Method and tool support for classifying software languages with Wikipedia

Ralf Lämmel, Dominik Mosen and Andrei Varanovich
Software Languages Team, University of Koblenz-Landau

http://softlang.uni-koblenz.de/wikitax/

Why and how to classify software languages?

planet-sl.org/sle2013/

The term "software language" refers to artificial languages used in software development. These include general-purpose programming languages, domain-specific languages, modeling and metamodeling languages, data models and ontologies. Examples include general purpose modeling languages such as SysML and UML, metamodeling frameworks such as Ecore, MOF or GOPRR, domain-specific modeling languages for business process modeling, such as BPMN, or embedded systems, such as Simulink or Modelica, and specialized XML-based and OWL-based languages and vocabularies. The term "software language" is intentionally broad; besides the above categories and examples, it also encompasses implicit approaches to language definition, such as APIs and collections of design patterns.

planet-sl.org/**slebok**/

SL(E) BOK 2.0

Group Announcements Discussions Photos Videos Members

SL(E)BOK 2.0

- Home
- SL(E)
- BOK
- 2.0

Welcome

SL(E)BOK 2.0 is an emerging community-based collaborative-oriented **project** that aims at creating and maintaining a **Body Of Knowledge**(BOK) about **Software Languages, Software Linguistics and Software Language Engineering** (SL(E)).

Agenda

September 2012

SL(E)BOK @ SLE2012 takes place on September 25th, 2012, at Dresden, Germany

101companies: the emerging hitchhiker's guide through the software galaxy

http://101companies.org/wiki/**Software_language**

- Language:Parsec *instanceOf* this
- this *instanceOf* Namespace:Concept
- this *instanceOf* Vocabulary:Software language engineering
- Data manipulation language *isA* this
- Query language *isA* this
- Style sheet language *isA* this
- Tool-defined language *isA* this
- Transformation language *isA* this
- XML language *isA* this

BTW, where is the Wikipedia for “Software Language”?

:-)

Anyone?

Problem-specific *exploration* tools

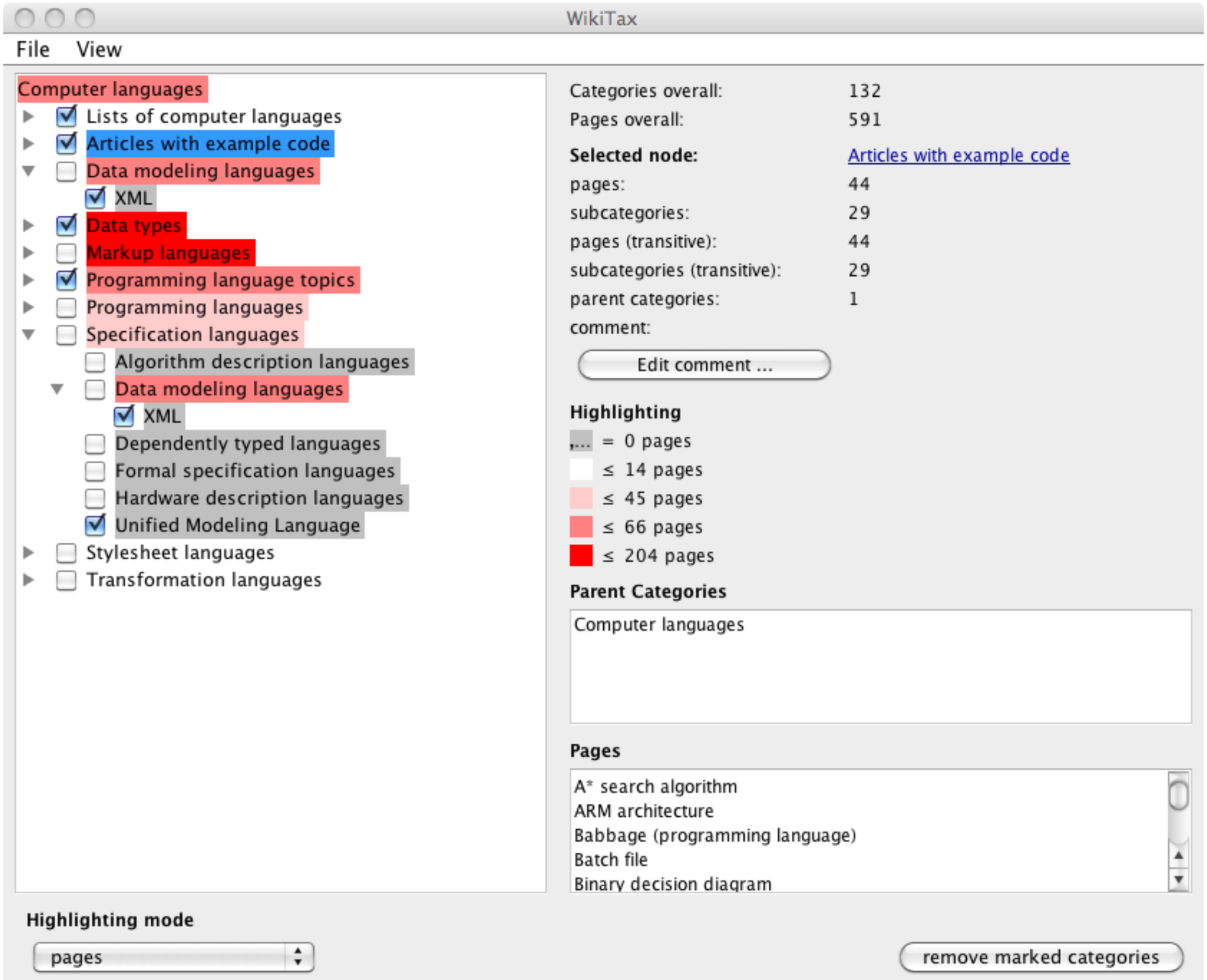
- Baskaya et al.: A tool for ontology-editing and **ontology**-based information exploration. ESAIR 2010.
- Haun et al.: CET: A tool for creative exploration of **graphs**. ECML/PKDD 2010.
- Dumas et al.: ViDaX: an interactive **semantic data** visualisation and exploration tool. AVI 2012.
- Hora et al.: Bug Maps: A tool for the visual exploration and analysis of **bugs**. CSMR 2012.
- De Roover et al: Multi-dimensional exploration of **API** usage. ICPC 2013.

Category graph exploration with

WikiTax

http://softlang.uni-koblenz.de/wikitax/

Category graph exploration with *WikiTax*



- Computer languages
  - [x] Lists of computer languages
  - [x] Articles with example code
  - [ ] Data modeling languages
    - [x] XML
  - [x] Data types
  - [ ] Markup languages
  - [x] Programming language topics
  - [ ] Programming languages
  - [ ] Specification languages
    - [ ] Algorithm description languages
    - [ ] Data modeling languages
      - [x] XML
    - [ ] Dependently typed languages
    - [ ] Formal specification languages
    - [ ] Hardware description languages
    - [x] Unified Modeling Language
  - [ ] Stylesheet languages
  - [ ] Transformation languages

Result of 2 levels of *extraction* with some categories marked for *exclusion*

Categories overall:	132
Pages overall:	591
Selected node:	Articles w
pages:	44
subcategories:	29
pages (transitive):	44
subcategories (transitive):	29
parent categories:	1
comment:	

Edit comment ...

Highlighting

- = 0 pages
- ≤ 14 pages
- ≤ 45 pages
- ≤ 66 pages
- ≤ 204 pages

Parent Categories

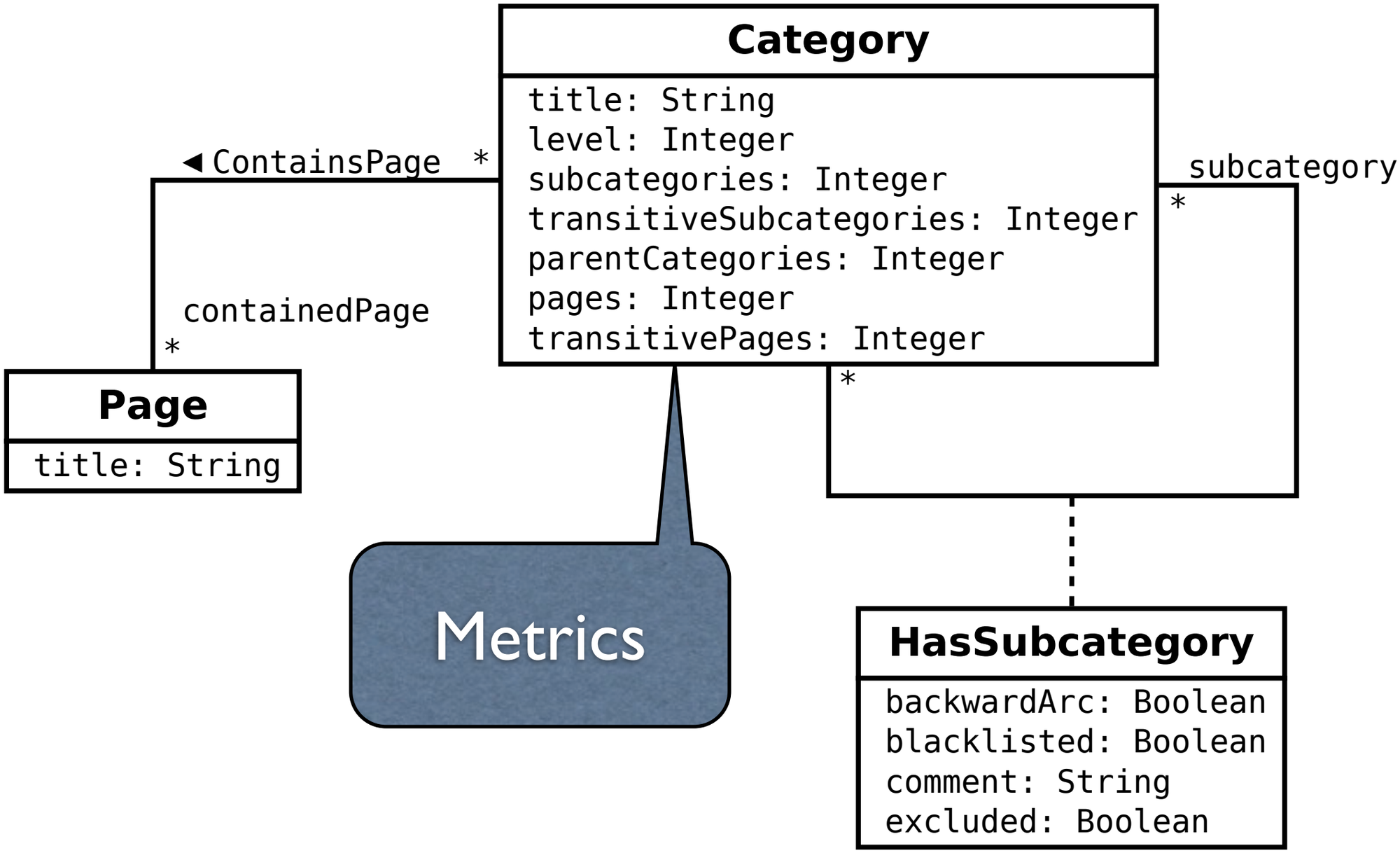
Computer languages

Pages

The *WikiTax* approach

- Category graph **extraction** (from *Wikipedia*)
- ... **reduction** (by the exclusion of categories)
- ... **visualization** (using simple metrics)
- ... **export** (for external processing)

Metamodel of the *WikiTax* category graph

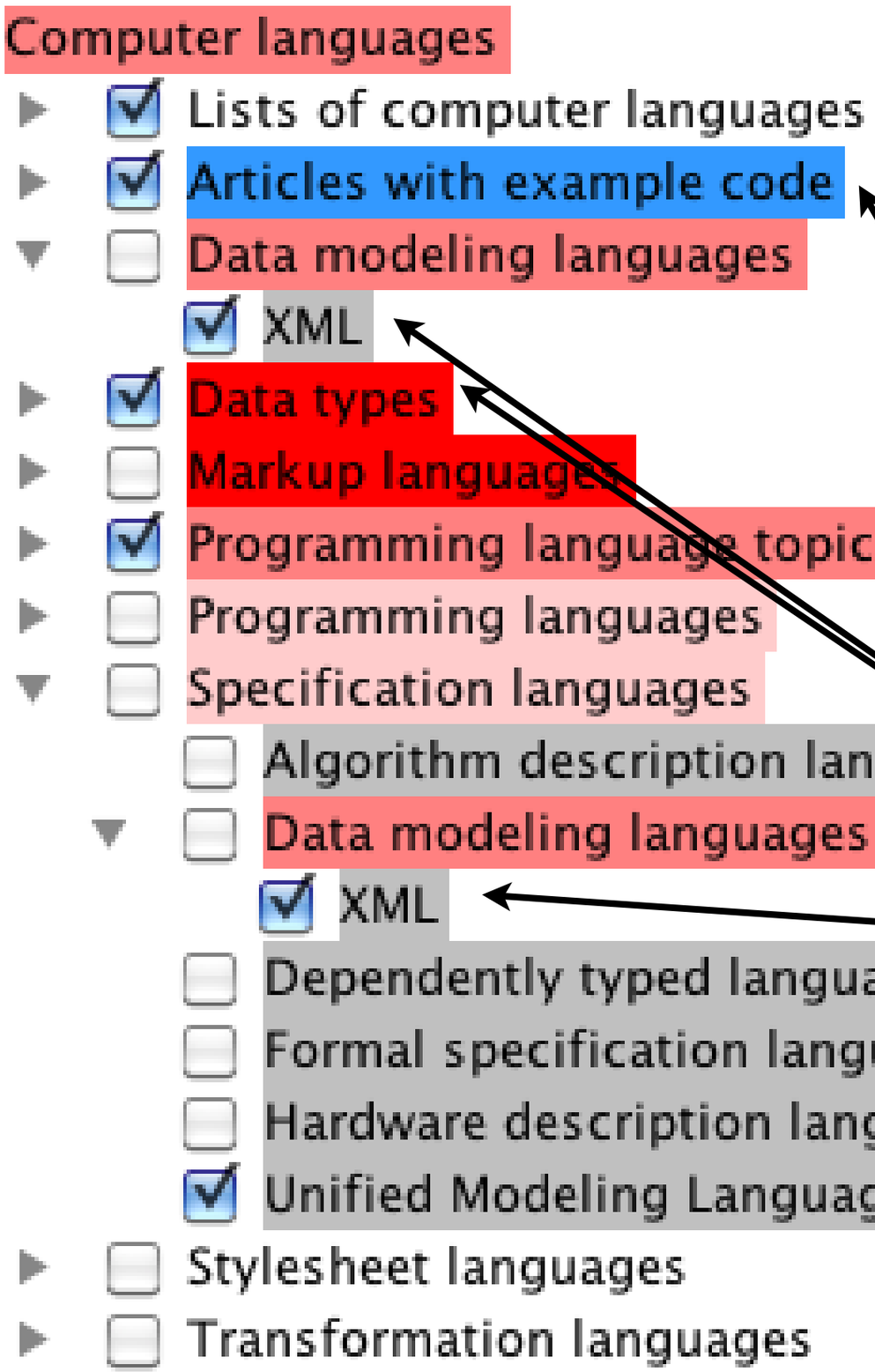


Implementation of *WikiTax*

- Wikipedia API
- TGraphs, JSON, and CSV
- Java (Swing)
- JGraLab

See GitHub URL etc.:
http://softlang.uni-koblenz.de/wikitax/
(Open Source and Open Data)

Problem-specific concern: ***exclusion***



Why and what and how to exclude?

Are these true classifiers in terms of software concepts?

Exclusion types

- Alternative classifier (unrelated to software concepts)
  - e.g., *Academic programming languages*
- Deviating classifier (in fact, non-classifier)
  - e.g., *Articles with example code*
- Singleton classifier (focusing on one language)
  - e.g., *Cascading Style Sheets*
- List classifier (collecting list pages)
  - e.g., *Lists of programming languages*
- Maintenance classifier
  - e.g., *Uncategorized programming languages*

Exclusion types for levels 1 and 2 of Computer languages

Category	Exclusion type
Academic programming languages	Alternative classifier
Articles with example code	Deviating classifier
Cascading Style Sheets	Singleton classifier
Data types	Deviating classifier
Discontinued programming languages	Alternative classifier
DocBook	Singleton classifier
Esoteric programming languages	Alternative classifier
Experimental programming languages	Alternative classifier
HTML	Singleton classifier
JSON	Singleton classifier
Lists of computer languages	List classifier
Lists of programming languages	List classifier
Markup language comparisons	Deviating classifier
Markup language stubs	Maintenance classifier
Non-English-based programming languages	Alternative classifier
Programming language families	Deviating classifier
Programming language standards	Deviating classifier
Programming language topics	Deviating classifier
Programming languages by creation date	Alternative classifier
Programming languages conferences	Deviating classifier
Software by programming language	Deviating classifier
SyncML	Singleton classifier
TeX	Singleton classifier
Text Encoding Initiative	Singleton classifier
Troff	Singleton classifier
Uncategorized programming languages	Maintenance classifier
Unified Modeling Language	Singleton classifier
Wikipedia categories named after programming languages	Deviating classifier
XML	Singleton classifier

Computer languages

Category	**Subcategories**
Data modeling languages	–
Markup languages	*Declarative markup languages*, *GIS file formats*, *Knowledge representation languages*, *Lightweight markup languages*, *Mathematical markup languages*, *Musical markup languages*, *Page description markup languages*, *Playlist markup languages*, *User interface markup languages*, *Vector graphics markup languages*, *Web syndication formats*, *XML markup languages*
Programming languages	*.NET programming languages*, *Agent-based programming languages*, *Agent-oriented programming languages*, *Concatenative programming languages*, *Concurrent programming languages*, *Data-structured programming languages*, *Declarative programming languages*, *Dependently typed languages*, *Domain-specific programming languages*, *Dynamic programming languages*, *Extensible syntax programming languages*, *Formula manipulation languages*, *Function-level languages*, *Functional languages*, *High Integrity Programming Language*, *High-level programming languages*, *ICL programming languages*, *Intensional programming languages*, *Low-level programming languages*, *Multi-paradigm programming languages*, *Nondeterministic programming languages*, *Object-based programming languages*, *Pattern matching programming languages*, *Procedural programming languages*, *Process termination functions*, *Prototype-based programming languages*, *Reactive programming languages*, *Secure programming languages*, *Set theoretic programming languages*, *Statically typed programming languages*, *Synchronous programming languages*, *Term-rewriting programming languages*, *Text-oriented programming languages*, *Tree programming languages*, *Visual programming languages*, *XML-based programming languages*
Specification languages	*Algorithm description languages*, *Dependently typed languages*, *Formal specification languages*, *Hardware description languages*
Stylesheet languages	–
Transformation languages	*Macro programming languages*

Category	**Subcategories**
Data modeling languages	–
Markup languages	*Declarative markup languages*, *GIS file formats*, *Knowledge representation languages*, *Lightweight markup languages*, *Mathematical markup languages*, *Musical markup languages*, *Page description markup languages*, *Playlist markup languages*, *User interface markup languages*, *Vector graphics markup languages*, *Web syndication formats*, *XML markup languages*
Programming languages	*.NET programming languages*, *Agent-based programming languages*, *Agent-oriented programming languages*, *Concatenative programming languages*, *Concurrent programming languages*, *Data-structured programming languages*, *Declarative programming languages*, *Dependently typed languages*, *Domain-specific programming languages*, *Dynamic programming languages*, *Extensible syntax programming languages*, *Formula manipulation languages*, *Function-level languages*, *Functional languages*, *High Integrity Programming Language*, *High-level programming languages*, *ICL programming languages*, *Intensional programming languages*, *Low-level programming languages*, *Multi-paradigm programming languages*, *Nondeterministic programming languages*, *Object-based programming languages*, *Pattern matching programming languages*, *Procedural programming languages*, *Process termination functions*, *Prototype-based programming languages*, *Reactive programming languages*, *Secure program-*

	native programming languages, *Concurrent programming languages*, *Data-structured programming languages*, *Declarative programming languages*, *Dependently typed languages*, *Domain-specific programming languages*, *Dynamic programming languages*, *Extensible syntax programming languages*, *Formula manipulation languages*, *Function-level languages*, *Functional languages*, *High Integrity Programming Language*, *High-level programming languages*, *ICL programming languages*, *Intensional programming languages*, *Low-level programming languages*, *Multi-paradigm programming languages*, *Nondeterministic programming languages*, *Object-based programming languages*, *Pattern matching programming languages*, *Procedural programming languages*, *Process termination functions*, *Prototype-based programming languages*, *Reactive programming languages*, *Secure programming languages*, *Set theoretic programming languages*, *Statically typed programming languages*, *Synchronous programming languages*, *Term-rewriting programming languages*, *Text-oriented programming languages*, *Tree programming languages*, *Visual programming languages*, *XML-based programming languages*
Specification languages	*Algorithm description languages*, *Dependently typed languages*, *Formal specification languages*, *Hardware description languages*
Stylesheet languages	–
Transformation languages	*Macro programming languages*

An aside:
Where are the query languages?

Let’s compare SQL and XSLT!

http://en.wikipedia.org/wiki/**SQL**

Categories: Database management systems Computer languages Data modeling languages Declarative programming languages **Query languages** Relational database management systems SQL

Category:Query_languages

Categories: Domain-specific programming languages; Data management Databases

A level-2 subcategory of computer languages

http://en.wikipedia.org/wiki/**XSLT**

Categories: Declarative programming languages Functional languages Markup languages Transformation languages World Wide Web Consortium standards XML-based programming languages XML-based standards

A level-1 subcategory of computer languages

Category:Transformation_language

Categories: Computer languages

Programming languages -- all levels

- Initial extraction
  - 423 categories, 7515 pages, 8 levels
- 1st pruning phase
  - 29 excluded categories as discussed earlier
  - 288 categories, 6671 pages
- 2nd pruning phase
  - 79 categories, 1560 pages, 4 levels

Two hours of work

Pages

Programming languages
- ICL programming languages
- Agent-based programming languages
- Agent-oriented programming languages
- Concatenative programming languages
- Concurrent programming languages
- Data-structured programming languages
- Declarative programming languages
- Dependently typed languages
- Domain-specific programming languages
- Dynamic programming languages
- Extensible syntax programming languages
- Formula manipulation languages
- Function-level languages
- Functional languages
- High Integrity Programming Language
- High-level programming languages
- Intensional programming languages
- Low-level programming languages
- Multi-paradigm programming languages
- .NET programming languages
- Nondeterministic programming languages
- Object-based programming languages
- Pattern matching programming languages
- Procedural programming languages
- Process termination functions
- Prototype-based programming languages
- Reactive programming languages
- Secure programming languages
- Set theoretic programming languages
- Statically typed programming languages
- Synchronous programming languages
- Term-rewriting programming languages
- Text-oriented programming languages
- Tree programming languages
- Visual programming languages
- XML-based programming languages

Categories

Programming languages
- ICL programming languages
- Agent-based programming languages
- Agent-oriented programming languages
- Concatenative programming languages
- Concurrent programming languages
- Data-structured programming languages
- Declarative programming languages
- Dependently typed languages
- Domain-specific programming languages
- Dynamic programming languages
- Extensible syntax programming languages
- Formula manipulation languages
- Function-level languages
- Functional languages
- High Integrity Programming Language
- High-level programming languages
- Intensional programming languages
- Low-level programming languages
- Multi-paradigm programming languages
- .NET programming languages
- Nondeterministic programming languages
- Object-based programming languages
- Pattern matching programming languages
- Procedural programming languages
- Process termination functions
- Prototype-based programming languages
- Reactive programming languages
- Secure programming languages
- Set theoretic programming languages
- Statically typed programming languages
- Synchronous programming languages
- Term-rewriting programming languages
- Text-oriented programming languages
- Tree programming languages
- Visual programming languages
- XML-based programming languages

Pages

Programming languages

- ICL programming languages
- Agent-based programming languages
- Agent-oriented programming languages
- Concatenative programming languages
- Concurrent programming languages
- ▶ Data-structured programming languages
- ▶ Declarative programming languages
- Dependently typed languages
- ▶ Domain-specific programming languages
- ▶ Dynamic programming languages
- Extensible syntax programming languages
- Formula manipulation languages
- Function-level languages
- Functional languages
- High Integrity Programming Language
- High-level programming languages
- Intensional programming languages
- Low-level programming languages
- ▶ Multi-paradigm programming languages
- .NET programming languages
- Nondeterministic programming languages
- ▶ Object-based programming languages
- Pattern matching programming languages
- ▶ Procedural programming languages
- Process termination functions
- Prototype-based programming languages
- Reactive programming languages
- Secure programming languages

Categories

Programming languages

- ICL programming languages
- Agent-based programming languages
- Agent-oriented programming languages
- Concatenative programming languages
- Concurrent programming languages
- ▶ Data-structured programming languages
- ▶ Declarative programming languages
- Dependently typed languages
- ▶ Domain-specific programming languages
- ▶ Dynamic programming languages
- Extensible syntax programming languages
- Formula manipulation languages
- Function-level languages
- Functional languages
- High Integrity Programming Language
- High-level programming languages
- Intensional programming languages
- Low-level programming languages
- ▶ Multi-paradigm programming languages
- .NET programming languages
- Nondeterministic programming languages
- ▶ Object-based programming languages
- Pattern matching programming languages
- ▶ Procedural programming languages
- Process termination functions
- Prototype-based programming languages
- Reactive programming languages
- Secure programming languages

- [ ] Declarative programming languages
- [ ] Dependently typed languages
- [ ] Domain-specific programming languages
- [ ] Dynamic programming languages
- [ ] Extensible syntax programming languages
- [ ] Formula manipulation languages
- [ ] Function-level languages
- [ ] Functional languages
- [ ] High Integrity Programming Language
- [ ] High-level programming languages
- [ ] Intensional programming languages
- [ ] Low-level programming languages
- [ ] Multi-paradigm programming languages
- [ ] .NET programming languages
- [ ] Nondeterministic programming languages
- [ ] Object-based programming languages
- [ ] Pattern matching programming languages
- [ ] Procedural programming languages
- [ ] Process termination functions
- [ ] Prototype-based programming languages
- [ ] Reactive programming languages
- [ ] Secure programming languages
- [ ] Set theoretic programming languages
- [ ] Statically typed programming languages
- [ ] Synchronous programming languages
- [ ] Term-rewriting programming languages
- [ ] Text-oriented programming languages
- [ ] Tree programming languages
- [ ] Visual programming languages
- [ ] XML-based programming languages

- [ ] Declarative programming languages
- [ ] Dependently typed languages
- [ ] Domain-specific programming languages
- [ ] Dynamic programming languages
- [ ] Extensible syntax programming languages
- [ ] Formula manipulation languages
- [ ] Function-level languages
- [ ] Functional languages
- [ ] High Integrity Programming Language
- [ ] High-level programming languages
- [ ] Intensional programming languages
- [ ] Low-level programming languages
- [ ] Multi-paradigm programming languages
- [ ] .NET programming languages
- [ ] Nondeterministic programming languages
- [ ] Object-based programming languages
- [ ] Pattern matching programming languages
- [ ] Procedural programming languages
- [ ] Process termination functions
- [ ] Prototype-based programming languages
- [ ] Reactive programming languages
- [ ] Secure programming languages
- [ ] Set theoretic programming languages
- [ ] Statically typed programming languages
- [ ] Synchronous programming languages
- [ ] Term-rewriting programming languages
- [ ] Text-oriented programming languages
- [ ] Tree programming languages
- [ ] Visual programming languages
- [ ] XML-based programming languages

Thanks! Questions?

http://softlang.uni-koblenz.de/wikitax/