Ontology Lifecycle
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

Ontology

AcademicStaff

PostDoc

Prof

cooperate_with

rdfs:domain

rdfs:range

rdfs:subClassOf

rdfs:subClassOf

instance of

Cooperate_with

Links have explicit meanings!

Annotation

PostDoc

rdf:ID="sha"

name=Siegfried Handschuh

cooperate_with rdf:resource="http://www.aifb.uni-karlsruhe.de/WBS/sst#sst"

... 

Prof

name=Steffen Staab

... 

Web Page

Siegfried Handschuh

He is working together with Steffen Staab in the Knowledge Management Group

URL

http://www.aifb.uni-karlsruhe.de/WBS/sha

Research:

Semantic Web, Knowledge Management, Natural Language

http://www.uni-koblenz.de/~staab
Explicit vs. Implicit Knowledge

Explicit Knowledge

Implicit Knowledge

Socialisation

Externalisation

Internalisation

Combination / Integration

From

To
Case study: OntoWeb.org

Portal Generation

Navigation

Query/Search

Content

Integration

Collect metadata from participating partners
Ontology-based Processes

Knowledge Meta Process
Design, Implementation, Evolution of Ontology

Knowledge Process
Usage of Ontology
OTK Methodology: Knowledge Meta Process

- **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
OTK Methodology: Knowledge Meta Process

Feasibility study
- Identify
  1. Problems & opportunities
  2. Focus of KM application
  3. (OTK-) Tools
  4. People

Kickoff
- 15. Capture requirements specification in ORSD
- 6. Create semi-formal ontology description

Refinement
- 17. Refine semi-formal ontology description
- 8. Formalize into target ontology
- 9. Create Prototype

Evaluation
- 10. Technology-focused evaluation
- 11. User-focused evaluation
- 12. Ontology-focused evaluation

Application & Evolution
- 13. Apply ontology
- 14. Manage evolution and maintenance

Knowledge Management Application

Human Issues
Software Engineering

Ontology Development

ISWeb – Lecture „Semantic Web“ (8)
OTK Methodology: Knowledge Meta Process

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

- OntoKick: Capture Requirements Specification
- Mind2Onto: Brainstorming
- OntoFiller: Documentation & Translation
- OntoClean: Formal Ontology Evaluation
- SesamePlugin: Storage & Versioning

Ontology Development

Kickoff -> Refinement -> Evaluation -> Evolution

Ontology Development
Tools

- OntoKick: Capture Requirements Specification
- Mind2Onto: Brainstorming
- OntoFiller: Documentation & Translation
- OntoClean: Formal Ontology Evaluation
- SesamePlugin: Storage & Versioning

Ontology Development

Kickoff → Refinement → Evaluation → Evolution

OntoEdit (Infrastructure)
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 study

Current State:
Skills Management

• Employee data distributed over many systems

• Different schemata for data

• Incomplete data
Feasibility study

Intended state: Skills Management

Expert search

Knowledge gap analysis

Personal development

Intellectual Capital Assessment
OTK Methodology: Knowledge Meta Process

Feasibility study

- Kickoff
- Refinement
- Evaluation
- Application & Evolution

Knowledge Management Application

Human Issues
Software Engineering

Ontology Development

Identify...
1. Problems & opportunities
2. Focus of KM application
3. (OTK-) Tools
4. People

OTK Methodology:
Knowledge Meta Process
Ontology Kickoff

• Ontology Requirements Specification Document (ORSD)
  1. Domain & Goal
  2. Design guidelines
  3. Available knowledge sources
  4. Potential users and user scenarios
  5. Applications supported by the ontology

• Analyze knowledge sources
• Develop baseline ontology description

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

E.g. Competency questions

Ontology Learning!
ORS – Ontology Requirements Specification

• 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
### CQ – Competency Questionnaire

<table>
<thead>
<tr>
<th>CQ Nr.</th>
<th>Competency Question</th>
<th>Concepts</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CQ1</td>
<td>What are the subsidiaries, divisions and locations of company X?</td>
<td>company, subsidiary, division, location</td>
<td>company has subsidiary company has division company has location</td>
</tr>
<tr>
<td>CQ2</td>
<td>Which companies acquired company X?</td>
<td>company, acquisition</td>
<td>company makes acquisition acquisition has buyer acquisition has seller</td>
</tr>
<tr>
<td>CQ3</td>
<td>Which companies merged in 1990 in the rubber industry?</td>
<td>company, merger, year, industry</td>
<td>company makes merger company isPartOf industry merger happensIn year</td>
</tr>
<tr>
<td>CQ4</td>
<td>Who is CEO of company X?</td>
<td>CEO, company</td>
<td>company has CEO</td>
</tr>
</tbody>
</table>
• 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
Requirement specification
Requirement specification

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

03-01-2001

Completion deadline

12-31-2001
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
Knowledge Sources

1. Domain & Goal
   - Design Guidelines
   - Knowledge Sources
   - Users & Use Cases
   - Deployment

2. Source List
   - Source
   - Type
   - Status
   - File (OXML, HTML, XML, DOC, TXT)

3. NEW SOURCE
   - Competency Questionnaire
   - Ontology
   - Word Document
   - XML Document
   - RDF Document
   - Text Document

Steffen Staab
ISWeb – Lecture „Semantic Web“ (23)
Knowledge Sources

1. Domain & Goal
2. Design Guidelines
3. Knowledge Sources
4. Users & Use Cases
5. Deployment

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

Kick-off
Knowledge Sources
Competency questions

New competency questionnaire

Concept hierarchy
- Root
  - Ding
  - Immaterielles
  - Massen_Konzept
  - Mathematisches_Konzept
  - Raumliches_Konzept
  - Situation
  - Zahlenbares_Konzept

Knowledge Engineer
- York Sure

Domain Expert
- Claus Boyens

Edition Date
- 10-7-2001

Match Pattern
- 4 letters
- Activate stemming

Edit Question
- Gibt es ein Luxushotel in Rostock?

Question List
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 Surfshule auf Usedom?
5. Gibt es einen weißen Sandstrand auf Rügen?

ADD TO LIST
CHANGE
REMOVE
IMPORT...

DONE
CANCEL
Competency questions

New competency questionnaire

Concept hierarchy
- Root
  - Ding
  - Immaterialles
  - Massen_Konzept
  - Mathematisches_Konzept
  - Raumliches_Konzept
  - Situation
  - Zählbares_Konzept

Knowledge Engineer: York Sure
Domain Expert: Claus Boyens
Edition Date: 10-7-2001

Match Pattern: 4 letters, Activate stemming

Edit Question:
Gibt es ein Luxushotel in Rostock?

Question List:
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?

ADD TO LIST
CHANGE
REMOVE
IMPORT...

DONE CANCEL
Competency questions
Competency questions
Competency questions
Competency questions
Traceability
Traceability
Brainstorming, Structuring, Formalisation

Kick-off
Mind2Onto

• **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
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

Mind2Onto Kickoff
Steffen Staab
ISWeb – Lecture "Semantic Web" (37)

- 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

Mind2Onto

Kick off

MindManager:
Standard software for the creation of electronic MindMaps

Advantage:
Intuitive, understandable

Problem:
Semantics of MindMaps only vaguely defined

Export to OntoEdit
OntoEdit/OntoFiller

OntoFiller: Support for translation and documentation of concepts and relations in multiple languages
OTK Methodology: Knowledge Meta Process

Feasibility study
Kickoff
Refinement
Evaluation
Application & Evolution
Knowledge Management Application

Ontology Development

Human Issues
Software Engineering

OTK Methodology:
Knowledge Meta Process

Go / No Go?
Sufficient requirements?
Meets requirements?
Roll-out?
Changes?

Common KADS Worksheets
ORSD + Semi-formal ontology description
Target ontology
Evaluated ontology
Evolved ontology

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

- 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**
**Mind2Onto**

![Mind2Onto Interface](image)

<table>
<thead>
<tr>
<th>Concept hierarchy</th>
<th>Relations</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT_ROOT_CONCEPT</td>
<td>author</td>
<td>STRING</td>
</tr>
<tr>
<td>OntoWebPortal</td>
<td>dcContributor</td>
<td>STRING</td>
</tr>
<tr>
<td>Agent</td>
<td>dcCoverage</td>
<td>STRING</td>
</tr>
<tr>
<td>Person</td>
<td>dcCreator</td>
<td>STRING</td>
</tr>
<tr>
<td>Software</td>
<td>dcDate</td>
<td>STRING</td>
</tr>
<tr>
<td>Event</td>
<td>dcDescription</td>
<td>STRING</td>
</tr>
<tr>
<td>News</td>
<td>dcFormat</td>
<td>STRING</td>
</tr>
<tr>
<td>Organization</td>
<td>dcIdentifier</td>
<td>STRING</td>
</tr>
<tr>
<td>Project</td>
<td>dcLanguage</td>
<td>STRING</td>
</tr>
<tr>
<td>Publication</td>
<td>dcPublisher</td>
<td>STRING</td>
</tr>
<tr>
<td>Topic</td>
<td>dcRelation</td>
<td>STRING</td>
</tr>
<tr>
<td>Methodology</td>
<td>dcRights</td>
<td>STRING</td>
</tr>
<tr>
<td>BusinessScenario</td>
<td>dcSource</td>
<td>STRING</td>
</tr>
<tr>
<td>Language</td>
<td>dcSubject</td>
<td>STRING</td>
</tr>
<tr>
<td>Ontology</td>
<td>dcTitle</td>
<td>STRING</td>
</tr>
<tr>
<td>EducationalResource</td>
<td>dcType</td>
<td>STRING</td>
</tr>
</tbody>
</table>

Refine-ment
Mind2Onto
Inferencing

Theoretical Issues

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

Practical Issues

• Namespace mechanism: 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))
Exploit Inferencing

- Hook in existing resources with inferencing
  - Jdbc
  - Rules

- Construct axiom libraries
  - Temporal reasoning
  - PartWhole reasoning
  - ... Contrast: OiEd

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

Feasibility study

Kickoff

Refinement

Evaluation

Application & Evolution

Knowledge Management Application

Ontology Development

OTK Methodology:
Knowledge Meta Process

- Go / No Go?
- Sufficient requirements?
- Meets requirements?
- Roll-out?
- Changes?

Knowledge Meta Process:

1. 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
7. Refine semi-formal ontology description
8. Formalize into target ontology
9. Create Prototype

Human Issues

Software Engineering
Evaluation

• Check requirements (ORSD)
  – Are all CQs answered?
  – Is the ontology within the scope?

• Test in target application
  – Analyze usage patterns

• Deploy application(s)
OntoClean

- **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
OntoClean: Definitions

„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. ...
# OntoClean:
## Classification & ideal structure

<table>
<thead>
<tr>
<th>+O</th>
<th>+I</th>
<th>+R</th>
<th>+D</th>
<th>-D</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>-O</td>
<td>+I</td>
<td>+R</td>
<td>+D</td>
<td>-D</td>
<td>Quasi-type</td>
</tr>
<tr>
<td>-O</td>
<td>+I</td>
<td>~R</td>
<td>+D</td>
<td>-D</td>
<td>Material role</td>
</tr>
<tr>
<td>-O</td>
<td>+I</td>
<td>~R</td>
<td>-D</td>
<td></td>
<td>Phased sortal</td>
</tr>
<tr>
<td>-O</td>
<td>+I</td>
<td>~R</td>
<td>+D</td>
<td>-D</td>
<td>Mixin</td>
</tr>
<tr>
<td>-O</td>
<td>-I</td>
<td>+R</td>
<td>+D</td>
<td>-D</td>
<td>Category</td>
</tr>
<tr>
<td>-O</td>
<td>-I</td>
<td>~R</td>
<td>+D</td>
<td>-D</td>
<td>Formal Role</td>
</tr>
<tr>
<td>-O</td>
<td>-I</td>
<td>~R</td>
<td>-D</td>
<td>+D</td>
<td>Attribution</td>
</tr>
<tr>
<td>+O</td>
<td>-I</td>
<td>~R</td>
<td>-D</td>
<td>+D</td>
<td>incoherent</td>
</tr>
<tr>
<td>+I</td>
<td>~R</td>
<td>-R</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

See:
[Welty & Guarino, 2001]
OntoClean: Classification & ideal structure

Non-sortals

Attributions

Sortals

Mixins

Backbone Taxonomy

Categories

Top Types

Types & Quasi-Types

Material Roles

Formal Roles

Phased Sortals

See: [Welty & Guarino, 2001]
OntoClean: Layering

- **meta ontology**
- **ontology**
- **metadata**

Eva-
uation

- **Type**
- **Formal Role**

Instance of

- **Agent**
- **Person**
- **York**

+D –I ~R –U

–D +O +R +U

Subclass of

ontological metadata
OntoCleanPlugin: Formalisation of meta ontology
OntoCleanPlugin: Formalisation of meta ontology

Uppermost concept „Property“ of the meta ontology has attached all relations necessary for classifying concepts of an ontology
OntoCleanPlugin: Formalisation of axioms

- Anti-rigid concepts (~R) cannot have rigid subconcepts (+R)
- Etc.
OntoCleanPlugin: Cleaning example
Def.: Being an active participant in some event.

<table>
<thead>
<tr>
<th>Agent</th>
<th>Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>Person</td>
</tr>
</tbody>
</table>

**OntoCleanPlugin:**

Cleaning example

<table>
<thead>
<tr>
<th>+D –I ~R –u</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>-D +O +R +u</th>
</tr>
</thead>
</table>
OntoCleanPlugin: Cleaning example


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

„Is York an agent?“
OntoCleanPlugin: Cleaning example

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

Feasibility study

Kickoff

Refinement

Evaluation

Application & Evolution

Knowledge Management Application

Ontology Development

1. 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
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

Human Issues

Software Engineering
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

• ala ontology

Rule of thumb – costs:
• Hardware 1
• Software 10
• Daten 100

• 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.
Coordination of metadata & ontology

• Match or mismatch between the two,
  – E.g. classification only, but ontology about transitive relationships
Type-1 Error

- 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
Type-2 Error

- 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“
Refined Error types (Halo Project)

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
Refined Error types II (Halo Project)

- 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 Evolution: Technical aspects

- 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
Requirements for ontology evolution

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
Ontology Evolution Process

Discovery → Representation → Semantics of change → Implementation → Propagation → Validation

- Core component

Refinement requirement

Functional requirement

Interaction requirement
Ontology Evolution – Change representation

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

MoveConcept ≠ (RemoveSubConcept + AddSubConcept)

• Composite changes
  ➢ They are more powerful
  ➢ They have coarser granularity
  ➢ They have often more meaningful semantics
Ontology Evolution – Change representation

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

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

• Enables resolution of changes in a systematic manner, ensuring consistency of the whole ontology
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.
Evolution Strategies

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

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;
- …

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

**Applications & Services**

| OIModeler - Ontology and Metadata Engineering Tool | KAON Portal and other User Interface Applications and Services |

**Middleware**

<table>
<thead>
<tr>
<th>KAON Access Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Discovery</td>
</tr>
<tr>
<td>Interaction Logging</td>
</tr>
<tr>
<td>Evolution Strategy</td>
</tr>
<tr>
<td>Reversibility Services</td>
</tr>
<tr>
<td>Evolution Logging</td>
</tr>
</tbody>
</table>

| KAON API |
| RDF API  |
| KAON RDF Server |

**Data and Remote Services**

| Persistence, Transactions, Security |

**http://kaon.semanticweb.org**

---

Steffen Staab  
ISWeb – Lecture „Semantic Web“ (80)
Resolution points
Resolution points:

Elementary evolution strategies:

- Orphaned concepts will be...
  - ...deleted.
  - ...reconnected to ontology root.
  - ...reconnected to superconcepts.

- Properties will not be propagated.
- ...may contain subconcepts of other domain/range concepts.
Resolution points

Elementary evolution strategies
Evolution wrap-up

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
OTK Methodology: Knowledge Meta Process

Feasibility study
- Identify...
  1. Problems & opportunities
  2. Focus of KM application
  3. (OTK-) Tools
  4. People

Kickoff
- Go / No Go?
- Common KADS Worksheets

Refinement
- Sufficient requirements?
- ORSD + Semi-formal ontology description

Evaluation
- Meets requirements?
- Target ontology

Application & Evolution
- Roll-out?
- Evaluated ontology
- Changes?
- Evolved ontology

Ontology Development

Knowledge Management Application

Human Issues

Software Engineering

1. Capture requirements specification in ORSD
2. Create semi-formal ontology description
3. Refine semi-formal ontology description
4. Formalize into target ontology
5. Create Prototype
6. Technology-focused evaluation
7. User-focused evaluation
8. Ontology-focused evaluation
9. Apply ontology
10. Manage evolution and maintenance
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

Creation

Import

Ontology

Use

Capture

Retrieval / Access

Search

Query

Inferencing

Apply

Summarize

Analyze

Generate

Views

Steffen Staab
ISWeb – Lecture „Semantic Web“ (90)
OTK Case Study @ BT
Users Portal

OntoShare
Sesame
OMM
BOR

OntoExtract
OntoWrapper

QuizRDF
Spectacle
OTK Architecture

- QuizRDF
- OntoShare
- Spectacle
- OntoEdit
- User
- Knowledge Engineer

- OMM
- BOR
- Sesame
- OIL-Core ontology repository
- Annotated Data Repository
  - RDF
  - pers05
  - tel 731
  - par05
  - about
  - car

- OntoWrapper
- OntoExtract

Data Repository (external)