objects, relations
- extension
 - Macros, Generics (future), SystemExtensionPrelude
- re-use
 - structural sharing
 - Packages
- tool support
- development and presentation environment

Goals of the MMiSS Project

- develop **slides**, lecture notes, **courses**
 - DetailAttribute
- work on the **board**, with slides, interactively with **tools**
 - PresentationAttribute
- embed mathematical formulae, **programs**, etc.
 - L^AT_EX, Interactive use of tools
- manage **English** and **German** documents in parallel
 - variants, LanguageAttribute

Goals of the MMiSS Project

- publish **complete** and **consistent** “packages”
 - versions, configuration management
- (partially) **re-use** the slides of a colleague
 - re-use by structural sharing, views
- be made aware of the **changes** made by your colleague
 - sustainable development via change management
- agree with your colleague on a **uniform terminology**
 - ontology

Join the
International MMiSS Forum
www.mmiss.de
Open Source!