SEMANTIC CONFLICT DETECTION IN META-DATA

A RULE-BASED APPROACH

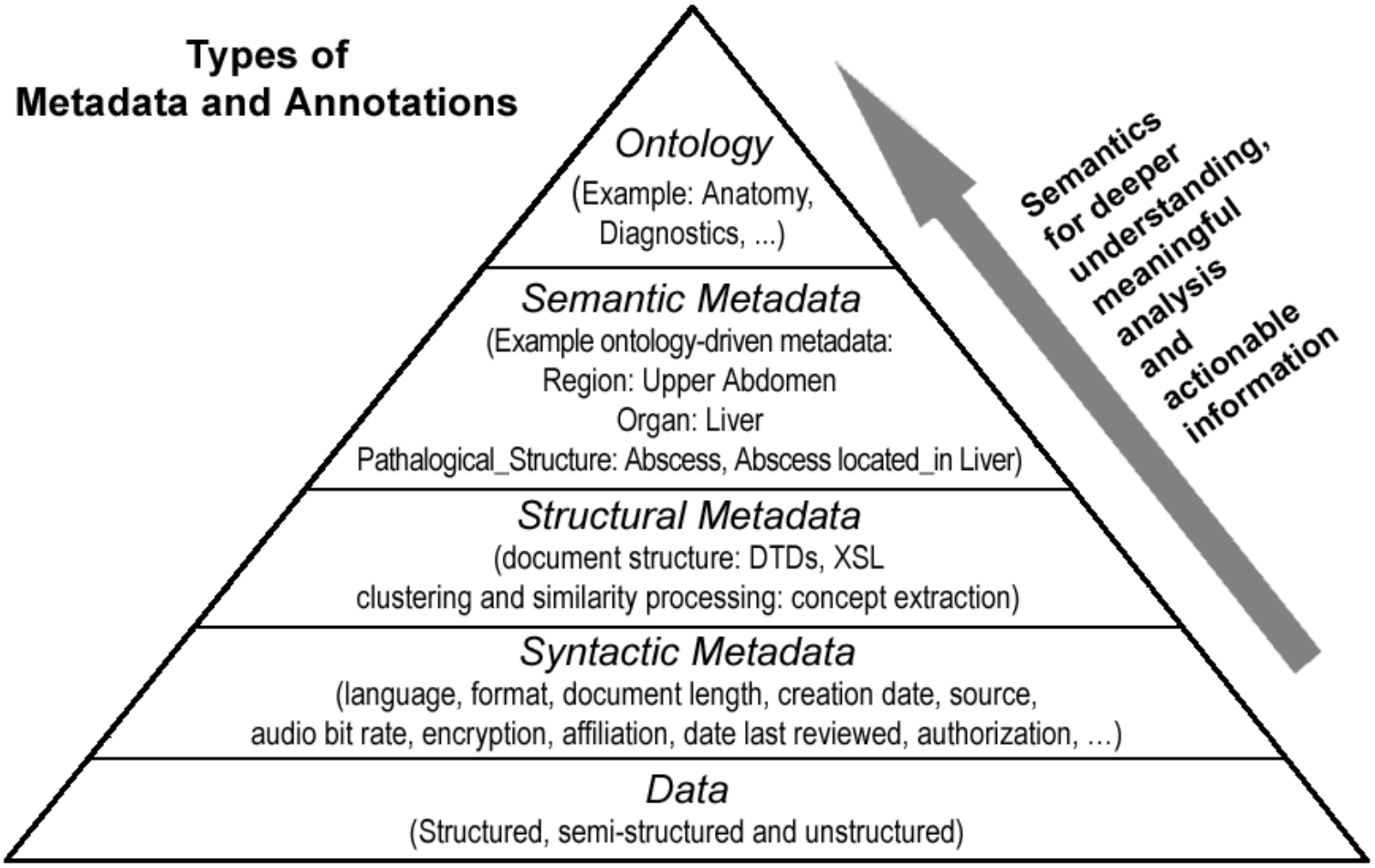
KARTHIKEYAN GIRILOGANATHAN

ADVISOR: Dr. I. BUDAK ARPINAR
COMMITTEE: Dr. AMIT P. SHETH
Dr. KHALED M. RASHEED

Introduction

- Massive amount of data is available on the Web
- Ability to annotate, extract, and query semantic meta-data has increased:
  - SWETO (Semantic Web Technology Evaluation Ontology):
    - populated with over 800,000 entities and 1.5 million explicit relationships between them in RDF or OWL
  - Freedom (Semagix):
    - uses SWETO and other domain ontologies to semantically annotate millions of documents or Web pages
  - Web Fountain (IBM):
    - annotated and disambiguated data from over a billion documents

Evolution of Meta-Data



[Sheth 2003]

Meta-Data Concerns

- Next generation tools will focus on actionable information (with associated sources and supporting evidence) from existing (meta-)data
- Concerns about usage of meta-data
  - High quality (i.e., reliable, accurate, and trustworthy) semantic meta-data
    - Entity disambiguation
    - Inconsistency checking in OWL
    - Conflict detection

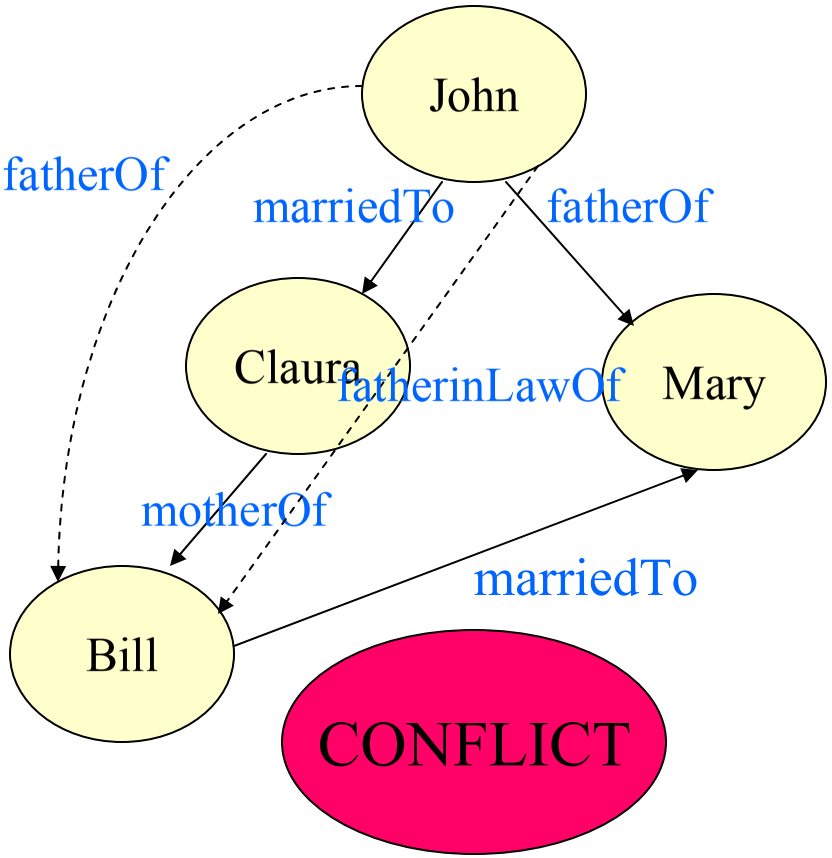
Motivating Factors

- "Representing, identifying, discovering, validating, and exploiting complex relationships are important issues related to realizing the full power of the Semantic Web, and can help close the gap between highly separated information retrieval and decision-making steps" [Sheth, Arpinar & Kashyap 2003]

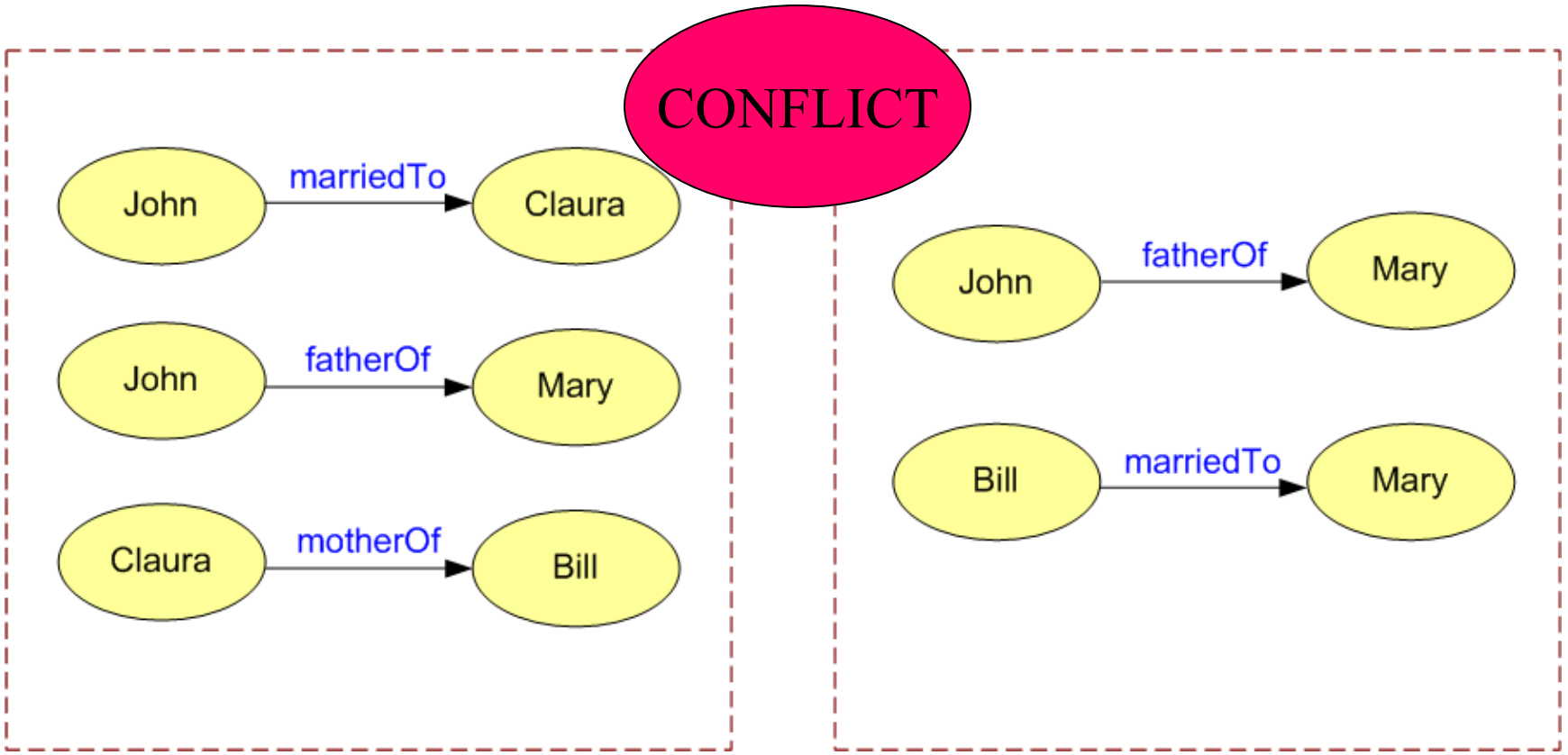
- "The Web is decentralized, allowing anyone to say anything. As a result, different viewpoints may be contradictory, or even false information may be provided. In order to prevent agents from combining incompatible data or from taking consistent data and evolving it into an inconsistent state, it is important that inconsistencies can be detected automatically" [W3C 2004]

- "… these problems manifest themselves in various ways, including poor recall of available resources and inconsistency of search results. They arise due to errors, omissions and ambiguities in the metadata…" [Currier & Barton 2003]

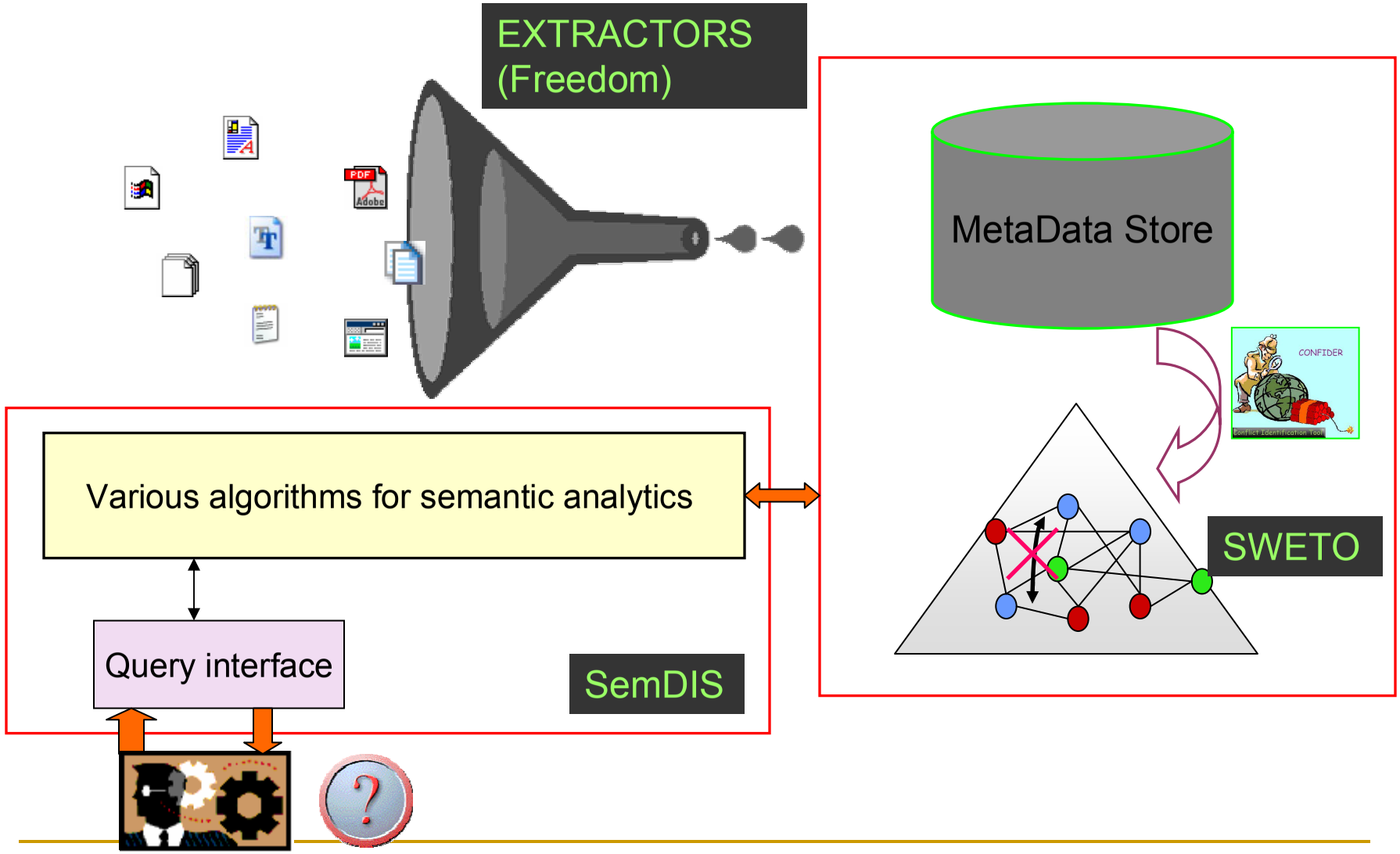
Semantic Conflict Identification



Conflict illustration through Simplification



Motivating Scenario

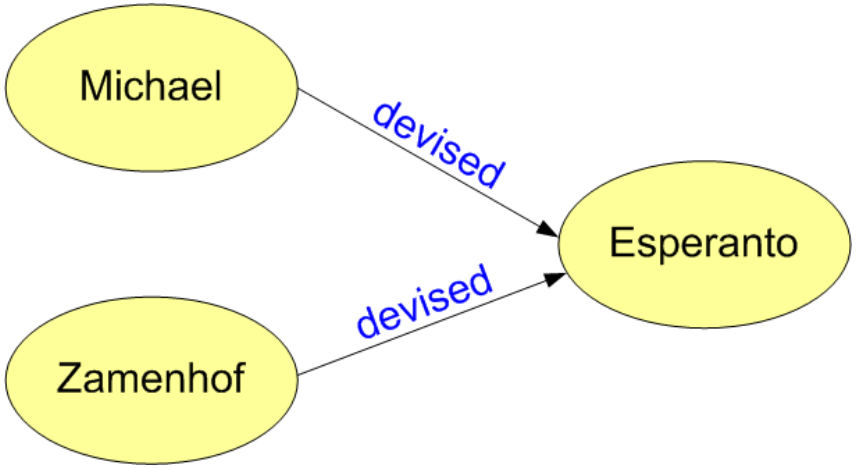


Outline

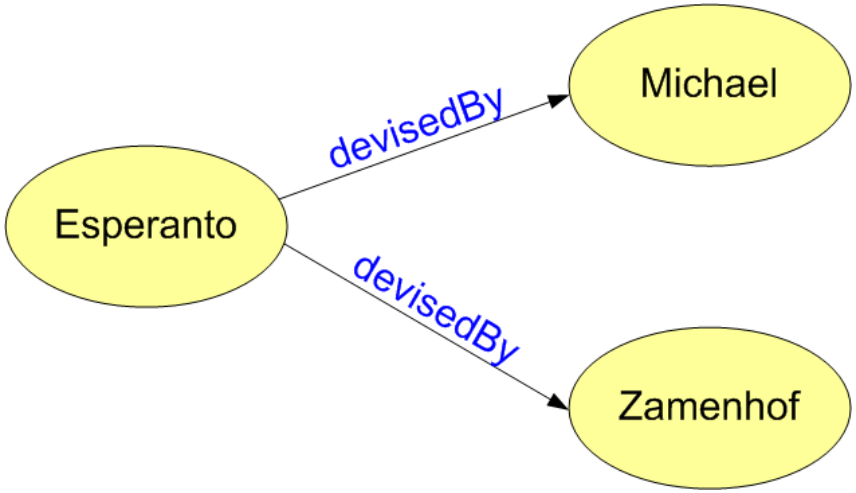
- Conflict types and definitions
- Simplification process
- System architecture
- Experimental results
- Conclusion and future work

Property Assertion Conflicts

- ‘daml:unambiguous’ or ‘owl:inverseFunctional-Property’ violation



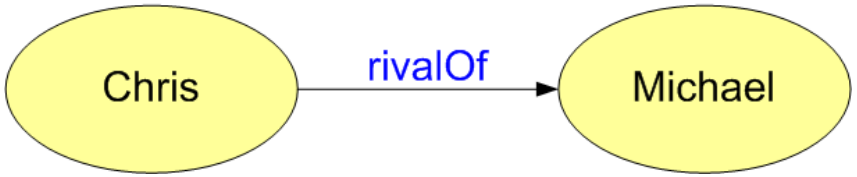
- ‘daml:unique’ or ‘owl:Functional-Property’ violation



Property Assertion Conflicts

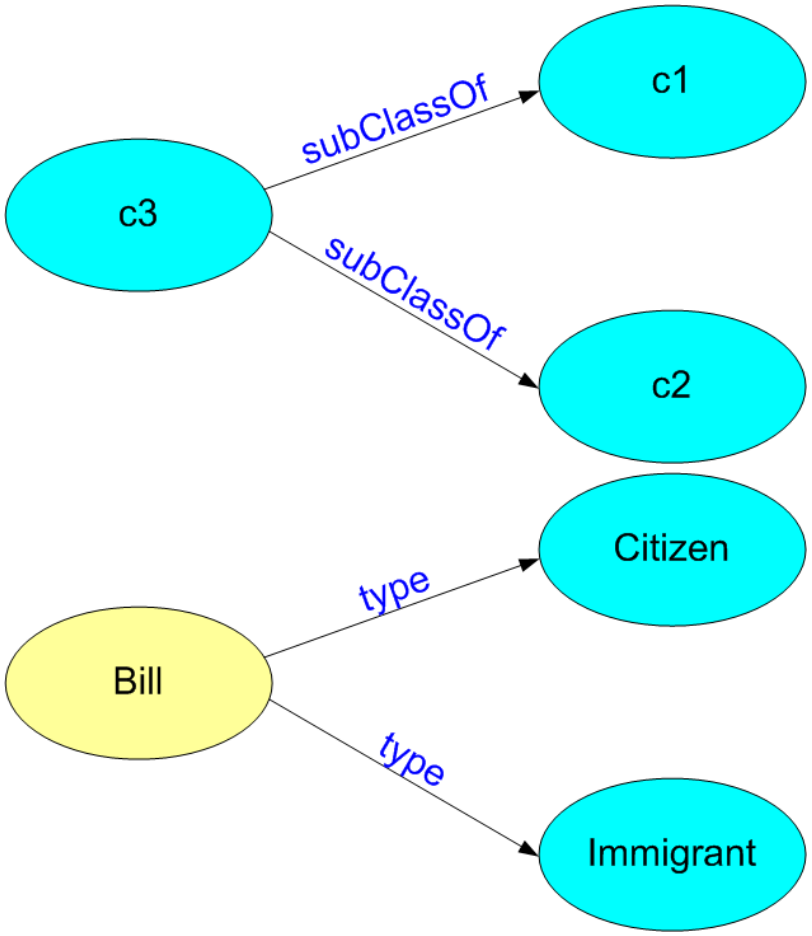
- ‘asymmetric’ property violation

- ‘disjoint’ property violation



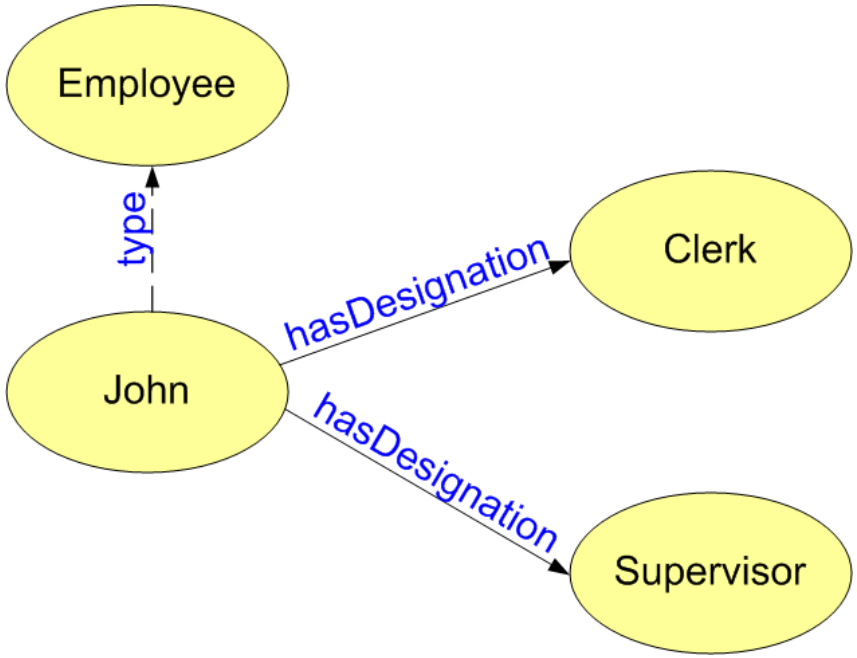
Class Assertion Conflicts

- Classes c1 and c2 are ‘daml:disjoint’ or ‘owl:disjoint’
- Classes ‘Citizen’ and ‘Immigrant’ are ‘daml:disjoint’ or ‘owl:disjoint’



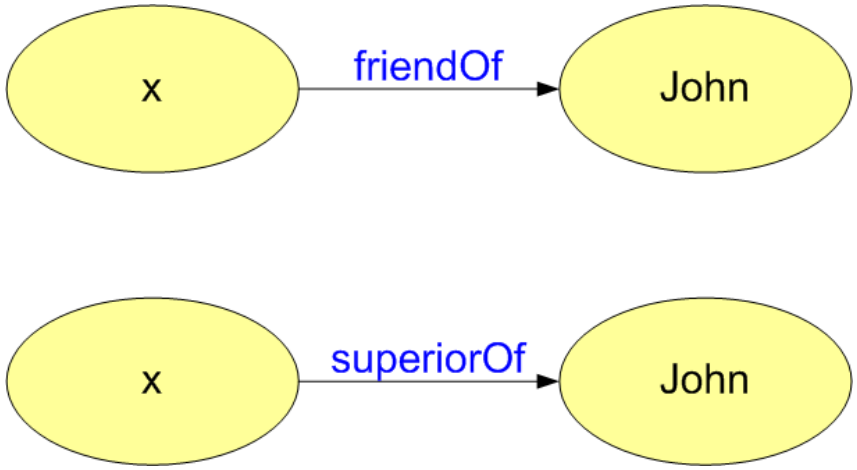
Class Assertion Conflicts

- Class ‘Employee’ has a OWL or DAML restriction ‘maxCardinality’ of ‘1’ on a relation ‘*hasDesignation*’



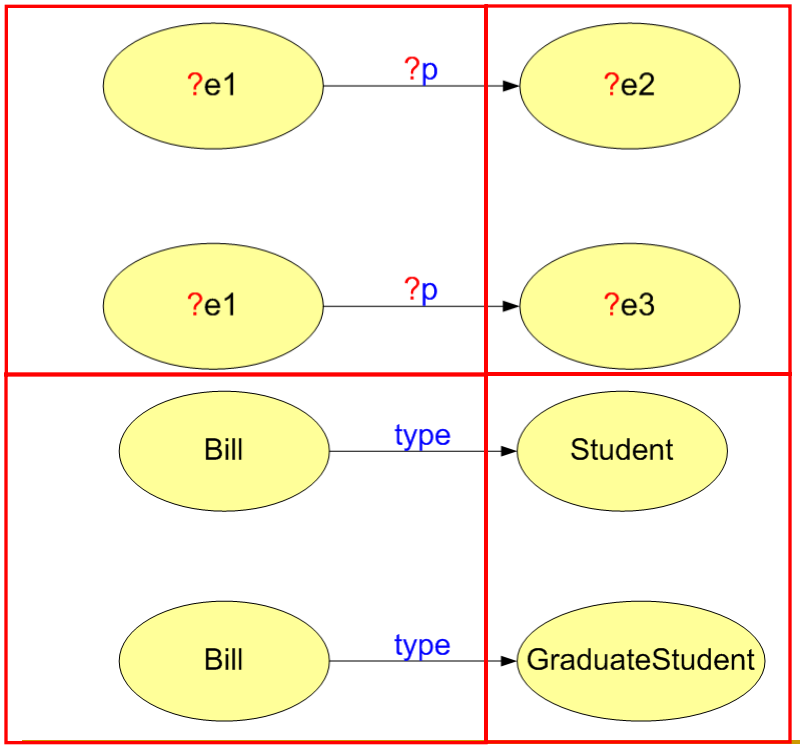
Statement Assertion Conflicts

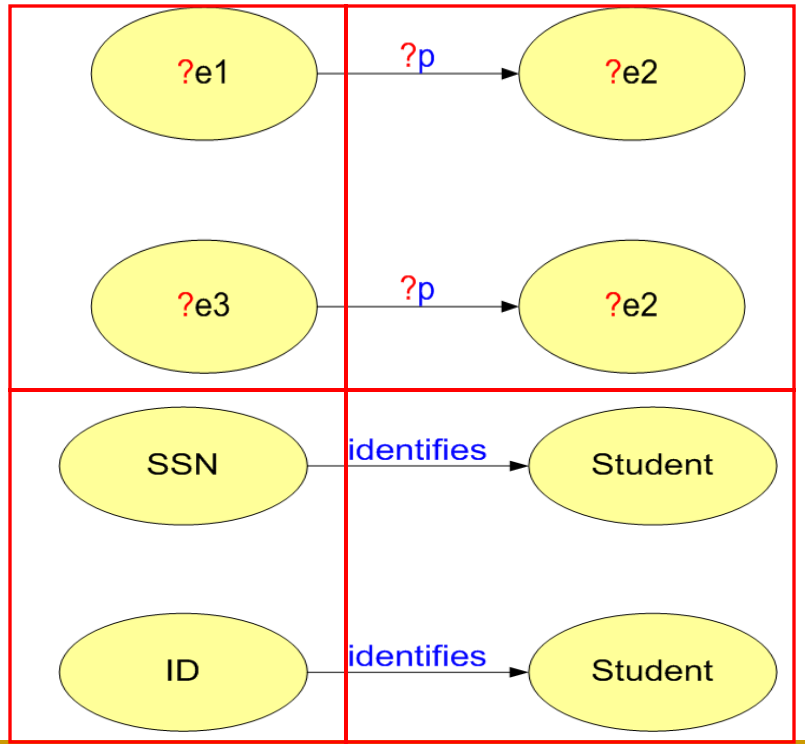
- We want to say that a person cannot be a superior and a friend to “John” at the same time.



Non-Assertional Conflicts

- Either the subject or the object alone is different between two RDF triples.
- Subjective Conflict





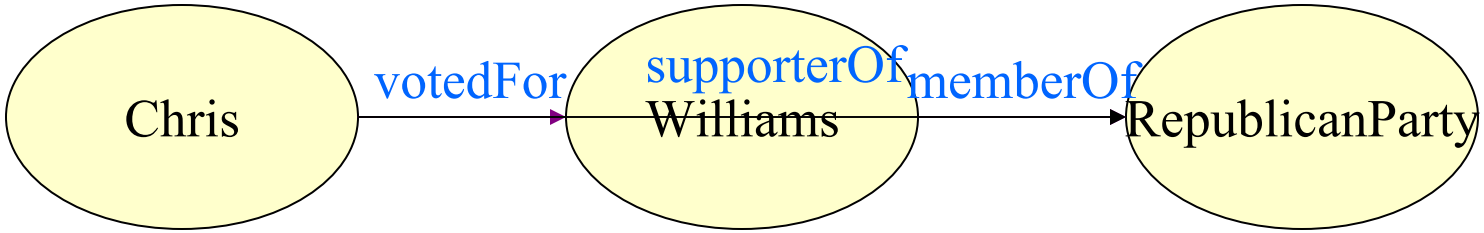
Conflict Definitions

- Two sets of triples T1 and T2 are said to be in *conflict* if their *simplifications* S(T1)→s1 and S(T2)→s2 are mutually *non-agreeable*.
- Two *simplifications* s1 and s2 are mutually *non-agreeable* if taken together they are in violation of U or E.

T	A set of triples
S	A function denoting the process of *simplification*
s	The result of *simplification* (S(T)→s)
U	Constraints expressed in an ontology (e.g., the property 'biologicalMother' is unique)
E	Constraints supplied by an expert (e.g., person(x) can never do action(y))

Simplification Types

- An RDF triple is trivially a *simplification* because it is the most basic piece of knowledge
- Composition of relations leads to simplification



Composition of Relations

Consider a set of Triples T,

Let

E	-	set of entities
P	-	set of relations

Then,

E	=	$\{e1, e2 \ldots en\}$
P	=	$\{p1, p2 \ldots pm\}$.

Let

C	-	set of ordered relation tuples that can be composed to a single relation
R	-	set of relations obtained by substituting the composed relation for the composable relations.

Then,

C	=	$\{(p1, pk), \ldots, (pa, pb, pc, \ldots)\}$.
R	=	$\{r1, r2 \ldots rn\}$, where $r1, r2 \ldots rn$ are results of the composition.

The triple (ei rk ej) is a simplification if rk $\in$ R and ei,ej $\in$ E.

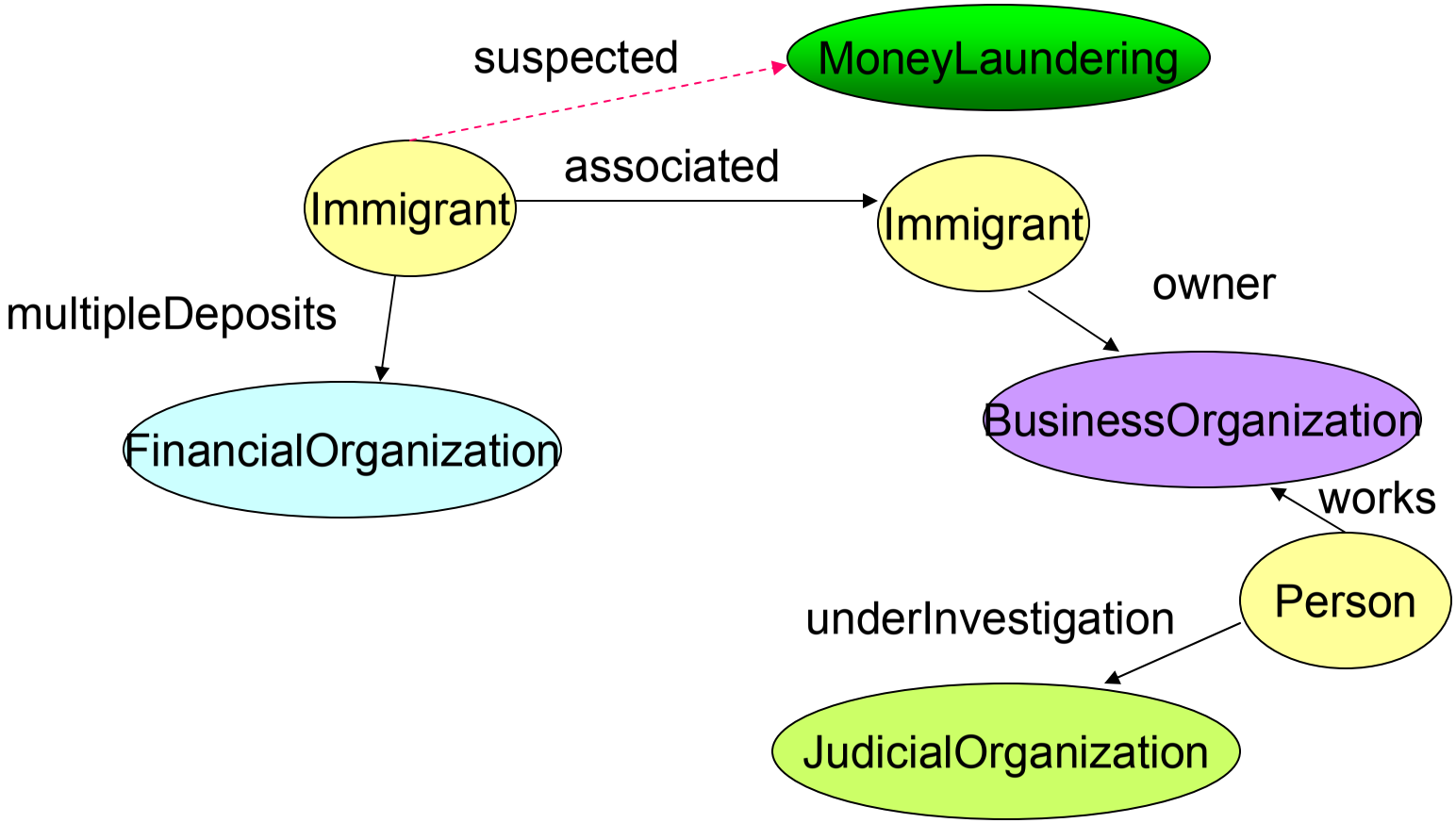
Statement Simplification

- There could be background knowledge based simplifications of the form:

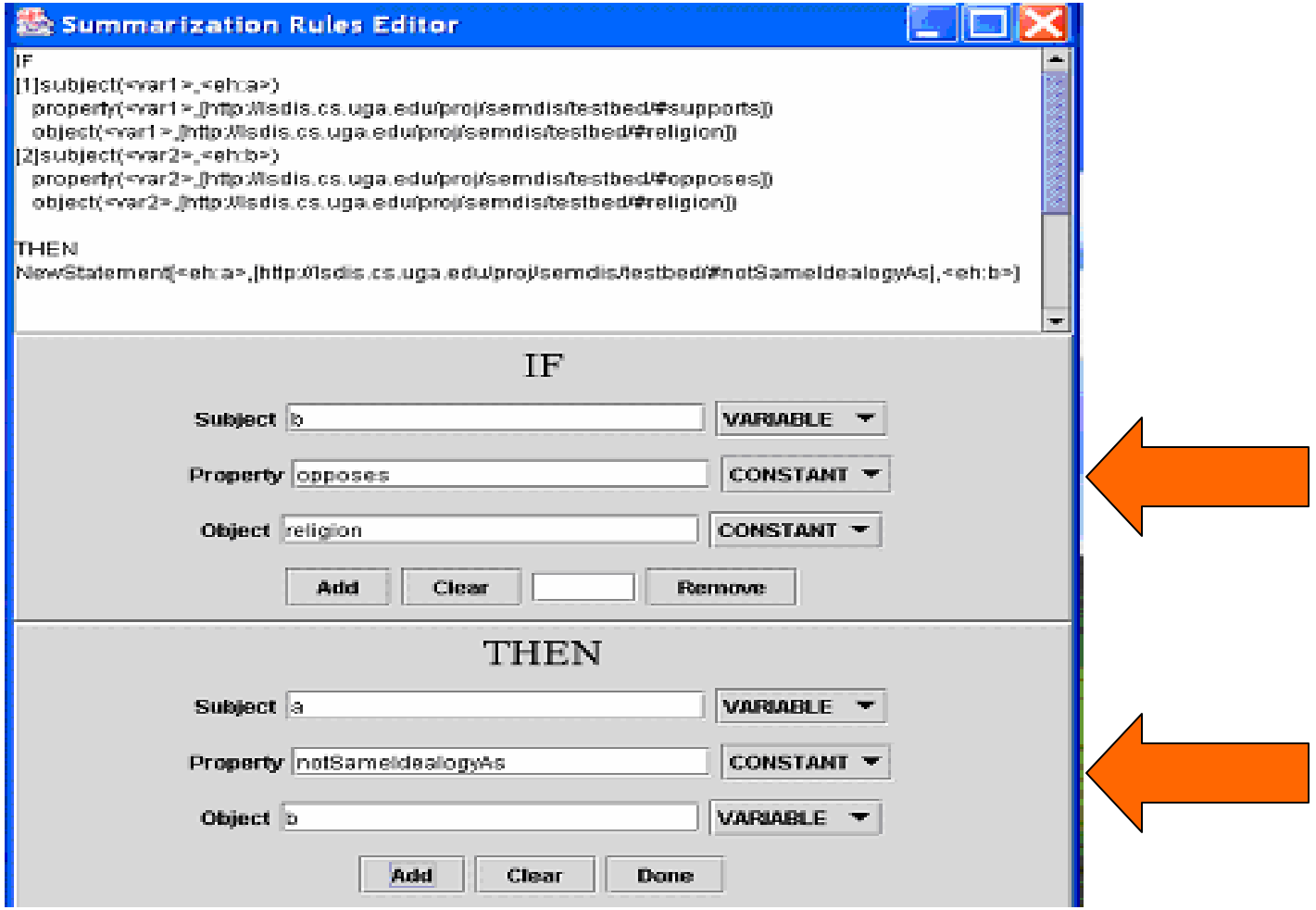
$$statement_1 \wedge statement_2 \wedge \ldots statement_n \rightarrow statement_t$$

- In this case *statement*$_t$ is a simplification.
  - This type of simplification will depend on expert knowledge.

Statement Simplification



Defining Statement Simplification Rules



RuleML (Overview)

- Explore rule systems suitable for the Web
- The syntax (in XML and RDF form)
- Semantics
- Tractability/efficiency
- Transformation
- Compilation
- Enable inferencing on Web data & interchange of rules between intelligent systems (ontology integration etc.)

http://www.ruleml.org/

We use RuleML

- Any inference engine that understands RuleML can evaluate our rules.
- We do not need to think about representation and translation.

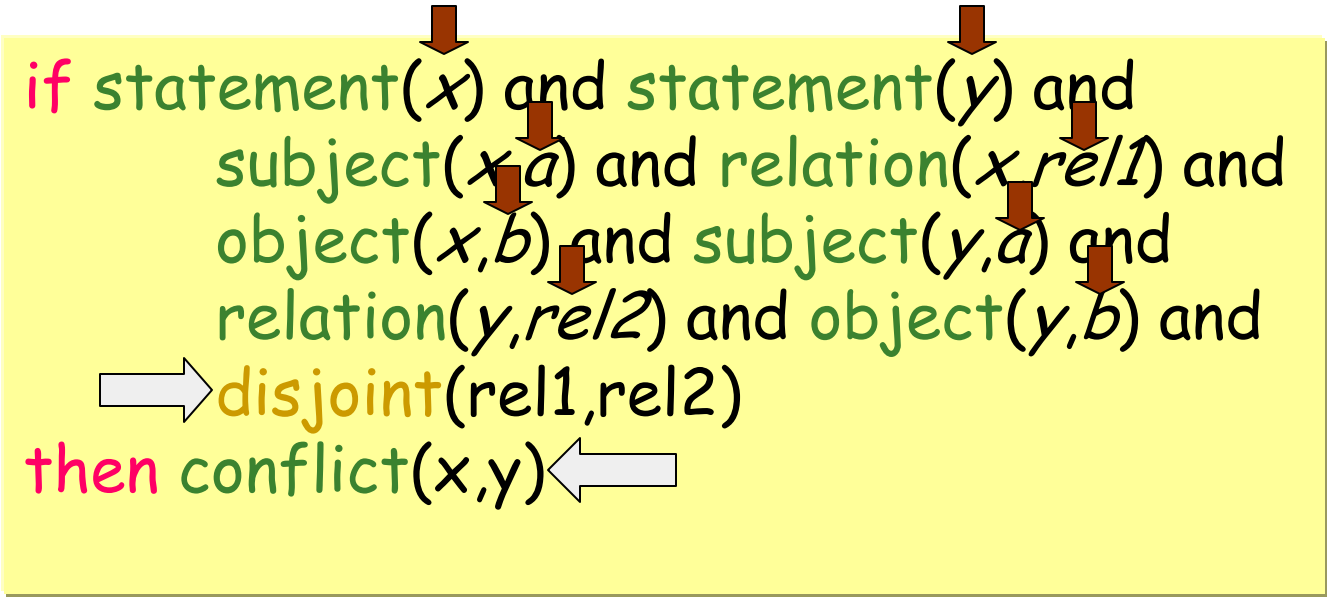
Our Representation of an RDF triple

<*subject*><*property*><*object*>

- Statement(x)
- Subject(x,*subject*)
- Property(x,*property*)
- Object(x,*object*)

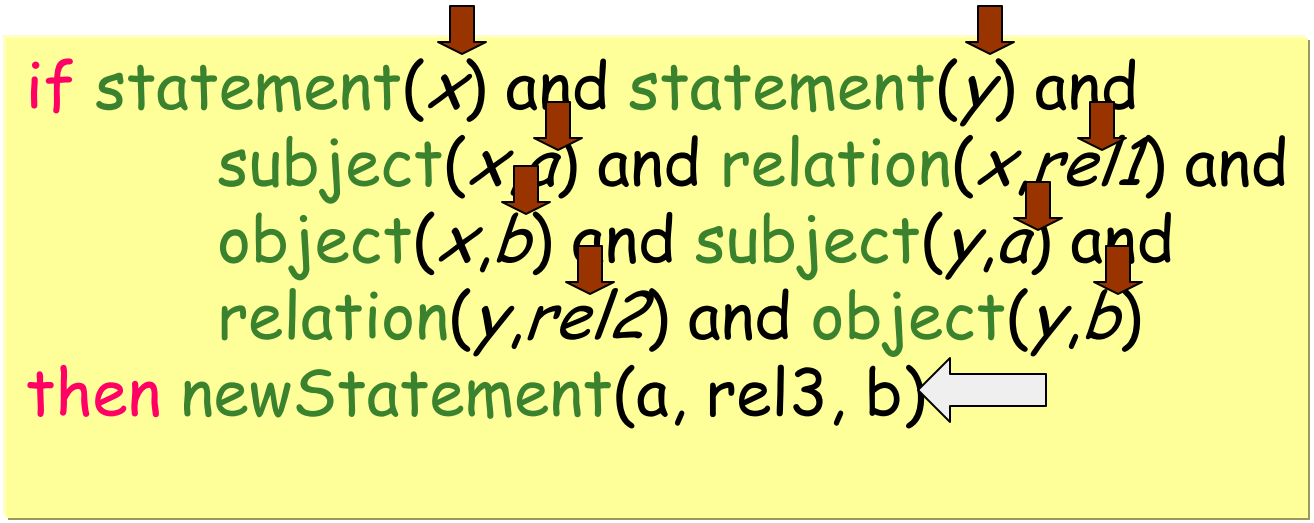
Conflict Rules

- Can be classified as Integrity Constraint Rules.



Simplification Rules

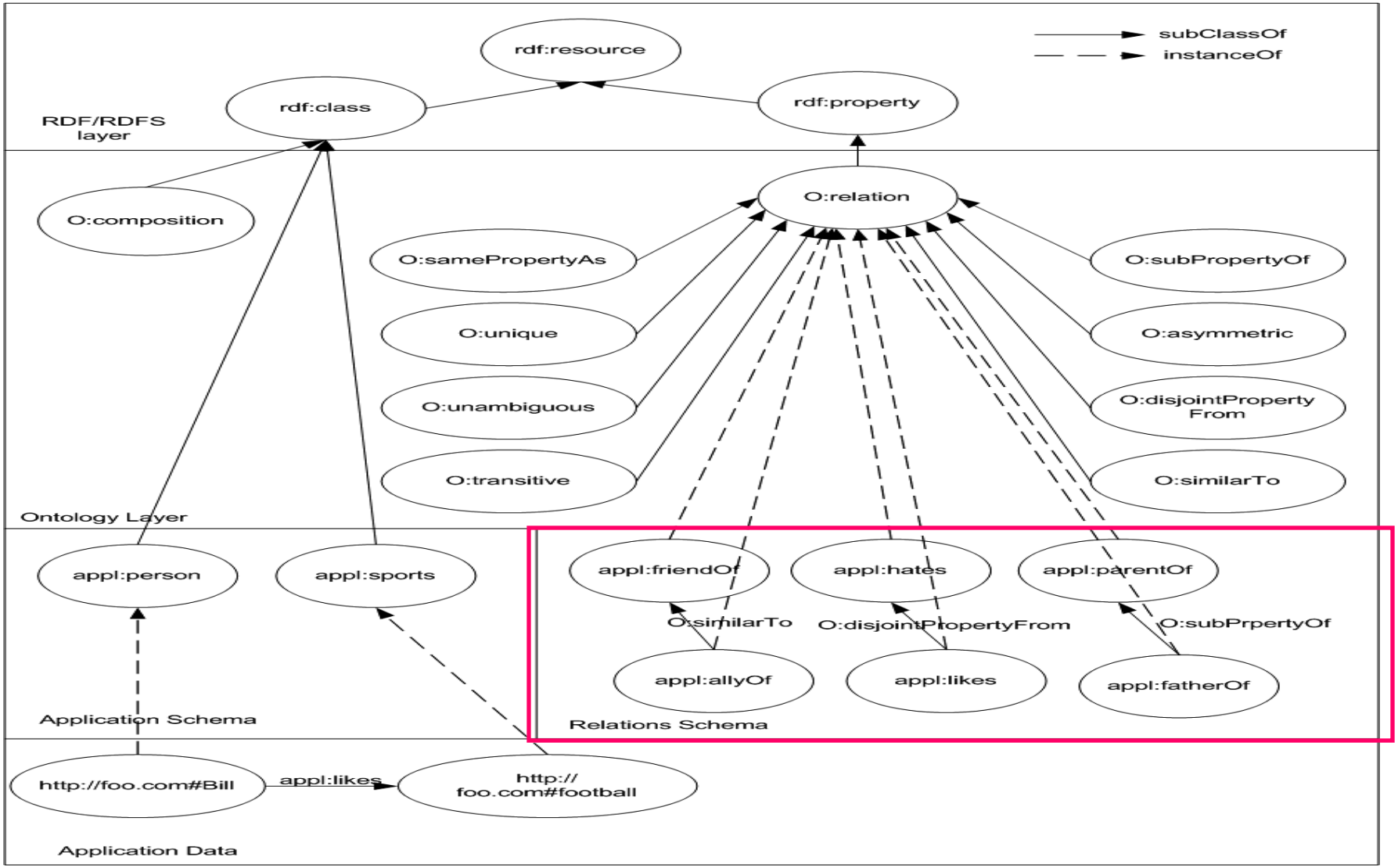
- Can be classified as Production Rules



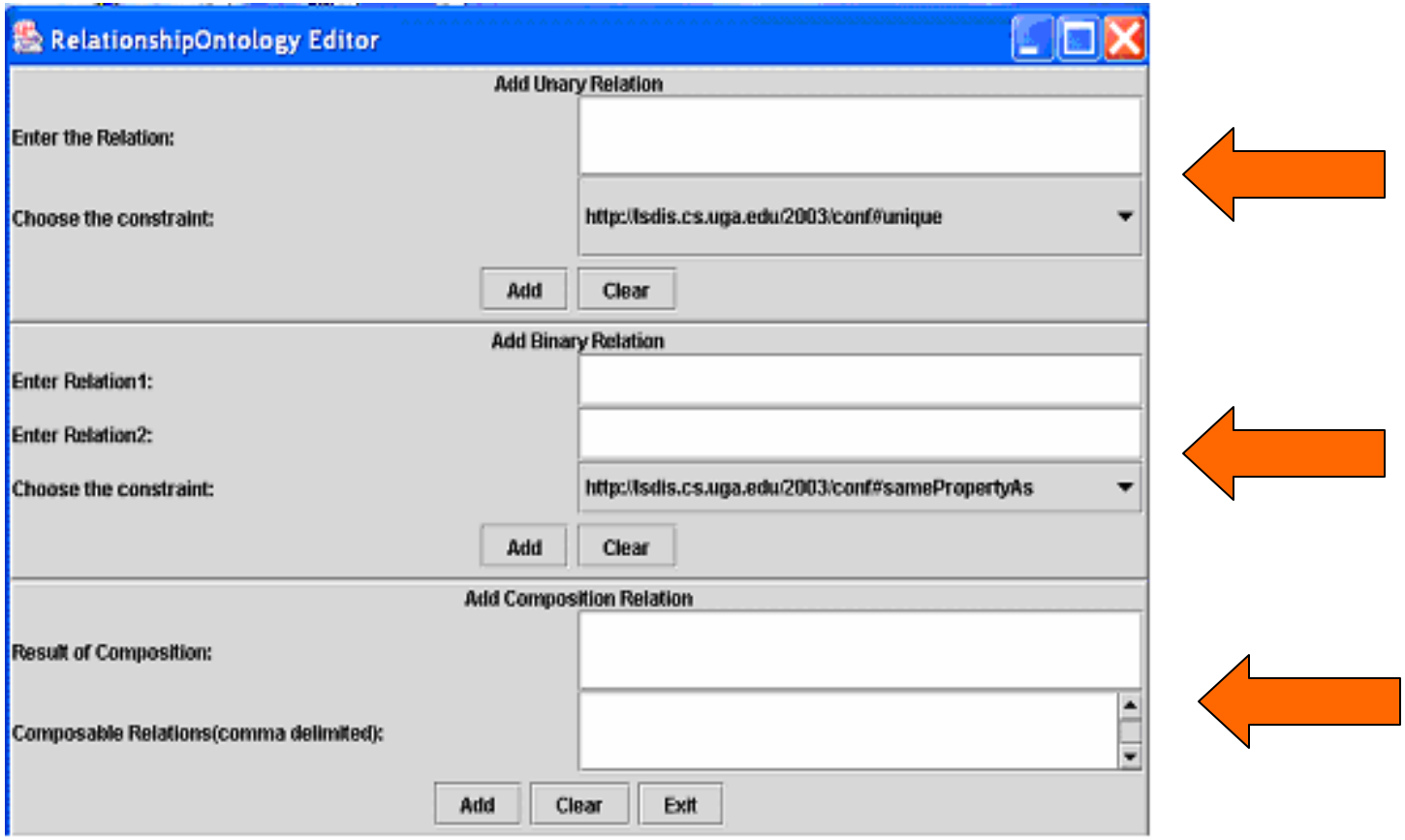
Relationship Ontology

- Relations are at the heart of semantic Web [Sheth, Arpinar & Kashyap 2003]
- Relations among relations need to be specified
  - Hierarchy of relations is similar to a taxonomy
- Just as we have moved from taxonomy to ontology the idea is to have an ontology for relations also.

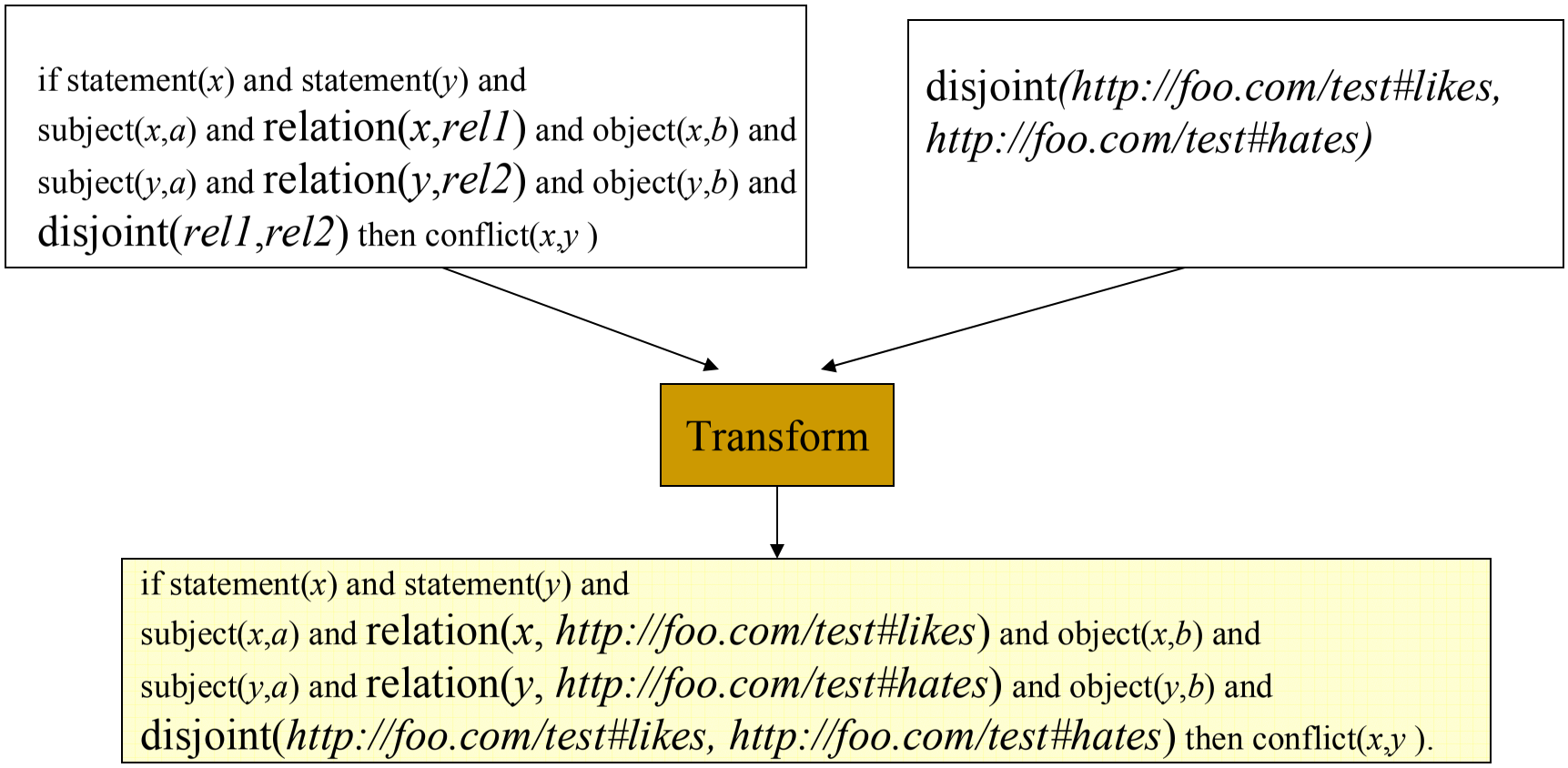
Relationship Ontology



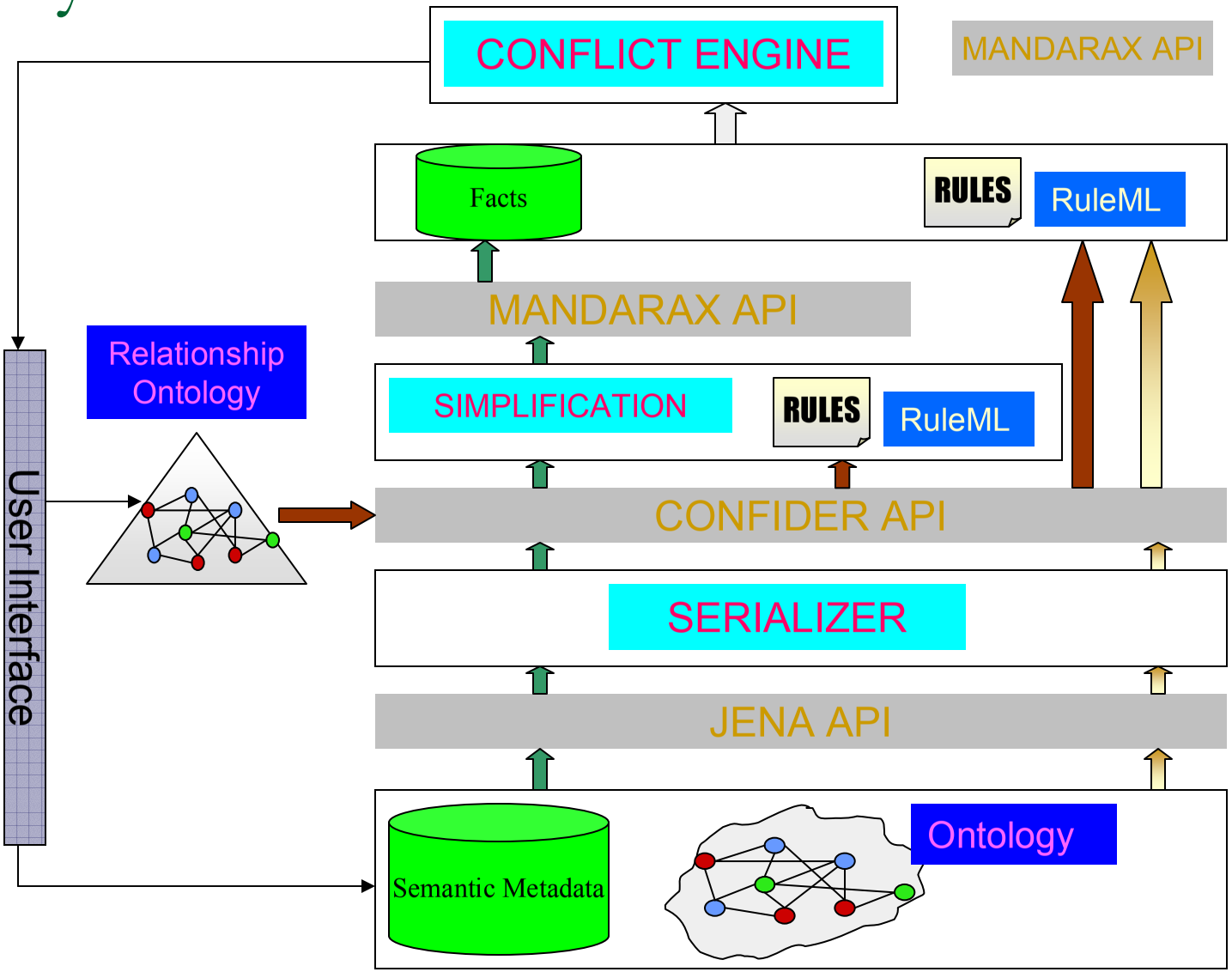
Editing/Populating the Relationship Ontology



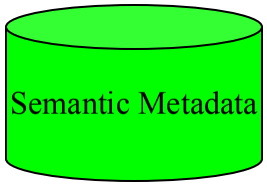
Template Based Rule Transformation

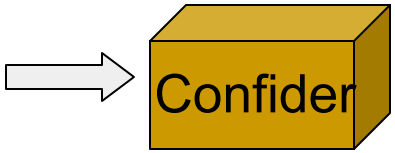


System Architecture



Algorithm





❑Serialize the statements to triples and assign id to each triple
❑Convert triples to FOL predicates

Simplification Rules?

yes

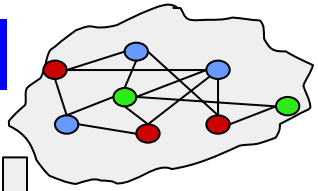
NO

While new statements produced

❑Apply simplification rules to get new stmts
❑Assign internal ids to these stmts
❑Attach the derivation to each stmt

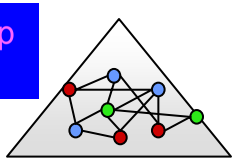
FACTS

Ontology



❑Find statements that correspond to constraints
❑Convert those statements to **conflict rules**
❑Convert the rest of the statements to facts

Relationship Ontology



❑Find relations that correspond to constraints
❑Generate **Conflict rules**
❑Find relations that correspond to Compostition
❑Generate **Simplification rules**

Conflict Rules

CONFLICT ENGINE

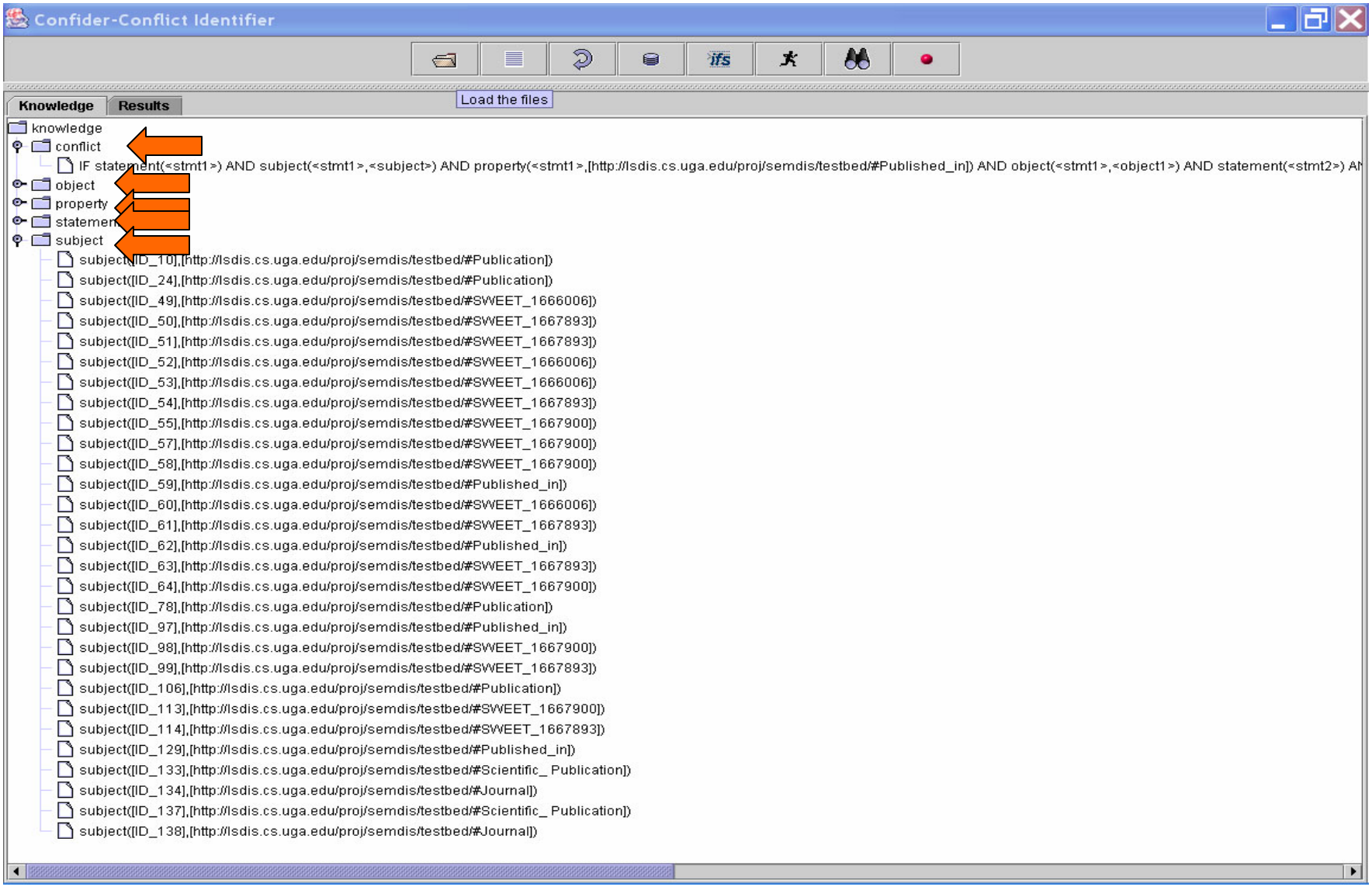
❑List of Statement pairs that are in conflict
❑Derivation available for conflicting statements
❑Derivation available if stmts generated internally

Mandarax

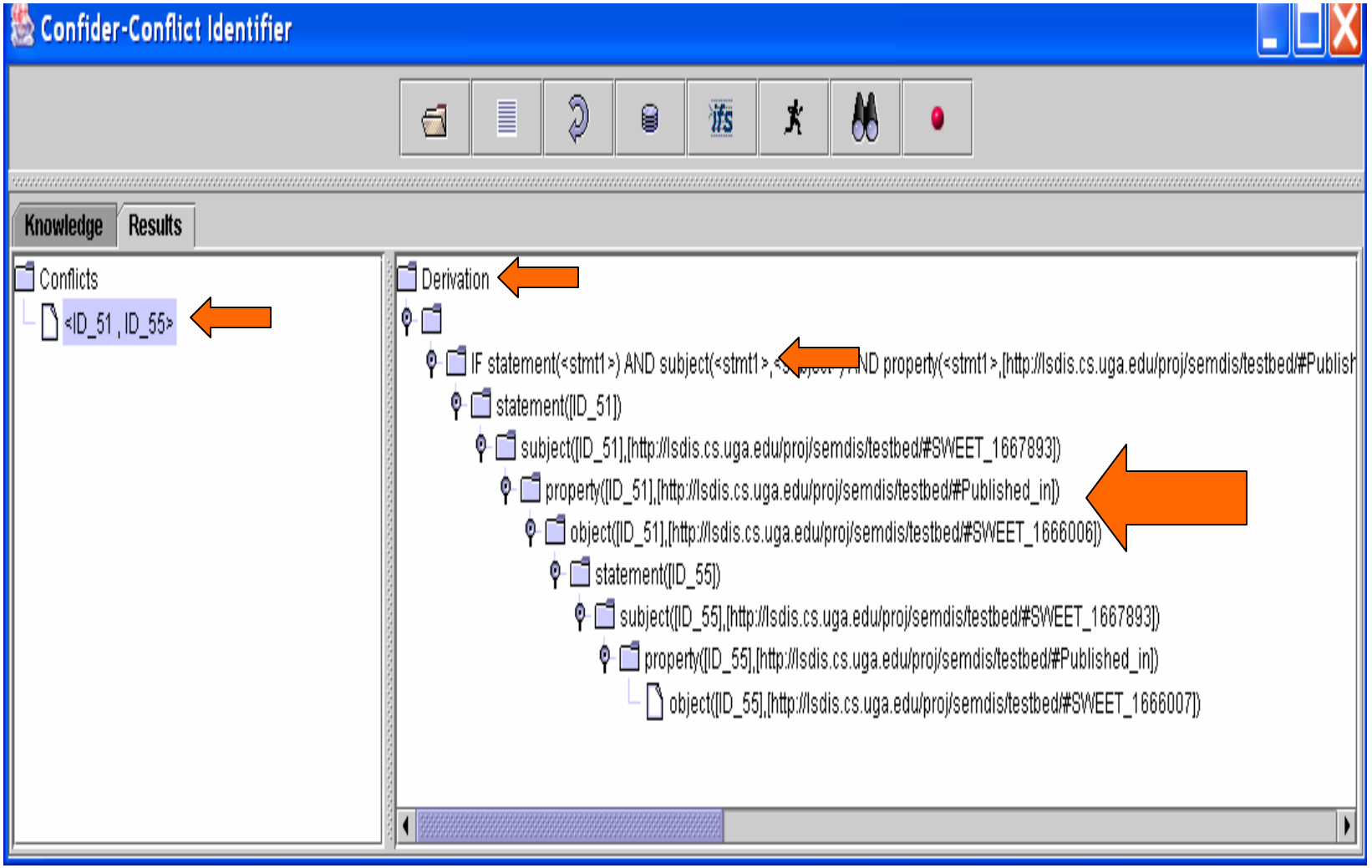
- A open source java class library for deduction rules
- OO
  - Not a translation of a prolog interpreter from c to java
- Based on backward reasoning
- Easy integration of various databases
- Support for Web services, and EJB
- Rules specified as RuleML
- Jens Dietrich (Massey University, New Zealand)
  - A list of contributors available at http://mandarax.sourceforge.net/

www.mandarax.org

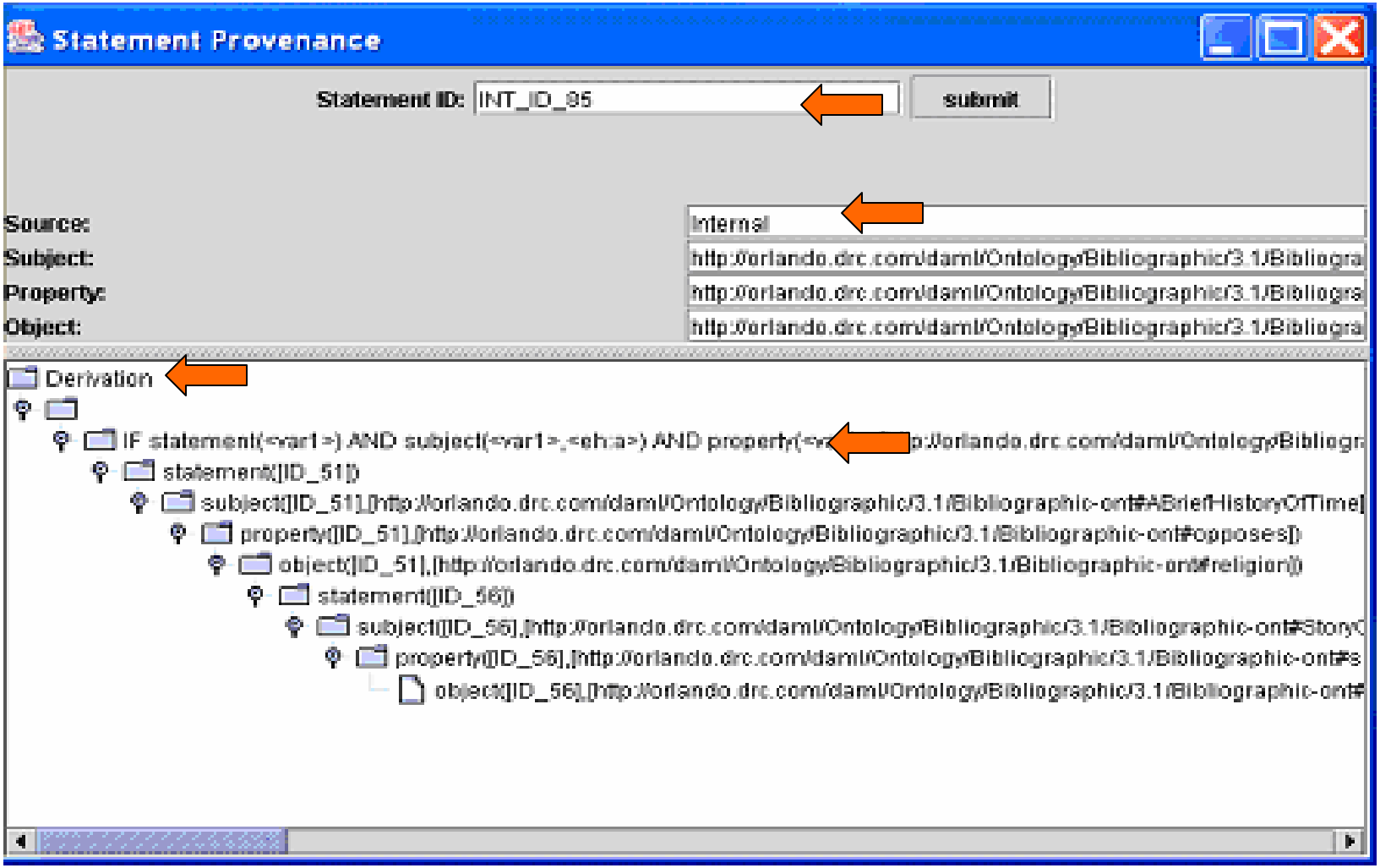
Knowledgebase with Facts and Rules



Conflict Identification Results

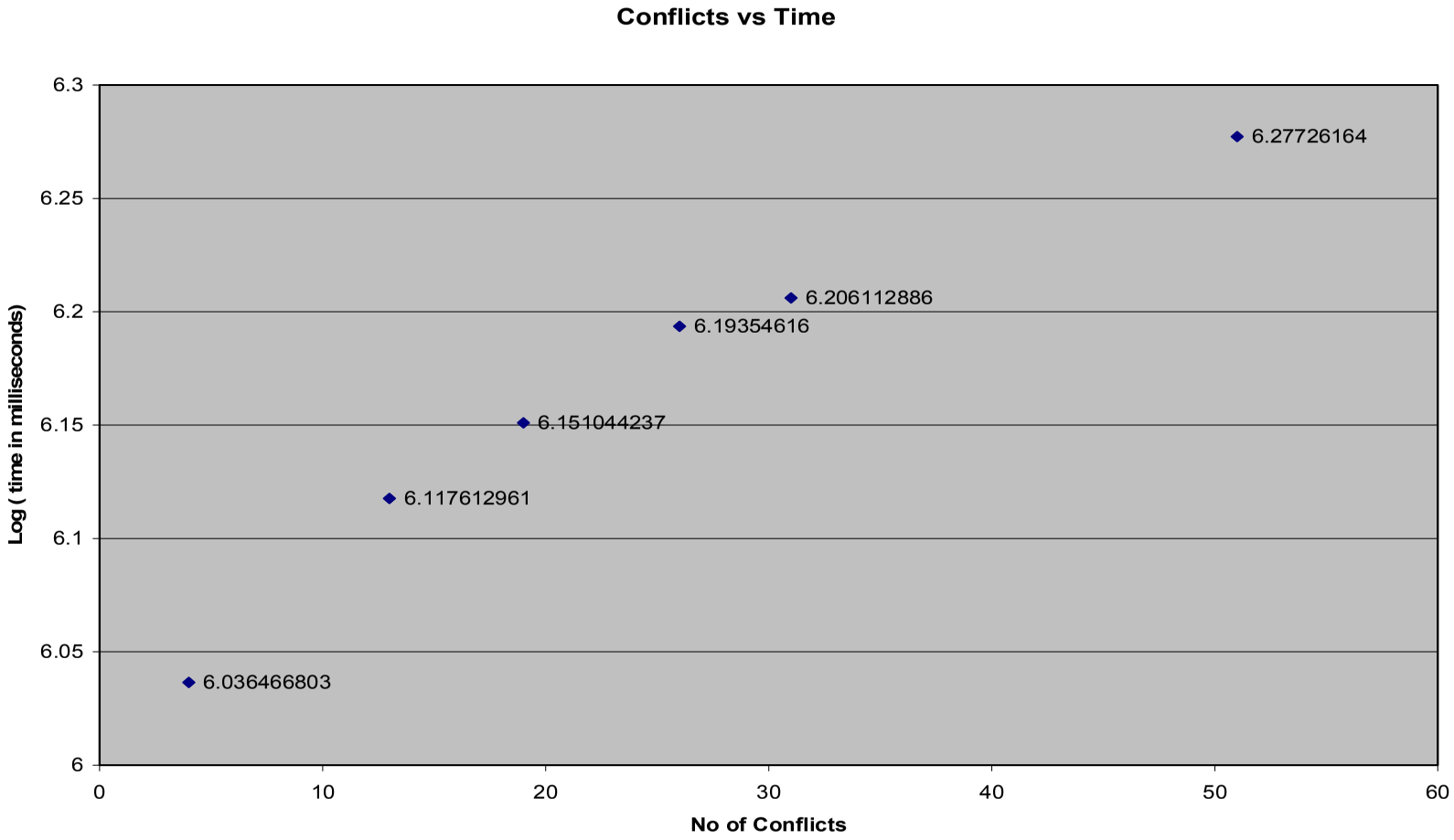


Statement Provenance



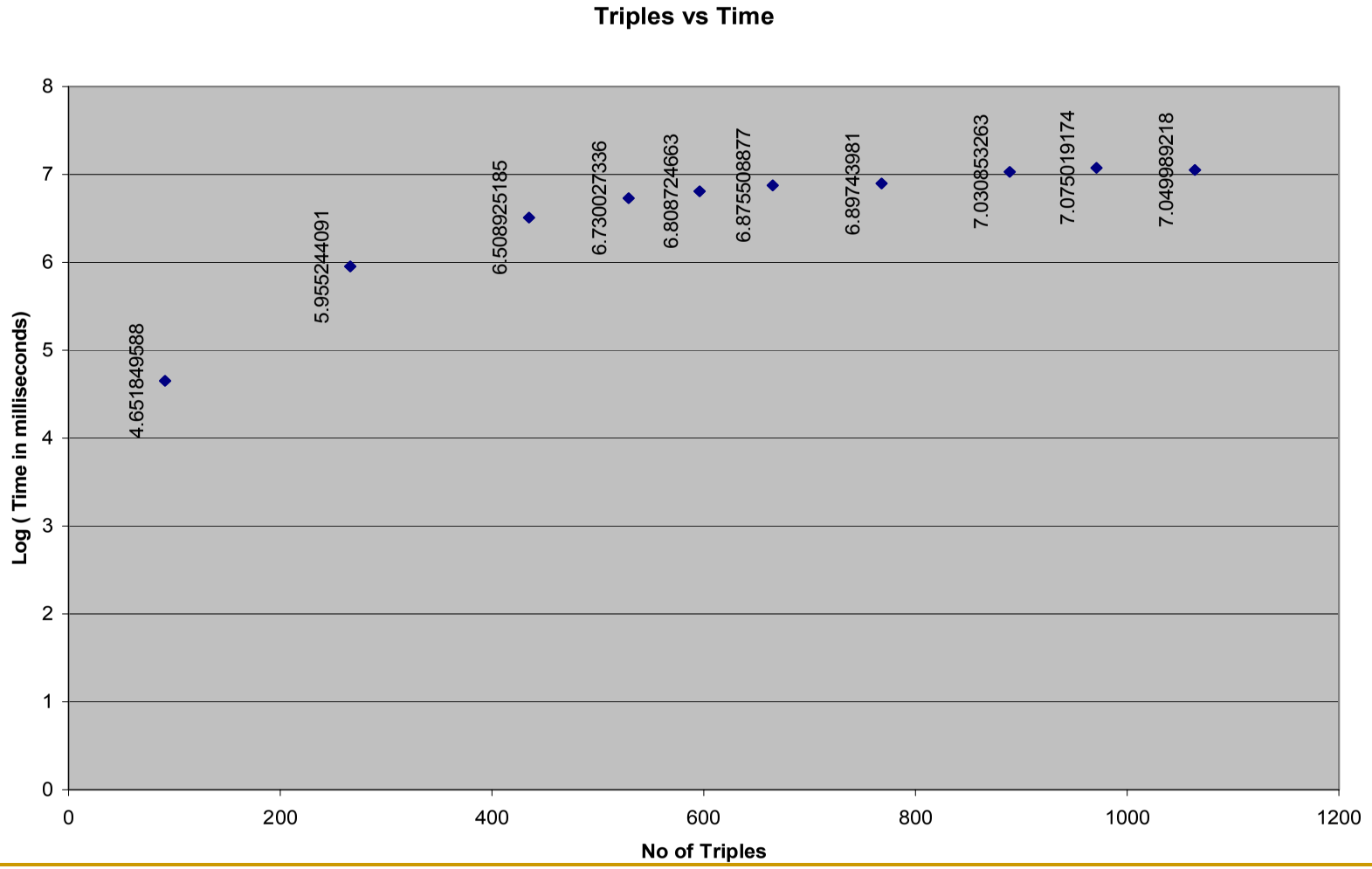
Performance Evaluation (1)

with increase in number of conflicts (500 triples)



Performance Evaluation (2)

with increase in number of triples (10 conflicts)



Conclusion

In this work we have:

- Defined conflicts in semantic meta-data and classified them.
- Discussed a rule-based approach to identify the conflicts.
- Shown the use of relations between relations to simplify the triples and identify conflicts.
- Demonstrated the applicability of the approach over a limited data set using a prototype.

Future Work

Our future work directions include developing:

- Scalable conflict identification techniques for large amounts of semantic data and conflict rules
- Investigation of other rule evaluation methods to improve performance
- Experiment with ways of representing an RDF triple in predicate form to compare performance
- A mechanism for expressing, evaluating, and adjusting trust dynamically based on conflict detection

Questions?

Thank You