AIFB

Intelligent Systems on the World Wide Web

Applications

Lecture Slides
Marc Ehrig
Steffen Staab, Raphael Volz
Institute for Applied Computer Science and Formal Description Methods (AIFB)
Karlsruhe University

AIFB

Inhalt

- RDF – Friend of a Friend
- RDF Schema - Bibster
- OWL – Web Service Matching
- F-Logic – Halo Chemistry

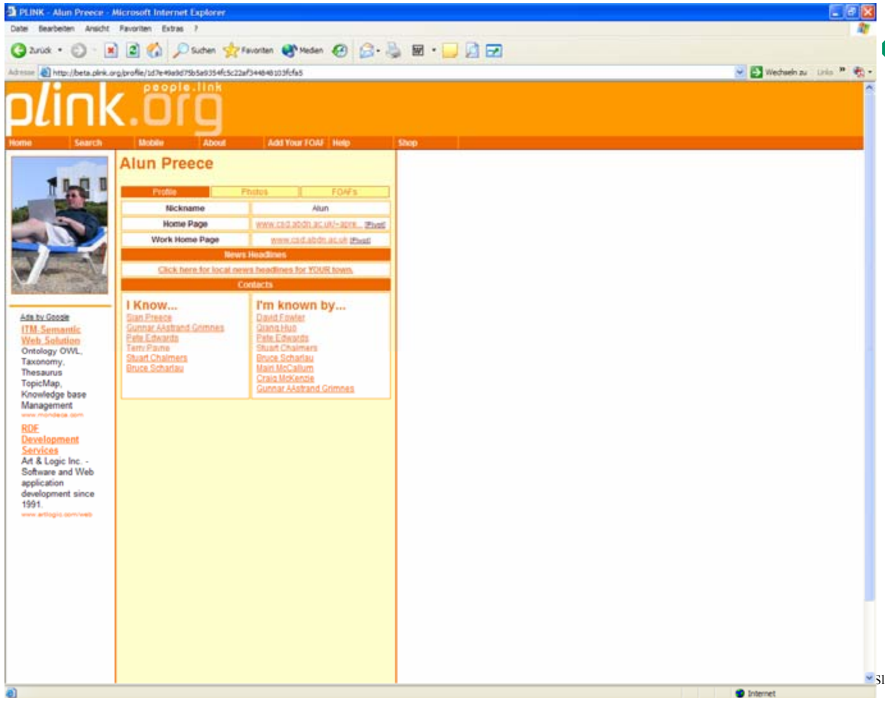
AIFB

RDF - FOAF

AIFB

FOAF File example

```
<!-- imagine you're Alice and this is the markup in your FOAF about you -->
<foaf:Person>
   <foaf:nick>Alice</foaf:nick>
   <foaf:knows rdf:nodeID="bob"/> <!-- a pointer to the 'bob' entry below -->
        <foaf:made rdf:resource=""/> <!-- this says: I made this FOAF file -->
        <foaf:made rdf:resource="http://alice.example.com/why-bob-is-
   great.html"/>
</foaf:Person> <!-- some markup describing a document, and its topic -->
<foaf:Document rdf:about="http://alice.example.com/why-bob-is-great.html">
   <dc:title>A page all about Bob, by Alice</dc:title> <foaf:topic> <!-- here is a
   chunk of data about bob --> <foaf:Person rdf:nodeID="bob">
   <foaf:nick>Bob</foaf:nick> <!-- other stuff about bob could go here -->
   </foaf:Person> </foaf:topic> </foaf:Document>
```



AIFB

RDF + RDFS - Bibster

AIFB

Agenda

1. Introduction:
   The Bibliographic Scenario

2. The Bibster System
   System Architecture
   Semantic Methods in Bibster

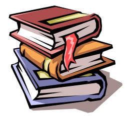
3. Evaluation

AIFB

1. Introduction: Scenario

- **Scenario**: Sharing of bibliographic metadata in a Peer-to-Peer network
  - Bibliographic metadata is created and maintained in a **decentralized** manner,
    centralized solution not applicable
  - Researchers are willing to share their data
  - Use of semantics is crucial in this setting

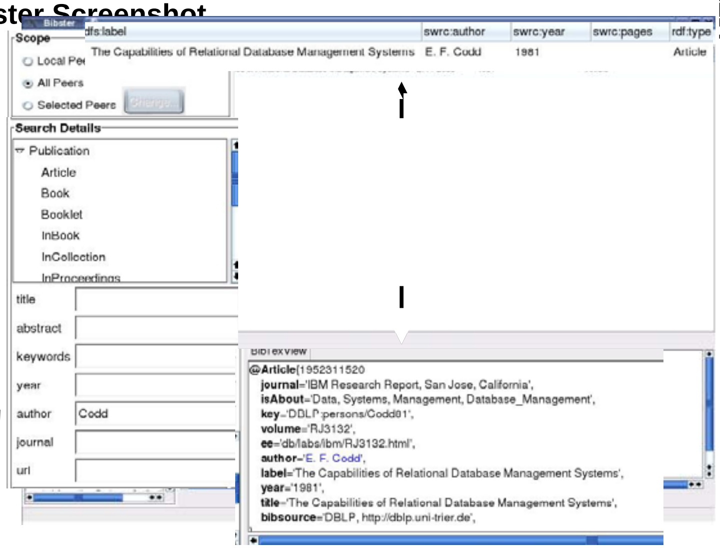
- The Bibster system allows to:
  - **Easily share** bibliographic data
  - Save work in finding this data
  - Avoid re-typing this data by hand



2. Bibster Screenshot

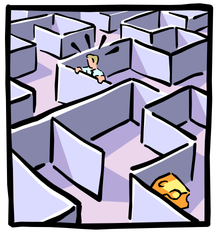
Query Scope

Semantic Search



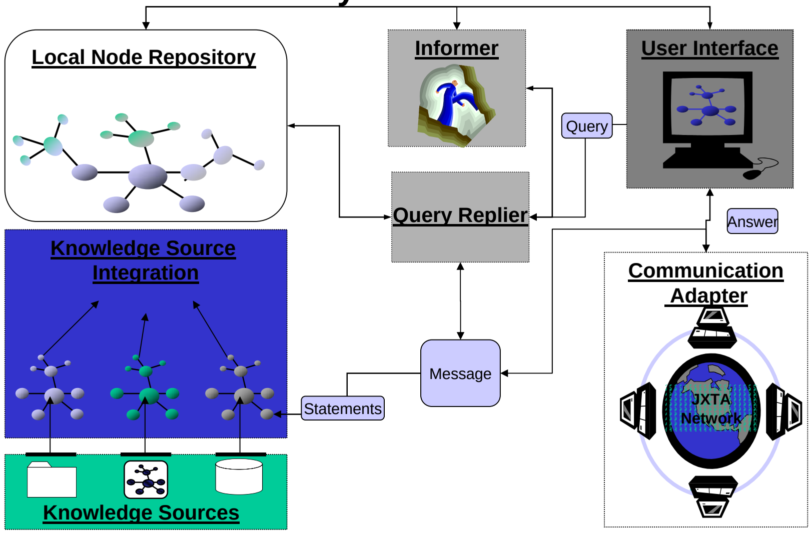
2. Semantic Methods in Bibster

- Semantic representation and querying of **metadata**
  - Extraction and classification from e.g. BibTeX files
  - **Semantic Web Research Community Ontology** and **ACM Topic hierarchy** as light-weight ontologies



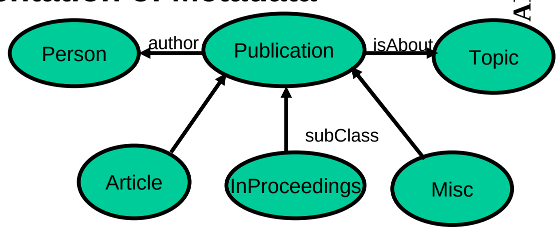
- **Peer selection** using semantic topologies
  - Scalability requires intelligent query routing
  - Semantic descriptions of peers´ expertise to build semantic topologies as basis for peer selection
- Semantic **duplicate detection**
  - Highly redundant and inconsistent representation of bibliographic metadata
  - Semantic similarity measures to detect duplicates

2. Bibster / SWAP System Architecture



Semantic Representation of Metadata

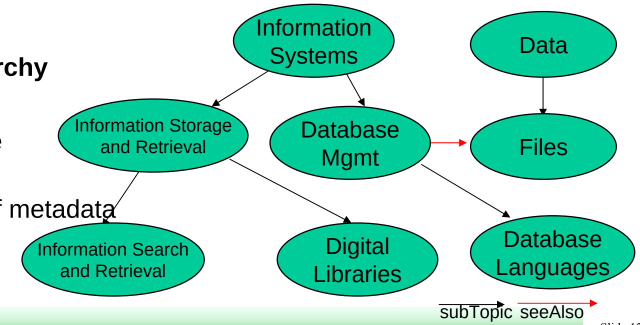
SWRC as base ontology: models domain of research community



ACM Topic hierarchy

1287 topics from Computer Science Domain

for classification of metadata (linked by hasSubtopic)



AIFB

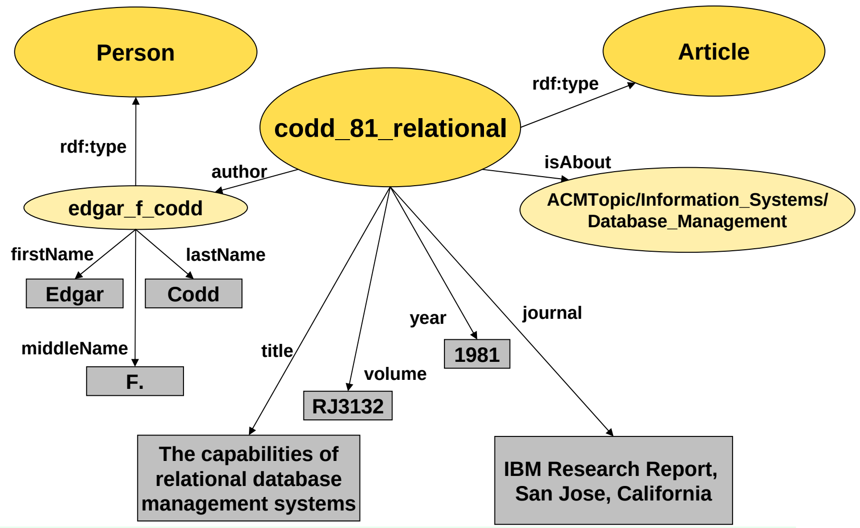
Sample BibTeX Entry

```
@ARTICLE{codd81relational,
 author = {Edgar F. Codd},
 title = {The capabilities of relational database
   management systems},
 journal = {IBM Research Report, San Jose, California},
 volume = {RJ3132},
 year = {1981}
}
```


© S. Staab, 2003

AIFB

Sample Entry

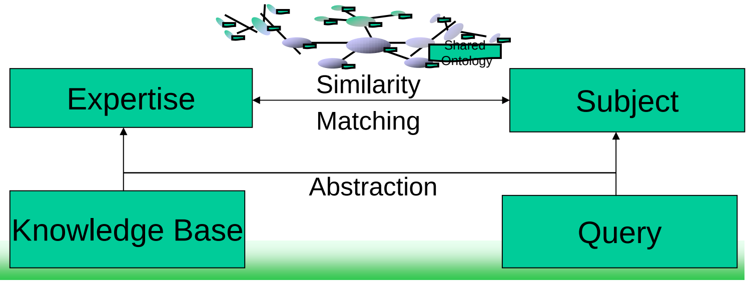


© S. Staab, 2003

AIFB

Expertise-Based Peer Selection

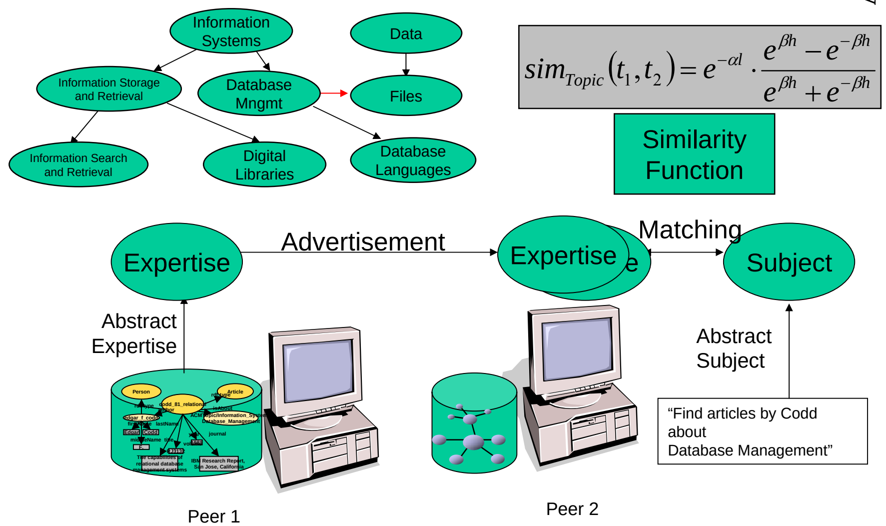
- **Advertisements** to promote semantic descriptions of expertise
  in the network
- **Semantic topology:** formed by the knowledge about other peers' expertise, peers with similar expertise are clustered
- **Peer Selection**: ranks peers according to similarity between their expertise and query subject



© S. Staab, 2003

AIFB

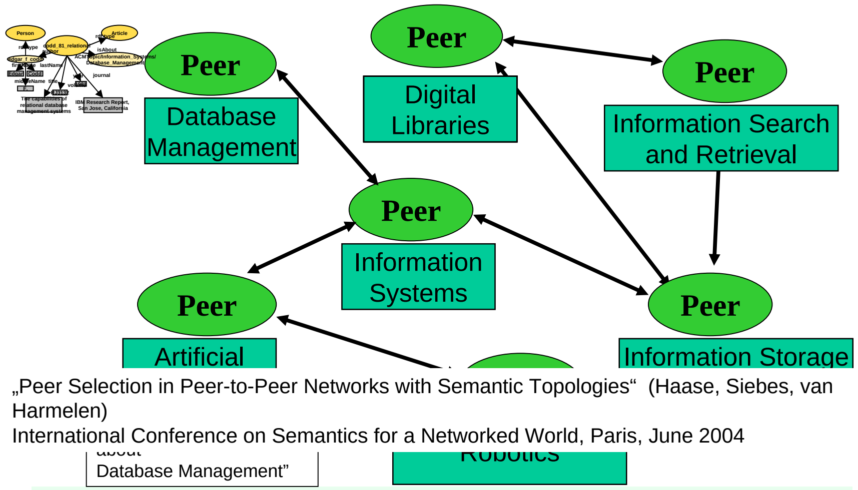
Expertise-Based Peer Selection



$$sim_{Topic}(t_1,t_2)=e^{-\alpha l}\cdot\frac{e^{\beta h}-e^{-\beta h}}{e^{\beta h}+e^{-\beta h}}$$

AIFB

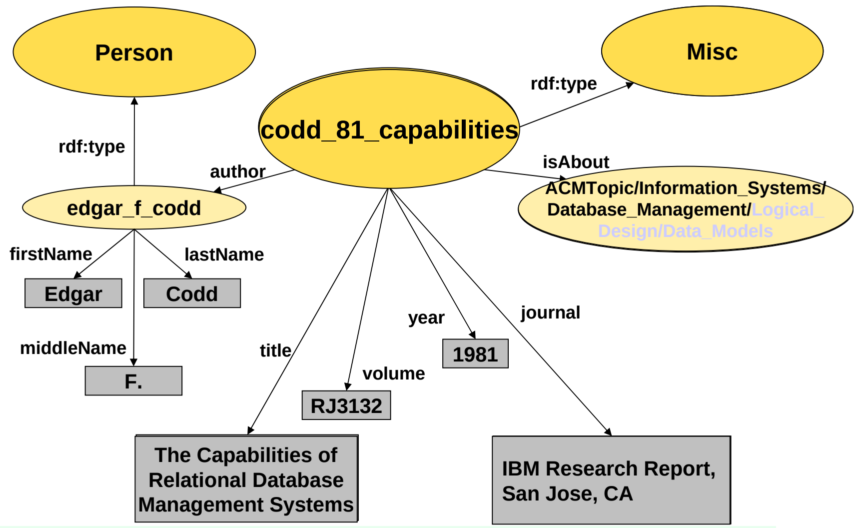
Semantic Topologies



„Peer Selection in Peer-to-Peer Networks with Semantic Topologies" (Haase, Siebes, van Harmelen)
International Conference on Semantics for a Networked World, Paris, June 2004

AIFB

Duplicate Entry



AIFB

Semantic Duplicate Detection

- Individual similarity functions
  - Syntactic: Levenshtein similarity for titles etc.
  - Graph Structure: Sequence of authors
  - Ontological Structure: ACM Topic hierarchy
  - Domain Knowledge: e.g. Type Misc often corresponds to Unknown
- Aggregated similarity function:
  - e.g. weighted average
- Duplicates: Entries with similarity above a specified threshold
- Clusters of duplicate entries
- Merging of entries based on heuristics

AIFB

3. Evaluation

- We arrange a case study with Bibster to
  - evaluate the scalability, functionality, and the performance of a semantics based Peer-to-Peer system
  - validate results from simulation experiments
- How do we evaluate?
  - User queries and routing information will be logged for evaluation purposes
  - After the case study we will provide a user questionnaire to evaluate the usability of our system
- Evaluation is open to public:
  You may join the case study team and work with Bibster
  - We will provide full support during the case study (May – July 04)
  - See **http://bibster.semanticweb.org/**

© S. Staab, 2003

AIFB

4. Conclusion

- Fully implemented semantic P2P system for the bibliographic domain
- Exploitation ontologies in all steps:
  - Importing and representing the data
  - Querying the data
  - Routing requests
  - Integrating heterogeneous results
- Evaluation using simulation experiments and real-life case study

© S. Staab, 2003

AIFB

OWL – Web Service Matching

Courtesy of Stephan Grimm & Boris Motik

© S. Staab, 2003

AIFB

Description

© S. Staab, 2003

AIFB

Overview

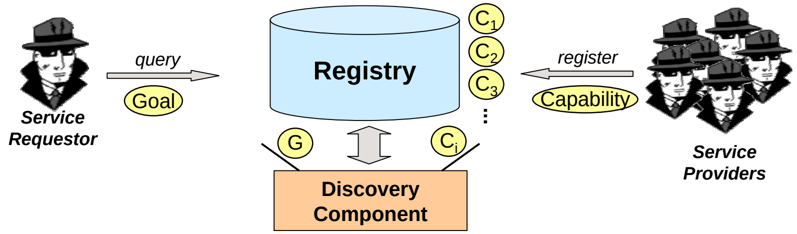
- Discovery **Environment**
- Service **Description** w.r.t. Discovery
- **Matching** for Discovery
- **Outlook** on future work

Environment

for Discovery

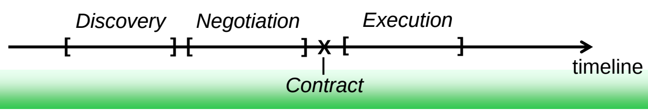
Notion of Discovery in SWWS

- Discovery :
  Task of locating services that meet a requestor's needs
- Base Discovery on descriptions of business-level service semantics
- Architecture



Discovery in contrast to Negotiation

- Pre-contractual phases
  - Discovery
    - **outcome is just the fact that Request and Capability are compatible**
    - **does not directly lead to a concerte Service Instance**
  - Negotiation
    - **optimally negotiate the parameters of the service to be carried out**
    - **leads to one concrete Service Instance, if successful**



Description

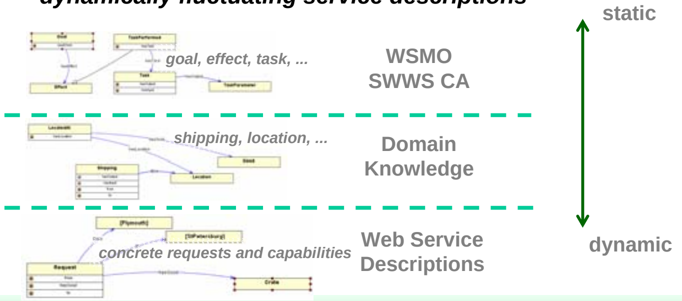
with respect to Discovery

Approaches for using upper-level service ontologies

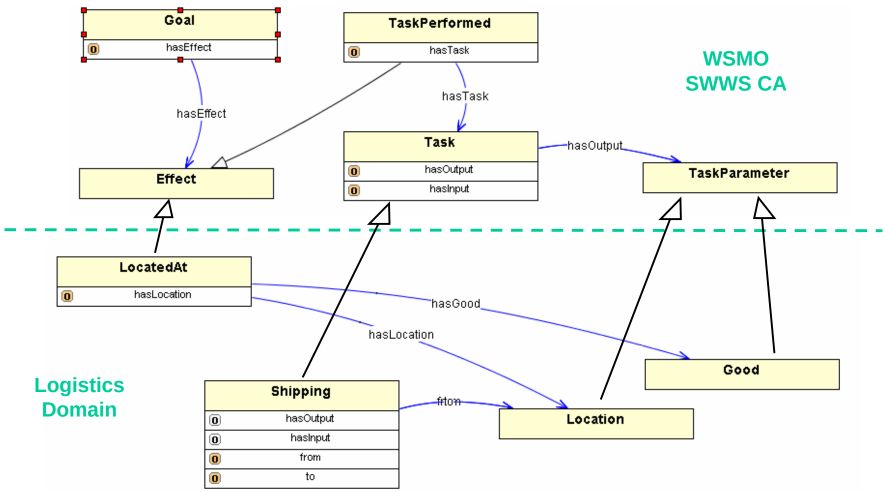
- Consider an upper-level ontology for Web Services
- Two different approaches for modelling SWS w.r.t. discovery:
  - Model service descriptions on the level of concepts & relations and base discovery on subsumption or disjointness checks
  - Model service descriptions on the level of instances and base discovery on querying concept extensions

3-layer ontological architecture

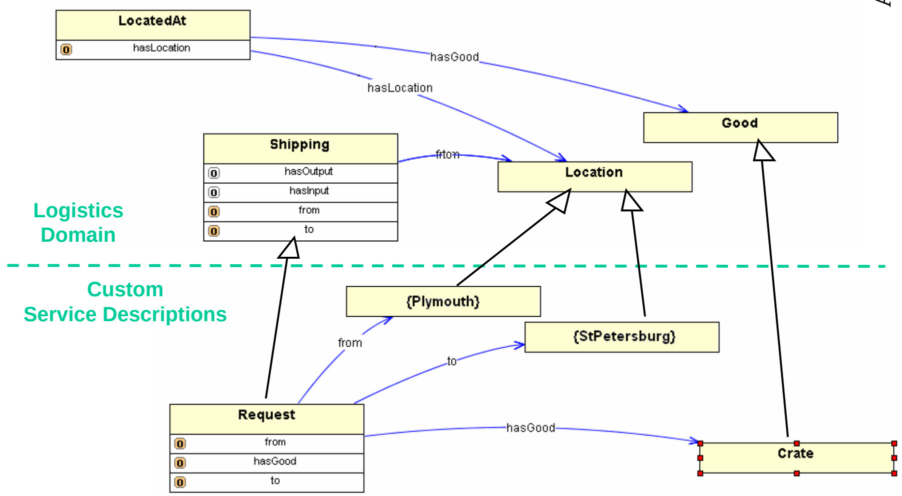
- 3 layers for ontological description of Web Services
  - Upper layer
    - ***top-level Web Service concepts WSMO, SWWS-CA***
  - Domain layer
    - ***static domain-specific knowledge modellers of requests and capabilities refer to***
  - Custom description layer
    - ***dynamically fluctuating service descriptions***



Upper layer ⇒ Domain layer



Domain ⇒ Custom Web Service descriptions



Procedure of modeling WS semantics

- Initially
  - Start out with upper-level ontology for Web Services
  - Align domain-ontological concepts and relation on this upper-level ontology by means of subsumption
- When offering or requesting WS capabilities
  - Use concepts and relations on the domain-level to describe the semantics of the desired service
  - Possibly introduce new domain-level concepts and associate them to already existing ones via subsumption relations
  - Specify what task(s) should be performed by the service in terms of domain-level concepts
  - Further restrict properties of the desired service

Example of Service Descriptions in DL

Goal ⊑ ∃hasEffect.TaskPerformed

TaskPerformed ⊑
Effect ⊓
= 1 hasAction ⊓
∀ hasGood.Crate
. . .

Upper layer

Shipping ⊑ Transportation
Shipping ⊑ ∀from.Location
Shipping ⊑ ≤ 1 from
. . .

Domain Knowledge

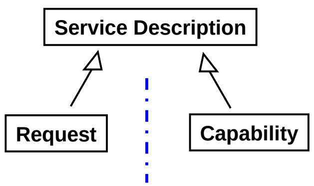
Request ≡
Shipping ⊓
∃from.{Plymouth} ⊓
∃to.{StPetersburg} ⊓
= 1 hasGood ⊓
∀ hasGood.Crate ⊓
. . .

Capability_A ≡
Transportation ⊓
∃from.UKCity ⊓
∃to.EUCity ⊓
∀hasGood.(Good ⊓
¬HazardousGood) ⊓
. . .

Custom Web Service Descriptions

Symmetry of Capabilities and Requests

- Requests and Capabilities are symmetric specializations of Service Description
- Both equally describe desired service instances
- The only difference lies in the role they play
  - Request originates from the service consumer
  - Capability originates from the service provider

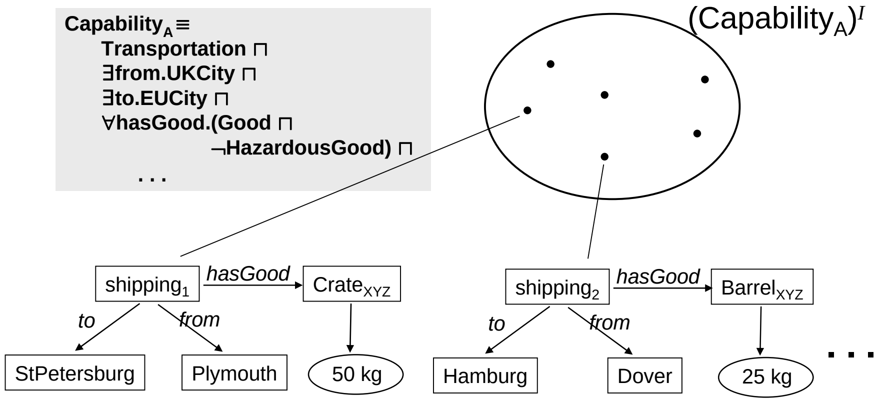


Service Instances

- Finally a concrete ***Service Instance*** is being performed
- Service Descriptions are templates for Service Instances
- A Service Description (potentially) allows several Service Instances which are accepted by the provider/requestor
- This introduces variance that has to be controlled in Service Descriptions

Variance in Service Descriptions

- A Service Description allows several Service Instances



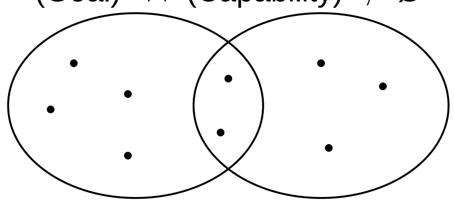
Matching

of Requests and Capabilities to perform the task of Discovery

Idea of Matching for Discovery

- Solving the Discovery task by matching Requests and Capabilities
- Request and Capability allow common Service Instances

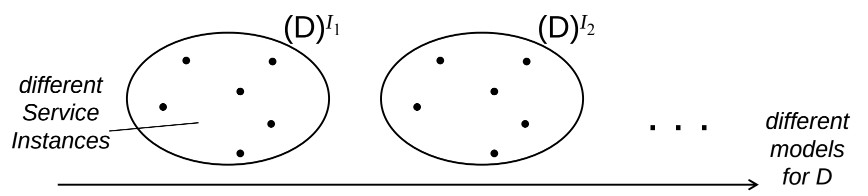
$$(\text{Goal})^I \cap (\text{Capability})^I \neq \emptyset$$



- Requestor and Provider can (potentially) do business with each other

Variance in Service Descriptions

- Use DL-formalism for description and matching
- Variance in Service Descriptions maps to variance in model-theoretic interpretations of DL concepts
- Two kinds of variance for a Service Description D :



Don‘t know vs don‘t care

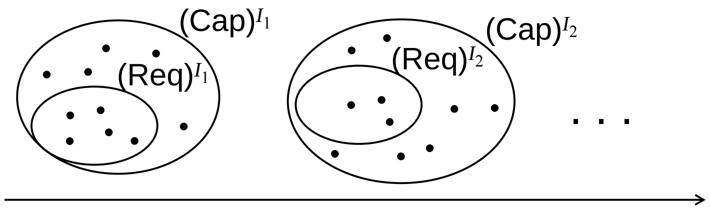
DL Inferences for Matching

- Assume a T-Box *KB* containing domain-level facts, and a matching criteria α
- Resolving variance in different Service Instances
  - Subsumption $\alpha = D_1 \sqsubseteq D_2$
  - Non-Disjointness $\alpha = D_1 \sqcap D_2 \not\sqsubseteq \bot$
- Resolving possible world variance
  - Entailment $T \vDash \alpha$
  - Satisfiability $T \cup \{\alpha\}$ is coherent

Subsumption

- Entailment of concept Subsumption

  $KB \vDash$ Request $\sqsubseteq$ Capability



- (Request)$^I \subseteq$ (Capability)^I in every possible world

Subsumption

- Characteristic of Subsumption
  - Strong requirement

    (Request ⊓ ¬Capability) is empty given *KB*

    i.e. *KB* ⊔ i:(Request ⊓ ¬Capability) is insatisfiable

KB :
CreditCard ⊑ EPayment
Shipping ⊑ ≤ 1 from
Plymouth : UKCity
. . .

Request ≡
Shipping ⊓
∃from.{Plymouth} ⊓
∀payment.EPayment ⊓
. . .

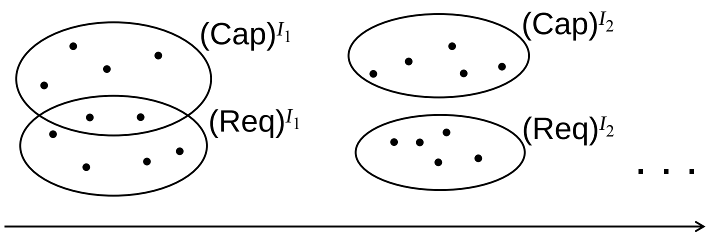
Capability ≡
Shipping ⊓
∃from.UKCity ⊓
∃payment.CreditCard ⊓
. . .

no match !

Intersection

- Satisfiability of concept Conjunction

  (Request ⊓ Capability) is satisfiable w.r.t. *KB*



- (Request)$^I \cap$ (Capability)$^I \neq \varnothing$ in <u>some</u> possible world

AIFB

Intersection - deficiencies

- Characteristic of Intersection
  - Weak requirement

(Request ⊓ Capability) is satisfiable w.r.t. *KB*

KB :
UKCity ⊑ City
EUCity ⊑ City
Shipping ⊑ ≤ 1 from
. . .

Request ≡
Shipping ⊓
∃from.UKCity ⊓
. . .

Capability ≡
Shipping ⊓
∃from.USCity ⊓
. . .

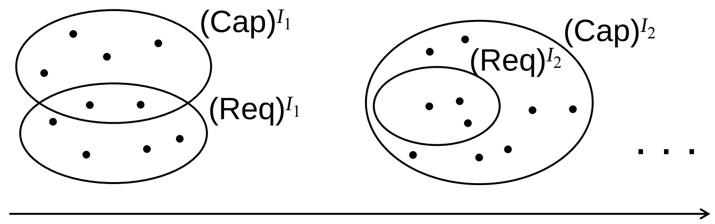
false positive match !

AIFB

Non-Disjointness

- Entailment of concept Non-Disjointness

KB ⊨ Request ⊓ Capability ⋢ ⊥



- (Request)^{I} ∩ (Capability)^{I} ≠ Ø in every possible world

AIFB

Non-Disjointness - deficiencies

- Characteristic of Non-Disjointness
  - Rather strong

KB ⊨ Request ⊓ Capability ⋢ ⊥

KB :
⊤ ⊑ ≤ 1 payment
Shipping ⊑ = 1 payment
. . .

Request ≡
Shipping ⊓
∀payment.(BankTransfer ⊔ CreditCard) ⊓
. . .

Capability ≡
Shipping ⊓
∃payment.(CreditCard ⊔ Cash) ⊓
. . .

no match !

AIFB

F-Logic - Project Halo

Overview

AIFB

Project
- Scenario / Participants

Technical Approach
- Performance strategies
- Metareasoning / Justifications

Encoding / Knowledge Base
- Encoding Method
- Architecture of the KB

Challenge
- Results
- Question encoding, fidelity
- Failures, brittleness

Project

AIFB

- task:
  development of a query answering system containing the knowledge about 50 pages of chemistry book
- evaluation:
  - sequestration of system
  - 100 novel questions (AP exam)
  - encoding of the questions
  - chemistry professors graded results
- participants: SRI, Cyc Corp, Ontoprise/Unikarl

Overview

AIFB

Project
- Scenario / Participants

Technical Approach
- Performance
- Metareasoning / Justifications

Encoding / Knowledge Base
- Encoding Method
- Architecture of the KB

Challenge
- Results
- Question encoding, fidelity
- Failures, brittleness

Operationalization - F –logic

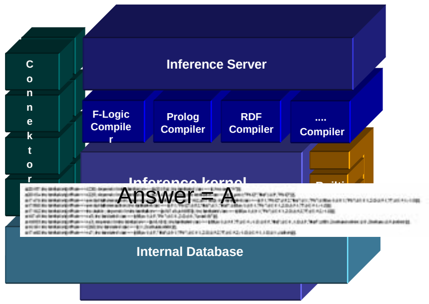
AIFB

dynamic filtering, M. Kifer

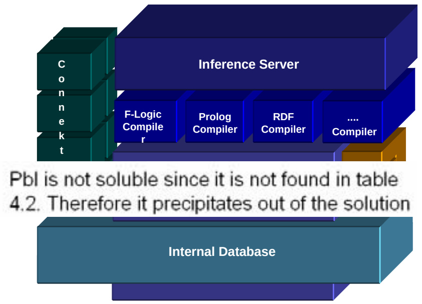
- set-oriented evaluation (vs. tuple-oriented)
- mix of top-down and bottom-up
  - **top-down propagating of constants restricts result set**
  - **bottom-up generation of results (join-operations)**

Metareasoning & Answer Justification

FORALL R,P <- R:PrecipitationReaction[precipitate->>P]



knowledge base



explanation knowledge base

Syllbus question no. 35 inferencing process

FORALL Answer <- exists R,P unify(Answer,"A")
and R:PrecipitationReaction[fromMixture->>m1; precipitate->>P] and color(P,yellow).

F1=PbI2, F2=Na, X=Pb, M=m1, K1=Pb, CK1=2.0, A1=I, CA1=-1.0

M=m1, AF=I, CA=-1.0, KF=Na, CK=1.0, F=NaI, M1=Sodiumiodide,
K=Sodium, A=Iodide

FORALL M,F,K,A,M1,KF,AF,CK,CA ion(M,AF,CA,anion) and ion(M,KF,CK,cation)
<- M:Mixture[hasComponent->>F]
and M1:IonicMolecule[hasCation->>K; hasAnion->>A; hasFormula->>F]
and K:Cation[hasFormula->>KF;hasCharge->>CK]
and A:Anion[hasFormula->>AF;hasCharge->>CA]. true

Metareasoning & Answer Justification
Proof Tree

a20157:Instantiation[ofRule->>r230; dependsOnInstantiation->>{a20154}; instantiatedVar
a20154:Instantiation[ofRule->>r225; dependsOnInstantiation->>{a17476,a17560}; instant

a17476:Instantiation[ofRule->>precipitationreaction;
dependsOnInstantiation->>{a17162,a17460};
instantiatedVars->>{i(F1,"PbI2"),i(F2,"Na"),i(X,"Pb"),
i(M,m1),i(K1,"Pb"),i(CK1,2.0),i(A1,"I"),i(CA1,-1.0)}].

a17560:Instantiation[ofRule->>precipitationreaction; instantiatedVars->>{i(F1,"PbI2"),i(F2,
a17162:Instantiation[ofRule->>insoluble; dependsOnInstantiation->>{a16749,a16953}; ins
a16749:Instantiation[ofRule->>r45; instantiatedVars->>{i(M,m1),i(F,"Pb"),i(CK,2.0),i(K,"Le
a16953:Instantiation[ofRule->>r43; dependsOnInstantiation->>{a16181}; instantiatedVars
1.0),i(KF,"Na"),i(CK,1.0),i(F,"NaI"),i(M1,Sodiumiodide),i(K,Sodium),i(A,Iodide)}].
a16181:Instantiation[ofRule->>r260; instantiatedVars->>{i(X,Sodiumiodide)}].
a17460:Instantiation[ofRule->>r47; instantiatedVars->>{i(M,m1),i(F,"Na"),i(K1,"Pb"),i(CK1

Metareasoning & Answer Justification
Explanations

the products of this reaction are PbI2 and Na because PbI2
precipitates out of the solution

an ionic molecule consisting of cation Pb and anion I is
not known to be soluble and is thus guessed to be unsoluble

AIFB

Metareasoning & Answer Justification

- In the reaction above **H2PO4,HPO4** and **H2BO3,HBO3** are **conjugate acid-base pairs**. **H2BO3** and **H2PO4 are acting as acids**, and **HBO3** and **HPO4 are acting as bases**.
- if the **equilibrium** of the reaction is on the **right** then the **acid H2PO4 is stronger than the acid H2BO3** and the **base HBO3 is stronger than base HPO4**
  - HBO3 is the stronger base than HPO4 therefore the **equilibrium is on the right** (on the products side)
  - H2PO4 is the stronger acid than H2BO3 therefore the **equilibrium is on the right** (on the products side)
  - the **equilibrium constant** is **larger than 1** and thus the **equilibrium is on the right** (on the products side)

redundant (applies to the first two sub-items)

AIFB

Metareasoning & Answer Justification

Reasoning for generating explanations

- integrating additional knowledge into explanations
- generating abstractions
- avoiding redundancies
- considering context and user profile

AIFB

Overview

Project

- Scenario / Participants

Technical Approach

- Performance strategies
- Metareasoning / Justifications

Encoding / Knowledge Base

- Encoding Method
- Architecture of the KB

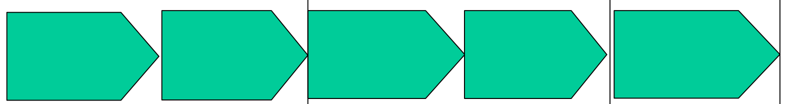
Challenge

- Results
- Question encoding, fidelity
- Failures, brittleness

AIFB

Encoding

Modelling procedure



Model ground operations

Verify ground operations by example questions from Brown *et al.*

40 test cases

Verify by syllabus questions

add explanation rules

refinement of modeling

refinement of explanations

testing

2002/2003

dry run

challenge run

Question encoding

A 0.3M solution of acetic acid has a pH of 2.63.
The ionization constant of this acid is

a) 1.8 x 10-5
b) 7.0 x 10-4
c) 1.1 x 10-6
d) 7.8 x 10-3
d) 1.9 x 10-6

Question encoding – multiple choice strategy

Input facts

```
m1:Mixture[hasComponents->>{"HCl","Ba(OH)2"}].
m2:Mixture[hasComponents->>{"HCl","CaCO3"}].
m3:Mixture[hasComponents->>{"HCl","CuSO4"}].
m4:Mixture[hasComponents->>{"HCl","Na3PO4"}].
m5:Mixture[hasComponents->>{"HCl","NaCl"}].
```

definition of alternatives

```
answer("A") <- exists P P:GaseousReaction[fromMixture->>m1].
answer("B") <- exists P P:GaseousReaction[fromMixture->>m2].
answer("C") <- exists P P:GaseousReaction[fromMixture->>m3].
answer("D") <- exists P P:GaseousReaction[fromMixture->>m4].
answer("E") <- exists P P:GaseousReaction[fromMixture->>m5].
```

ask for alternative

```
FORALL X <- answer(X).
```


Question encoding – multiple choice strategy

all possible answers

```
possibleanswer("A").
possibleanswer("B").
possibleanswer("C").
possibleanswer("D").
possibleanswer("E").
```

definition of wrong alternatives

```
FORALL X wronganswer("A") <- checkequation(ra,X).
FORALL X wronganswer("B") <- checkequation(rb,X).
FORALL X wronganswer("C") <- checkequation(rc,X).
FORALL X wronganswer("D") <- checkequation(rd,X).
FORALL X wronganswer("E") <- checkequation(re,X).
```

```
FORALL Answer <- possibleanswer(Answer) and not wronganswer(Answer).
```

exclusion of wrong alternatives

Question encoding – Detailed Answer Section

MC48

1.0 L of a buffer formed by mixing 0.25 moles of ammonia solution with
0.25 moles of ammonium nitrate has a pH of
(For ammonia, Kb = 1.8 x 10-5)

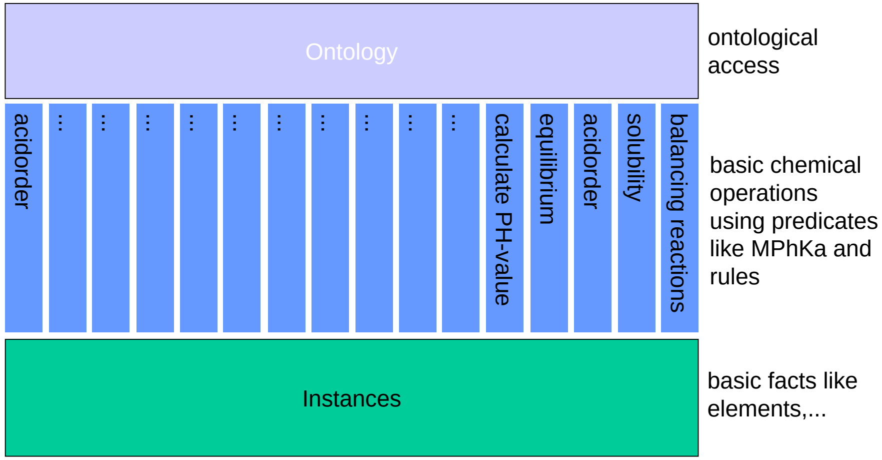
```
FORALL X,Y
m1:BufferSolution[hasComponents->>X;hasMole->>c(X,0.25);
hasVolume->>c(X,1.0)] and Y[hasKb->>1.8E-5]
        <- Y[hasName->>"Ammonia";hasFormula->>X].
FORALL X,Y
m1:BufferSolution[hasComponents->>X;hasMole->>c(X,0.25);
hasVolume->>c(X,1.0)]
        <- Y[hasName->>"ammonium nitrate";hasFormula->>X].

FORALL Ph <- m1[hasPHValue->>Ph].
```

Encoding - Basic Chemical Operations

Classify compound as ionic (aequous.flo, utils.flo)
Balance chemical equation (balancing.flo)
Determine solubility (acidbase.flo)
Determine equilibrium expression
Rank strength of metal ions as lewis acids
.....
.....
.....
Determine acid/base conjugate
Determine strengths of acids/bases
Determine products of reaction and type of reaction
Calculate PH (ph.flo)
Determine position of equilibrium
Describing substances
Naming

Architecture of KB



Example for Architecture

Strong Acid – Strong Base Titration (ch. 17.3, p. 671)

FORALL Acid,Base,AcidQuantity,AcidMole,BaseQuantity,BaseMole,
PH,OHMole,HMole,Volume,HAfterReaction,HConc,PH,A,B

strongacidstrongbasetitration(Acid,AcidQuantity,AcidMole, Base,BaseQuantity,BaseMole,PH)
<-
A:StrongAcid[hasFormula->>Acid] and B:WeakBase[hasFormula->>Base] and
multiply(BaseQuantity,BaseMole,OHMole) and
multiply(AcidQuantity,AcidMole,HMole) and
greater(HMole,OHMole) and
add(AcidQuantity,BaseQuantity,Volume) and
add(HAfterReaction,OHMole,HMole) and
multiply(HConc,Volume,HAfterReaction) and
phH(HConc,PH).

Example for Architecture

define OO - Wrapper

FORALL M,PH,F1,F2,Q1,Q2,M1,M2

M:Mixture[hasPHvalue->>PH] <-

M:Mixture[hasComponent->>{F1,F2}; hasQuantity->>c(F1,Q1); hasQuantity->>c(F2,Q2); hasMole->> c(F1,M1); hasMole->> c(F2,M2)]

and **strongacidstrongbasetitration (F1,Q1,M1,F2,Q2,PH).**

AIFB

Summary Architecture KB

- chemical operations:
  independent knowledge chunks
  - collaborative development of KB
  - reduce complexity
  - reduce testing effort

- OO – Wrapper: ontological access
  - eases access
  - closer to NL

AIFB

Overview

Project
- Scenario / Participants

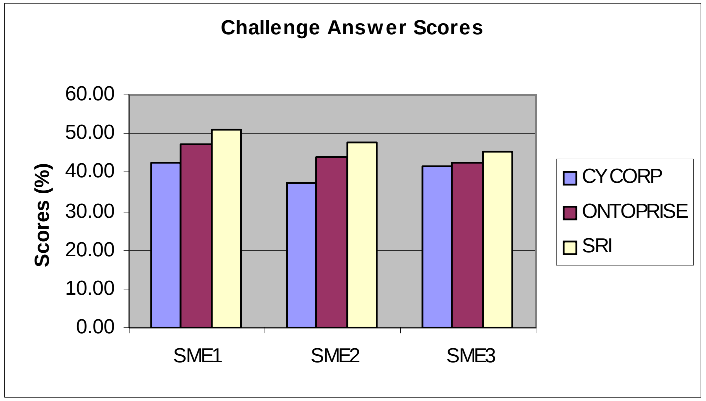
Technical Approach
- Performance
- Metareasoning / Justifications

Encoding / Knowledge Base
- Encoding Method
- Architecture of the KB

Challenge
- Results
- Question encoding, fidelity
- Failures, brittleness

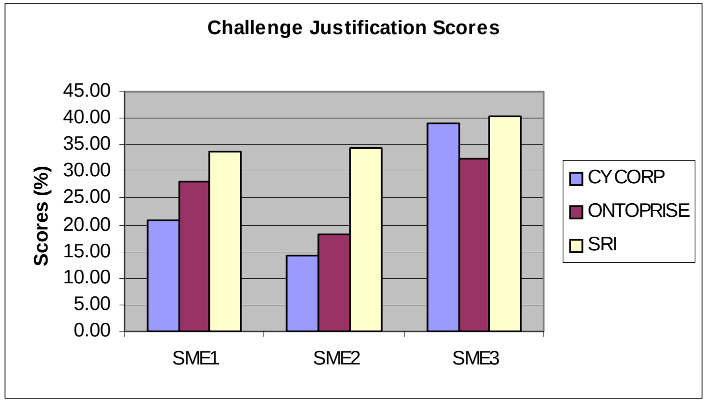
AIFB

Results



AIFB

Results



Performance

Team End-To-End Challenge Run Times		
Team	**Sequestered**	**Improved**
Cycorp	> 12 hours	> 27 hours
Ontoprise	2 hours	9 minutes
SRI	5 hours	38 minutes

Question encoding - fidelity

MC4

When calcium carbonate is heated it decomposes forming:
Calcium carbonate reacts with acids to produce gas

a) Solid Ca and CO2 gas
b) Gaseous CaCO and CO2 gas
c) Solid CaO and CO2 gas
d) Gaseous Ca and CO2
e) Solid CaO and liquid CO2

not encoded

B-MOD-1 or out of scope ?

Question encoding - fidelity

MC12

The spectator ions in the reaction of barium nitrate with sodium sulfate are:

a) Sodium ions and barium ions.
b) Sodium ions and sulfate ions.
c) Nitrate ions and sulfate ions.
e) Sodium ions and nitrate ions.
f) These are not ionic compounds so there are no spectator ions.

```
FORALL I spectator(I) <-
EXISTS R,X1,X2,X3,X4 R:Reaction
        [hasReactantsIon->>c(I,X1,X2);hasProductsIon->>c(I,X3,X4)].
```

encodes definition of spectator as part of answer

Question encoding - fidelity

MC12

When methane, CH4, gas reacts with oxygen, the following changes occur

burn("CH4").

reacts with oxygen = burn

AIFB

Question encoding - fidelity

MC21

The pOH of a solution containing 2.250 g of LiOH, a compound used in space craft and submarines to remove excess carbon dioxide from the atmosphere, in 250.0 mL of solution is:

m1:Mixture[hasComponents->>"LiOH";hasMass->>c("LiOH",2.25);
hasVolume->>c("LiOH",0.25)].

no converting of measures

AIFB

Question encoding - fidelity

MC48

1.0 L of a buffer formed by mixing 0.25 moles of ammonia solution with 0.25 moles of ammonium nitrate has a pH of
(For ammonia, Kb = 1.8 x 10-5)

```
FORALL X,Y
m1:Mixture[hasComponents->>X;hasMole->>c(X,0.25);
hasVolume->>c(X,1.0)] and Y[hasKb->>1.8E-5]
        <- Y[hasName->>"Ammonia";hasFormula->>X].
FORALL X,Y
m1:Mixture[hasComponents->>X;hasMole->>c(X,0.25);
hasVolume->>c(X,1.0)]
        <- Y[hasName->>"ammonium nitrate";hasFormula->>X].
```

buffer solution not encoded

AIFB

Failure Analysis

- (MOD) Knowledge Modeling
- (IMP) Knowledge Implementation/Modeling Language
- (INF) Inference and Reasoning
- (KFL) Knowledge Formation and Learning
- (SCL) Scalability:
- (MGT) Knowledge Management
- (QMN) Query Management
- (ANJ) Answer Justification
- (QMT) Quality Metrics (MTA) Meta Capabilities

AIFB

Brittleness – out of scope

MC4

When calcium carbonate is heated it decomposes forming:
Calcium carbonate reacts with acids to produce gas

a) Solid Ca and CO2 gas
b) Gaseous CaCO and CO2 gas
c) Solid CaO and CO2 gas
d) Gaseous Ca and CO2
e) Solid CaO and liquid CO2

Brittleness – out of scope

MC6

Which solution has the highest conductivity?

a) 0.5M NH_3
b) 0.5M $NaOH$
c) 0.5M Na_3PO_4
d) 0.5M HCl
e) 0.5M HCN

not decidable by the knowledge in the corpus

Brittleness – B – MOD -1

rule oxnoOxygen: FORALL M1,C1,Y
oxno(Y,"O",-2) and oxno(Y,M1,C1)
<- molecule(M) and regexp1("O",Y) and not (equal(Y,"H2O2") or
equal(Y,"O2F2")) and remainder(Y,0,8,-2,M1,C1).

instead of

rule oxnoOxygen: FORALL M1,C1,Y
oxno(Y,"O",-2) and oxno(Y,M1,C1)
<- molecule(Y) and regexp1("O",Y) and not (equal(Y,"H2O2") or
equal(Y,"O2F2")) and remainder(Y,0,8,-2,M1,C1).

MC14, MC16, MC17, DAS6b

Brittleness – B – MOD -1

MC11

Which of the following compounds is insoluble in water?

a) $Pb(NO_3)_2$
b) Li_2CO_3
c) $(NH_4)_3PO_4$
d) $Ba(OH)_2$
e) $BaSO_4$

Rule:

compounds containing OH^- **except** compounds of Ba^+ are insoluble

is missing in the KB

Brittleness – not expected question type

DA18

Ascorbic acid, $H_2C_6H_6O_6$, is a diprotic acid with a Ka1 value of 8.9 x 10-5. The pH of a 0.125 M solution of ascorbic acid is 2.48 and the concentration of $C_6H_6O_6^{2-}$ is 1.6 x 10-12 M.

Determine the value of Ka2.

basic operation modeled to determine the pH-value given the Ka-value and **not** vice versa

Brittleness – B – MOD -1

rule **conjugateAcidBase1**:
FORALL A,B,V1,V2,V3 conjugateacidbase(A,B)
<- chemparse1(A,V1) and chemparse1("H",V2) and
chemparse1(B,V3) and addvector(V3,V2,V1).

missing explanation rule

MC19, MC22

Brittleness – B – MOD -1

rule **oxnoIonicComp**:
FORALL KF,CK,AF,CA,Y
oxno(Y,YKF,CK) and oxno(Y,AF,CA)
<- molecule(Y) and component(Y,KF,CK,AF,CA).

typo in explanation rule

FORALL I,M1,KF1,CK1,AF1,CA1,EX1
explain(EX1,I) <-
I:Instantiation[ofRule->>**oxnoIonicComp**;
instantiatedVars->>{i(M,M1),i(KF,KF1),i(CK,CK1),i(AF,AF1),i(CA,CA1)}] and
EX1 is ("The ionic components of molecule "+M1+" are <b>anion "+AF1+"</

MC 17

Lessons learned

- additional language primitives for structuring KB
- reification of rules
  inferencing to check for
  - problems with explanation rules
  - typos in rules
- extending rule editor by explanation rule generator
- testing with additional questions
- explanation kb

Summary

- similar results of teams
- high and scalable performance
- only 2 MY used

AIFB

Next Steps: Halo II

Halo II

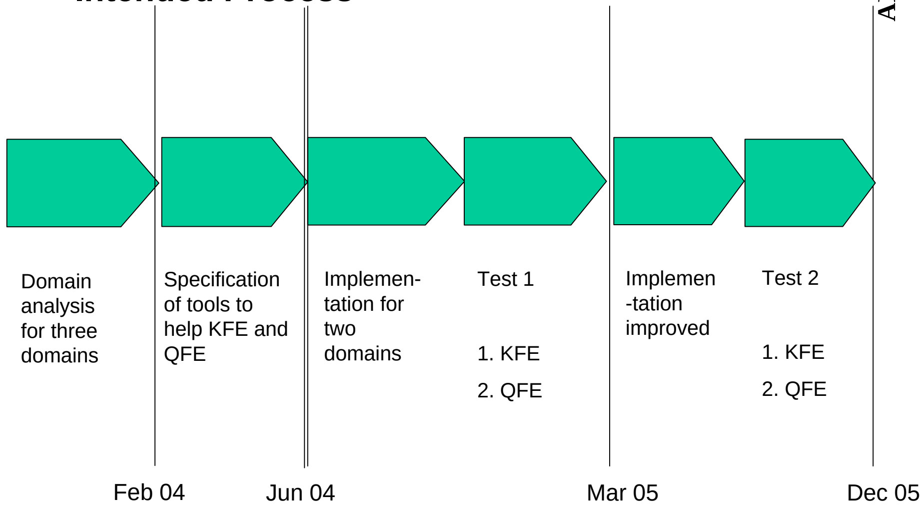
development of tools for

domain experts

to capture knowledge

AIFB

Intended Process



AIFB

Teams

- Ontoprise/Unikarl with CMU, UoColorado at Boulder, Georgia Tech, EML, Phi-T
- SRI with UoTexas, Boeing
- ISX with UoSCalifornia (ISI), Isoco, Stanford U

AIFB

1. Textbooks

- review textbooks
- selects few textbooks (in coordination with other teams, based on density of information ...)
- analyze content
  feasable ... \ / in principal
  intermediate ... >to formulate ... < using FLogic (i.e. by a KE)
  unfeasable ... / \ using our to-be created system (i.e. by the KFE)
- rate different kinds of knowledge qualitative vs. quantative
  textual vs. graphical vs. tanular structured vs. unstructured
  explicit vs. implicit

2. Questions

- review AP questions
- analzye AP questions
  feasable ... \ / in principal
  intermediate ... >to formulate ... < using FLogic (i.e. by a KE)
  unfeasable ... / \ using our to-be created system (i.e. by the QFE)
- judge wether AP questions are easy, hard not to answer by our system

3. Pump Priming

- identify basic epistemological primitives needed to model the domain (or the subset of the textbooks identified)
- identify top level ontology (which might lead to some kind of modularization of the domain)
- define domain terminology (maybe the same as top level ontology)
- identify extra-logical constructs (algorithmic)
- list external knowledge sources that might be good starting points for pump priming via wrapping

4. Misc & 5. Evaluation

4. Misc
   enumerate relevant tools, systems in their domain (what is their apprach, philosophy, UI?)

5. Evaluation
   review design and mock-up
   test system as SME and/or QFE, maybe with help of students