

# Intelligent Systems on the World Wide Web

## Ontology Lifecycle



Steffen Staab  
 Institute AIFB, Universität Karlsruhe (TH)  
<http://www.aifb.uni-karlsruhe.de/WBS>

Acknowledgements to York Sure



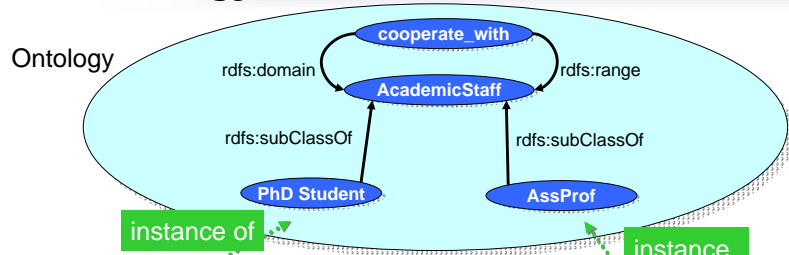
## Ontology

„People can't share knowledge if they do not speak a common language.“ [Davenport & Prusak, 1998]

„An ontology is an explicit specification of a conceptualization.“ [Gruber, 1993]

- Ontologies enable a **better communication** between Humans/Machines
- Ontologies **standardize** and **formalize** the meaning of words through concepts

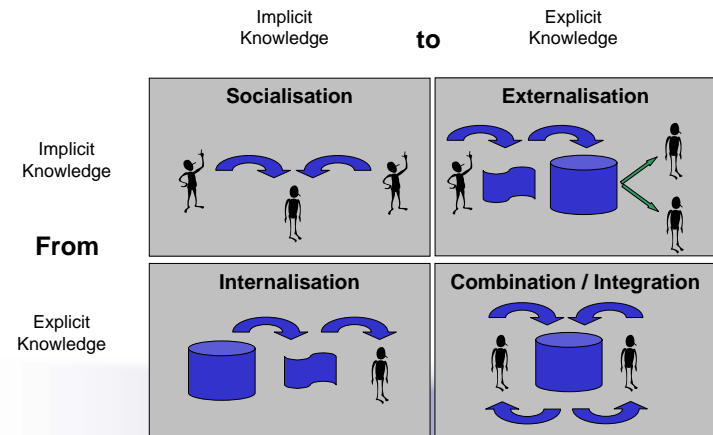
## Ontology & Metadata



Anno-tation	<pre>                 &lt;swrc:PhD_Student rdf:ID="sha"&gt;                 &lt;swrc:name&gt;Siegfried                 Handschuh&lt;/swrc:name&gt;                 &lt;swrc:cooperate_with rdf:resource =                 "http://www.aifb.uni-                 karlsruhe.de/WBS/sst#sst"/&gt;                 ...             </pre>	<pre>                 &lt;swrc:AssProf rdf:ID="sst"&gt;                 &lt;swrc:name&gt;Steffen Staab                 &lt;/swrc:name&gt;                 ...                 &lt;/swrc:AssProf&gt;             </pre>
	<p>Cooperate_with</p>	
Web Page	<p><b>Siegfried Handschuh</b></p> <p>He is working together with Steffen Staab in the Knowledge Management Group</p>	<p><b>Research:</b></p> <p>Semantic Web, Knowledge Management, Natural Language</p>
URL	<a href="http://www.aifb.uni-karlsruhe.de/WBS/sha">http://www.aifb.uni-karlsruhe.de/WBS/sha</a>	<a href="http://www.aifb.uni-karlsruhe.de/WBS/sst">http://www.aifb.uni-karlsruhe.de/WBS/sst</a>

Links have explicit meanings!

## Explicit vs. Implicit Knowledge





# Case study: OntoWeb.org

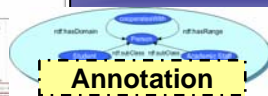
Steffen Staab, 2004



**Portal Generation**  
Navigation  
Query/Seerach  
Content



Collect metadata from participating partners



# Ontology-based Processes

Steffen Staab, 2004



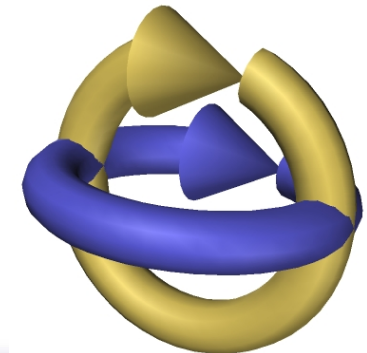
## Knowledge Meta Process

Design, Implementation, Evolution of Ontology

## Knowledge Process



Usage of Ontology



Slide 6



# OTK Methodology: Knowledge Meta Process

Steffen Staab, 2004



- Task: Build ontology based KM applications
- Problems:
  - Collaboration between domain experts and knowledge engineers
  - Evaluation of ontologies

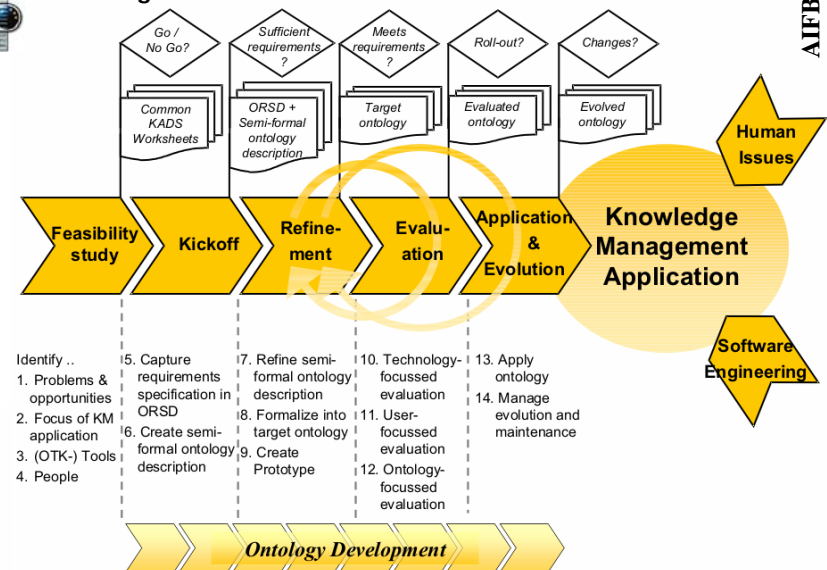
- Process-oriented, cyclic
- Pre-defined decisions and outcomes for each step
- Links to further existing methodologies for substeps

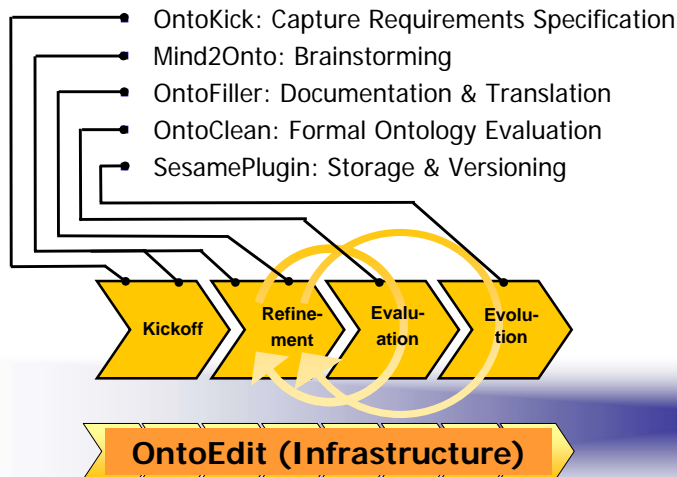
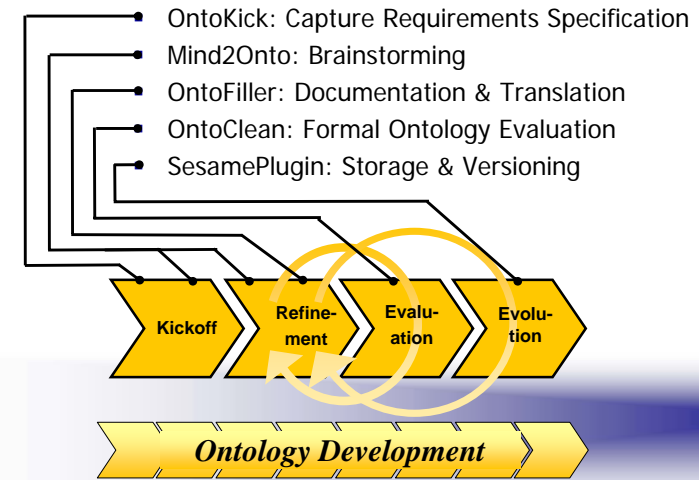
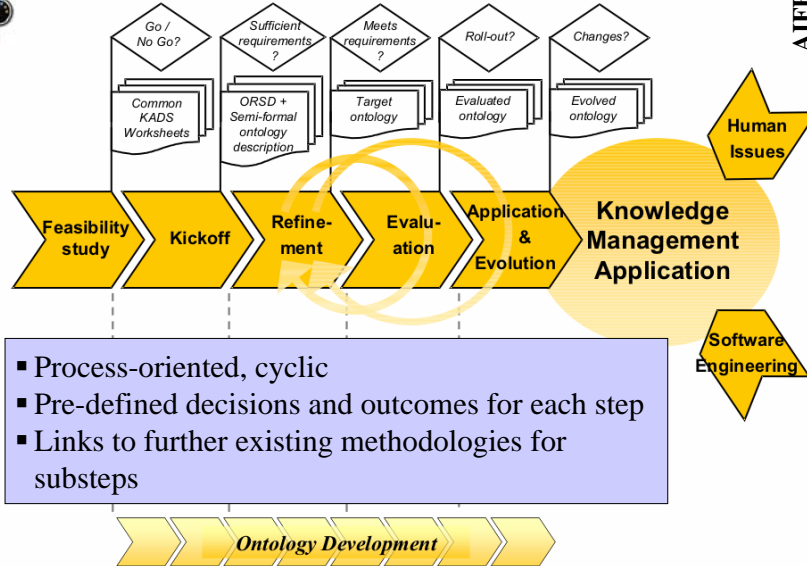
Slide 7



# OTK Methodology: Knowledge Meta Process

Steffen Staab, 2004





## Feasibility Study

- KM systems only function satisfactorily if they are properly integrated into the organization
- Many factors other than technology determine the success of such a system
- (Based on CommonKADS)

- Focus domain for ontology
- Identify people involved
- GO / No GO decision

Feasibility Feasibility study Steffen Staab, 2004

AIFB

## Current State: Skills Management

- Employee data distributed over many systems
- Different schemata for data
- Incomplete data

Rentenanstalt  
Swiss Life

Slide 13

Feasibility Feasibility study Steffen Staab, 2004

AIFB

## Intended state: Skills Management

Rentenanstalt  
Swiss Life

Slide 14

ON knowledge TO Steffen Staab, 2004

AIFB

## OTK Methodology: Knowledge Meta Process

Identify ..

- Problems & opportunities
- Focus of KM application
- (OTK-) Tools
- People

Ontology Development

Kick-off Ontology Kickoff Steffen Staab, 2004

AIFB

- Ontology Requirements Specification Document (ORSD)
  - Domain & Goal
  - Design guidelines
  - Available knowledge sources
  - Potential users and user scenarios
  - Applications supported by the ontology

E.g. Competency questions

Ontology Learning!

- Analyze knowledge sources
- Develop **baseline ontology description**

Draft version, typically most important concepts and relations are identified and described as an untyped graph

Slide 16

# ORS – Ontology Requirements Specification

Steffen Staab, 2004



- **Goal of the ontology:**
  - Tracking and analyzing corporate business histories
- **Domain and Scope:**
  - Merger & acquisition, restructurings, management changes and other strategic activities in the chemical industry
- **Supported Applications:**
  - Web-based Corporate History Analyzer
- **Knowledge Sources:**
  - Research analysts (domain experts)
  - Document: c:/mydocuments/superdokument.doc
  - URL: http://www.webpage.com
- **Users and Use Cases:**
  - Users: Research analysts, strategic consultants
  - Use Case 1: Track strategies of specific companies
  - Use Case 2: Analyze strategic moves of competitors
- **Competency Questions:**
  - Attached Competency Questionnaire
- **Potentially reusable ontologies:**
  - not known

Slide 17

# CQ – Competency Questionnaire

Steffen Staab, 2004



CQ Nr.	Competency Question	Concepts	Relation
CQ1	What are the subsidiaries, divisions and locations of company X?	company, subsidiary, division, location	company <i>has</i> subsidiary company <i>has</i> division company <i>has</i> location
CQ2	Which companies acquired company X?	company, acquisition	company <i>makes</i> acquisition acquisition <i>has</i> buyer acquisition <i>has</i> seller
CQ3	Which companies merged in 1990 in the rubber industry?	company, merger, year, industry	company <i>makes</i> merger company <i>isPartOf</i> industry merger <i>happensIn</i> year
CQ4	Who is CEO of company X?	CEO, company,	company <i>has</i> CEO
CQ5	Which activities of company X	activity,	company <i>performs</i>

Slide 18



## Kick-Off

Steffen Staab, 2004



- Ontology workshop to train domain experts in ontology modelling for
  - .. IT
  - .. Private customer insurance
  - .. Human Resource Management
- First version of domain ontology by expert
  - Manual development of ontology
  - Brainstorming (Mind Maps)
  - Middle-out approach
- Result: approx 700 Concepts in about 4 weeks



Slide 19



## Requirement specification

Steffen Staab, 2004



Slide 20



# Requirement specification

Steffen Staab, 2004



New Ontology

Concepts & Relations | Instances | Relation axioms | Disjoint concepts | Requirement Specification | Identification | Metadata

### Domain & Goal

Domain description

Sports & Recreation

- Food / Food Processing
- Hotel & Restaurant Equipment
- Industrial Equipment, Services & Supplies
- Information Technology / Robotics / Telecommunications
- Materials
- Medical / Scientific Products & Equipment
- Mining, Oil & Gas
- Sports & Recreation

Kick-Off date: 08-01-2001

Completion deadline: 12-31-2001

< Back Forward >



# Requirement specification

Steffen Staab, 2004



New Ontology

Concepts & Relations | Instances | Relation axioms | Disjoint concepts | Requirement Specification | Identification | Metadata

### Design Guidelines

Design instructions

- Write all concepts with capital letters.
- Write all relations with small letters.
- If you are using more than one word for defining a concept or relation, use an underscore (e.g. "Power\_plant")
- If you have only one concept as a subconcept, rethink your modelling decision!

Estimated number of concepts: 500

Maximal depth of concept hierarchy: 4

< Back Forward >



# Knowledge Sources

Steffen Staab, 2004



OntoPrise Ontology Engineering Workbench OntoEdit V1.0.3

File Edit Windows Help

C:\Programme\VisualAge for Java\ide\project\_resources\OntoEdit5.14\getess.owl

Concepts & Relations | Instances | Relation axioms | Disjoint concepts | Identification | Requirement Specification | About...

### Knowledge Sources

1 Domain & Goal

Design Guidelines

3 Knowledge Sources

Users & Use Cases

Deployment

2 < Back Forward >

Source	Type	Status
CQ Clinic Review, 6-21-2001	Competency Questionnaire	NEW SOURCE
unive	Ontology	NEW SOURCE
word	Word Document	NEW SOURCE
xmlD:	XML Document	NEW SOURCE
rdfDa	RDF Document	NEW SOURCE
textDa	Text Document	NEW SOURCE

Slide 23



# Knowledge Sources

Steffen Staab, 2004



OntoPrise Ontology Engineering Workbench OntoEdit V1.0.3

File Edit Windows Help

C:\Programme\VisualAge for Java\ide\project\_resources\OntoEdit5.14\getess.owl

Concepts & Relations | Instances | Relation axioms | Disjoint concepts | Identification | Requirement Specification | About...

### Knowledge Sources

1 Domain & Goal

Design Guidelines

3 Knowledge Sources

Users & Use Cases

Deployment

2 < Back Forward >

Ontomat Options

- General
- OntoKick
- Ontology Server

General Preferences

Tool selection

OXML-Files: OntoEdit

HTML-Files: HTML-Tool

XML-Files: XML-Tool

RDF-Files: RDF-To-Onto

Word-Documents: Text Tool

Text Document: Text Tool

Slide 24

# Kick-off Knowledge Sources

Steffen Staab, 2004



Slide 25

# Kick-off Competency questions

Steffen Staab, 2004



#	Question
1	Gibt es ein Luxushotel in Rostock?
2	Welche touristischen Attraktionen gibt es in Schwerin?
3	Wo ist das beste Restaurant in Warnemünde?
4	Gibt es eine Surfschule auf Usedom?
5	Gibt es einen weißen Sandstrand auf Rügen?

Slide 26

# Kick-off Competency questions

Steffen Staab, 2004



Slide 27

# Kick-off Competency questions

Steffen Staab, 2004



Slide 28

# Kick-off Competency questions

Slide 29

# Kick-off Competency questions

Slide 30

# Kick-off Competency questions

#	Question
1	Gibt es ein Luxushotel in Rostock?
2	Welche touristischen Attraktionen gibt es in Schwerin?
3	Wo ist das beste Restaurant in Warnemünde?
4	Gibt es eine Surfschule auf Usedom?
5	Gibt es einen weißen Sandstrand auf Rügen?

Slide 31

# Kick-off Traceability

Concept hierarchy	Relations	Range
Luxushotel	Adresse	Adresse
	Anzahl_Betten	INTEGER
	Haustiere_erlaubt	BOOLEAN
	Klassifizierung	STRING
	Name	STRING
	Verpflegung	STRING
	behindertenfreundlich	BOOLEAN
	aktivtaet	Aktion
	dienstleistung	Dienstleistung
	ereignis	Ereignis
	freizeiteinrichtung	Freizeiteinrichtung
	laub	Urlaub
	stattung	Nichtprivate Ausstattung der Unterkunft
	taetlichkeit	Raumliches Konzept
	sonabhaengigkeit	Qualitatives Zeitkonzept
	nat_Zimmer	Zimmer
	in_Gebiet	Gebiet
	liegt_bei	Sehenswuerdigkeit

Slide 32

## Kick-off Traceability

Steffen Staab, 2004



Slide 33

## Kick-off Brainstorming, Structuring, Formalisation

Steffen Staab, 2004



34

## Kickoff Mind2Onto

Steffen Staab, 2004



- **Task:** Collaborative capturing of domain knowledge through domain experts and modelling experts
- **Problem:** Collaboration with domain experts who have:
  - **No experience** with modelling
  - **No time** for modelling

Slide 35

## Kickoff Mind2Onto

Steffen Staab, 2004



Slide 36

**Kickoff** Mind2Onto Steffen Staab, 2004

**MindManager:**  
Standard software for the creation of electronic MindMaps

**Export to OntoEdit**

**Advantage:**  
Intuitive, understandable

**Problem:**  
Semantics of MindMaps only vaguely defined

**Kickoff** OntoEdit/OntoFiller Steffen Staab, 2004

**OntoFiller:** Support for translation and documentation of concepts and relations in multiple languages

Slide 38

**ON knowledge TO** **OTK Methodology:** Knowledge Meta Process Steffen Staab, 2004

**Human Issues**

**Software Engineering**

**Knowledge Management Application**

**Ontology Development**

Identify ..

1. Problems & opportunities
2. Focus of KM application
3. (OTK-) Tools
4. People
5. Capture requirements specification in ORSD
6. Create semi-formal ontology description

**Refinement** Steffen Staab, 2004

**Kickoff**

- Knowledge elicitation with domain experts
  - Refine concepts and relations
  - Typically axioms are identified
- Formalize
  - E.g. F-Logic, DAML+OIL
  - Axioms depend on language capabilities
- Develop and refine *ontology*

Slide 40

**Refinement** Mind2Onto Steffen Staab, 2004

OntoEdit for Beta Tester

File Edit View Tools Windows Help

Connect to Sesame Generate ontology

check

http://www.OntoWeb.org/extended (C:\home\projects\ontoweb\ExtendedOntoWeb.owl)

Inferencing Analyzer Visualizer Debugger Domain-Lexicon OntoFiller Or  
Concepts & Relations Instances Relation axioms Query Tool Disjoint concept

Concept hierarchy

- DEFAULT\_ROOT\_CONCEPT
  - OntoWebPortal
    - Agent
      - Person
      - Software
    - Event
    - News
    - Organization
    - Project
    - Publication
    - Topic
    - Methodology
    - BusinessScenario
    - Language
    - Ontology
    - EducationalResource
    - Product

Relations	Range
author	STRING
dcContributor	STRING
dcCoverage	STRING
dcCreator	STRING
dcDate	STRING
dcDescription	STRING
dcFormat	STRING
dcIdentifier	STRING
dcLanguage	STRING
dcPublisher	STRING
dcRelation	STRING
dcRights	STRING
dcSource	STRING
dcSubject	STRING
dcTitle	STRING
dcType	STRING

de 41

**Refinement** Mind2Onto Steffen Staab, 2004

OntoEdit for Beta Tester

File Edit View Tools Windows Help

Connect to Sesame Generate ontology

check

http://www.OntoWeb.org/extended (C:\home\projects\ontoweb\ExtendedOntoWeb.owl)

Inferencing Analyzer Visualizer Debugger Domain-Lexicon OntoFiller Or  
Concepts & Relations Instances Relation axioms Query Tool Disjoint concept

Concept hierarchy

- DEFAULT\_ROOT\_CONCEPT
  - OntoWebPortal
    - Agent
      - Person
      - Software
    - Event
    - News
    - Organization
    - Project
    - Publication
    - Topic
    - Methodology
    - BusinessScenario
    - Language
    - Ontology
    - EducationalResource
    - Product

Relations	Range
author	STRING
dcContributor	STRING
dcCoverage	STRING
dcCreator	STRING
dcDate	STRING
dcDescription	STRING
dcFormat	STRING
dcIdentifier	STRING
dcLanguage	STRING
dcPublisher	STRING
dcRelation	STRING
dcRights	STRING
dcSource	STRING
dcSubject	STRING
dcTitle	STRING
dcType	STRING

de 42

**Refinement** Inferencing Steffen Staab, 2004

AIFB

**Theoretical Issues**

- F-Logic
  - Object-oriented
  - Deductive Database-oriented
  - Well-founded semantics

**Practical Issues**

- Namespaces/Ontologies/Ontology Parts -> modules
- Switch-off definitions:
  - For testing
  - For fast executions without consistency checks
- DB Connectors: map DB tables via JDBC
- User-definable built-Ins
- Extensive API:
  - remotely connect to the inference engine
  - import and export several standards (e.g., RDF(S))

Slide 43

**Refinement** Exploit Inferencing Steffen Staab, 2004

AIFB

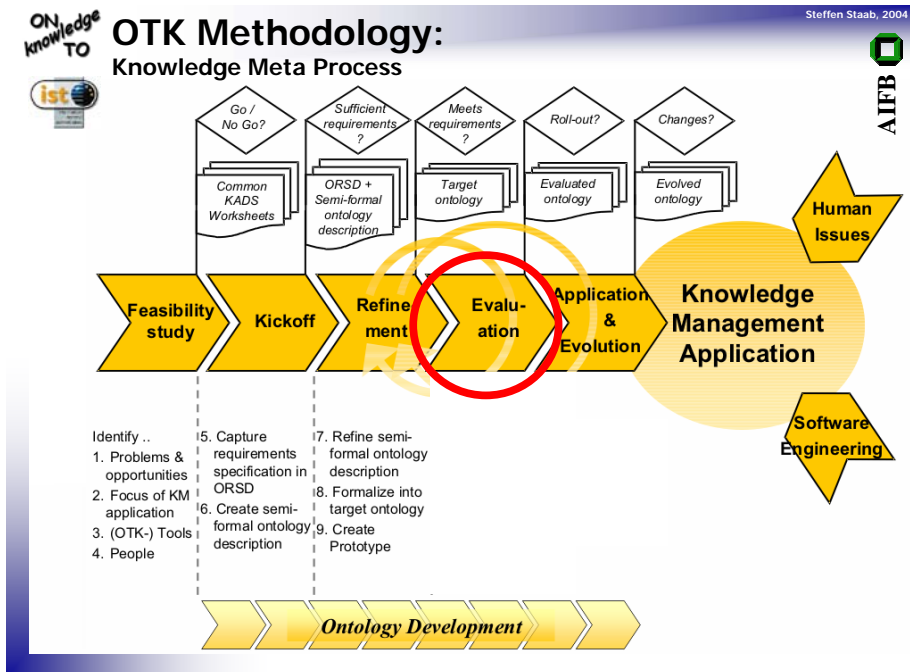
- Hook in existing resources with inferencing
  - Jdbc
  - Rules
- Construct axiom libraries
  - Temporal reasoning
  - PartWhole reasoning
  - ...

- Selective axiom applications
  - F-Logic semantics: E.g. type coercion at concept level
  - Domain specific consistency: non-cyclic hasPart
  - Axioms for modeling policies
  - Debugging

↔

Contrast: OilEd

Slide 44



Refinement

## Evaluation

Steffen Staab, 2004

- Check requirements (ORSD)
  - Are all CQs answered?
  - Is the ontology within the scope?
- Test in target application
  - Analyze usage patterns
- Deploy application(s)

Slide 46

AIFB

Evaluation

## OntoClean

Steffen Staab, 2004

- Task:** Formal evaluation of ontologies
- Well-known methodology:
  - OntoClean** [Welty & Guarino, 2001]
    - Aims at „cleaning“ of hierarchies
    - Based on philosophical notions „essence“, „rigidity“, „identity“, „unity“ ... etc.
- Implementations:** For F-Logic & OWL

Slide 47

AIFB

Evaluation

## OntoClean: Definitions

Steffen Staab, 2004

- „Essence“:** A property is essential for an individual *iff.* it necessarily holds for that individual.  
**Example:** York is *necessarily* a person.
- „Rigidity“**
  - A property is „rigid“ (+R) *iff.* it is **necessarily essential for all** its individuals.
  - A property is „non-rigid“ (-R) *iff.* it is **not essential for some** of its individuals.
  - A property is „anti-rigid“ (~R) *iff.* it is **not essential for all** its individuals.
- Example:** „Person“ is necessarily an essential property for all its individuals.
- There exist similar definitions for „identity“ (+I, -I, +O, -O), „unity“ (+U, -U, ~U), „dependency“ (+D, -D), ... etc. ...

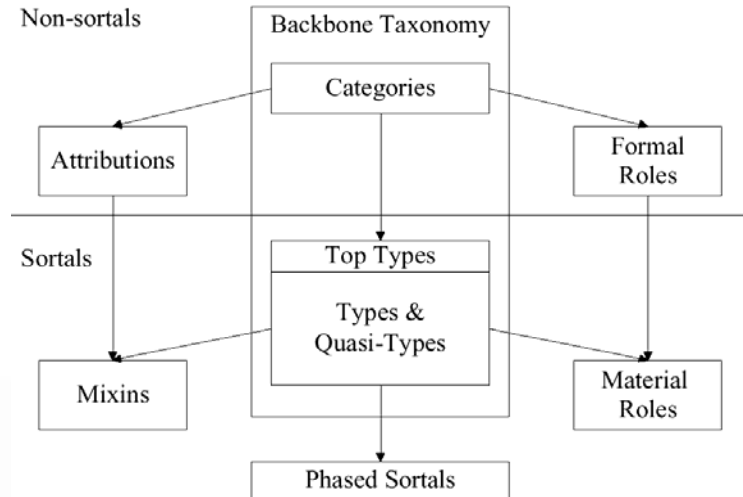
AIFB

# OntoClean: Classification & ideal structure

+O	+I	+R	+D	Type	Sortal
			-D		
-O	+I	+R	+D	Quasi-type	
			-D		
-O	+I	~R	+D	Material role	
			-D	Phased sortal	
-O	+I	~R	+D	Mixin	
			-D		
-O	-I	+R	+D	Category	Non-sortal
			-D		
-O	-I	~R	+D	Formal Role	
			-D		
-O	-I	~R	-D	Attribution	
			-D		
+O	-I			incoherent	
	+I	~R			
		-R			
		-R			

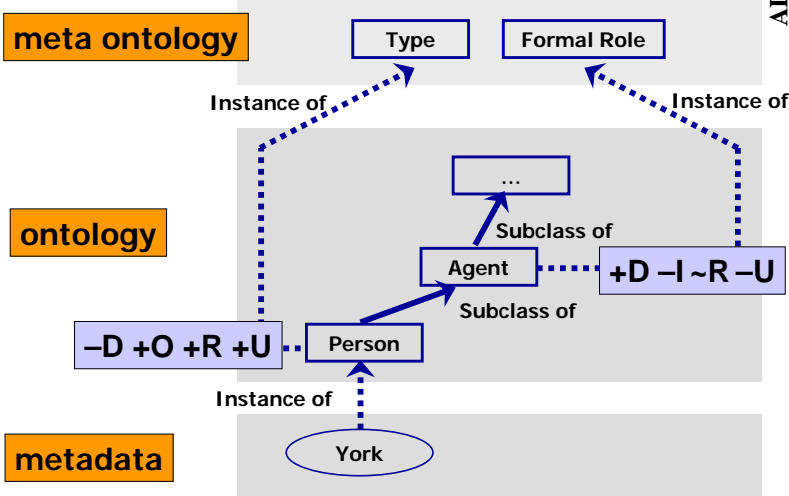
See: [Welty & Guarino, 2001]

# OntoClean: Classification & ideal structure

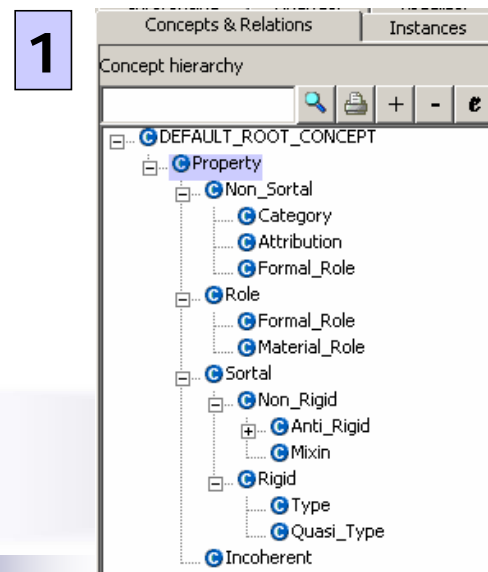


See: [Welty & Guarino, 2001]

# OntoClean: Layering



# OntoCleanPlugin: Formalisation of meta ontology





Uppermost concept „Property“ of the *meta ontology* has attached all relations necessary for classifying concepts of an *ontology*

Logic syntax: FORALL B ( check("Error: ",B," (~R) can't subsume ",C," (+R)!") <- ( EXISTS C{(C::B and (B[#antiR->>"true"] and C[#carryR->>"true"])})) ).

- Anti-rigid concepts (~R) cannot have rigid subconcepts (+R)
- Etc.

Concept	en	+D	-I	~R	-U
Agent	Agent	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Person	Person	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Event	Event	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Topic	Topic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Application	Application	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Methodology	Methodology	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Business	Business	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Language	Language	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Education	Educational	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Conference	Conference	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Lecture	Lecture	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Meeting	Meeting	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Workshop	Workshop	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Misc	Misc	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Association	Association	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Consortium	Consortium	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Department	Department	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Enterprise	Enterprise	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Institute	Institute	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Project	Project	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ResearchGroup	ResearchGroup	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
University	University	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Def.: Being an active participant in some event.

+D -I ~R -U

Agent Agent

Person Person

-D +O +R +U

# OntoCleanPlugin: Cleaning example



Concepts & Relations	Instances	Relation axioms	Query Tool	Disjoint concepts
Inferencing	Analyzer	Debugger	Domain-Lexicon	OntoFiller

Axioms

- inverse
- symmetric
- transitive
- rigidity\_subsumption
- unity\_subsumption
- definition\_4\_5
- definition\_6\_7
- definition\_4\_5\_6\_7
- check\_rigidity\_constraint\_1

```
FORALL V,W,X,Y,Z <- check(V,W,X,Y,Z).
```

Evaluating the query FORALL V,W,X,Y,Z <- check(V,W,:

Error: Agent (~R) can't subsume Person (+R)!



„Is York an agent?“

# OntoCleanPlugin: Cleaning example



Concepts & Relations	Instances	Relation axioms	Query Tool	Disjoint concepts
Inferencing	Analyzer	Debugger	Domain-Lexicon	OntoFiller

Axioms

- inverse
- symmetric
- transitive
- rigidity\_subsumption
- unity\_subsumption
- definition\_4\_5
- definition\_6\_7
- definition\_4\_5\_6\_7
- check\_rigidity\_0

```
FORALL V,W,X,Y,Z <- check(V,W,X,Y,Z).
```

Evaluating the query FORALL V,W,X,Y,Z <- check(V,W,:

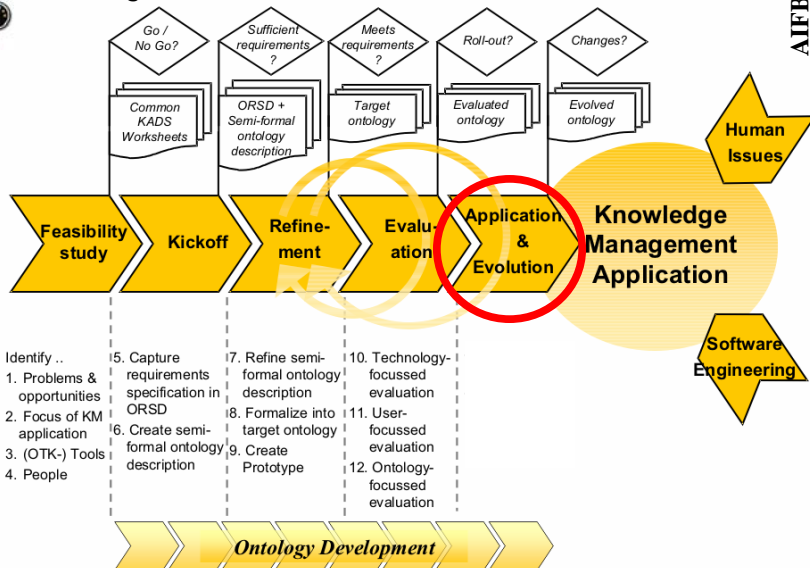
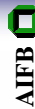
Error: Agent (~R) can't subsume Person (+R)!

Person should not be a subconcept of Agent!  
 Interpretation: Persons *can be* agents, but persons are not necessarily agents.



„Is York an agent?“

# OTK Methodology: Knowledge Meta Process



1. Problems & opportunities
2. Focus of KM application
3. (OTK-) Tools
4. People
5. Capture requirements specification in ORSD
6. Create semi-formal ontology description
7. Refine semi-formal ontology description
8. Formalize into target ontology
9. Create Prototype
10. Technology-focused evaluation
11. User-focused evaluation
12. Ontology-focused evaluation

# Worksheet for life cycle aspects of ontology



- Who is going to maintain it?
- Who is going to pay for it?
- What is the resulting quality (increase, decrease)?
- How large are the network costs (cost of negotiation grows quadratic with number of participants)?
- What is the expected life time of the ontology?
- How brittle is it with regard to updates?
- What error types will occur/are relevant?

## Worksheet for life cycle aspects of metadata

Steffen Staab, 2004



- ala ontology
- Co-ordinated change of data and metadata?
- Co-ordinated change of ontology and metadata?
- Cold start (chicken-and-egg) problem: A problem? How to overcome?
- Granularity of metadata envisaged: classification, identification of people/events/relationships/etc.

Faustregel – Kosten:

- Hardware 1
- Software 10
- Daten 100

Slide 61

## Coordination of metadata & ontology

Steffen Staab, 2004



- Match or mismatch between the two,
  - E.g. classification only, but ontology about transitive relationships

Slide 62

## Type-1 Error

Steffen Staab, 2004

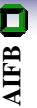


- False Positive
  - Often dominating problem in company internal IR
  - It can be more costly to learn about all low-price provider of pens than to just select from a sample

Slide 63

## Type-2 Error

Steffen Staab, 2004



- False negative: Positive example not detected as such
  - Often not critical for information retrieval  
„show me bookstores who sell the `CommonKADS` book“
  - Often critical for B2B operations  
„whether `6000 computer` is mapped to `IBM RS/6000 SP system` or to `HP OmniBook Laptop 6000` is a large difference with regard to price and performance“

Slide 64

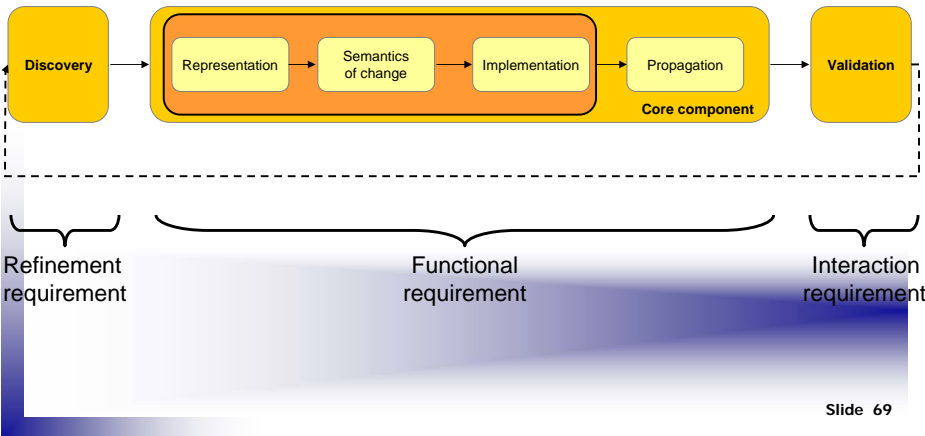
- 1. (MOD) Knowledge Modeling: the ability of the knowledge engineer to model information/write axioms
- 2. (IMP) Knowledge Implementation/Modeling Language: the ability of the representation language to accurately represent axioms
- 3. (INF) Inference and Reasoning: the ability of the inference engine to “find the needle in the haystack”
- 4. (KFL) Knowledge Formation and Learning: the ability of the system (KB + inference engine) to acquire and merge knowledge through automated and semi-automated techniques
- 5. (SCL) Scalability: the ability of the KB to scale

- <http://www.haloproject.com>

- 6. (MGT) Knowledge Management: the ability of the system to maintain, track changes, test, organize, document; the ability of the knowledge engineer to search for knowledge
- 7. (QMN) Query Management: the ability of the system to robustly answer queries
- 8. (ANJ) Answer Justification: the ability of the system to provide justifications for answers in the correct context and resolution
- 9. (QMT) Quality Metrics: the ability of the developers to determine how “good” the knowledge base is at any given point in its evolution
- 10. (MTA) Meta Capabilities: the system's ability to utilize meta-reasoning or meta-knowledge

- Ontology development is necessarily an **iterative** and a **dynamic process**
- Ontologies must be able to **evolve** for a number of reasons:
  - **Application domains and user's needs are changing**
  - **System can be improved**
- Developing ontologies is **expensive**, but evolving them is even **more expensive**

- Basic requirement {
  - **Functional requirement:**
    - enable the handling of the required changes
    - ensure the consistency of the underlying ontology and all dependent artifacts
- Extended requirements {
  - **Interaction requirement** – supports the user to manage changes more easily
  - **Refinement requirement** – offers advice to the user for continual system refinement



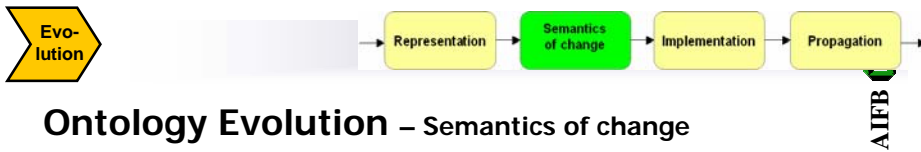
- Elementary changes
  - They can not be decomposed into simpler ones
  - They heavily depend on the underlying ontology model

$MoveConcept \neq (RemoveSubConcept + AddSubConcept)$

- Composite changes
  - They are more powerful
  - They have coarser granularity
  - They have often more meaningful semantics

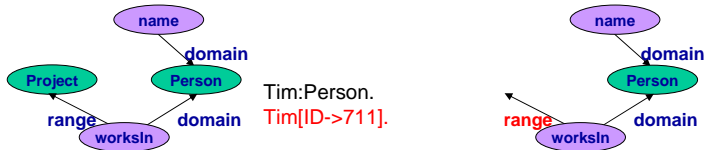
Composite change	Description
Move concept	Move a concept from one parent to another.
Merge concepts	Replace several concepts with one and aggregate all instances.
Extract subconcepts	Split a concept into several subconcepts and distribute properties among them.
Extract superconcept	Create a common superconcept for a set of unrelated concepts and transfer common properties to it.
Extract related concept	Extract related information into a new concept and relate it to the original concept.
Shallow concept copy	Duplicate a concept with all its properties.
Deep concept copy	Recursively apply shallow copy to all subconcepts of a concept.
Pull up properties	Move properties from a subconcept to a superconcept.
Pull down properties	Move properties from a superconcept to a subconcept.
Move properties	Move properties from one concept to another concept.
Shallow property copy	Duplicate a property with same domain and range.
Deep property copy	Recursively apply shallow copy to all subproperties of a property.
Move Instance	Moves an instance from one concept to another.

Composite change	Description
Move concept	Move a concept from one parent to another.
Merge concepts	Replace several concepts with one and aggregate all instances.
Extract superconcept	Create a common superconcept for a set of unrelated concepts and transfer common properties to it.
Extract related concept	Extract related information into a new concept and relate it to the original concept.
Deep concept copy	Recursively apply shallow copy to all subconcepts of a concept.
Pull up properties	Move properties from a subconcept to a superconcept.
Pull down properties	Move properties from a superconcept to a subconcept.
Move properties	Move properties from one concept to another concept.
Shallow property copy	Duplicate a property with same domain and range.
Deep property copy	Recursively apply shallow copy to all subproperties of a property.
Move Instance	Moves an instance from one concept to another.

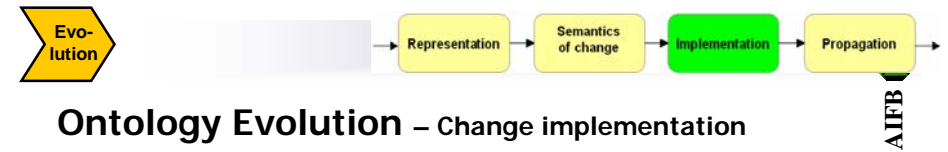


## Ontology Evolution – Semantics of change

- Enables resolution of changes in a systematic manner, ensuring consistency of the whole ontology

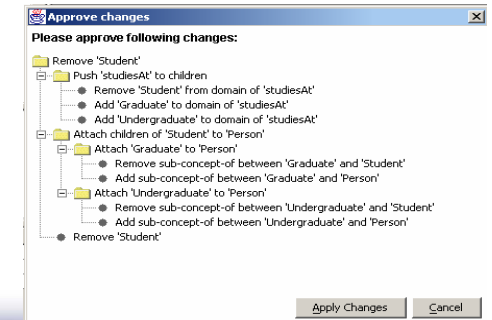


Slide 73



## Ontology Evolution – Change implementation

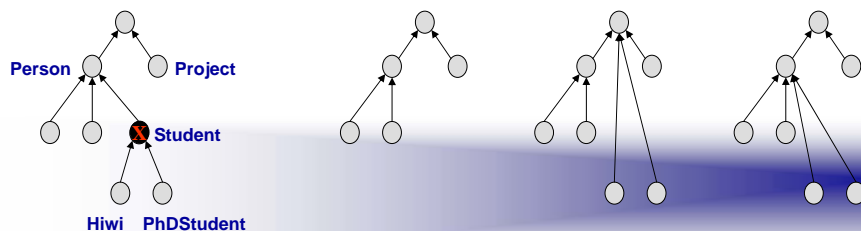
- After **user's approval** all changes are applied to the ontology
- Since it is necessary to perform several changes together, the **transaction server** is needed.



Slide 74



An evolution strategy unambiguously defines the way how changes will be resolved



Slide 75



## Elementary evolution strategies

### Resolution points:

- how to handle orphaned concepts;
- how to handle orphaned properties;
- how to propagate properties to the concept whose parent changes;
- what constitutes a valid domain of a property;
- what constitutes a valid range of a property;
- whether a domain (range) of a property can contain a concept that is at the same time a subconcept of some other domain (range) concept;
- the allowed shape of the concept hierarchy;
- the allowed shape of the property hierarchy;
- ...

- delete
- reconnect to the root
- reconnect to the superconcepts

Common policy consisting of a set of elementary evolution strategies, each giving an answer for one resolution point, is an **evolution strategy**

Slide 76

**Evolution Example** Steffen Staab, 2004

**List of changes:**

- Delete Concept Hierarchy Edge

Slide 77

**Evolution Example**

**List of changes:**

- AddPropertyDomain has\_name, PhDStudent
- AddPropertyDomain has\_index, PhDStudent
- RemoveSubConcept PhDStudent, Student
- AddSubConcept PhDStudent, KAON:Root

**List of changes:**

- RemovePropertyInstance has\_name, PhDStudentBob, Bob
- RemovePropertyInstance has\_index, PhDStudentBob, 9352
- RemoveSubConcept PhDStudent, Student
- AddSubConcept PhDStudent, KAON:Root

Slide 78

**Evolution Advanced evolution strategies** Steffen Staab, 2004

Mechanism to prioritize and arbitrate among different evolution strategies, relieving the user of choosing them individually:

- **structure-driven strategy**
- **process-driven strategy**
- **instance-driven strategy**
- **frequency-driven strategy**

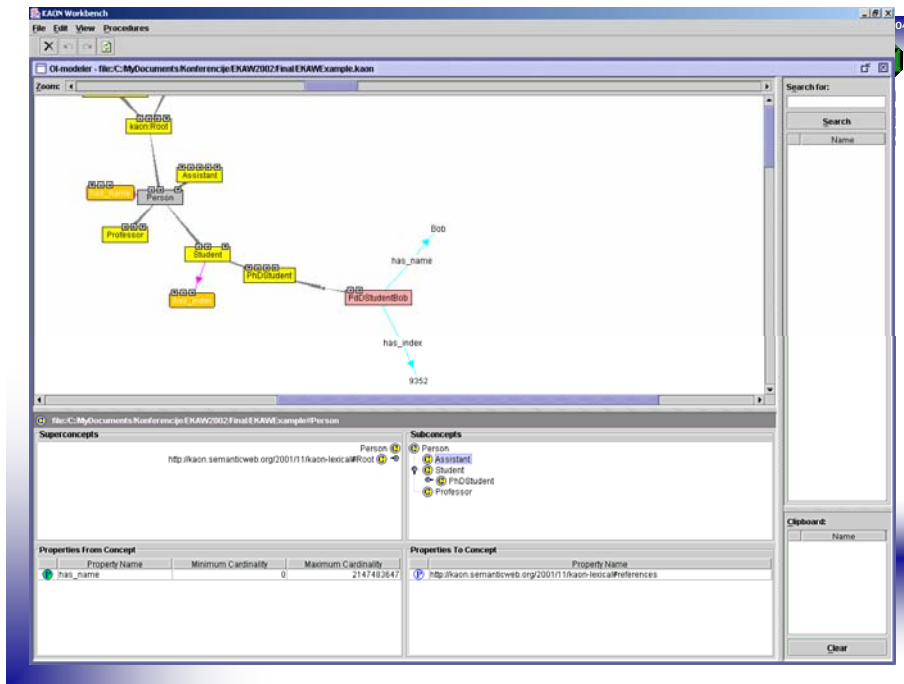
Slide 79

**Evolution Implementation** Steffen Staab, 2004

<http://kaon.semanticweb.org>

Applications & Services	OIModeler - Ontology and Metadata Engineering Tool	KAON Portal and other User Interface Applications and Services	
Middleware	KAON Access Interface		
	Evolution Strategy	Reversibility Services	Change Discovery
	Evolution Strategy	Reversibility Services	Interaction Logging
	Evolution Strategy	Reversibility Services	Evolution Logging
Data and Remote Services	KAON API		
	RDF API		
	KAON RDF Server		
Persistence, Transactions, Security			

Slide 80



This screenshot shows the same ontology graph as the previous image, but with an "Evolution Strategy Set-up" dialog box open. A black box labeled "Resolution points" has red arrows pointing to various options in the dialog. The dialog contains several sections with radio button options:

- Orphaned concepts will be...**
  - ...deleted.
  - ...reconnected to ontology root.
  - ...reconnected to superconcepts.
- When concept's parent is removed...**
  - ...properties will not be propagated.
  - ...all inherited properties will be added to the concept.
  - ...only parent's properties will be added to the concept.
- Properties without any domain concepts...**
  - ...may exist in the OI-model.
  - ...should be deleted from the OI-model.
- Instance consistency...**
  - ...should be enforced.
  - ...should not be enforced.

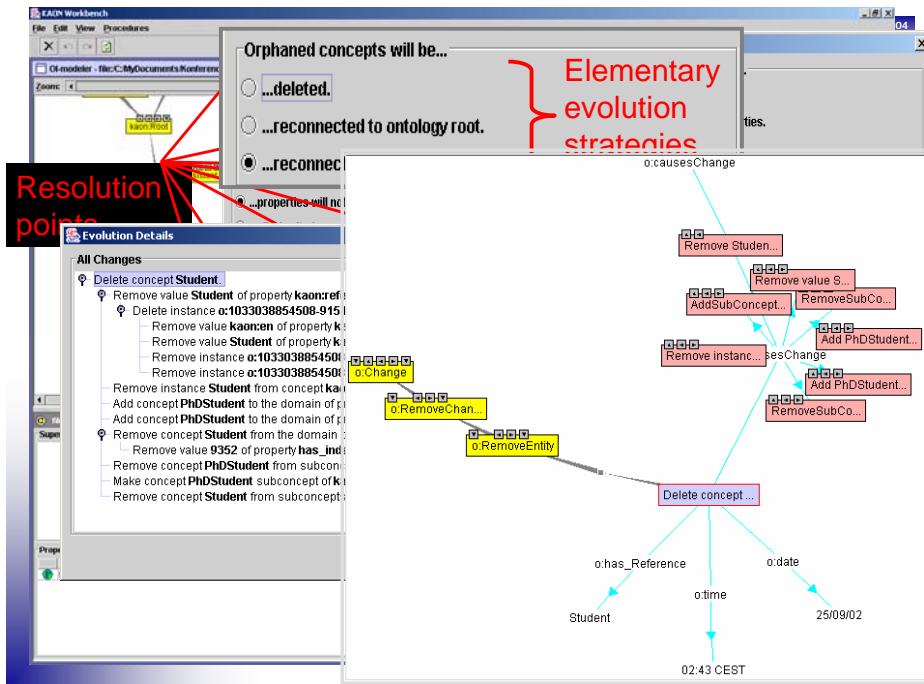
Other sections include "Domain/range of a property...", "Properties without any range concepts...", and "When creating a hierarchy path which already exists...".

This screenshot is similar to the previous one, but with a red bracket annotation on the right side of the dialog box. The bracket groups the "Orphaned concepts will be..." and "When concept's parent is removed..." sections, with the text "Elementary evolution strategies" written next to it. The "Resolution points" box and arrows are also present.

This screenshot shows the "Evolution Strategy Set-up" dialog box with the "reconnected to superconcepts" option selected. A new "Evolution Details" dialog box is open, listing the changes that will be applied:

- Delete concept `Student`
- Remove value `Student` of property `kaon:references` for instance `α:1033038854508-915157117`.
- Delete instance `α:1033038854508-915157117`.
- Remove value `Student` of property `kaon:language` for instance `α:1033038854508-915157117`.
- Remove instance `α:1033038854508-915157117` from concept `kaon:Label`.
- Remove instance `Student` from concept `kaon:Root`.
- Add concept `PhDStudent` to the domain of property `has_name`.
- Add concept `PhDStudent` to the domain of property `has_index`.
- Remove concept `Student` from the domain of property `has_index`.
- Remove value `9352` of property `has_index` for instance `PhDStudentBob`.
- Remove concept `PhDStudent` from subconcepts of `Student`.
- Make concept `PhDStudent` subconcept of `kaon:Root`.
- Remove concept `Student` from subconcepts of `Person`.

The "Elementary evolution strategies" annotation is also present on the left side of the dialog box.



**Evolution** Evolution wrap-up Steffen Staab, 2004

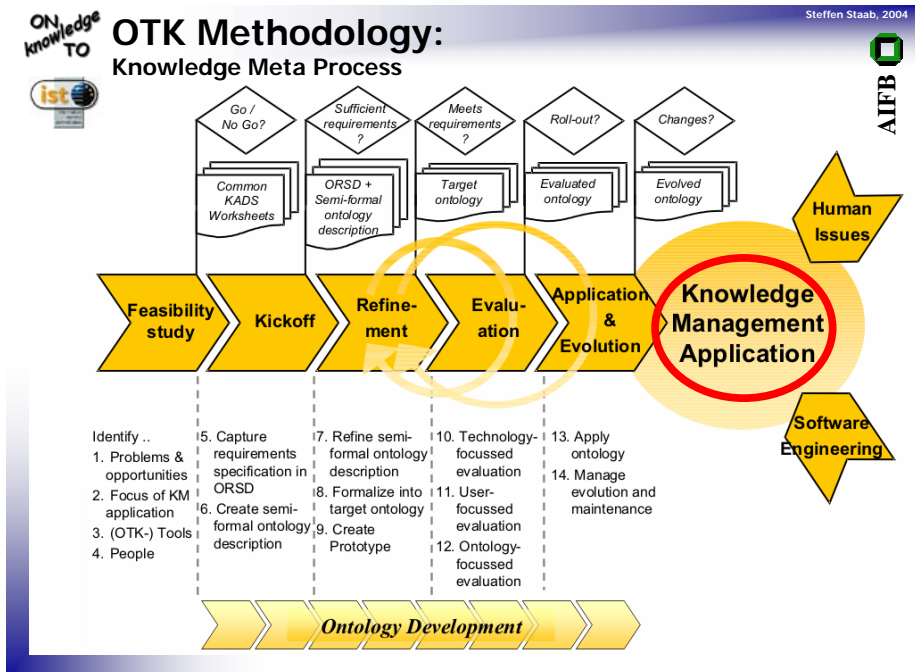
**OntoLogging:**

- process-based approach for ontology evolution
- Evolution strategies that enable the customisation of the ontology evolution process
- Implementation in KAON framework

**Ongoing work:**

- Evolution between distributed ontologies
- Change discovery

Slide 86



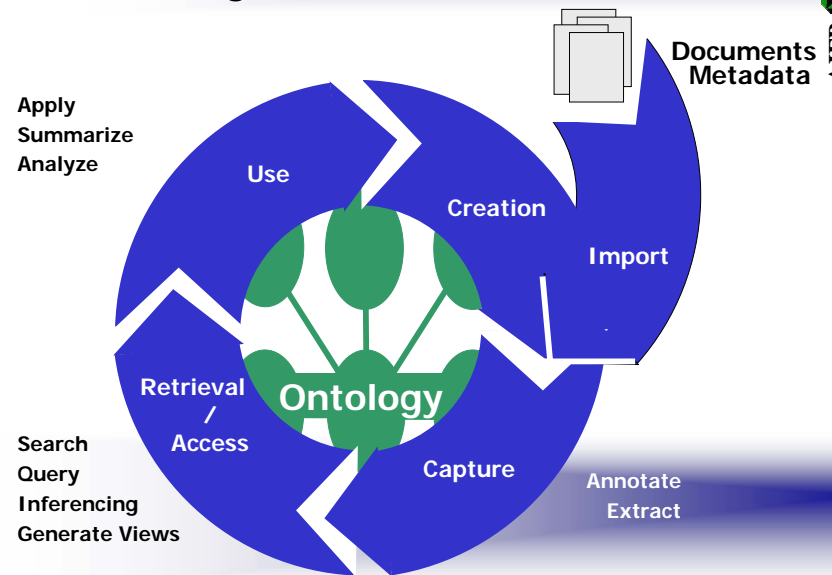
## Conclusions on Knowledge Meta Process

# Experiences from OTK Case Studies



- **Guidelines** for domain experts from industry have to be pragmatic
  1. Train the user about ontologies
  2. Show the concrete advantage of the KMS
  3. Model precisely – but allow for imprecise views (most users cannot distinguish classes vs instances or isa vs partOf)
- **Plan for Maintenance**
- **Avoid/Reduce chicken-and-egg problem**
  1. Plan für content that makes KMS interesting
  2. Show quick win
- **Collaborative ontology engineering** requires sophisticated tool support *and* physical presence
- **Brainstorming** is a valuable add-on during the early stages of ontology engineering

# Knowledge Process



# OTK Case Study @ BT

# Users Portal

