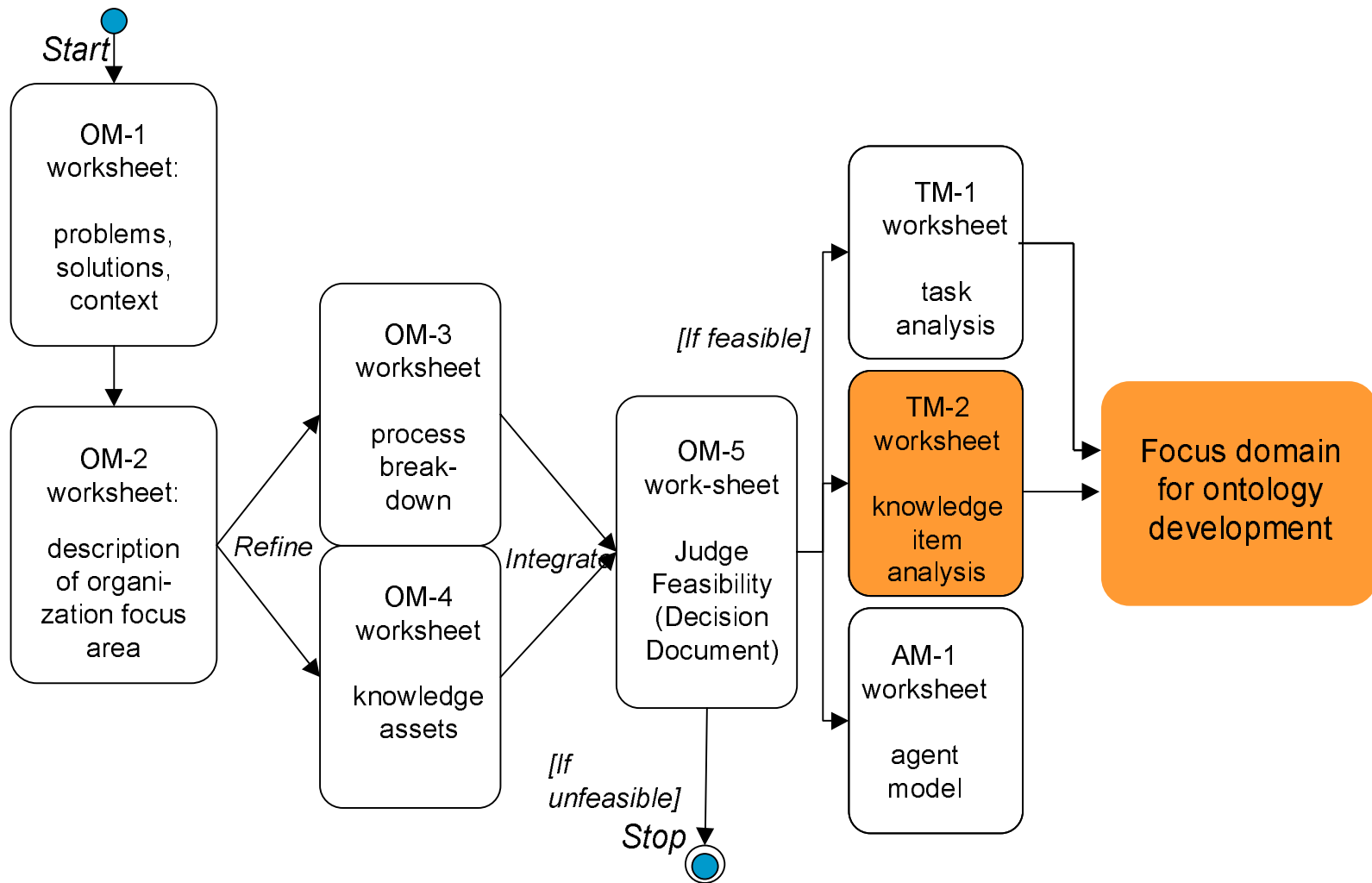


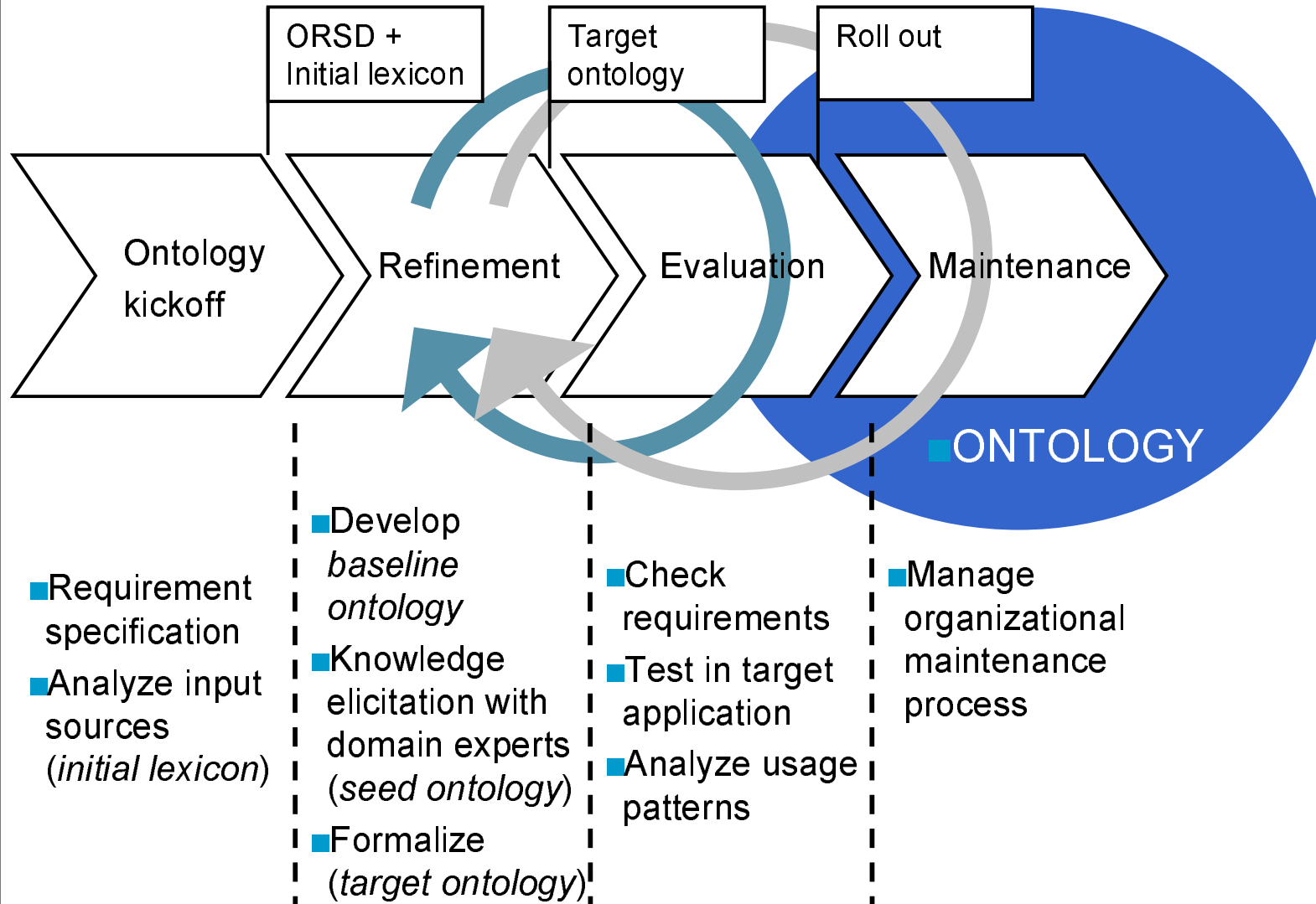
### 3.3.3 Methodology for Ontology Development

- Development of an ontology-based KM application involves ontology development as a very important subtask
- Outcome of CommonKADS feasibility study is starting point
  - TM-2 worksheet: **Knowledge item analysis**
- Ontology development is split up in several phases:
  - Kick-off phase
  - Refinement phase
  - Evaluation phase
  - Maintenance phase

# CommonKADS - Context Modeling Road Map



# Ontology Development Process



### 3.3.3.1 Kick-off Phase

- result is **Ontology Requirements Specification Document (ORSD)**
  - characterises the planned area of the ontology application
  - defines the scope of the ontology to be constructed
- ORSD is composed of parts:
  - **administrative information**
    - name
    - date
    - involved ontology engineer(s)
  - **requirements specification**

## 5

Wissensmanagement SS 2001

## Requirements Specification (I)

- **Domain and Goal**
  - what is the objective of the planned KM application
  - based on task analysis (TM-1)
- **Design Guidelines**
  - description of domain in use
  - estimation of size of ontology
  - exploit knowledge item worksheet (TM-2)
- **Supported Applications**
  - brief characteristics of planned application
  - specification of system environment

## Requirements Specification (II)

- **Knowledge Sources**

- types of knowledge sources may be very different
  - domain experts
  - (reusable) ontologies
  - documents / systems
    - dictionaries
    - thesauri
    - product descriptions
    - organisational charts
    - employee role descriptions
    - ...
- Knowledge item analysis is important input (TM-2)

## Requirements Specification (III)

- **Usage Scenarios** (Users and Use cases)
  - describe users/user groups
  - identify stakeholders
  - describe usage scenarios
    - how do they want to use the system?
    - what kind of support do they expect ?
    - use e.g. UML use-case diagrams
- **Competency Questions**
  - define collection of queries that should be supported by the system
  - analyze queries to find relevant lexical entries (concepts and relations)
  - explore scenarios
  - collect competency questionnaire



# Competency Questionnaire

| Competency Questionnaire No. 1 |   |  |                               |
|--------------------------------|---|--|-------------------------------|
| <b>Name:</b>                   |   | skill-man-ontology                         |                               |
| <b>Date:</b>                   |   | 2001/03/22                                 |                               |
| <b>Ontology Engineer:</b>      |   | T. Model                                   | <b>Domain Expert:</b> X. Pert |
| No.                            | Competency Question   | Lexical Entries                            | Type                          |
| Q1                             | Which of our consultants has experience with JAVA programming language? | consultant                                 | concept                       |
|                                |   | consultant <i>is a</i> employee            | <i>isA</i> relation           |
|                                |   | JAVA                                       | concept                       |
|                                |   | programming language                       | concept                       |
|                                |   | JAVA <i>is a</i> program-<br>ming language | <i>isA</i> relation           |
|                                |   | programming language <i>is a</i> skill     | <i>isA</i> relation           |
|                                |   | employee <i>has experience with</i> skill  | relation                      |
| Q2                             | What is the salary of a senior programmer?                              | salary                                     | concept                       |
|                                |   | ...  |                               |
| Q3                             |   |  |                               |

# Initial lexicon - example

Consultant

Employee

JAVA

Programming language

Experience

Skill

Programmer

Project

Customer

Industry

...

...

HasExperienceWith

WorksIn

Contains

| Competency Questionnaire No. 1 |   |   |                              |
|--------------------------------|---|---|------------------------------|
| <b>Name:</b>                   |   | skill-man-ontology                        |                              |
| <b>Date:</b>                   |   | 2001/03/22                                |                              |
| <b>Ontology Engineer:</b>      |   | T. Model                                  | <b>Domain Expert:</b> X.Pert |
| No.                            | Competency Question   | Lexical Entries                           | Type                         |
| Q1                             | Which of our consultants has experience with JAVA programming language? | consultant                                | concept                      |
|                                |   | consultant <i>is a</i> employee           | <i>isA</i> relation          |
|                                |   | JAVA                                      | concept                      |
|                                |   | programming language                      | concept                      |
|                                |   | JAVA <i>is a</i> programming language     | <i>isA</i> relation          |
|                                |   | programming language <i>is a</i> skill    | <i>isA</i> relation          |
|                                |   | employee <i>has experience with</i> skill | relation                     |
| Q2                             | What is the salary of a senior programmer?                              | salary                                    | concept                      |
|                                |   | ...                                       |                              |
| Q3                             | ...   |   |                              |

### 3.3.3.2 Refinement Phase

- Construct a mature application-oriented target ontology
- Step 1: **Gather baseline ontology**
  - derive **concepts** from lexical entries in initial lexicon
  - embed concepts into **is-a hierarchy** (taxonomy)
  - add additional concepts to taxonomy,  
e.g obvious generalisations of concepts

### 3.3.3.2 Refinement Phase

- Step 2: **Develop Seed Ontology**
  - add additional **concepts**
  - add **relations** between concepts of the baseline ontology
  - add **attributes** to concept descriptions
  - add **axioms** (informal descriptions)
  - perform knowledge elicitation process with domain experts
  
- Step 3: **Develop Target Ontology**
  - **formalize** seed ontology
  - choose appropriate representation language,  
e.g. Frame Logic, RDF Schema, OIL, DAML+OIL, ...

### 3.3.3.3 Evaluation Phase

- check target ontology with respect to:
  - ontology requirements specification document
  - competency questions
- collect feedback from beta users
  - track usage of ontology
    - which parts are used / not used
- feeds back to refinement phase

### 3.3.3.4 Maintenance Phase

- running application has to adapt to changing environment
  - ontology has to adapt as well (**evolving ontology**)
- evolving aspects:
  - new lexical entries show up in application
    - extend ontology
  - lexical entries change meaning
    - change reference function
  - parts of ontology became obsolete (not needed anymore)
- set up clearly defined **organisational process** for updating the ontology
  - feeds back to refinement phase
  - update ontology by ontology engineer based on collection of proposed changes

## Software environment

- Ontology development has to be supported by suitable software environment
  - **graphical** interface
  - **abstract** representation language (as abstract as possible)
  - **distributed** development by **several** persons
    - aspects of Computer-Supported Co-operative Work
  - **export** in suitable representation languages
    - Frame Logic, RDF Schema, OIL, DAML+OIL
  - **import** of ontologies

# Protegé-2000 Ontologie-Editor

newspaper Protégé-2000 [http://protege.stanford.edu/applet\_demo/Newspaper/newspaper]

Project Edit Window Help

Classes Slots Facets Forms Instances

Relationship: Subclass V C X

Relationship Hierarchy:

- THING
  - CLASS
    - FACET
      - SLOT
        - Author
          - Content
            - Advertisement
              - Personals\_Ad
              - Standard\_Ad
            - Article
              - Layout\_info
                - Billing\_Chart
                - Content\_Layout
                - Prototype\_Newspaper
                - Rectangle
                - Section
              - Library
              - Newspaper
              - Organization
              - Person

Superclasses

- Advertisement

Name: Standard\_Ad

Constraints

Role: Concrete

Template Slots

| Slot Name       | Type     | Cardinality | Default | Other Facets             |
|-----------------|----------|-------------|---------|--------------------------|
| ad_name         | String   | Single      |         |                          |
| content_layout  | Instance | Single      |         | classes={Content_Layout} |
| expiration_date | String   | Single      |         |                          |
| image           | String   | Single      |         |                          |
| page_number     | Integer  | Single      |         |                          |
| published_in    | Instance | Single      |         | classes={Newspaper}      |
| purchaser       | Instance | Single      |         | classes={Person}         |
| salesperson     | Instance | Single      |         | classes={Salesperson}    |
| section         | Instance | Single      |         | classes={Section}        |
| urgent          | Boolean  | Single      |         |                          |

Java Applet Window



# OntoEdit (AIFB)

The screenshot displays the OntoEdit V0.5 application window. The main area shows the 'GETESS Ontology 2.3' hierarchy with a search bar containing 'hotel'. The hierarchy is as follows:

- Organisation
  - Betrieb
    - Dienstleistungsbetrieb
      - Bahnhof
      - Bank
      - Flughafen
      - Fremdenverkehrsamt
      - Friseursalon
      - Geldwechselstelle
      - Geschaeft
      - Hafen
      - Kindergarten
      - Reiseveranstalter
    - Unterkunft
      - Appartement
      - Appartementanlage
      - Bauernhof
      - Campingplatz
      - Clubanlage
      - Feriendorf
      - Ferienhaus
      - Ferienwohnung
      - Gasthof
      - Hotel
      - Jugendherberge
      - Motel
      - Pension

On the right, the 'Relations of Unterkunft' window is open, showing a list of relations and their types:

| Name                    | Type                              |
|-------------------------|-----------------------------------|
| hat_Oertlichkeit        | Raeumliches_Konzept               |
| Adresse                 | Adresse                           |
| hat_Tageslicht          | Tageslicht                        |
| hat_Eintrittspreis      | Eintrittspreis                    |
| hat_Saisonabhaengigkeit | Qualitatives_Zeitkonzept          |
| hat_Zimmer              | Zimmer                            |
| hat_Ausstattung         | Nichtprivate_Ausstattung_der_U... |
| Anzahl_Betten           | INTEGER                           |
| Haustiere_erlaubt       | BOOLEAN                           |
| bietet_Dienstleistung   | Dienstleistung                    |
| in_Gebiet               | Gebiet                            |
| bietet_Ereignis         | Ereignis                          |
| behindertenfreundlich   | BOOLEAN                           |
| bietet_Aktivitaet       | Aktion                            |
| Verpflegung             | STRING                            |
| Klassifizierung         | STRING                            |

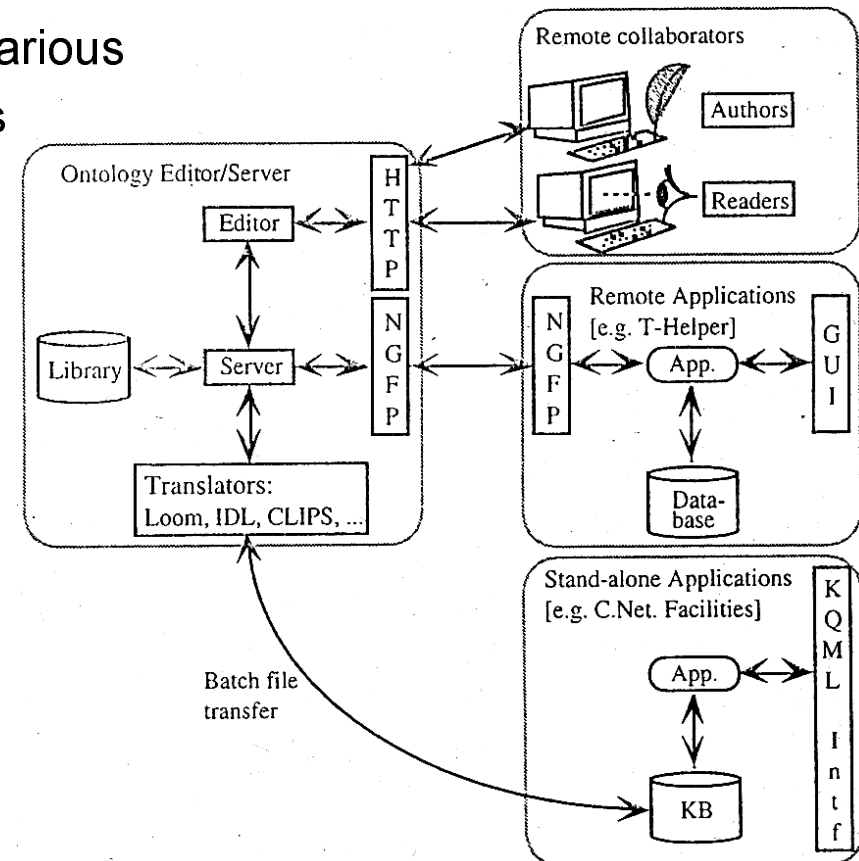
Below this, a window for defining relations is open, showing the 'normal' tab and a table of existing relations:

| Relation       | Domain | Range  | Type       | Comment |
|----------------|--------|--------|------------|---------|
| istVerwandtMit | Person | Person | symmetric  |         |
| istVerwandtMit | Person | Person | transitive |         |

### 3.3.4 Reusability of ontologies

#### Ontolingua-Server

- Development of a server, which provides a library of ontologies
- Server offers an interface for
  - application system: Application Programming Interfaces (API)
  - user: WWW-Browser
  - compiler: compiling in the various representation formalisms



[Farquhar et al. 97]  
<http://ontolingua.stanford.edu/>

## Reusability of ontologies (I)

- Supported by structuring-relations
- Examples for relations between ontologies:

inclusion

**Generic Products**  
Product  
Service-Agreement



**National Semiconductor**  
Operational-Amplifier  
Subclass-of: Product

restriction

**Numbers**  
+ is commutative  
and associative



**Integer Arithmetic**  
All numbers are integers

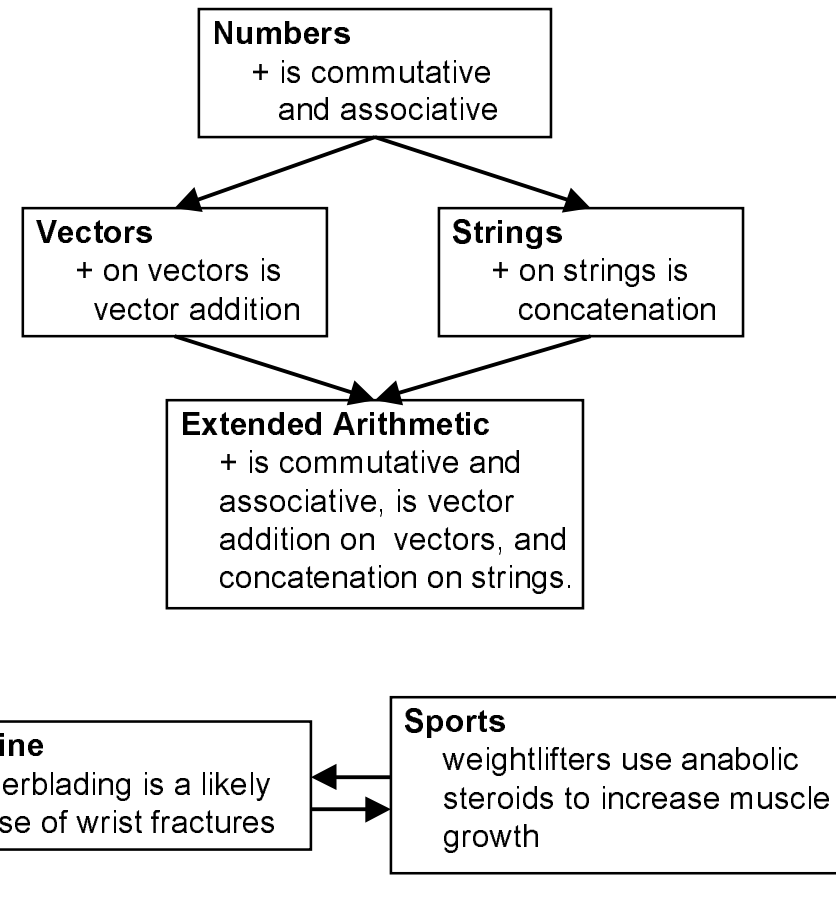
## Reusability of ontologies (II)

- Supported by structuring-relations

- Examples for relations between ontologies:

polymorphic  
refinement

cyclic  
dependence



## 3.3.5 Merging of Ontologies

### Motivation - Ontology Integration Trends

Ontologies are relevant for many applications:

- web-search applications (Yahoo, Lycos, Xift, ...)
- E-Commerce applications (Amazon, eBay, Virtual Vineyards, REI, VerticalNet, CommerceOne, etc.)
- configuration applications (Dell, PROSE, etc.)
- ...

## The Need For Ontology Merging

- Large-scale knowledge repositories will contain ontologies produced by multiple authors in multiple settings
- Ontologies for applications will be built by **aligning**, **assembling** and **extending** multiple modular ontologies from repositories
- Ontologies developed by multiple authors will frequently
  - Express overlapping knowledge in a common domain
  - Use **differing representations**, **vocabularies** and **structures**
- For such ontologies to be used together as building blocks - their representational and structural differences must be reconciled

## How Ontology Merging Tools Can Help (I)

- Combine input ontologies with name clashes
  - Treat each input ontology as a separate **namespace**
  
- Support merging of concepts and relations
  - Replace all occurrences by the merged concept or relation
  - Test for logical consistency of merge
    - **name resolution** in the case of pairs of concepts whose names are similar enough in some way
    - **taxonomy refinement** in the case of a concept that has subconcepts that came from multiple ontologies
    - **slot traversal** in the case of a concept that has slots (attributes) that came from multiple ontologies.

## How Ontology Merging Tools Can Help (II)

### ■ Focus attention

- Create an **agenda** of important aspects of the active ontology
- The items on the agenda are sorted according to a relevance heuristic such that the most important items are grouped at the top
- Users are guided through the agenda focusing their attention so that they can look at areas of the ontology that need fixes
  - e.g., portions of ontology where new relations are likely to be needed

### ■ Derive relationships among concepts and relations

- Disjointness, equivalence, subsumption,...



## Chimaera – A Merging and Diagnostic Ontology Environment

Web-based tool utilizing the KSL Ontolingua platform that supports:

- merging multiple ontologies found in distributed environments
- analysis of single or multiple ontologies
- focus attention in problematic areas
- simple browsing and mixed initiative editing

## Demo des Chimaera



Knowledge System Laboratory  
Stanford University

## Demo des Chimaera

### Upload

**Select KB content to upload:** ?

Please select the file to upload and select "Do it" to merge the contents of the file into the current KB.

**Upload a file:**

**Upload from a URL:**

**Load file on server:**

**Language:**

**Name of source KB:**

**Forms to upload:**


- AFFINIA
- ANSI KIF
- CLASSIC
- CML
- COOL/CLIPS
- CYC KIF FORMAT
- HPKB WITH ANSI KIF
- HPKB WITH KIF 3.0
- INDENTED INPUT
- KIF 3.0
- OCELOT
- OKBC**
- OKBC WITH ANSI KIF
- ONTOLINGUA
- PROTEGE
- SMITH

Chimaera can load KBs in many languages and formats

## Demo des Chimaera

**Analysis:** One active command  
**Class:** One active command  
**Decomposition:** No active commands  
**File:** 10 active commands KB  
**Mode:** 8 active commands  
**View:** 4 active commands  
**Name:** Pretty

Current KB is now Kb-Merge  
No local classes in KB Kb-Merge



OKBC cyc-agents.okbc  
OKBC cyc-platform-military.okbc  
OKBC cyc-products.okbc  
OKBC cyc-weapons.okbc  
OKBC individuals.hpkb  
OKBC merge-test1.okbc  
OKBC merge-test2.okbc  
OKBC merge-test3.okbc  
OKBC new-stuff.hpkb  
OKBC saic-agents-frame-ontology-downward  
OKBC saic-agents-frame-ontology-downward  
OKBC saic-agents-saic-downwards-with-suff  
OKBC saic-agents-saic-downwards.okbc  
OKBC saic-agents.okbc  
OKBC saic-commodities-steve.okbc  
OKBC saic-commodities.okbc  
OKBC saic-products-2.okbc  
OKBC saic-products-steve.okbc

## Demo des Chimaera

?

**Analysis:** 17 active commands

**Class:** 3 active commands

**Decomposition:** No active commands

**File:** 12 active commands    KB arg

**Mode:** 10 active commands    Numeric arg

**View:** 8 active commands

**Name:**    **Pretty name:**

There are 13 new roots.

- ▶ [Anti-Ship-Missile-Radar](#) {from [saic-products-2](#)}
- ▶ [Artillery](#) {from [saic-products-2](#)}
- ▶ [Bomb](#) {from [saic-products-2](#)}
- ▶ [Commodity](#) {from [saic-products-2](#)}
- ▶ [Conventional-Weapon](#) {from [saic-products-2](#)}
- ▶ [Mine](#) {from [saic-products-2](#)}
- ▶ [Naval-Countermine-Weapon](#) {from [saic-products-2](#)}
- ▶ [Nuclear-Technology](#) {from [saic-products-2](#)}
- ▶ [Projectile-Weapon](#) {from [saic-products-2](#)}
- ▶ [Reconnaissance-Equipment](#) {from [saic-products-2](#)}
- ▶ [Surface-Ship](#) {from [saic-products-2](#)}
- ▶ [Weapon-Of-Mass-Destruction](#) {from [saic-products-2](#)}
- ▶ [Weapon-Of-Short-Strike](#) {from [saic-products-2](#)}

Select KB content to upload

## Demo des Chimaera

The screenshot displays the Chimaera interface with the following settings on the left:

- Analysis:** 17 active commands
- Class:** 3 active commands
- Decomposition:** No active commands
- File:** 12 active commands (KB arg)
- Mode:** 10 active commands
- View:** 8 active commands
- Name:** (empty field) **Pretty**

A yellow status bar indicates: "There are 13 new roots."

Below this, a list of KB content to upload is shown, each preceded by a red triangle icon:

- ▶ Anti-Ship-Missile-Radar {from saic-products-2}
- ▶ Artillery {from saic-products-2}
- ▶ Bomb {from saic-products-2}
- ▶ Commodity {from saic-products-2}
- ▶ Conventional-Weapon {from saic-products-2}
- ▶ Mine {from saic-products-2}
- ▶ Naval-Countermine-Weapon {from saic-products-2}
- ▶ Nuclear-Technology {from saic-products-2}
- ▶ Projectile-Weapon {from saic-products-2}
- ▶ Reconnaissance-Equipment {from saic-products-2}
- ▶ Surface-Ship {from saic-products-2}
- ▶ Weapon-Of-Mass-Destruction {from saic-products-2}
- ▶ Weapon-Of-Short-Strike {from saic-products-2}

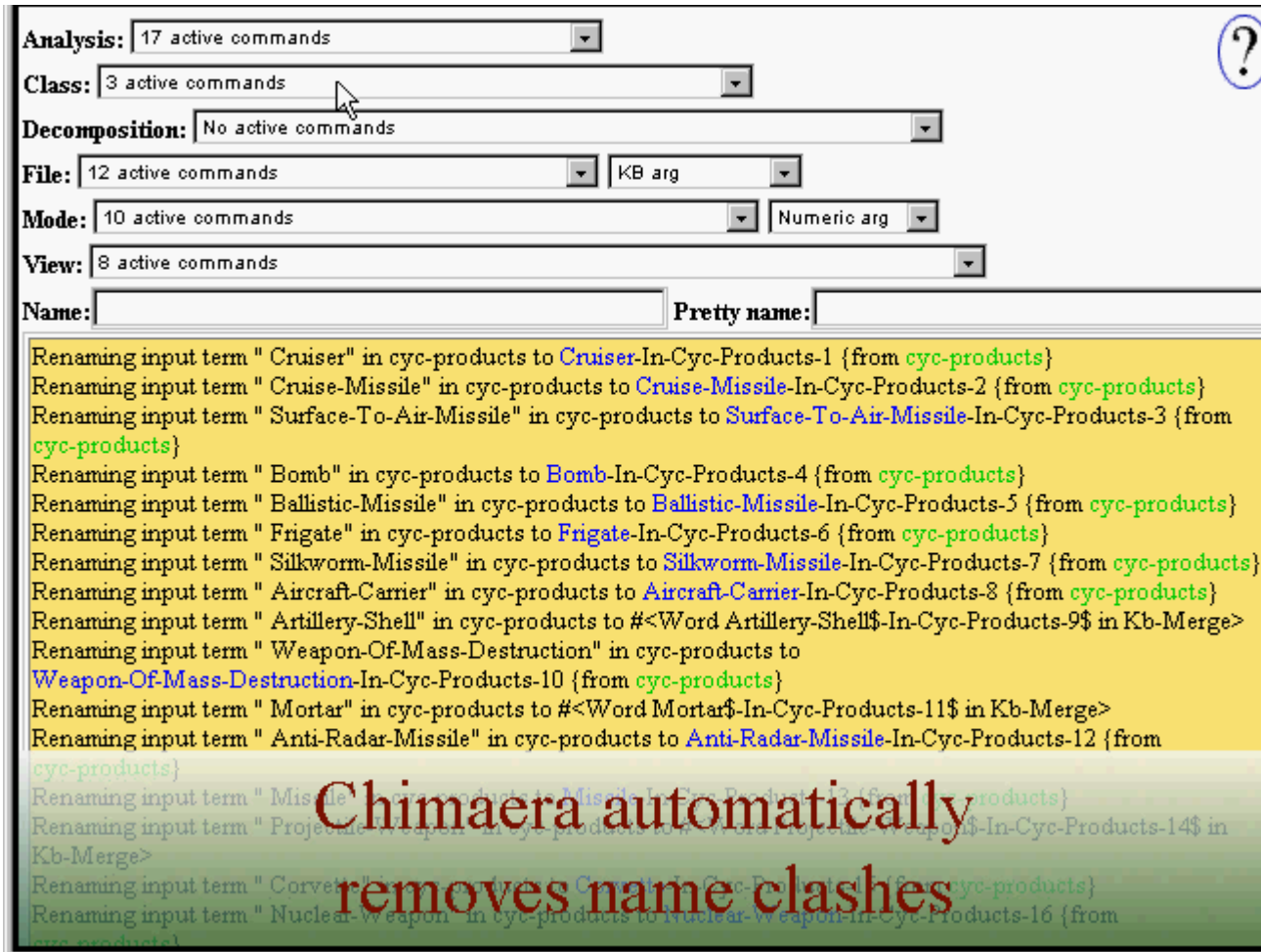
On the right side, a list of KB content is shown, each preceded by a red triangle icon and a status indicator (OK, BC, or both):

- OKBC cyc-agents.okbc
- OKBC cyc-platform-military.okbc
- OKBC **cyc-products.okbc**
- OKBC cyc-weapons.okbc
- OKBC individuals.hpkb
- OKBC merge-test1.okbc
- OKBC merge-test2.okbc
- OKBC merge-test3.okbc
- OKBC new-stuff.hpkb
- OKBC saic-agents-frame-ontology-downward
- OKBC saic-agents-frame-ontology-downward
- OKBC saic-agents-saic-downwards-with-suff
- OKBC saic-agents-saic-downwards.okbc
- OKBC saic-agents.okbc
- OKBC saic-commodities-steve.okbc
- OKBC saic-commodities.okbc
- OKBC saic-products-2.okbc

At the bottom, a large red text overlay reads: "Select KB content to upload".

## Demo des Chimaera

### Processing



The screenshot shows the Chimaera interface with the following settings:

- Analysis:** 17 active commands
- Class:** 3 active commands
- Decomposition:** No active commands
- File:** 12 active commands, KB arg
- Mode:** 10 active commands, Numeric arg
- View:** 8 active commands
- Name:** (empty)
- Pretty name:** (empty)

The output list shows the following renaming actions:

- Renaming input term "Cruiser" in cyc-products to **Cruiser-In-Cyc-Products-1** {from **cyc-products**}
- Renaming input term "Cruise-Missile" in cyc-products to **Cruise-Missile-In-Cyc-Products-2** {from **cyc-products**}
- Renaming input term "Surface-To-Air-Missile" in cyc-products to **Surface-To-Air-Missile-In-Cyc-Products-3** {from **cyc-products**}
- Renaming input term "Bomb" in cyc-products to **Bomb-In-Cyc-Products-4** {from **cyc-products**}
- Renaming input term "Ballistic-Missile" in cyc-products to **Ballistic-Missile-In-Cyc-Products-5** {from **cyc-products**}
- Renaming input term "Frigate" in cyc-products to **Frigate-In-Cyc-Products-6** {from **cyc-products**}
- Renaming input term "Silkworm-Missile" in cyc-products to **Silkworm-Missile-In-Cyc-Products-7** {from **cyc-products**}
- Renaming input term "Aircraft-Carrier" in cyc-products to **Aircraft-Carrier-In-Cyc-Products-8** {from **cyc-products**}
- Renaming input term "Artillery-Shell" in cyc-products to **#<Word Artillery-Shell\$-In-Cyc-Products-9\$ in Kb-Merge>**
- Renaming input term "Weapon-Of-Mass-Destruction" in cyc-products to **Weapon-Of-Mass-Destruction-In-Cyc-Products-10** {from **cyc-products**}
- Renaming input term "Mortar" in cyc-products to **#<Word Mortar\$-In-Cyc-Products-11\$ in Kb-Merge>**
- Renaming input term "Anti-Radar-Missile" in cyc-products to **Anti-Radar-Missile-In-Cyc-Products-12** {from **cyc-products**}
- Renaming input term "Missile" in cyc-products to **Missile-In-Cyc-Products-13** {from **cyc-products**}
- Renaming input term "Projectile-Weapon" in cyc-products to **#<Word Projectile-Weapon\$-In-Cyc-Products-14\$ in Kb-Merge>**
- Renaming input term "Corvette" in cyc-products to **Corvette-In-Cyc-Products-15** {from **cyc-products**}
- Renaming input term "Nuclear-Weapon" in cyc-products to **Nuclear-Weapon-In-Cyc-Products-16** {from **cyc-products**}

**Chimaera automatically removes name clashes**

## Wissensmanagement SS 2001





## Demo des Chimaera

Analysis: 17 active commands

Class: 3 active commands

Decomposition: No active commands

File: 12 active commands KB arg

Mode: 10 active commands Numeric arg

View: 8 active commands

Name: Pretty name:

Stubbing definition for class [Artillery-Shell-In-Cyc-Products-9](#) {from [cyc-products](#)}  
Stubbing definition for class [Heavy-Machine-Gun](#) {from [cyc-products](#)}  
Stubbing definition for class [Howitzer](#) {from [cyc-products](#)}  
Stubbing definition for class [Selective-Fire-Firearm](#) {from [cyc-products](#)}  
Stubbing definition for class [Midcourse-Guided-Projectile](#) {from [cyc-products](#)}  
Stubbing definition for class [Projectile-Weapon-In-Cyc-Products-14](#) {from [cyc-products](#)}  
Stubbing definition for class [Mortar-In-Cyc-Products-11](#) {from [cyc-products](#)}

- ▶ [Ab-205-Helicopter](#) {from [cyc-products](#)}
- ▶ [Ab-206-Helicopter](#) {from [cyc-products](#)}
- ▶ [Agricultural-Product](#) {from [cyc-products](#)}
- ▶ [Air-Base](#) {from [cyc-products](#)}
- ▶ [Airborne-Early-Warning-Aircraft](#) {from [cyc-products](#)}
- ▶ [Aircraft-Multi-Role](#) {from [cyc-products](#)}
- ▶ [Anti-Ship-Missile-Radar](#) {from [saic-products-2](#)}
- ▶ [Artificial-Intelligence-Program](#) {from [cyc-products](#)}
- ▶ [Artillery](#) {from [saic-products-2](#)}
- ▶ [Artillery-Shell-In-Cyc-Products-9](#) {from [cyc-products](#)}
- ▶ [Mortar-Shell](#) {from [cyc-products](#)}

# Name Resolution



## Demo des Chimaera

**Analysis:** 17 active commands

**Class:** 12 active commands

**Decomposition:** No active commands

**File:** 12 active commands KB arg

**Mode:** 12 active commands Numeric arg

**View:** 15 active commands

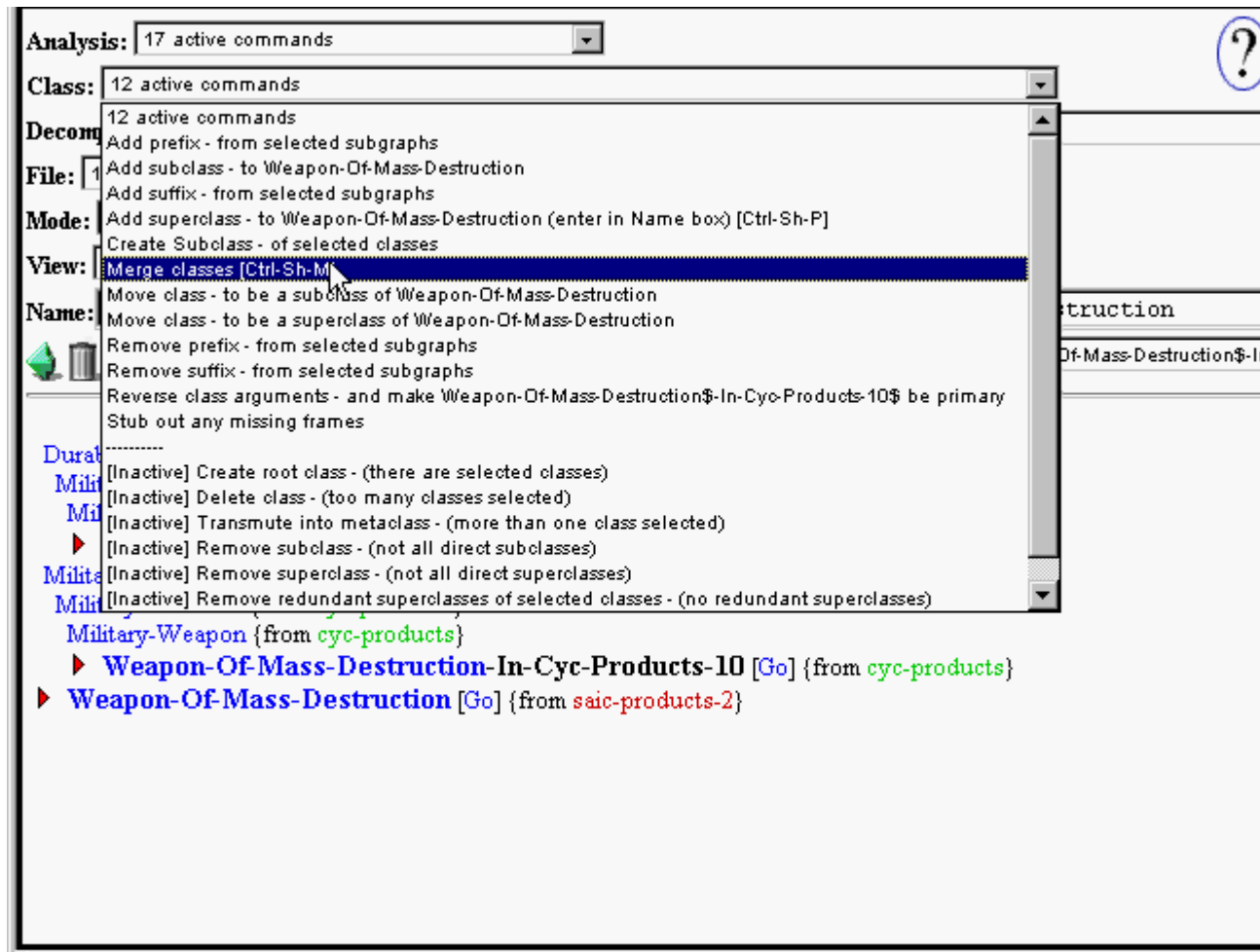
**Name:** WEAPON-OF-MASS-DESTRUCTION **Pretty name:** Weapon-Of-Mass-Destruction

**Names to resolve:** Shared the same name at load time: Weapon-Of-Mass-Destruction, Weapon-Of-Mass-Destruction-In-

Durable-Goods {from cyc-products}  
 Military-Hardware {from cyc-products}  
 Military-Weapon {from cyc-products}  
 ▶ **Weapon-Of-Mass-Destruction-In-Cyc-Products-10** [Go] {from cyc-products}  
 Military-Equipment {from cyc-products}  
 Military-Hardware {from cyc-products}  
 Military-Weapon {from cyc-products}  
 ▶ **Weapon-Of-Mass-Destruction-In-Cyc-Products-10** [Go] {from cyc-products}  
 ▶ **Weapon-Of-Mass-Destruction** [Go] {from saic-products-2}

**Merge the two selected classes**

## Demo des Chimaera



## Demo des Chimaera

Analysis: 17 active commands

Class: 3 active commands


Decomposition: No active commands



File: 12 active commands KB arg

Mode: 12 active commands Numeric arg

View: 11 active commands

Name: Pretty name:



Names to resolve: Shared the same name at load time: Weapon-Of-Mass-Destruction, Weapon-Of-Mass-Destruction\$-In

2 classes merged into **Weapon-Of-Mass-Destruction** {from **cyc-products**, **saic-products-2**}

Durable-Goods {from **cyc-products**}

Military-Hardware {from **cyc-products**}

Military-Weapon {from **cyc-products**}

▶ **Weapon-Of-Mass-Destruction** {from **cyc-products**, **saic-products-2**}

Military-Equipment {from **cyc-products**}

Military-Hardware {from **cyc-products**}

Military-Weapon {from **cyc-products**}

▶ **Weapon-Of-Mass-Destruction** {from **cyc-products**, **saic-products-2**}

## Demo des Chimaera

### Reports

Analysis: Analyze all KB

Class: 3 active commands


Decomposition: No active commands

File: 12 active commands KB arg

Mode: 11 active commands Numeric arg

View: 10 active commands

Name:  Pretty name:


Taxonomy: cyc-products, saic-products-2: Weapon-Of-Mass-Destruction

Device-One-Time-Use {from cyc-products}

Explosive-Device {from cyc-products}

Bomb-In-Cyc-Products-4 {from cyc-products}

Mine-Weapon {from cyc-products}

Nuclear-Weapon {from cyc-products, saic-products-2}

Point-Detonating-Weapon {from cyc-products}

Pressure-Sensitive-Detonating-Weapon {from cyc-products}

Proximity-Detonating-Weapon {from cyc-products}

Mine-Weapon {from cyc-products}

Durable-Goods {from cyc-products}

Military-Hardware {from cyc-products}

Aircraft-Weapon-Pylon {from cyc-products}

Military-Weapon {from cyc-products}

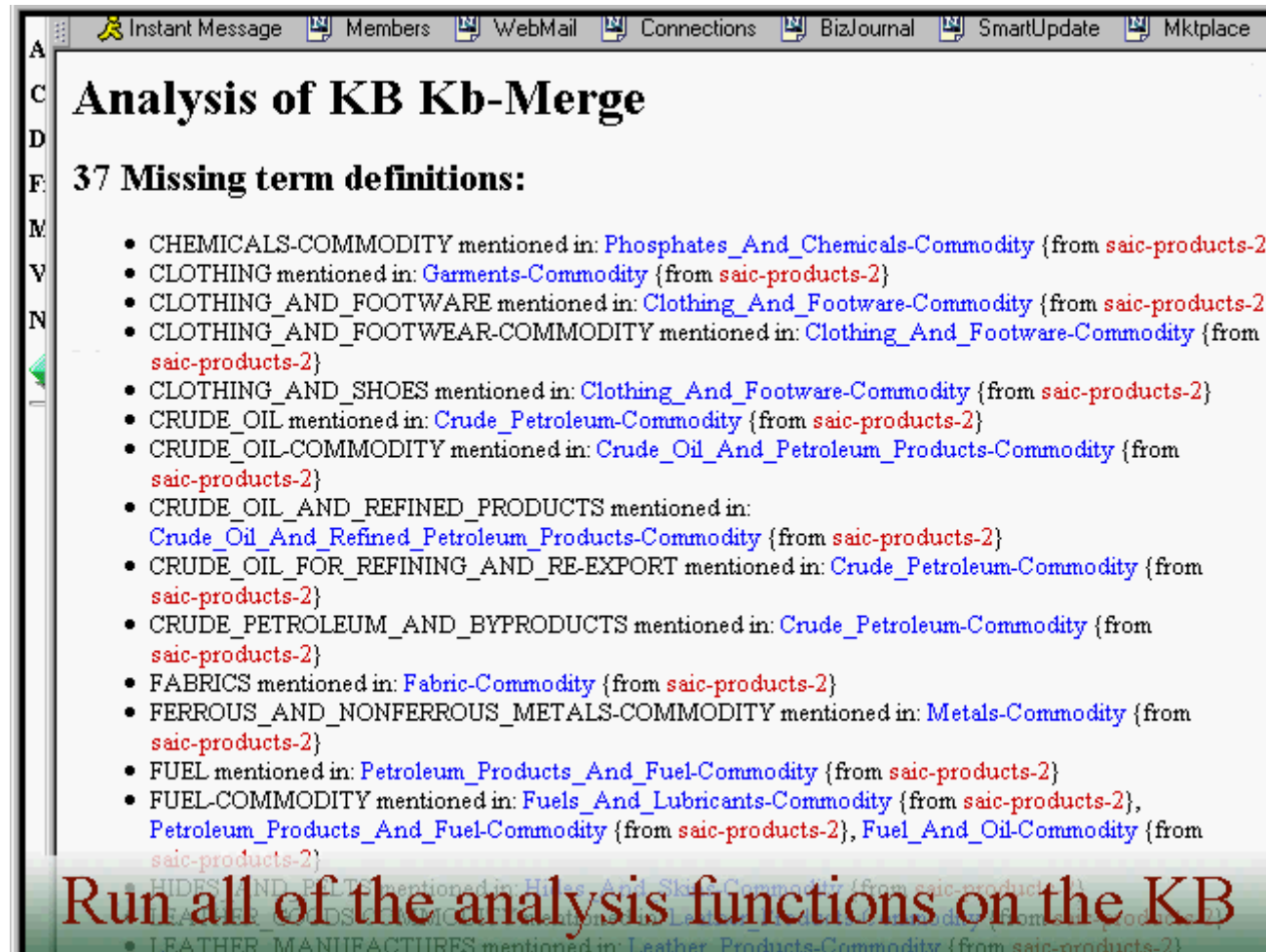
Air-Defense-Weapon {from cyc-products}

Anti-Armor-Weapon {from cyc-products}

Anti-Tank-Guided-Weapon {from cyc-products}

Run all of the analysis functions on the KB

## Demo des Chimaera



The screenshot shows a window titled "Analysis of KB Kb-Merge" with a menu bar containing: Instant Message, Members, WebMail, Connections, BizJournal, SmartUpdate, and Mktplace. The main content area displays a list of 37 missing term definitions. Each item consists of a term followed by its source in curly braces. The terms are color-coded: blue for the term name and red for the source identifier "saic-products-2".

**Analysis of KB Kb-Merge**

**37 Missing term definitions:**

- CHEMICALS-COMMODITY mentioned in: [Phosphates\\_And\\_Chemicals-Commodity](#) {from **saic-products-2**}
- CLOTHING mentioned in: [Garments-Commodity](#) {from **saic-products-2**}
- CLOTHING\_AND\_FOOTWEAR mentioned in: [Clothing\\_And\\_Footware-Commodity](#) {from **saic-products-2**}
- CLOTHING\_AND\_FOOTWEAR-COMMODITY mentioned in: [Clothing\\_And\\_Footware-Commodity](#) {from **saic-products-2**}
- CLOTHING\_AND\_SHOES mentioned in: [Clothing\\_And\\_Footware-Commodity](#) {from **saic-products-2**}
- CRUDE\_OIL mentioned in: [Crude\\_Petroleum-Commodity](#) {from **saic-products-2**}
- CRUDE\_OIL-COMMODITY mentioned in: [Crude\\_Oil\\_And\\_Petroleum\\_Products-Commodity](#) {from **saic-products-2**}
- CRUDE\_OIL\_AND\_REFINED\_PRODUCTS mentioned in: [Crude\\_Oil\\_And\\_Refined\\_Petroleum\\_Products-Commodity](#) {from **saic-products-2**}
- CRUDE\_OIL\_FOR\_REFINING\_AND\_RE-EXPORT mentioned in: [Crude\\_Petroleum-Commodity](#) {from **saic-products-2**}
- CRUDE\_PETROLEUM\_AND\_BYPRODUCTS mentioned in: [Crude\\_Petroleum-Commodity](#) {from **saic-products-2**}
- FABRICS mentioned in: [Fabric-Commodity](#) {from **saic-products-2**}
- FERROUS\_AND\_NONFERROUS\_METALS-COMMODITY mentioned in: [Metals-Commodity](#) {from **saic-products-2**}
- FUEL mentioned in: [Petroleum\\_Products\\_And\\_Fuel-Commodity](#) {from **saic-products-2**}
- FUEL-COMMODITY mentioned in: [Fuels\\_And\\_Lubricants-Commodity](#) {from **saic-products-2**}, [Petroleum\\_Products\\_And\\_Fuel-Commodity](#) {from **saic-products-2**}, [Fuel\\_And\\_Oil-Commodity](#) {from **saic-products-2**}
- HIDES\_AND\_SKIN mentioned in: [Hides\\_And\\_Skin-Commodity](#) {from **saic-products-2**}
- LEATHER mentioned in: [Leather\\_Products-Commodity](#) {from **saic-products-2**}
- LEATHER\_MANUFACTURES mentioned in: [Leather\\_Products-Commodity](#) {from **saic-products-2**}

**Run all of the analysis functions on the KB**

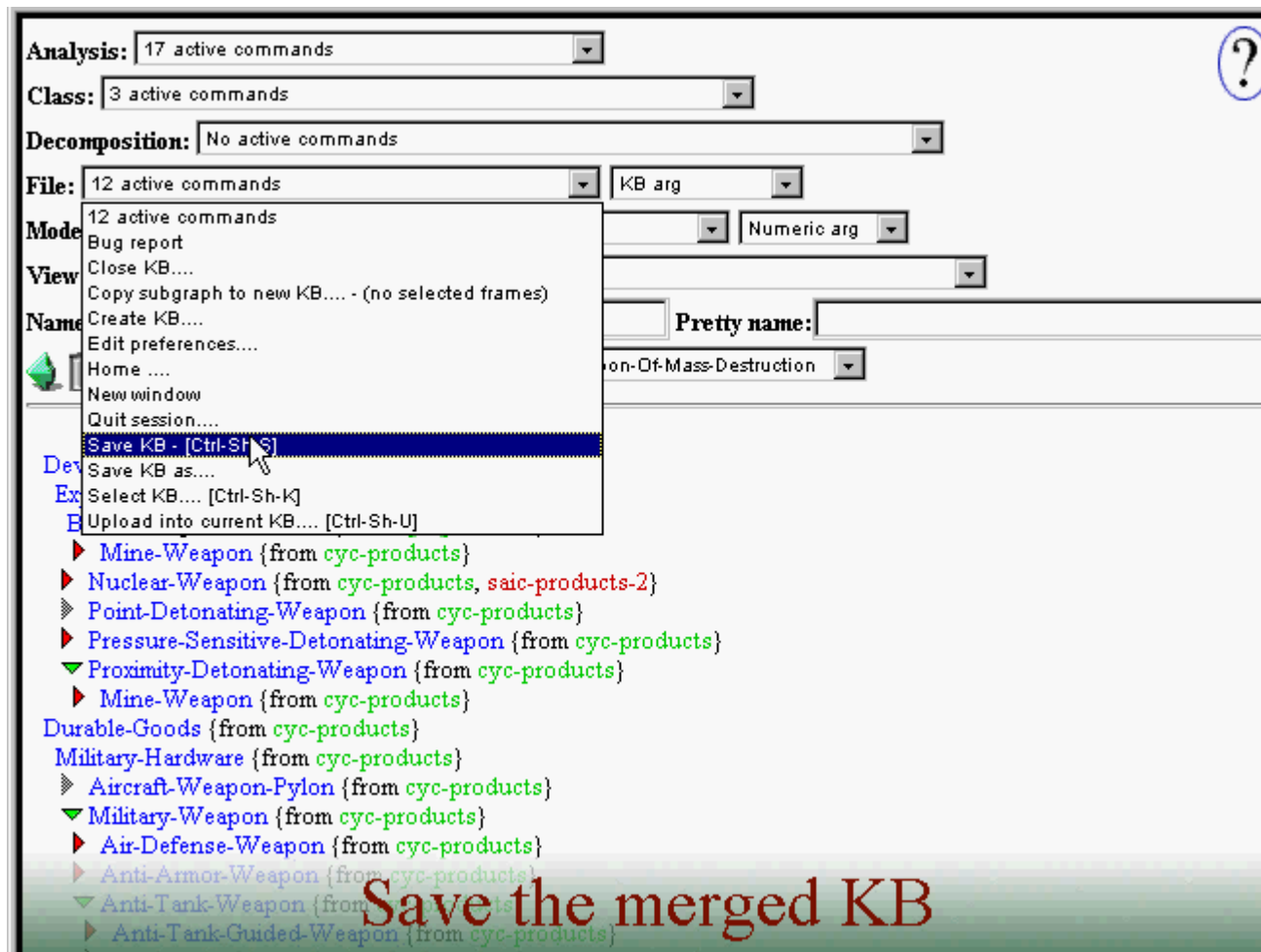
## Demo des Chimaera

### 132 Classes with redundant direct superclasses:

- **Warplane** {from **cyc-products**, **saic-products-2**}, has redundant direct superclasses: **Weapon** {from **saic-products-2**} because it is a superclass of **Conventional-Weapon** {from **saic-products-2**}
- **Torpedo** {from **saic-products-2**}, has redundant direct superclasses: **Weapon** {from **saic-products-2**} because it is a superclass of **Projectile-Weapon** {from **saic-products-2**}
- **Tank-Vehicle** {from **cyc-products**, **saic-products-2**}, has redundant direct superclasses: **Weapon** {from **saic-products-2**} because it is a superclass of **Armored-Combat-Vehicle** {from **saic-products-2**}, **Land-Transportation-Device** {from **cyc-products**, **saic-products-2**} because it is a superclass of **Armored-Combat-Vehicle** {from **saic-products-2**}, **Armored-Combat-Vehicle** {from **saic-products-2**} because it is a superclass of **Afv** {from **saic-products-2**}, **Transportation-Device-Vehicle** {from **cyc-products**, **saic-products-2**} because it is a superclass of **Armored-Combat-Vehicle** {from **saic-products-2**}
- **Small-Arms** {from **saic-products-2**}, has redundant direct superclasses: **Weapon** {from **saic-products-2**} because it is a superclass of **Conventional-Weapon** {from **saic-products-2**}
- **Oghab-Missile** {from **saic-products-2**}, has redundant direct superclasses: **Thing** {from **saic-products-2**} because it is a superclass of **Ballistic-Missile** {from **saic-products-2**}
- **Molotov-Cocktail** {from **saic-products-2**}, has redundant direct superclasses: **Thing** {from **saic-products-2**} because it is a superclass of **Bomb** {from **saic-products-2**}
- **Modern-Naval-Ship** {from **cyc-products**, **saic-products-2**}, has redundant direct superclasses: **Weapon** {from **saic-products-2**} because it is a superclass of **Conventional-Weapon** {from **saic-products-2**}
- **Missile** {from **saic-products-2**}, has redundant direct superclasses: **Weapon** {from **saic-products-2**} because it is a superclass of **Projectile-Weapon** {from **saic-products-2**}
- **Machine-Gun** {from **saic-products-2**}, has redundant direct superclasses: **Weapon** {from **saic-products-2**} because it is a superclass of **Conventional-Weapon** {from **saic-products-2**}
- **Ballistic-Missile** {from **saic-products-2**}, has redundant direct superclasses: **Thing** {from **saic-products-2**} because it is a superclass of **Surface-To-Surface-Missile** {from **saic-products-2**}
- **Artillery-Shell** {from **saic-products-2**}, has redundant direct superclasses: **Weapon** {from **saic-products-2**} because it is a superclass of **Projectile-Weapon** {from **saic-products-2**}
- **Anthrax Weapon** {from **saic-products-2**}, has redundant direct superclasses: **Biological-Weapon** {from **cyc-products**, **saic-products-2**} because it is a superclass of **Microorganism-Biological-Weapon** {from **saic-products-2**}
- **Air-Defense-Gun** {from **saic-products-2**}, has redundant direct superclasses: **Weapon** {from **saic-products-2**}



## Demo des Chimaera



## Zusammenfassung

- Ontologien stellen semantische Informationen über
  - **Commonsense** Wissen (Zeit, Raum, ...) und über
  - **Anwendungsbereiche** bereit
- Ontologien geben Hinweis auf **relevante** Begriffe, Attribute, Beziehungen beim Aufbau domänenspezifischer Wissensbasen bzw. konzeptueller Schemabeschreibungen
  - Beziehungen führen zu **weiteren** relevanten Begriffen
  - Attribute innerhalb einer Begriffsdefinition kennzeichnen **zusätzliche relevante Merkmale** für domänenspezifischen Begriffen
- Ontologien stellen Grundlage für semantisch-basierte Kommunikation bereit und sind deswegen auch relevant im Kontext von **WWW-Suchmaschinen, E-Commerce, Web Applikationen**
- Konstruktion einer Ontologie geschieht über Kombination einer Vielzahl anderer Ontologien
- Merging von Ontologien ist aktuelles Forschungsgebiet