

Semantic Web and Linked Data Technologies for E-Learning

Tobias Gleißner

Personal Information

- Name: Tobias Gleißner
- Studiengang: Informatik M. Sc.
- STPO: 2014/15
- Modul: Wissenschaftliches Arbeiten Praktische Informatik A
- Formen aktiver Teilnahme: "Vortrag, schriftliche Ausarbeitung, regelmäßige Diskussionsbeiträge"
- Modulprüfung: "Schriftliche Ausarbeitung (ca. 4 500 Wörter) mit mündlicher Präsentation (ca. 45 Minuten); die Modulprüfung wird nicht differenziert bewertet." (nicht differenziert bewertet = keine Benotung)

Goal

Create **individual** e-learning environments by

- Interlinking information on a topic
- Generating a useful presentation of the information for a user for
 - browsing the information space
 - searching for information
- Take user context into account

→ Link Generation



Title

Java Tutorial

☒ Control Flow Statements

Generalizations

☒ Language Basics

Details

[The for Statement](#)[The while and do-while Statements](#)[Exception Handling Statements](#)[The switch Statement](#)[The if/else Statements](#)[Branching Statements](#)

Summaries

[Summary of Control Flow Statements](#)

Exercises

[Questions and Exercises: Control Flow](#)

The Java™ Tutorial

[Start of Tutorial](#) > [Start of Trail](#) > [Start of Lesson](#)[Search](#)
[Feedback Form](#)**Trail:** Learning the Java Language**Lesson:** Language Basics

Control Flow Statements

When you write a program, you type statements into a file. Without control flow statements, the interpreter executes these statements in the order they appear in the file from left to right, top to bottom. You can use *control flow statements* in your programs to conditionally execute statements, to repeatedly execute a block of statements, and to otherwise change the normal, sequential flow of control. For example, in the following code snippet, the if statement conditionally executes the `System.out.println` statement within the braces, based on the return value of `Character.toUpperCase(aChar)`:

```
char c;  
  
if (Character.toUpperCase(aChar)) {  
    System.out.println("The character " + aChar + " is upper case.");  
}
```

The Java programming language provides several control flow statements, which are listed

Example - Personal Reader

- Goal: browse through information space
- Links are generated from
 - HTML structural elements
 - Annotations from the author
- User context
 - Read the preceding topics? Mark the section
 - Finished exercises? Highlight next topics

Unicode Characters

Securehttps://www.coursera.org/learn/python-databases/lecture/lvqrl/unicode-characters-and-strings

coursera

UNIVERSITY OF MICHIGAN

CatalogSearch catalog

For EnterpriseTobias

Back to Week 1Lessons

PrevNext

Welcome

Welcome to Using Databases with Python2 min

Help us learn more about you!10 min

Python Textbook10 min

Coming from Python 2 - Encoding Data in Python 310 min

Unicode Characters and Strings17 min

Notice for Auditing Learners: Assignment Submission10 min

Quiz: Using Encoded Data in Python 35 questions

Lectures

14.1 - Object Oriented Definitions and Terminology8 min

14.2 - Our First Class and Object7 min

14.3 - Object Life Cycle8 min

14.4 - Object Inheritance6 min

Assessments

Quiz: Object Oriented Programming11 questions

Bonus

Bonus: Interview - Software Engineering - Bertrand Meyer11 min

Network Programs - Part 3

PYTHON FOR EVERYBODY

ASCII

American Standard Code for Information Interchange

<https://en.wikipedia.org/wiki/ASCII>
<http://www.catonmat.net/download/ascii-cheat-sheet.png>

01:11 / 17:17

Unicode Characters and Strings

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

Search Transcript

English

0:09
So we started this entire course printing hello world and I just said quote, hello world, quote and out comes hello world. And it'd be nice if that was super simple.

0:22

Downloads

Lecture Videomp4

Subtitles (English)WebVTT

Transcript (English)txt

Would you like to help us translate the transcript and subtitles into additional languages?



Personal Learning Assistant



for

SMART SPACE FOR LEARNING

[Peter Dolog](#) & [Michael Sintek](#)

Personalized Search Service

Select user: Type in concept name(s):

(default)
Michael
Peter

intelli

Select one or more concepts from the ontology:

Computing Milieux
COMPUTERS AND SOCIETY
Electronic Commerce
 Electronic data interchange (EDI)
 Payment schemes
 Intellectual property
 Distributed commercial transactions
 Security
 Cybercash, digital cash
Social Issues
 Employment
 Handicapped persons and special needs
 Assistive technologies for persons with disabilities
 Abuse and crime involving computers
Organizational Impacts
 Reengineering
 Automation
 Employment
 Computer-supported collaborative work
General
Miscellaneous
Public Policy Issues
 Use and abuse of power
 Transborder data flow
 Privacy

☒ personal recommendation

Search

Reset

ELENA: PLA -- Personalized Search Service - Microsoft Internet Explorer

Datei Bearbeiten Ansicht Favoriten Extras ?



Adresse <http://localhost:8080/pla/result.jsp?user=michael&concept1=intelli&concept2=&concept3=&records=personal&action=Search>



Personal Learning Assistant

for

SMART SPACE FOR LEARNING

[Peter Dolog](#) & [Michael Sintek](#)

Personalized Search Service

User:

michael

Select one or more concepts:

Distributed artificial intelligence [in: ARTIFICIAL INTELLIGENCE << Computing Methodologies]
Intellectual property [in: Electronic Commerce << COMPUTERS AND SOCIETY << ...]
Intellectual property rights [in: Public Policy Issues << COMPUTERS AND SOCIETY << ...]
Intelligent Agents [in: Distributed artificial intelligence << ARTIFICIAL INTELLIGENCE << ...]
ARTIFICIAL INTELLIGENCE [in: Computing Methodologies]



Personal Learning Assistant

for

SMART SPACE FOR LEARNING

Peter Dolog & Michael Sintek



Personalized Search Service

User:

default

Selected concepts:

Intelligent Agents [In: Distributed artificial intelligence << ARTIFICIAL INTELLIGENCE << ...]

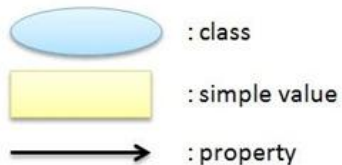
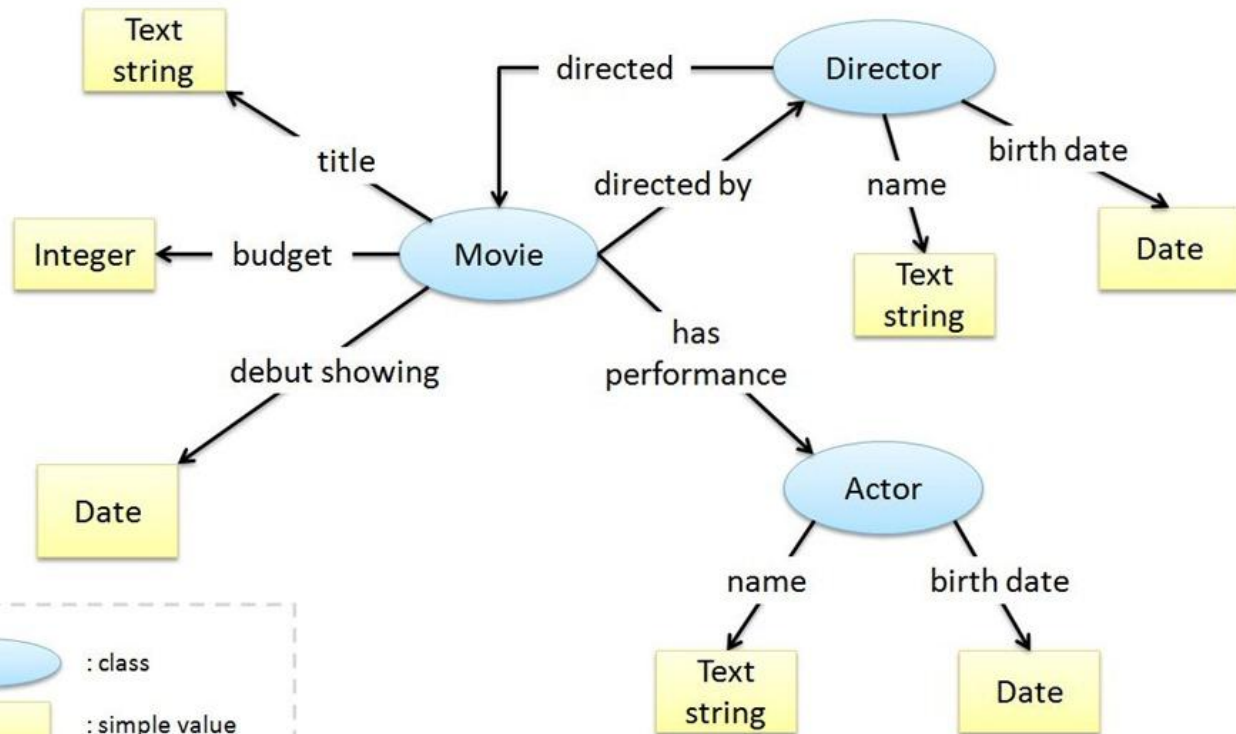
Query results:

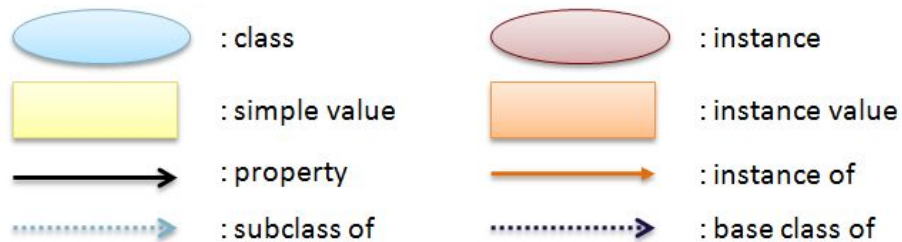
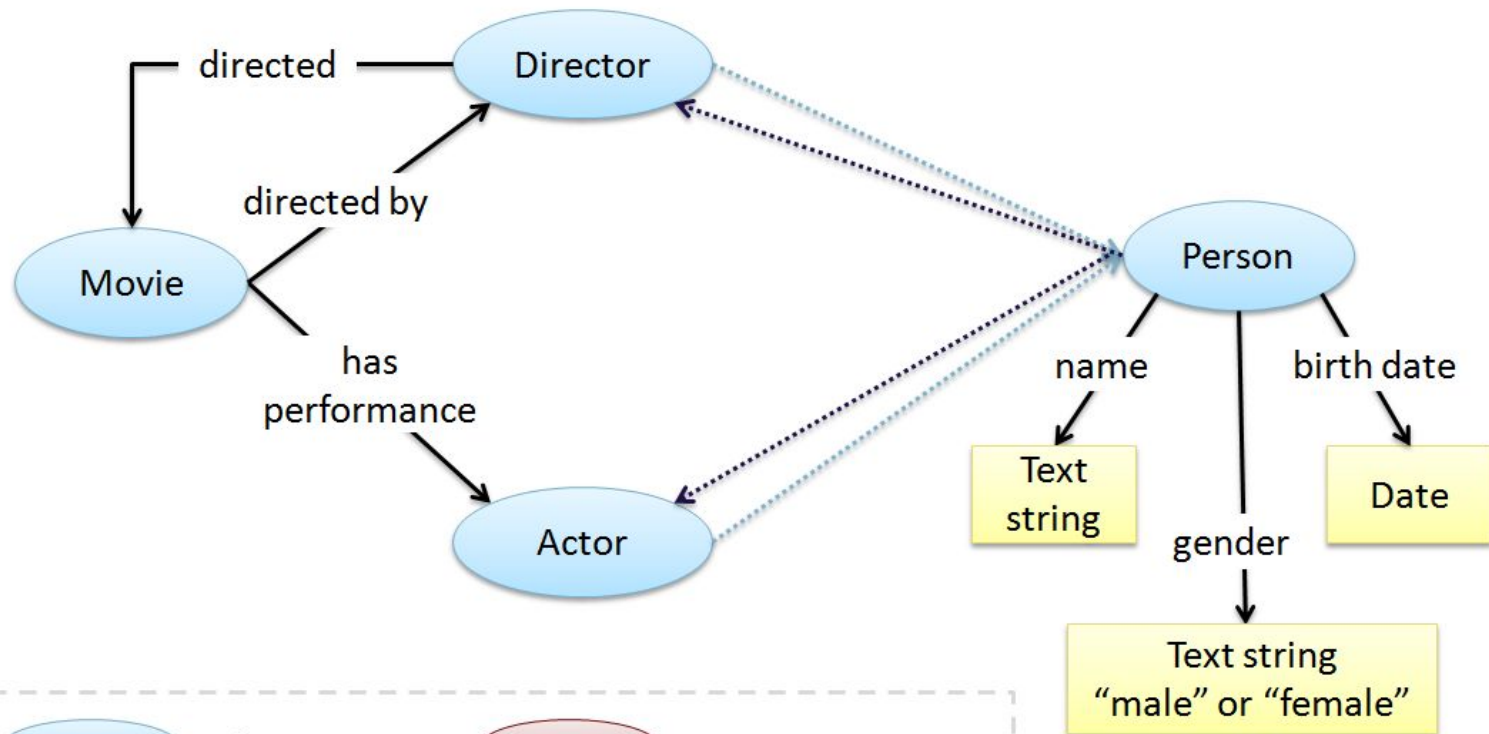
PReco	Reco	Title	Description	Concepts
		Aufgaben zum Thema Intelligente Agenten	Aufgaben, um den Stoff des Moduls zu vertiefen	Intelligent Agents
		Einige Fragen zum Thema Intelligente Agenten	Fragen, die Ihnen helfen sollen, den Stoff besser zu verstehen	Intelligent Agents
		Vorlesung Künstliche Intelligenz WS 2002 : Stichworte zum Thema Umgebungen	Was stellen die verschiedenen Grundtypen Intelligenter Agenten vor und ihre prinzipielle Programmierung	Intelligent Agents
		Weiterführende Materialien	Eine Sammlung von weiterführenden Links zum Thema Künstliche Intelligenz und Intelligente Agenten	Special-purpose; Intelligent Agents

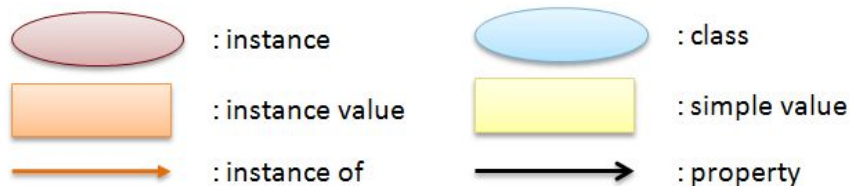
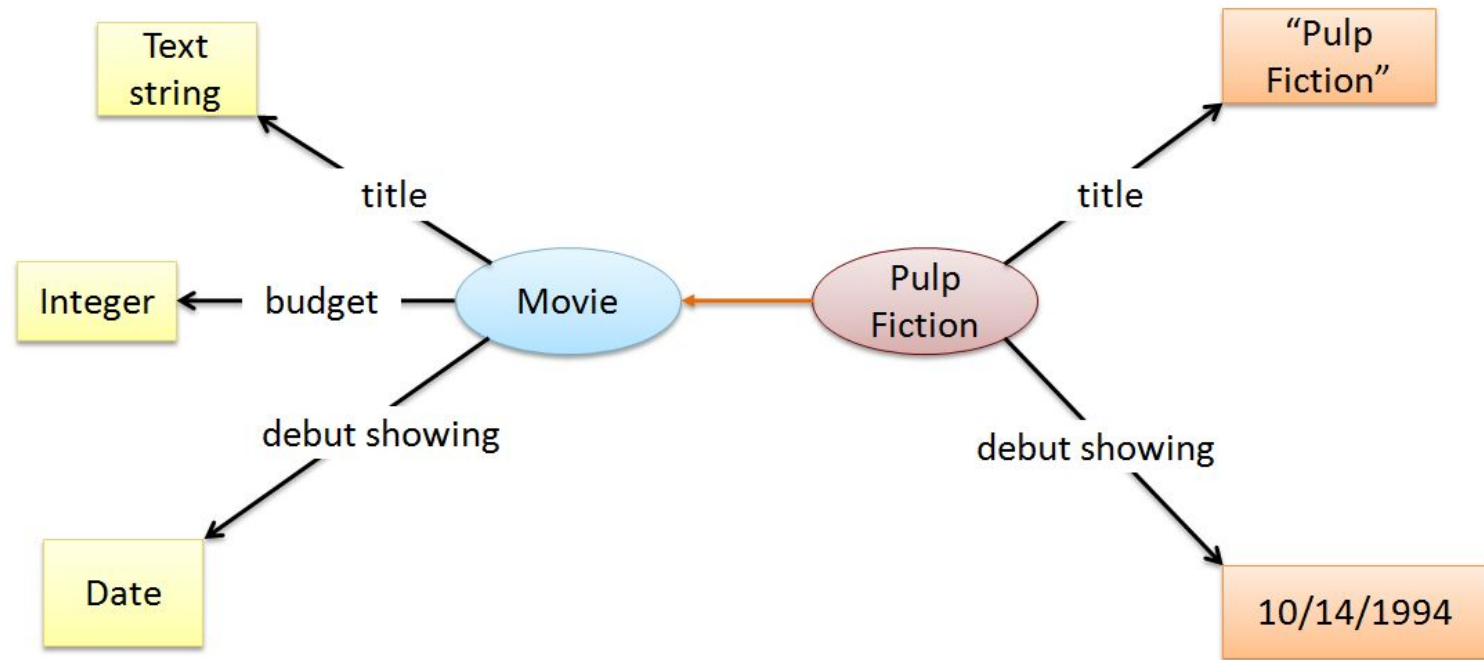
Example - Elena

- Goal: search for related information
- Links are generated from
 - Taxonomy which is based on keywords of the documents
- User context
 - Selected concepts from the taxonomy

Example: A Film Ontology







Ontology - Overview

- Models concepts within a domain as classes and relationships
 - Types of ontologies:
 - Domain ontologies
 - Upper ontologies
 - Ontology creation
 - Manually
 - Automatically (machine learning, merging)
 - The user of an ontology has to know how to interpret it
- Reuse existing ontologies

Populate an E-Learning Ontology

- Sources of metadata for e-learning resources
 - Already existing structure
 - Annotations by the author
 - Automation (e.g. keyword extraction, transcript generation, etc.)
- Sources of metadata for the user context
 - User explicitly provides data (selects topics, difficulty, etc.)
 - Observe user behaviour (time spent on topic, exercises, etc.)

Resource Description Framework (RDF)

- Machine-readable XML-based format for describing resources as triples (subject, predicate, object)
- Predefined concepts and relations for
 - Constructing ontologies (e.g. a cat is owned by a person)
 - Instantiating ontologies (e.g. Mephistopheles is a cat)

RDF Example - Constructing the Ontology

There are cats and persons. A cat might be owned by a person.

```
<rdfs:Class rdf:ID="Cat" />
<rdfs:Class rdf:ID="Person" />
<rdf:Property rdf:ID="ownedBy">
    <rdfs:domain rdf:resource="#Cat" />
    <rdfs:range rdf:resource="#Person" />
</rdf:Property>
```

RDF Example - Instantiating the Ontology

Mephistopheles is a cat. Harry is a person. Harry owns Mephistopheles.

```
<rdf:Description rdf:about="#Mephistopheles">
```

```
  <rdfs:type rdf:resource="#Cat" />
```

```
</rdf:Description>
```

```
<rdf:Description rdf:about="#Mephistopheles">
```

```
  <ownedBy rdf:resource="#Harry" />
```

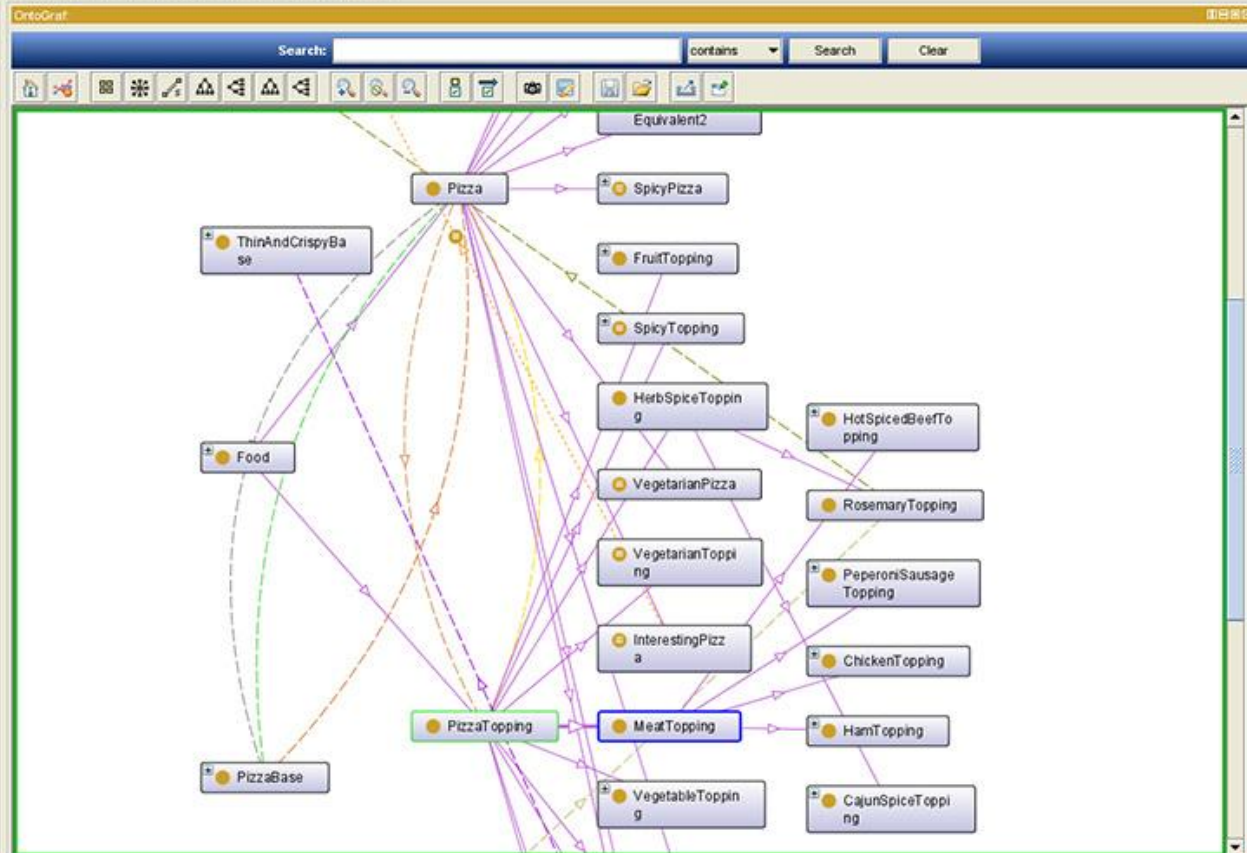
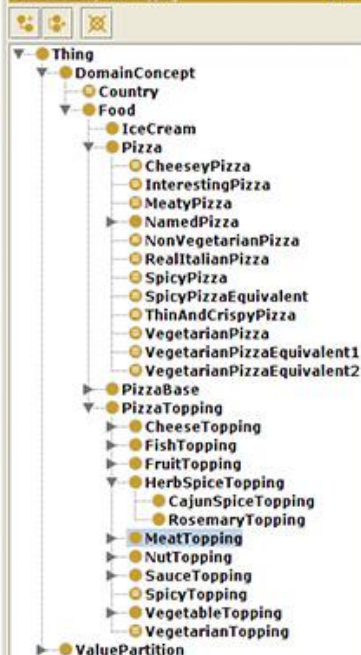
```
</rdf:Description>
```

Web Ontology Language (OWL)

- Machine-readable format for ontologies
- Overcomes some drawbacks of RDF
- Comes in different reasoning flavours

→ Reuse existing ontologies

(Dublin Core, Cyc, Schema.org, FOAF, ...)



Querying and Reasoning in Ontologies

- Model the domain of interest
- Store in a specialized database (e.g. triple store)
- Define goal
- Use reasoning engine to find out about intermediary goals

SPARQL Query Example

```
PREFIX ex: <http://example.com/exampleOntology#>

SELECT ?capital ?country

WHERE {

    ?x    ex:cityname          ?capital      ;
          ex:isCapitalOf      ?y            .
    ?y    ex:countryname       ?country      ;
          ex:isOnContinent    ex:Africa     .

}
```

Heterogeneous Sources

- Approaches
 - Convert ontologies to upper ontology
 - Merge ontologies
- Methods
 - Define a transition
 - Clustering and classification

Slightly Off-Topic Topics

- Recommender systems
- Motivational considerations: statement of accomplishment / gamification
- Speech to text / text to speech
- Automated summation tools

Summary

- Goal: individualized e-learning environment
 - Interlink information according to user's needs
- Ontologies as modelling tool for
 - E-learning resources
 - User context

Technology Summary

- Modelling languages
 - RDF
 - OWL
- Modelling software
 - Protégé
- Storage
 - TDB (triple store)
- Reasoners
 - Cyc inference engine
- Query languages
 - SPARQL
- Common ontologies
 - Dublin Core
 - Cyc
 - Schema.org
 - FOAF
- Software suites
 - Apache Jena

References

1. Stefan Dietze, Hong Qing Yu, Daniela Giordano, Eleni Kaldoudi, Nikolas Dovrolis, and Davide Taibi. Linked education: interlinking educational resources and the web of data. In The 27th ACM Symposium On Applied Computing SAC-2012), Special Track on Semantic Web and Applications, March 2012.
2. Peter Dolog and Wolfgang Nejdl. Semantic Web Technologies for the Adaptive Web, pages 697–719. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
3. Nicola Henze. Personal readers: Personalized learning object readers for the semantic web. In Proceedings of the 2005 Conference on Artificial Intelligence in Education: Supporting Learning Through Intelligent and Socially Informed Technology, pages 274–281, Amsterdam, The Netherlands, The Netherlands, 2005. IOS Press.
4. Ora Lassila and Ralph R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. W3c recommendation, W3C, February 1999.
5. Boris Motik, Bijan Parsia, and Peter F. Patel-Schneider. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. W3C recommendation, W3C, October 2009.
6. protege.stanford.edu. Protégé, 2012.
7. Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF. W3C Recommendation, January 2008.
<http://www.w3.org/TR/rdf-sparql-query/>
8. Sergey Sosnovsky and Darina Dicheva. Ontological technologies for user modelling. International Journal of Metadata, Semantics and Ontologies, 5(1):32–71, 2010.
9. <https://en.wikipedia.org/wiki/SPARQL>
10. https://en.wikipedia.org/wiki/Web_Ontology_Language
11. <https://www.cambridgesemantics.com/>