AIFB

3.4 Anwendungen

3.4.1 Ontobroker

3.4.2 SEAL - Semantic Portal

AIFB

3.4.1 Ontobroker

- Online-Suche im WWW heute:
  - Browsing
  - Suche nach Stichworten (Suchmaschinen wie Alta Vista, Google, Yahoo etc.)
- große Nachteile:
  - große Trefferzahl durch Falsch-Positive
  - mangelnde Präzision
  - Information kann verstreut vorliegen
  - Informationen teilweise nicht explizit im WWW dargestellt

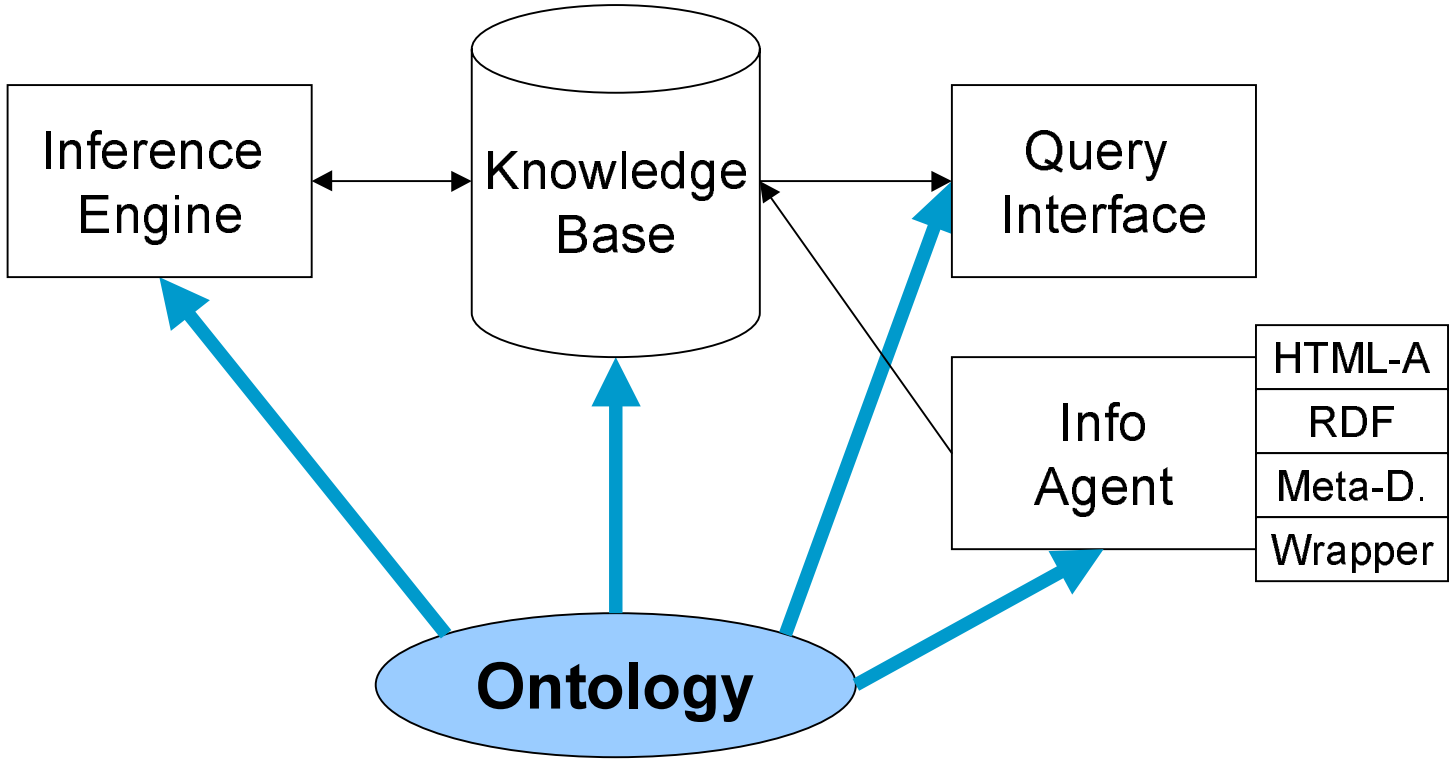
➡ Ontobroker löst diese Probleme

AIFB

Idee von Ontobroker

- nur ein **Teil des WWW** wird berücksichtigt
  - thematisch eingegrenzt
  - Metapher einer Newsgroup (z.B. Teil des Intranet)
- **Ontologie**
  - gemeinsame Sprache für Informationsanbieter und Informationssuchende
  - definiert relevante
    - Begriffe
    - Relationen
    - Regeln
- Informationen werden in einer **zentralen Wissensbasis** gesammelt
- Suche = Anfrage an Wissensbasis

Architektur von Ontobroker



AIFB

InfoAgent

- importiert Informationen in die Wissensbasis
  - verschiedenartige Quellen
  - stark strukturiert
    - Wrapper
  - RDF- und andere Meta-Daten
  - HTML-Seiten
    - „normale“ Inhalte
    - plus semantische/ontologische Metadaten
    - Onto-O-Mat als Annotierungswerkzeug
- **Ontologie ermöglicht den Import**

AIFB

Inference Engine

- leitet neues Wissen ab
- ausgehend von gegeben Fakten (aus dem WWW)
- verwendet Ableitungsregeln der Ontologie
- z.B.
  - jeder Manager auch bei Anfragen nach Personen gefunden
  - Symmetrie der Kooperations-Beziehung berücksichtigt

Query Interface

- im WWW:
  - http://ontobroker.aifb.uni-karlsruhe.de
- Anfrage ist logischer Ausdruck (FLogic)
- graphisches und tabellarisches Interface (verstecken Logik)
- **Ontologie dient der Unterstützung der Suchanfrage**
- Ontologie definiert Begriffe, nach denen gefragt werden kann

Beispiel (OntoIce)

Konzepte	Beziehungen	Regeln

Object [ ].
  Person :: Object.
  Employee :: Person.
  Manager :: Employee.
  Consultant :: Employee.
 	
Project :: Object.
 	
Company :: Object.
  Manufacturer :: Company.
  FinanceComp :: Company.
  Insurer :: FinanceComp.
  LifeInsurer :: Insurer.
  Bank :: FinanceCompany.
 	
Location :: Object.
	Person [
	  firstName =>> String;
	  lastName =>> String;
	  email =>> String;
	  phone =>> String;
	  participantOf =>> Project;
	  hasCompExperience =>> Company;
	  address =>> Location]
	
	Project [
	  projectname =>> String;
	  projectgoal =>> String;
	  client =>> Company;
	  member =>> Person;
	  leader =>> Person].
		FORALL Proj1, Pers1
		  Proj1 : Project
		  [member ->> Pers1]
		↔
		  Pers1 : Person
		  [participantOf ->> Proj1].
		
		FORALL Pers1, Proj1, Comp1
		  Proj1 : Project
		  [member ->> Pers1,
		  client ->> Comp1]
		→
		  Pers1 : Person
		  [hasCompExperience ->> Comp1].

- Welche Consultants kennt das System?
- Gib mir zu allen Consultants die Email-Adresse, den Vor- und den Nachnamen!
- Welche Telefonnummer hat "Steffen Staab"?
- Wer arbeitet mit "Steffen Staab" zusammen?

AIFB

Beispiel-Anfragen in F-Logic

- Welche Consultants kennt das System?

```
FORALL Variable1 <- Variable1:Consultant.
```

- Gib mir zu allen Consultants die Email-Adresse, den Vor- und den Nachnamen!

```
FORALL V1, V2, V3, V4 <-
    V1:Consultant[firstName->>V2] AND
    V1:Consultant[lastName->>V3] AND
    V1:Consultant[email->>V4].
```

AIFB

Beispiel-Anfragen in F-Logic

- Welche Telefonnummer hat "Steffen Staab"?

```
FORALL V1, V2 <-
    V1:Person[firstName->>"Steffen"] AND
    V1[lastName->>"Staab"] AND
    V1[phone->>V2].
```

- Wer arbeitet mit "Steffen Staab" zusammen?

```
FORALL V1 <- EXISTS V2,V3
    V2:Person[firstName->>"Steffen"] AND
    V2[lastName->>"Staab"] AND
    V2[participantOf->>V3:Project] AND
    V1:Person[participantOf->>V3].
```

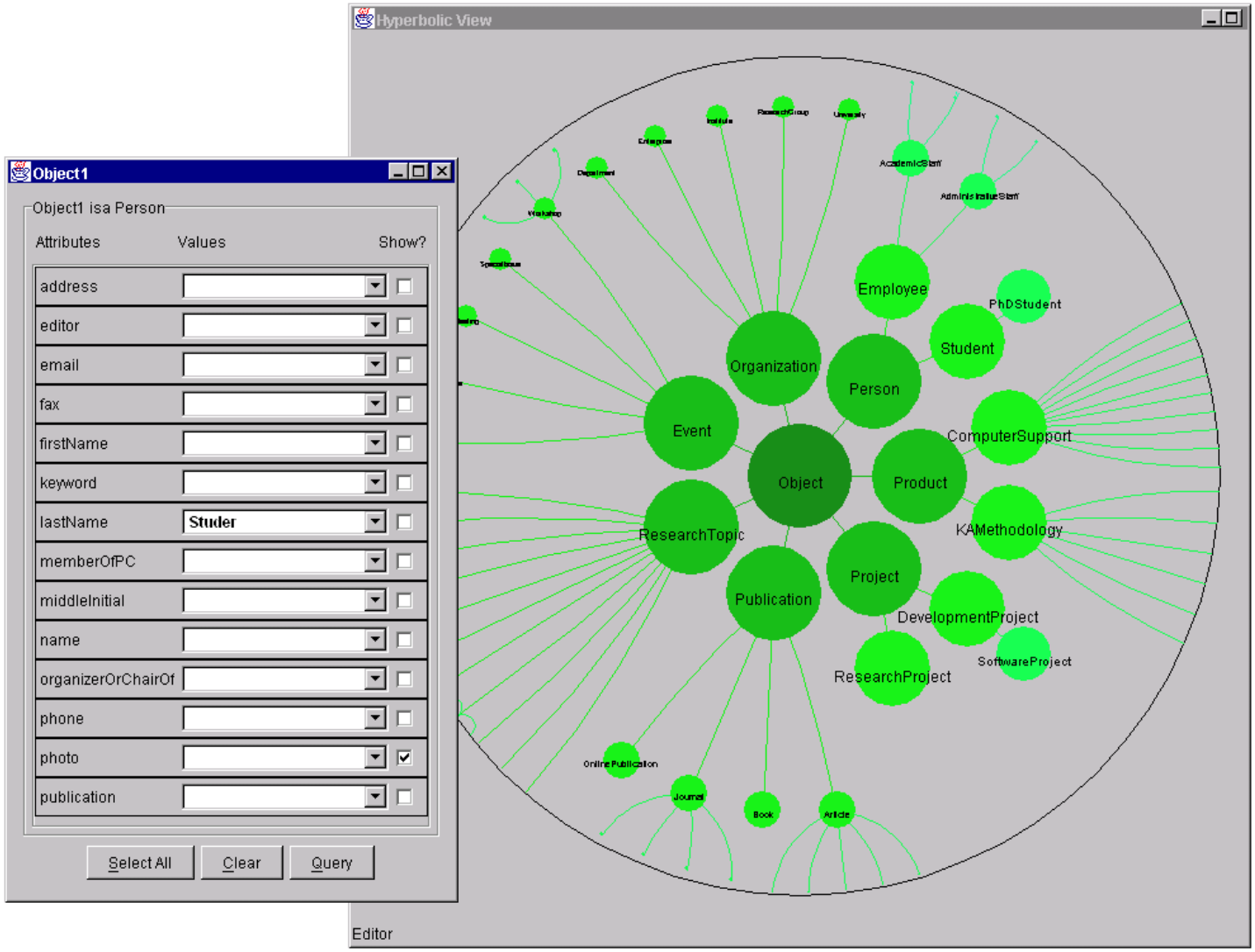
AIFB

Hyperbolic View Interface

A query may also be generated by using the hyperbolic view interface

- It visualises the ontology as a hierarchy of concepts
- The presentation is based on hyperbolic geometry:
  - nodes in the center are depicted with a large circle,
  - nodes at the border of the surrounding circle are only marked with a small circle.
- This visualisation technique allows:
  - a survey over all concepts,
  - a quick navigation to nodes far away from the center, as well as
  - a closer examination of nodes and their vicinity

Hyperbolic View Interface



When a user selects a node from the hyperbolic view, a form is presented which allows the user to select attributes or to insert values for the attributes

Ontobroker as a Search Tool

- Keyword Search
  - delivers **documents** that contain the requested strings
- Keyword Search and Thesaurus
  - delivers **documents** that contain terms that are defined within the thesaurus
- Database query
  - delivers **facts** that are explicitly stored in the database
- Ontobroker
  - delivers answers that subsume all the above cases, derived facts are delivered in addition

Comparison of Search Tools

	Query	Answer
Keyword Search	„OntoIce and Nordic Life“	Documents with Highlighting
Keyword Search with Thesaurus	„OntoIce:Project and Nordic Life:Insurance company“	Documents with Highlighting
Database query	Smith was a team member in which projects?	Smith was part of OntoIce...
Ontobroker	Which consultant has insurance industry experience?	Smith knows about Nordic Life

AIFB

3.4.2 Semantic Portals

- Web Portal
  - central source of information
  - structures information
- Community Web Portal
  - Web Portal
  - homogeneous group of users
  - similar interests/goals
  - community provides **and** uses information
- Semantic Community Web Portal - Semantic Portal
  - Community Web Portal
  - provides multiple views to information
  - information are accessible to human and software agents on a semantic basis
  - uses semantic means to structure information
    - Ontology a core component

Requirements for a Semantic Portal

- Put the “Semantic Web” into practice for communities of interest
  - present a structured **semantic view** onto the Web
  - **delivery** of data on a **semantic basis**
    - by establishing a higher quality of communication between the information provider (community) and the consumer (human and software agents)
  - **ontologies** provide a **sound semantic basis**
    - used to define the meaning of terms and hence to support intelligent **providing and access** to information on the Web.
  - reflect basic paradigm of the Web:
    - **self-organization**

3.4.2.1 Framework for developing Semantic Portals
Requirements

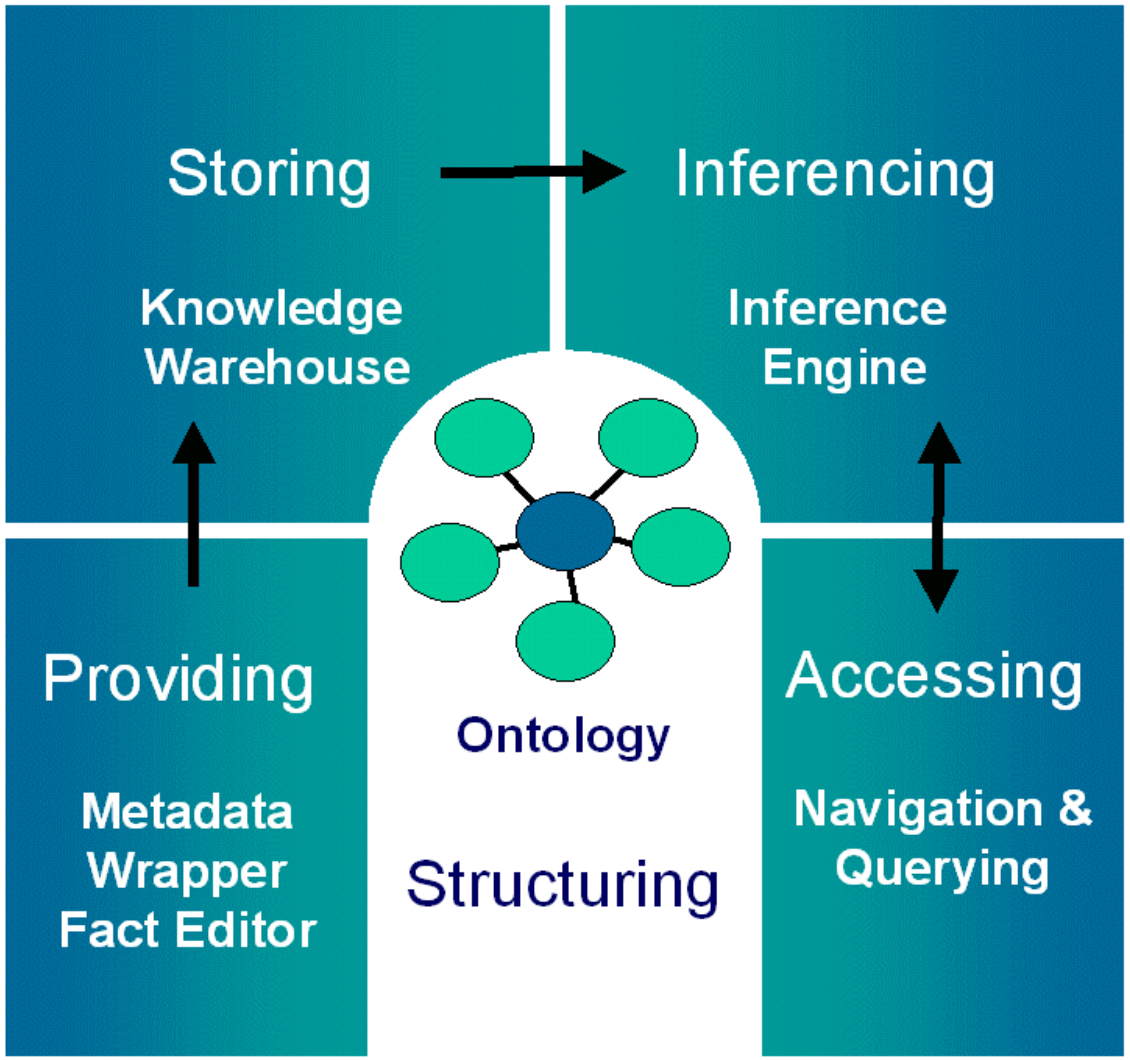
- Integrated approach has to cover two aspects:
  - **Development** and **maintenance** of the semantic backbone - **ontology**
    - methodology and associated tools have to be provided to manage huge and complex information sources
  - **Ontology supported** manipulation (providing, storing, accessing) of the information contained in a semantic portal

3.4.2.1 Framework for developing Semantic Portals
Requirements

- Ontology-supported manipulation covers several aspects:
  - **Information provisioning**
    - all community members must be able to provide information
  - **Portal access** by human agents
    - navigating
    - semantic querying
      - deliver integrated answers extended by derived facts
  - **Portal access** by software agents
    - “crawling” information represented in some metadata representation formalism, e.g. RDF

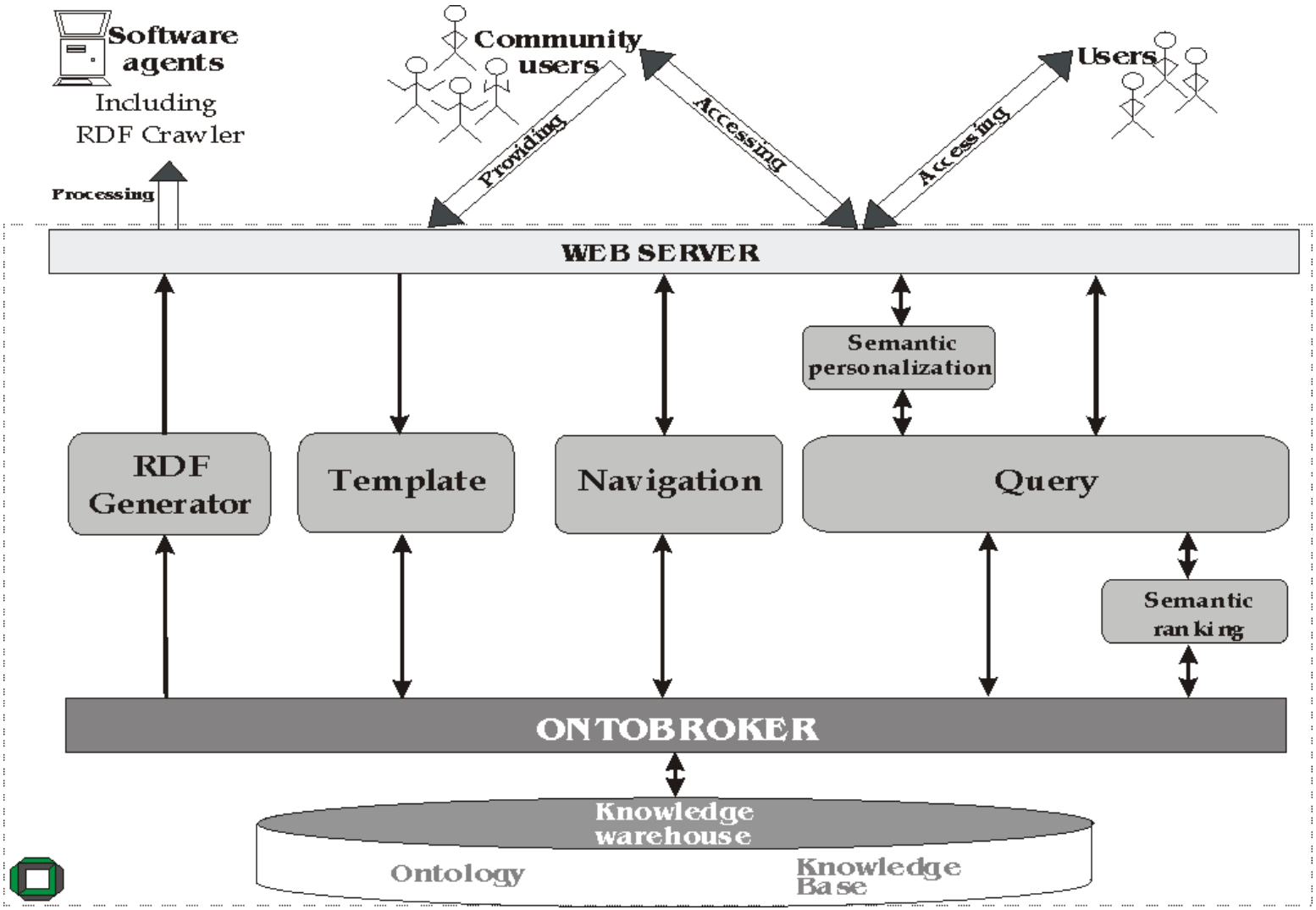
Framework for developing Semantic Portals - SEAL

Functional view



Framework for developing Semantic Portals - SEAL

Overall architecture and environment



Framework for developing Semantic Portals - SEAL
Core modules

- The **backbone** of the system consists of:
  - the knowledge warehouse, i.e. the ontology and knowledge base, and
  - the Ontobroker system, *i.e.* the principal inferencing mechanism
- Modules:
  - **Template** module for **providing** of information
  - **Navigation** module for **accessing** information by browsing
  - **Query** module for **accessing** information. It uses as pre- and post- processing, respectively:
    - Semantic personalization module and
    - Semantic ranking module
  - **RDF Generator** for **transforming** community information in **RDF metadata**

Ontobroker (summary)

- The Ontobroker system is a **deductive, object-oriented database** system operating either in main memory or on a relational database (via JDBC)
- It provides **compilers** for different languages to describe ontologies, rules and facts
  - it is also used as an **inference engine**:
    - reads input files containing the knowledge base and the ontology,
    - evaluates incoming queries, and
    - returns the results derived from the combination of ontology, knowledge base and query.
- The possibility to derive additional factual knowledge from given facts and background knowledge enables **providing of consistent and non redundant** information

AIFB

Knowledge warehouse

- Serves as **repository** for data represented in the form of F-Logic statements
- Hosts the **ontology**, as well as the **data** proper.
- It is organised around a relational database
  - facts and concepts are stored in a reified format,
  - relations and concepts are first-order objects
  - it is very flexible with regard to changes and amendments of the ontology.

3.4.2.2 Providing Information for the Portal

- Contribution of information has to be possible for all members of the community
  - support a decentralized, self-organizing approach
- Cope with the diversity of information sources
  - **metadata** based information, that enrich documents with semantic information
    - use predefined **templates**, which generate HTML pages with embedded metadata information (annotated HTML pages)
    - use **tools** for metadata annotation of existing HTML pages (Onto-O-Mat)
  - **wrapper** based information
  - **direct provisioning** of semantic facts

Metadata in HTML - HTML-A

Goal: explicitly describe contents of HTML documents on a semantic basis

- HTML-A: Enhance HTML with few semantically relevant extensions
  - *tagging instances of concepts*
  - *relating these instances*
  - *setting their properties*

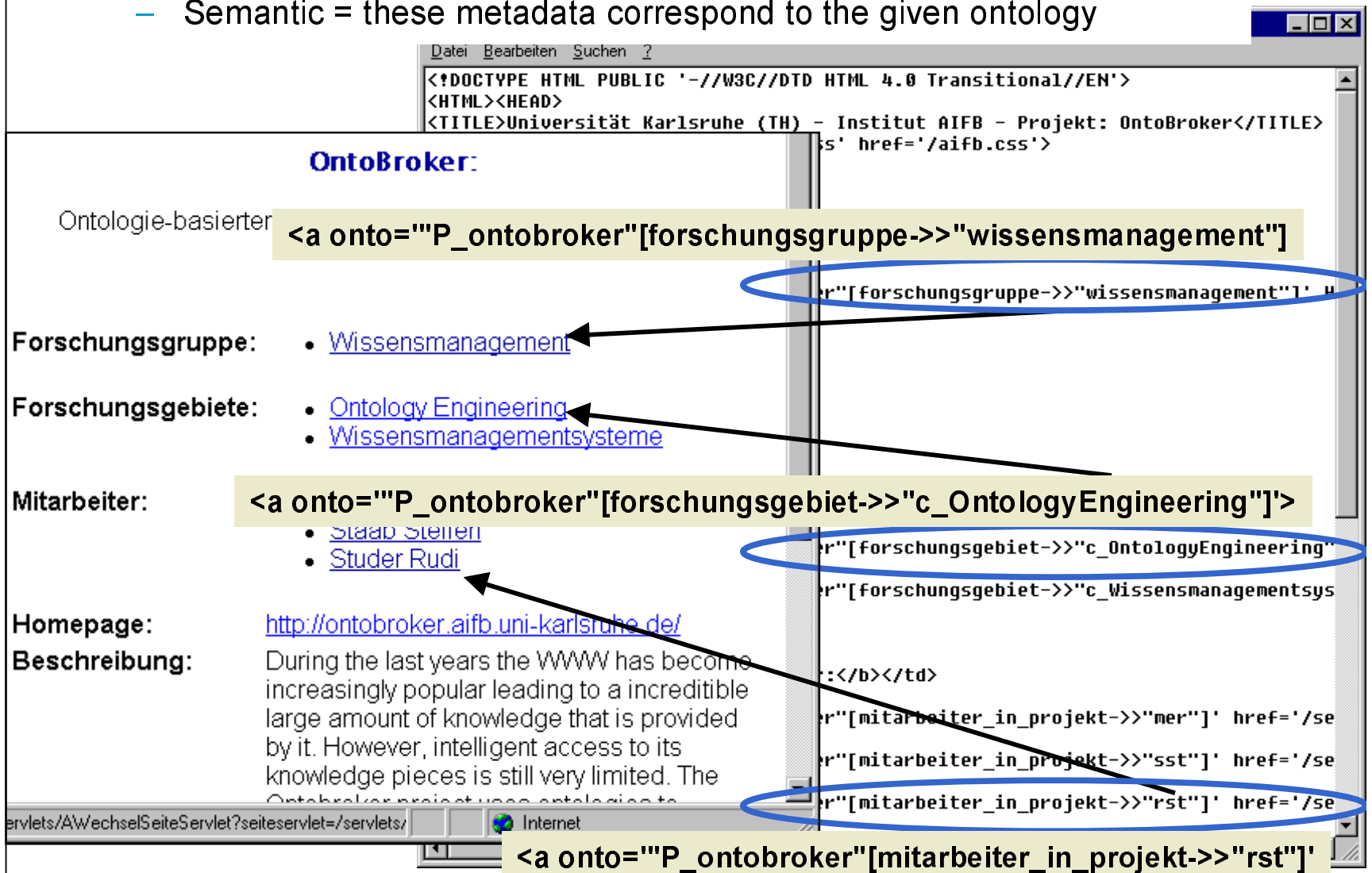
- Example:

```
<A onto="'http://www.aifb.uni-karlsruhe.de/studer':Researcher"></A>
<A onto="'www9':Proceedings"></A>
<A onto="page:Institute"></A>
```

- <A onto = “O:C“></A> represents O as an instance of C.

Metadata in HTML - Semantic Annotation

- Annotation: Embedding metadata information into HTML pages
  - Semantic = these metadata correspond to the given ontology



AIFB

Metadata as Input for the Portal

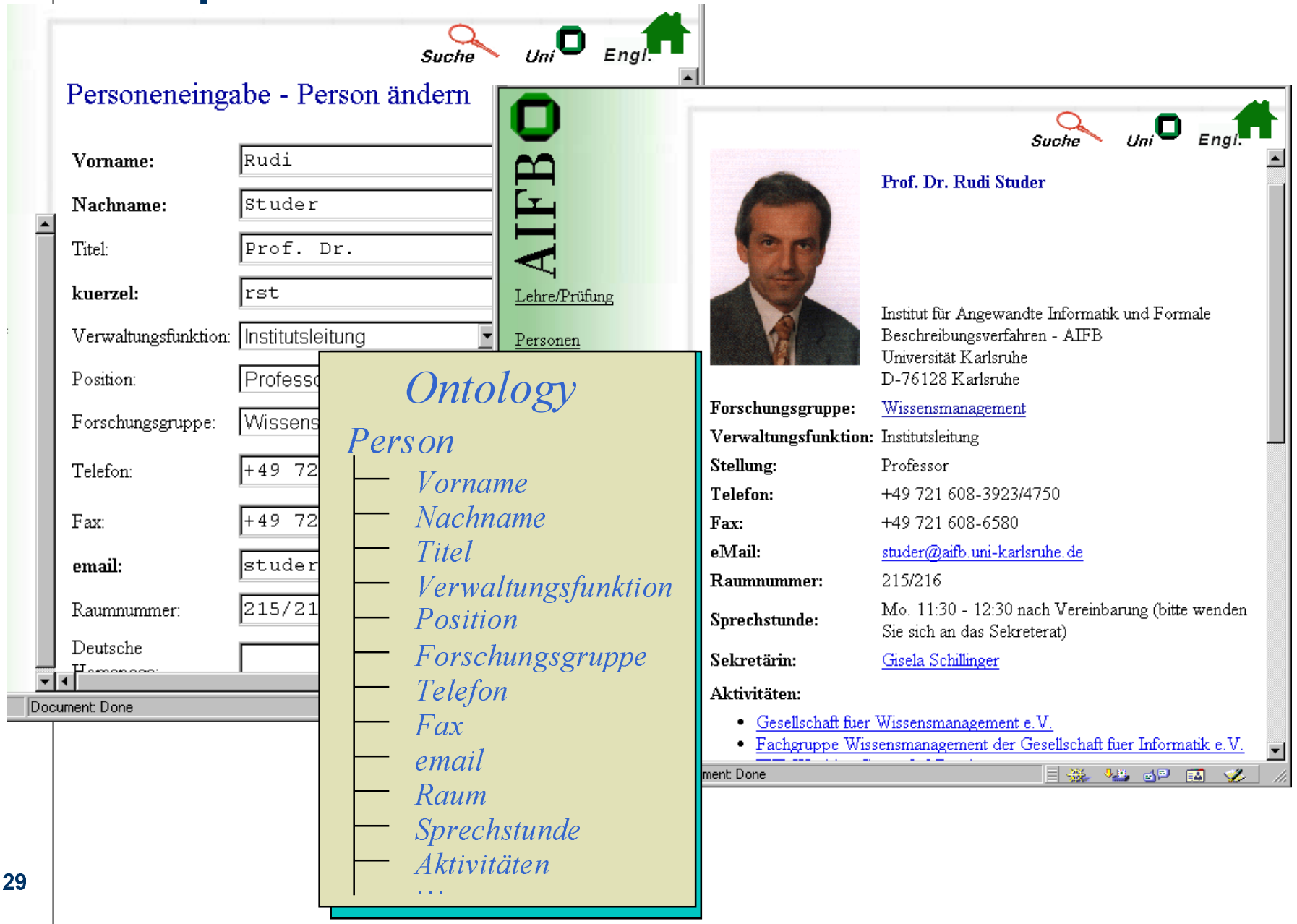
- General objective:
  - reduce overhead as far as possible
- exploit **templates** that are annotated with ontological concepts
  - ontological facts come for free
- copy-and-paste annotation tool: Onto-O-Mat
  - extend **HTML**-pages with **semantic annotations**
- generate XML DTDs out of ontologies
  - **XML tags** have semantic underpinning
- **RDF** delivers a standardization for metadata specifications
  - support reuse and exchange of metadata specifications
- exploit **text mining** to extract ontological facts from given text sources

AIFB

Template module

- Facilitate the contribution of information by community users, by generating an input HTML form for each concept that a user may instantiate, e.g. Person, Project
  - using inverse axiom from ontology avoids providing of redundant information
- Structure information provided by users according to definition of concepts in the ontology and store information in knowledge warehouse
- Generate HTML page for each instantiated concept
  - templates for each HTML page are based on ontological relations between concepts (used for hyperlinking)
  - metadata are embeded in HTML

Template module



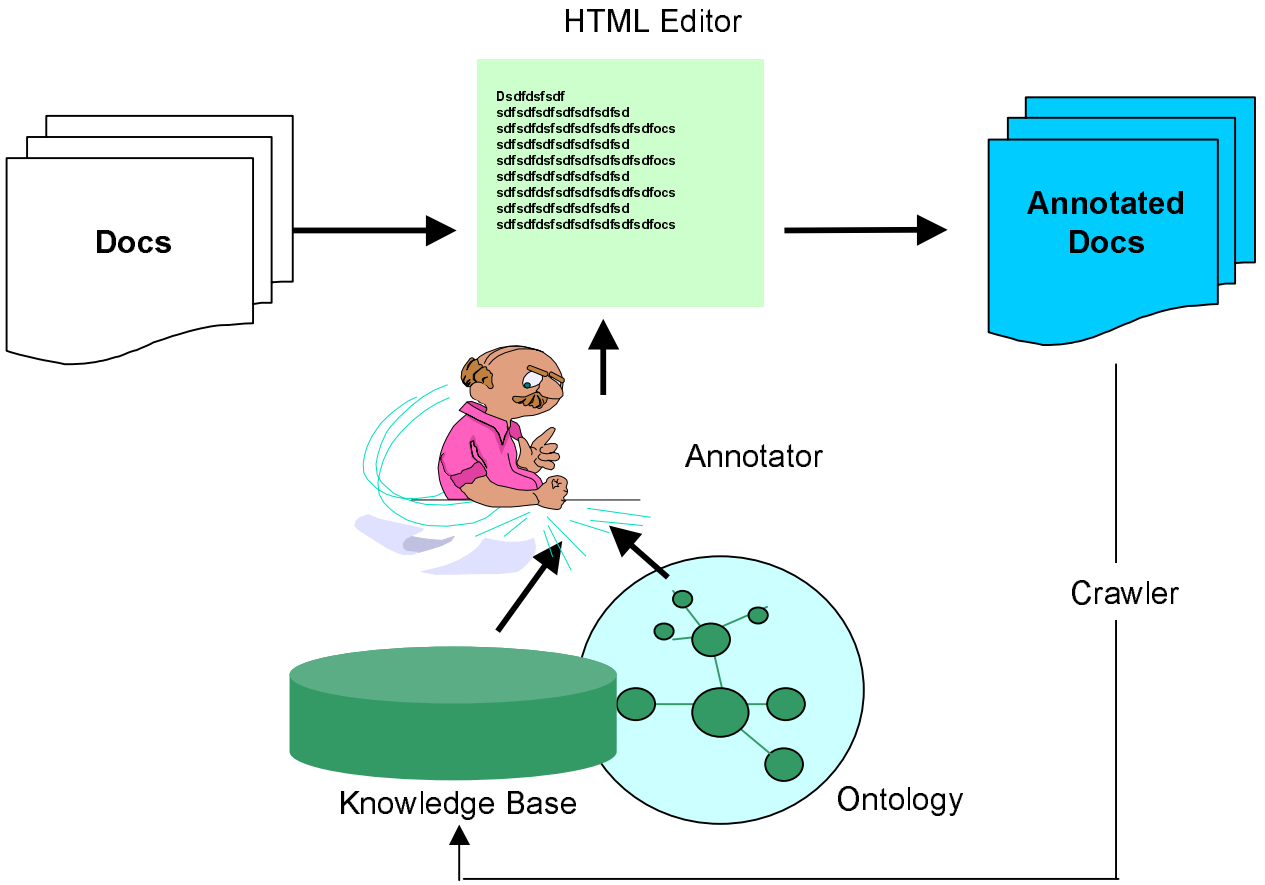
Template module

HTML-Dokument ← *Ontology*

<html> …

Name:
>"Rudi"]>Rudi
>"Studer"]>Studer
…
Position:
>body]'>Professor
Forschungsgruppe:
>
"wissensmanagement"]
…
Telefon:
>body]>+49 721 608-3923/4750
E-Mail:
>body]>studer@aifb.uni-karlsruhe.de
… </html>

Metadata annotation of existing sources
Manual Semantic Annotation



AIFB

Manual Semantic Annotation

Problems:

- syntax errors are too often (e.g. missing brackets or quotation marks)

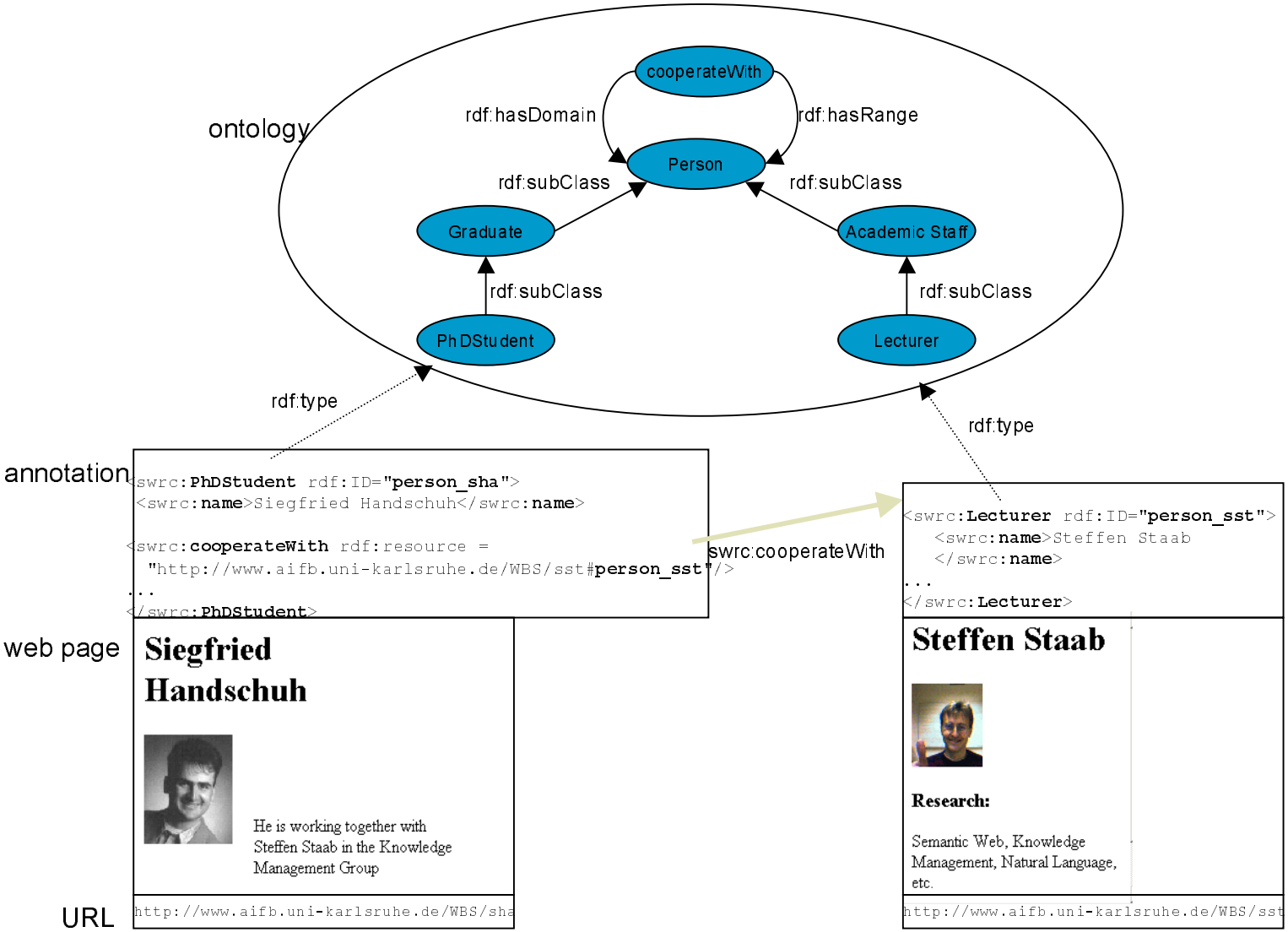
- the annotations have to strictly conform to the concept and relation names defined in the ontology (typing errors, e.g. online-Version instead of onlineVersion)

- semantics of annotations
  e.g. a paper has been categorised as Publication instead of JournalArticle,TechnicalReport

=> Support annotators with interactive, graphical tools!

Metadata annotation tools - Onto-O-Mat

Onto-O-Mat - Annotation example



AIFB

Onto-O-Mat

Characteristics:

Consistency: Semantic structures should adhere to a given ontology in order to allow for better sharing of knowledge

Proper Reference: Identiers of instances, e.g. of persons, institutes or companies, should be unique

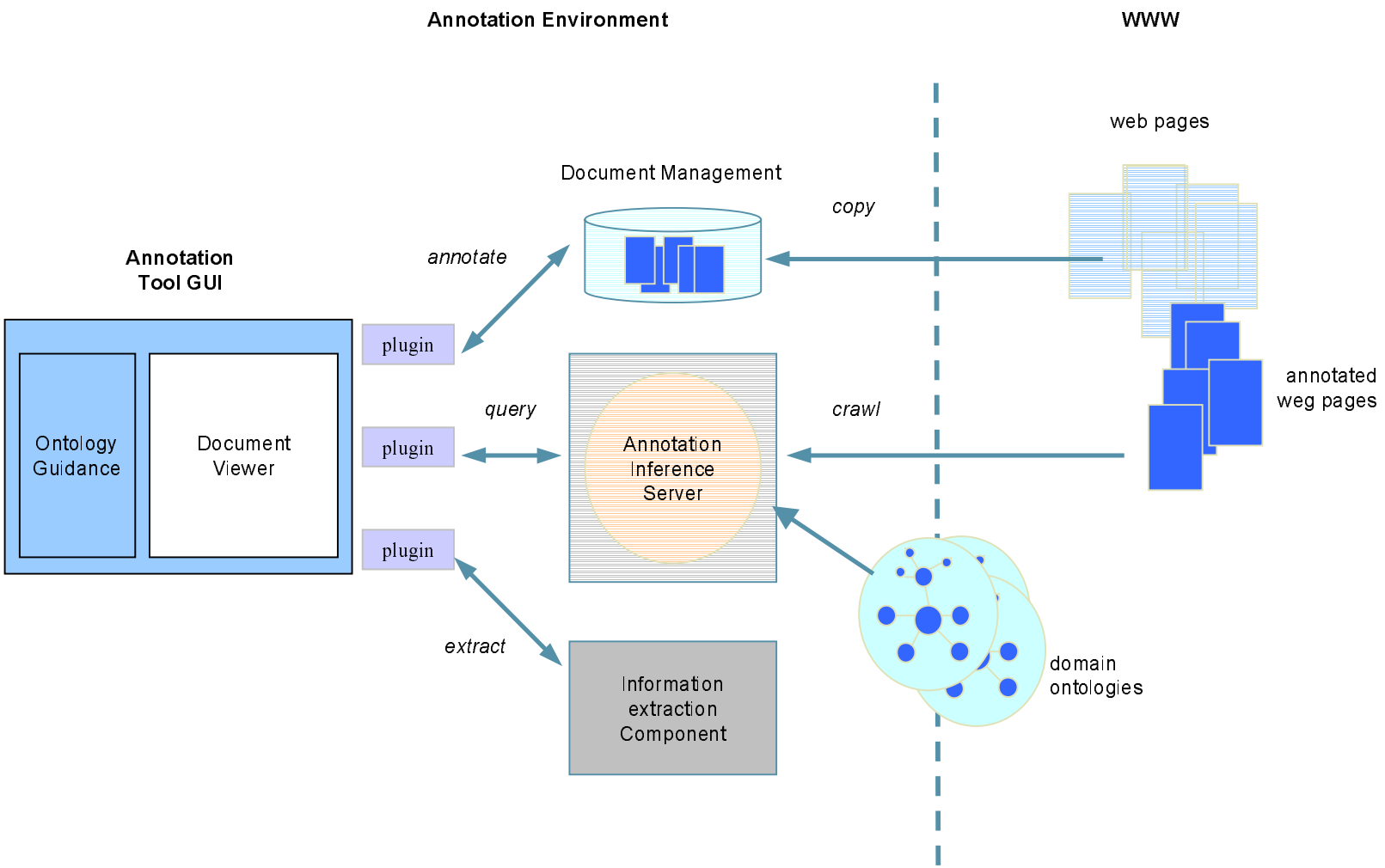
Avoid Redundancy: Decentralized knowledge provisioning should be possible, but avoiding laborious redundant annotations.

Maintenance: Knowledge markup needs to be maintained.

Ease of use: It is obvious for an annotation environments to be useful. However, it is not trivial, because it involves intricate navigation of semantic structures.

Efficiency: The more efficiently a tool supports the annotation, the more metadata a user will produce.

Onto-O-Mat - Architecture



Onto-O-Mat - Linking Challenges with Required Modules

Requirement / General Problem	Document Viewer	Ontology Guidance	Storage: Replication: Crawler	Storage: Replication: Annotation Inference Server	Storage: Document Management	Information Extraction
Consistency		X		X		
Proper Reference			X	X		
Avoid Redundancy			X	X	X	
Relational Metadata		X	X	X		
Maintenance					X	X
Ease of use	X	X				X
Efficiency	X	X	X	X	X	X

Wrapper based information

Goal: align regularities found in documents or data structures with the corresponding semantic background knowledge

- Annotation may be **automated** when one finds **regularities** in a larger number of documents
- The principle idea behind wrapper-based information:

  there are large information collections that have a similar structure
- We here distinguish between:
  - semi-structured information sources (e.g. HTML) and
  - structured information sources (e.g. relational databases)

Expert mode - Fact editor

Goal: to enable direct provisioning of semantic facts

- F-Logic statements are typed in by the user
  - the most technical (but also most powerful and flexible) way for providing information in the portal
- Only appropriate for users who are very familiar with
  - F-Logic and
  - the ontology.
- Hyperbolic interface tool may be used as a Fact Editor
  - In this mode its forms are used to insert values for attributes of instances of corresponding concepts from the ontology

AIFB

3.4.1.3 Accessing the Portal

- Access has to be conceptually founded
  - exploit semantic structure of ontology
  - e.g. persons, projects, publications
- Provide multitude of **views**
  - adapt to specific application needs
  - hide complex queries behind hypertext links
- Three forms of accessing are supported:
  - **navigating** through the portal by exploiting semantic hyperlink structure of documents
  - **searching** for information by posting queries
  - **processing** community information (in RDF) by software agents
- Backbone is inference engine for Frame Logic
  - provide integrated results
  - derive new knowledge by exploiting the ontology

AIFB

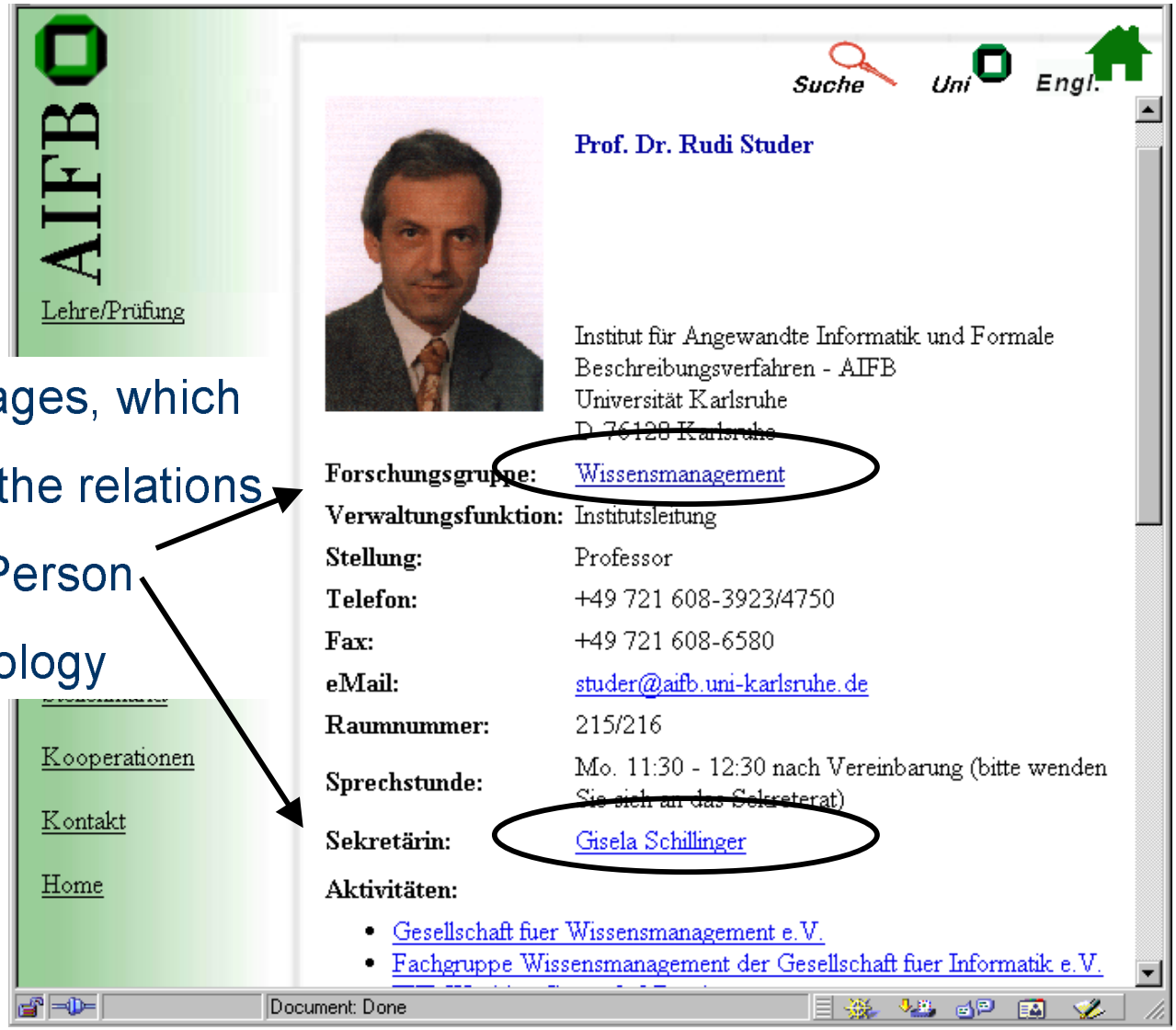
Navigation module

- The hyperlink structure is partially given by the portal builder, but it may be extended with the help of the **navigation** module
- The navigation module exploits inferencing capabilities of the inference engine in order to **construct semantic hyperlink structures**
  - it enables complex graph-based semantic hyperlinking, based on ontological relations between concepts in the domain
  - the conceptual approach to hyperlinking is based on the assumption that semantically relevant hyperlinks from a web page correspond to conceptual relations, such as memberOf or hasPart, or to attributes, like hasName.

=> Thus, hyperlinks in HTML pages are generated automatically to all (ontologically) related instances - HTML pages

Navigation module

Hyperlinks to pages, which corresponds to the relations of the concept Person in the given ontology



AIFB

Querying the Portal

- Compose a query by a click and select process
  - exploit concepts, attributes, instances
  - use check boxes, selection list, …
- Exploit **hyperbolic view interface**
  - visualize concept hierarchy
  - specialize query by concept attributes
- **Reuse** queries by storing them as semantic bookmarks
- Support **personalization** of portal
  - adapt to specific user needs
- Support **ranking** of retrieved information
  - exploit ontological background

AIFB

Querying the Portal - Techniques

- A choice of concepts, instances, or combinations of both may be issued to **search-forms in HTML**
- A **predefined hypertext link** that contains a query which is dynamically evaluated when one clicks on the link
- A query may also be generated by using the **hyperbolic view** interface
- The predefined queries can be stored as **semantic bookmarks**
- **Expert mode**: querying the Portal with native F-Logic statements - similar to Fact editor (see Providing)

Query module

- Puts an **easy-to-use** interface on the query capabilities of the F-Logic query interface of Ontobroker
- The portal builder models web pages that serve particular query needs, such as:
  - querying for projects or
  - querying for people
- Query module offers to the user selection lists, that
  - restrict query possibilities
  - are compiled using knowledge from the ontology and/or the knowledge base
- Query module is based on the given ontology of the domain

Query module

For instance, the query interface for persons allows to search for people according to:

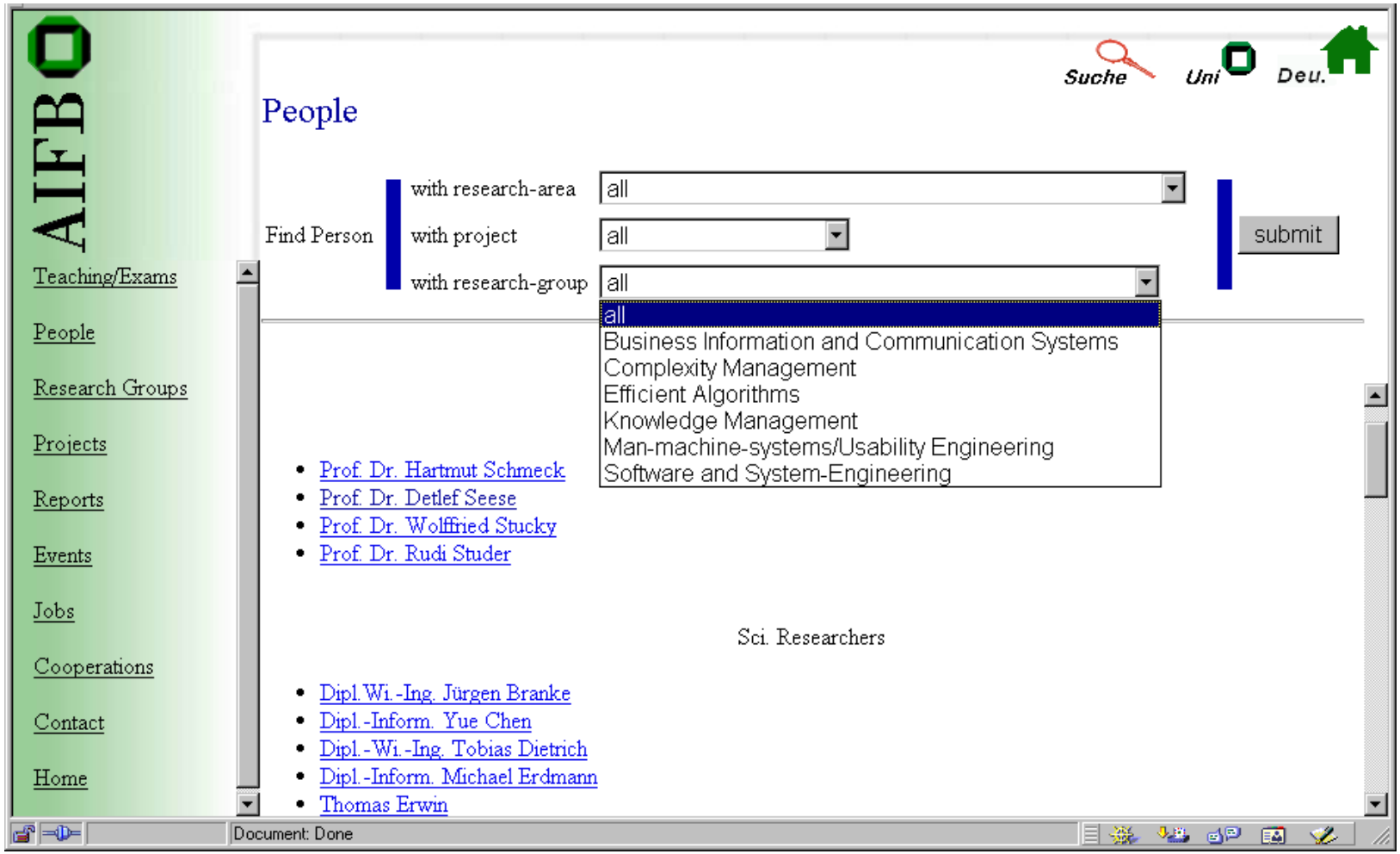
- research areas
- projects
- research groups they are members of (next slide).

These information are **ontology-based**:

Concept Person in the given ontology has relation with concepts:

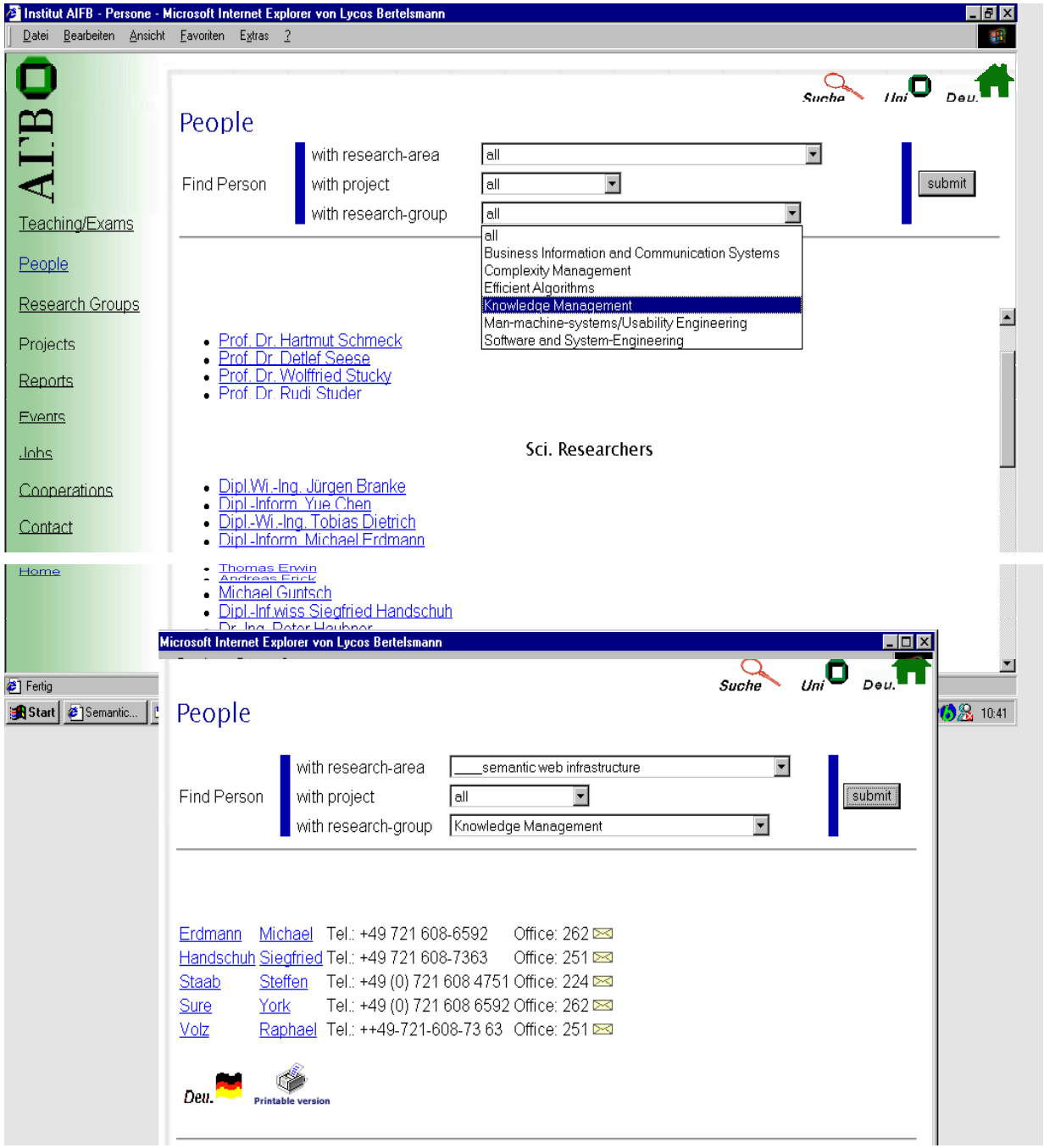
- Research-area
- Project
- Research group

Query module



Remark: The list of research groups is dynamically filled by an F-Logic query and presented to the user for easy choice by a drop-down list

Query module



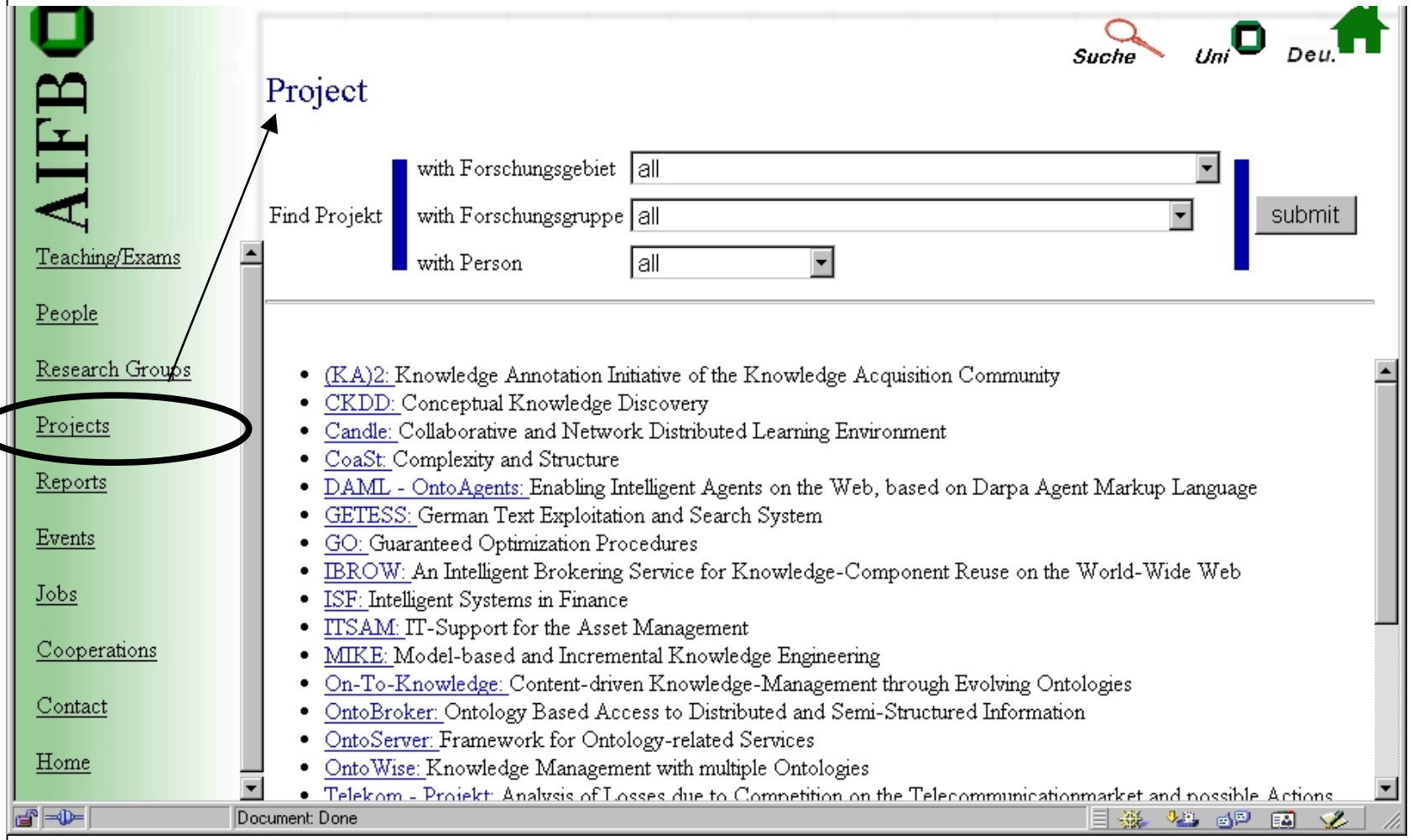
AIFB

Predefined queries

- A hypertext link may contain a F-Logic query which is dynamically evaluated when one clicks on the link

- Browsing is made possible through the definition of views onto top-level concepts of the AIFB ontology (Persons, Projects, Research groups)
  - Each of these topics can be browsed using predefined queries
  - For example, a click on the Projects hyperlink results in actions:
    - generation of a query to get name of all projects known at the portal,
    - evaluation of the query and
    - presentation of the results to the user in a form of a table

Predefined queries



Access by software agents -
RDF Generator module

- To enable the embedding of semantic portals into the Semantic Web, means for RDF-capable software agents to process the portal data are developed (RDF Generator and RDF Crawler)

- RDF GENERATOR
  - dynamically generates RDF statements on each of the static and dynamic pages of the semantic portal
  - RDF statements are generated according to the underlying ontology
  - for example,
    - for each person from the institute, contact information as well as professional information are available for processing from world-wide software agents, which understand this form of metadata representation

Access by software agents -
RDF Generator module

- RDF CRAWLER
  - used by software agents to crawl dedicated portion of the Web
  - enables collecting of RDF annotated information from dedicated sources
  - it is a tool which downloads interconnected fragments of RDF from the internet and builds a knowledge base from this data

AIFB

Issues

- logic formalism has to be hidden for most of the portal users
  - rather straightforward for concepts, attributes
- provide efficient inference engine
  - incremental delivery of answers
  - caching of derived facts
  - parallelization
- provide easy-to-use means for personalization
- user interface has to be tailored to application / user needs

AIFB

Running Example: AIFB Portal

- Semantic Portal of AIFB Institute
  - http://aifb.uni-karlsruhe.de
- Semantic backbone: AIFB Ontology
- Provide semantic access to
  - people,
  - projects and
  - research area information
- Provide access to students and researches services
  - teaching/exams
  - reports
  - events

AIFB Ontology

- Developed in a collaborative process of AIFB staff
  - according to presented OTK Methodology for Ontology Development
  - using ontology environment - OntoEdit

- simple centralized ontology is currently offered
  - around
    - 170 concepts
    - 75 relations
    - 10 axioms
  - still first version - good results
  - maintenance phase will come soon

Partial AIFB Ontology

(specified in Frame Logic)

Concepts	Relations	Axioms

Forschung[].
Forschungsgebiet :: Forschung.
Projekt :: Forschung.
Lehre[].
Diplomarbeit :: Lehre.
Lehrveranstaltung :: Lehre.
Seminar :: Lehrveranstaltung.
Organisation[].
Forschungsgruppe :: Organisation.
Institut :: Organisation.
Person[].
Mitarbeiter :: Person.
Wissenschaftliche_Hilfskraft :: Mitarbeiter.
Administrativer_Mitarbeiter :: Mitarbeiter.
Sekretaerin :: Administrativer_Mitarbeiter.
Techniker :: Administrativer_Mitarbeiter.
	Person[
	nachname=>>STRING ;
	vorname=>>STRING ;
	telefon=>>STRING ;
	fax=>>STRING ;
	email=>>STRING ;
	raumnummer=>>STRING ;
	titel=>>STRING].
	
	Wissenschaftlicher_Mitarbeiter[
	forschungsgruppe=>> Forschungsgruppe;
	sprechstunde=>>STRING;
	hat_publikation=>> Publikation;
	leitet_Lehre=>> Lehrveranstaltung;
	arbeitet_in_projekt=>> Projekt;
	forscht_in_gebiet=>> Forschungsgebiet].
		FORALL Per, Pro
		Per:Person
		[arbeitet_in_projekt->>Pro]
		<-
		Pro:Projekt
		[mitarbeiter_in_projekt->>Per].
		
		FORALL Per, Pro, RA
		Per:Person
		[forscht_in_gebiet ->>RA]
		<-
		Per:Person
		[arbeitet_in_projekt->>Pro]
		and
		Pro:Projekt
		[forschungsgebiet ->>RA].

AIFB

Conclusion

- providing information is still the crucial bottleneck
  - low level entry point is needed
  - **text mining** as a long term 80% solution
- ontology management becomes a major issue
  - evolution of ontologies
  - different versions of an ontology
- integration of different types of data is extremely important
- usability of tools is a must

AIFB

3.5 Zusammenfassung

- Ontologien
  - strukturien und erschließen Wissen
  - definieren "gemeinsame Sprache" für Personen(gruppen) und Anwendungssysteme
  - unterstützen
    - Suche, Navigation
    - reaktiv und proaktiv
  - stellen selbst Wissen dar
    - modellieren Commonsense, Anwendungsbereich
- Ausblick:
  - Integration verschiedenster Quellen im Web wird immer wichtiger (XML, RDF, ...)
  - Wissensmodellierung und Ontologien sind adäquate Technologien