



LSDIS

Large Scale Distributed Information Systems



University of Georgia
Computer Science Department

SEMANTIC CONFLICT DETECTION IN META-DATA

A RULE-BASED APPROACH

KARTHIKEYAN GIRILOGANATHAN

ADVISOR: Dr. I. BUDAK ARPINAR

COMMITTEE: Dr. AMIT P. SHETH

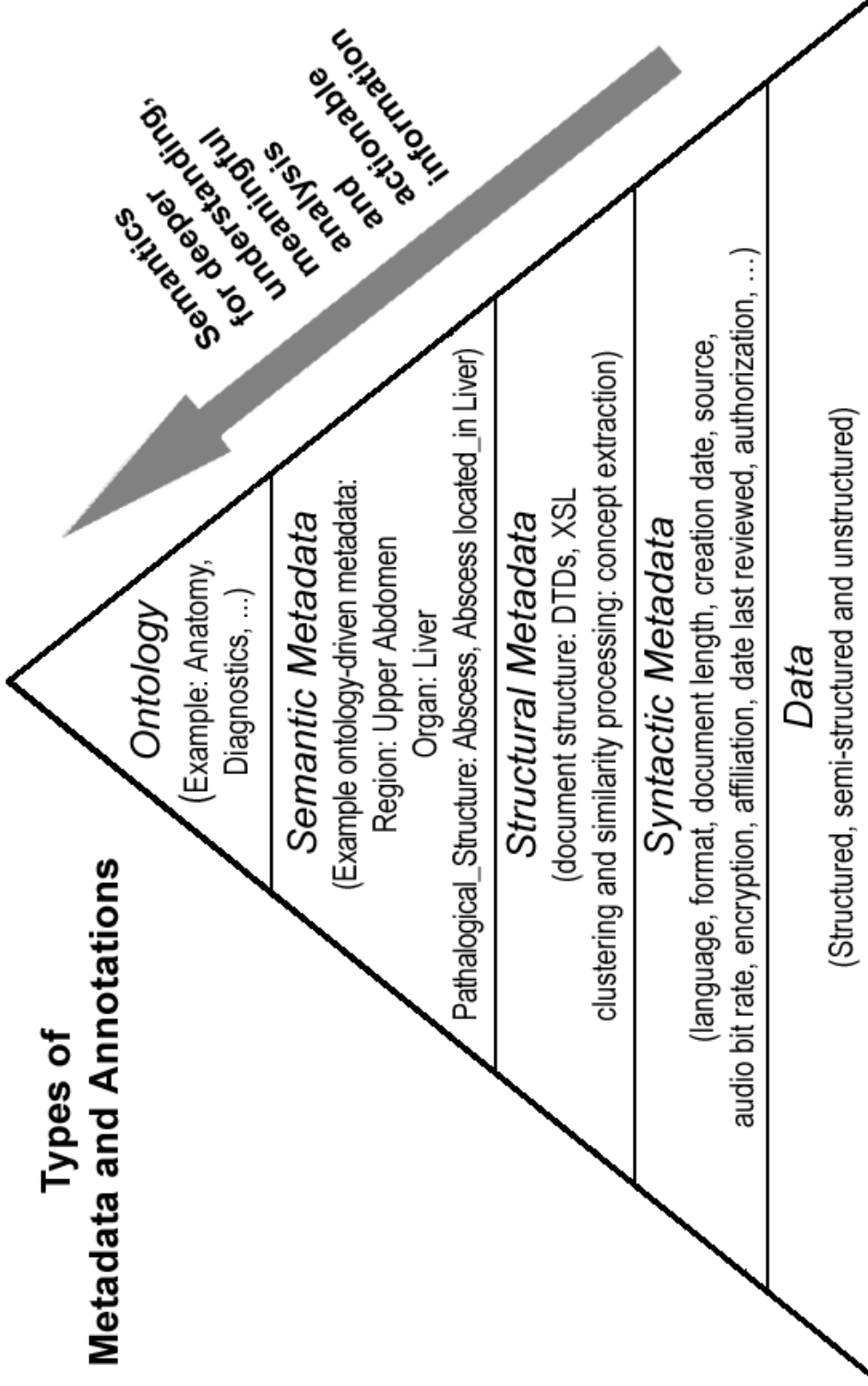
Dr. KHALED M. RASHEED

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

Types of Metadata and Annotations



[Sheth 2003]

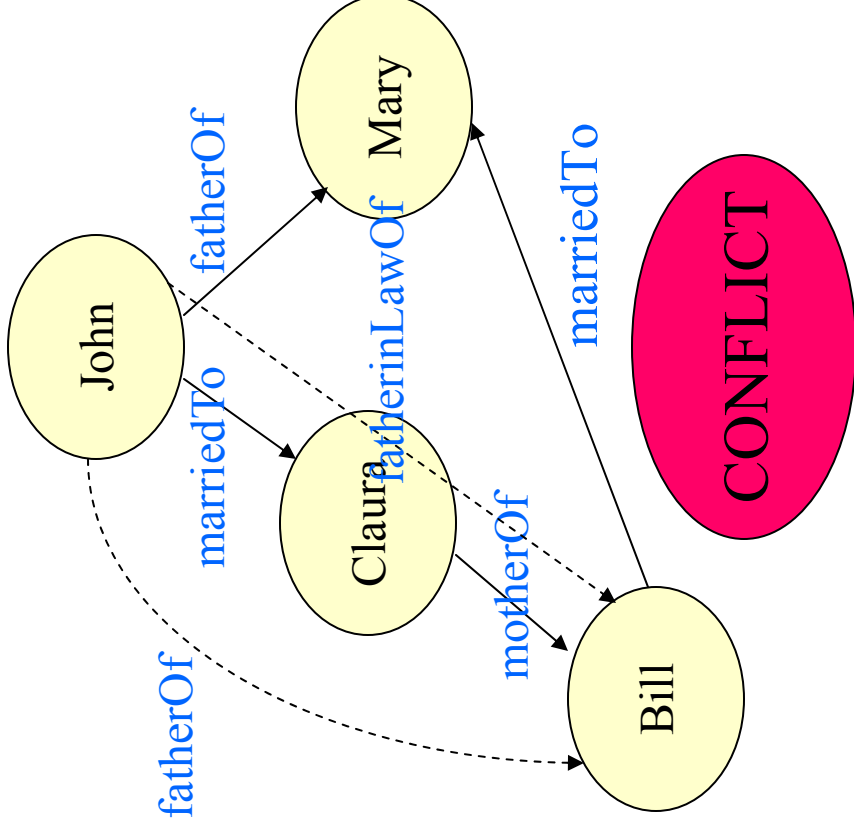
Meta-Data Concerns

- 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

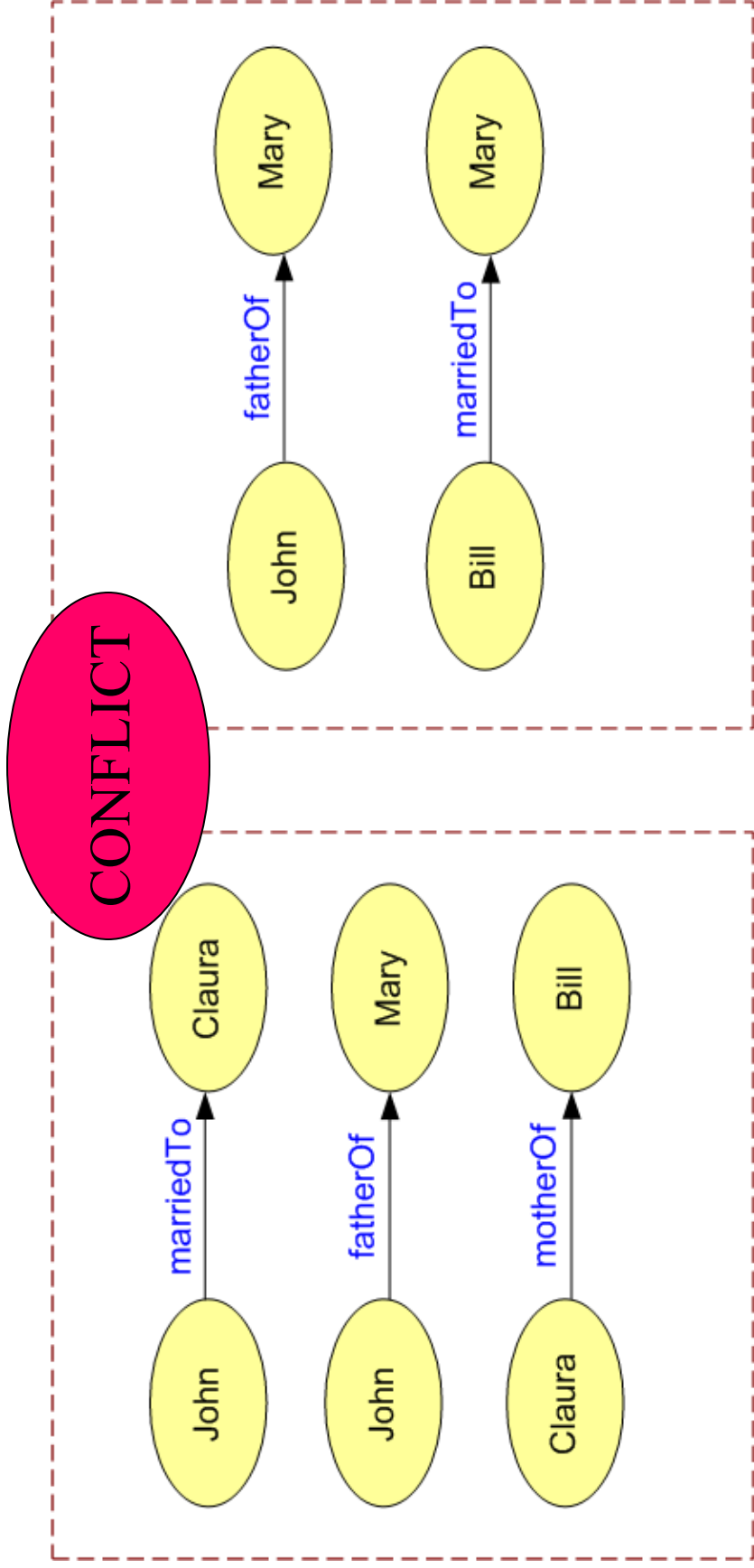
Motivating Factors

- “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 agent 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]

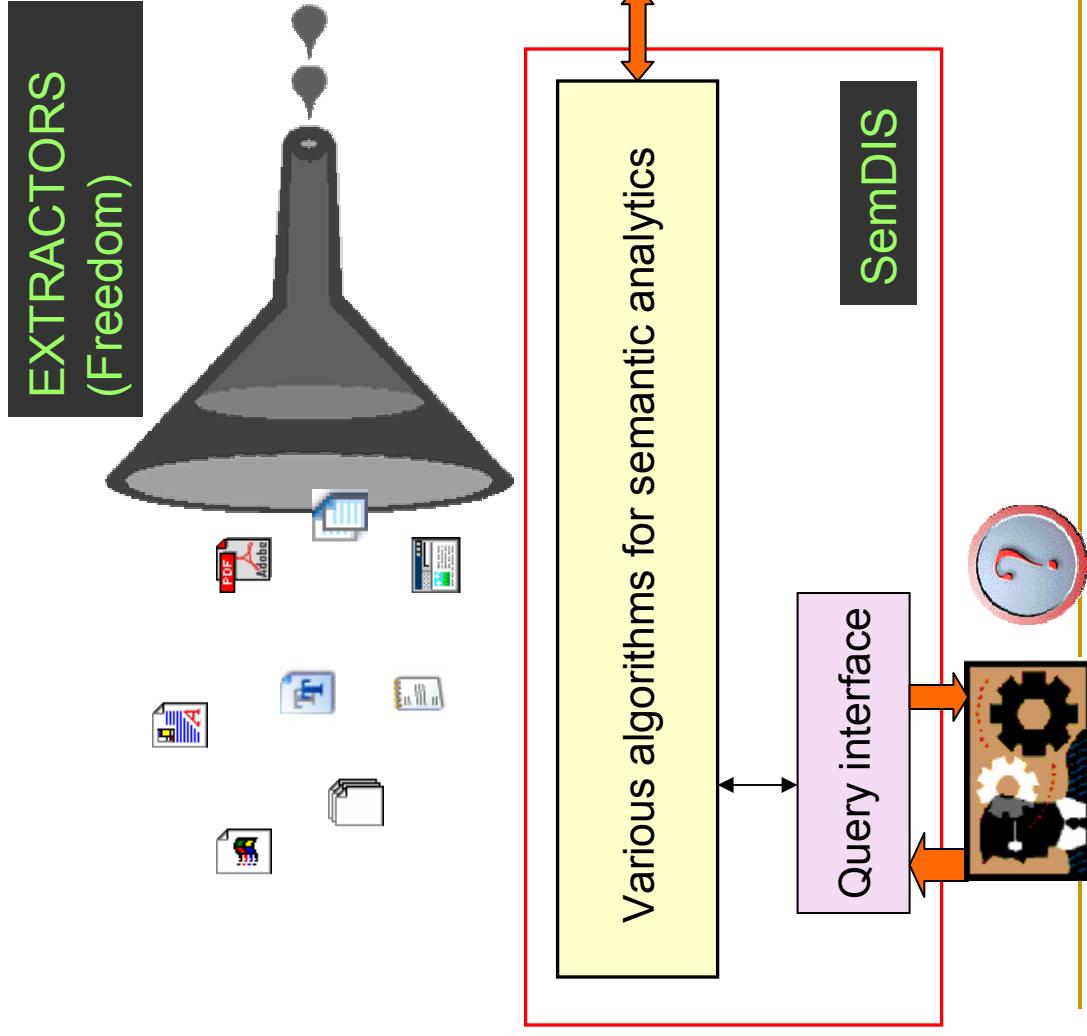
Semantic Conflict Identification



Conflict illustration through Simplification



Motivating Scenario

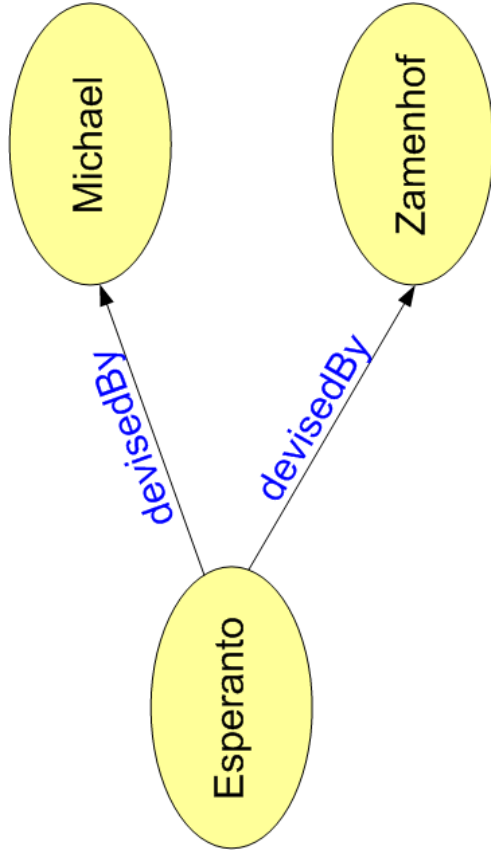
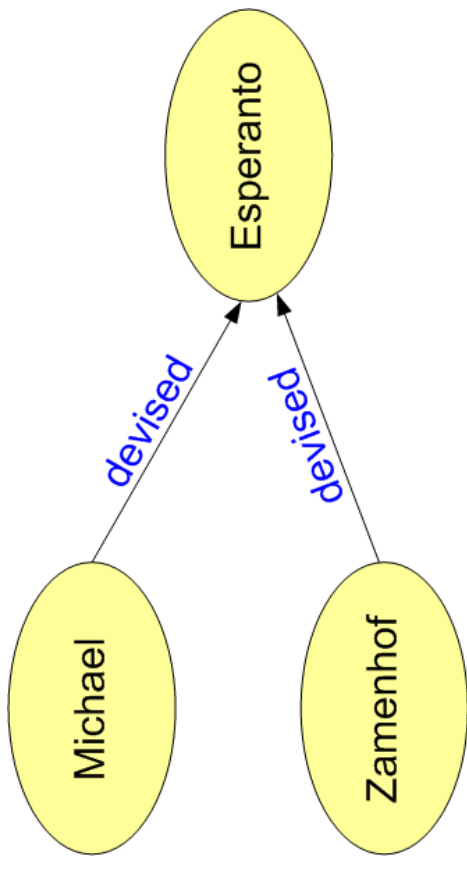


Outline

- 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

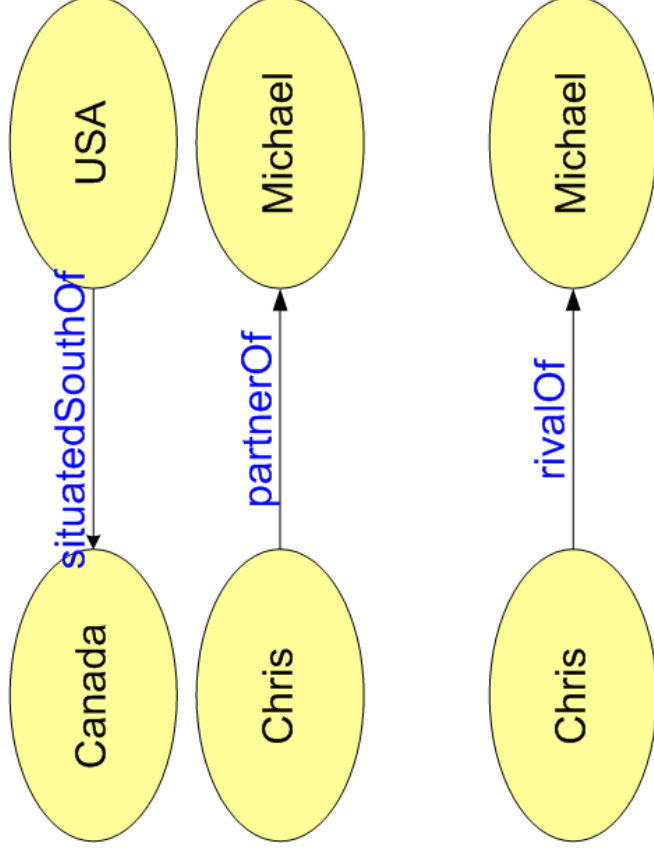


Property Assertion Conflicts

- 'asymmetric' property violation

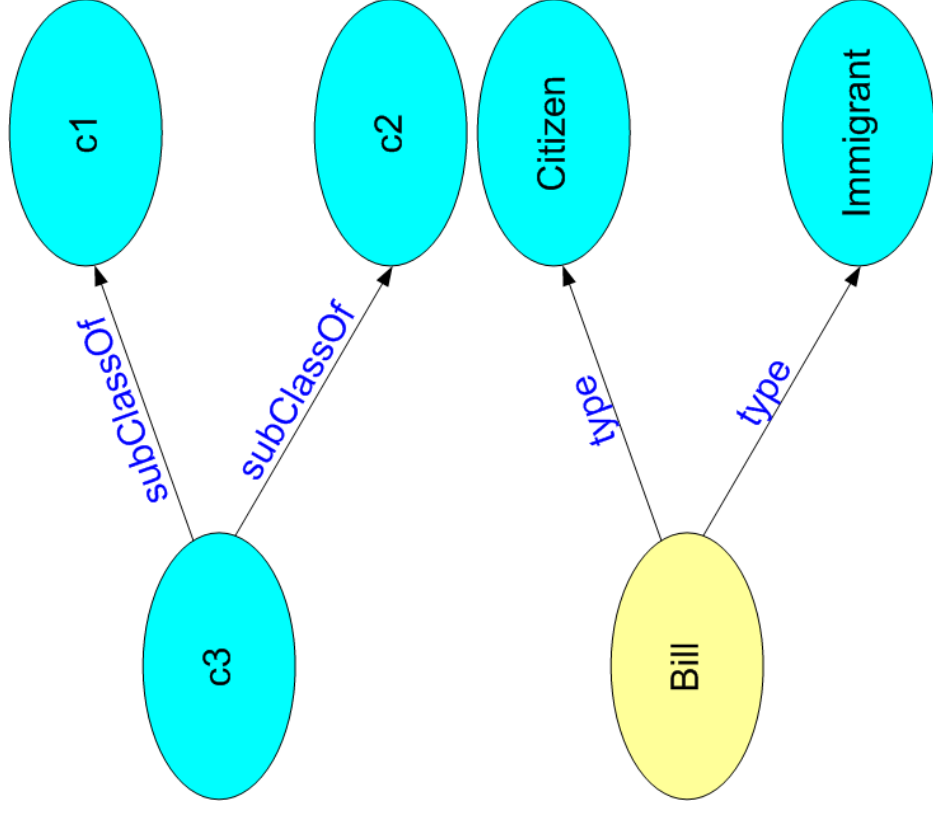


- 'disjoint' property violation



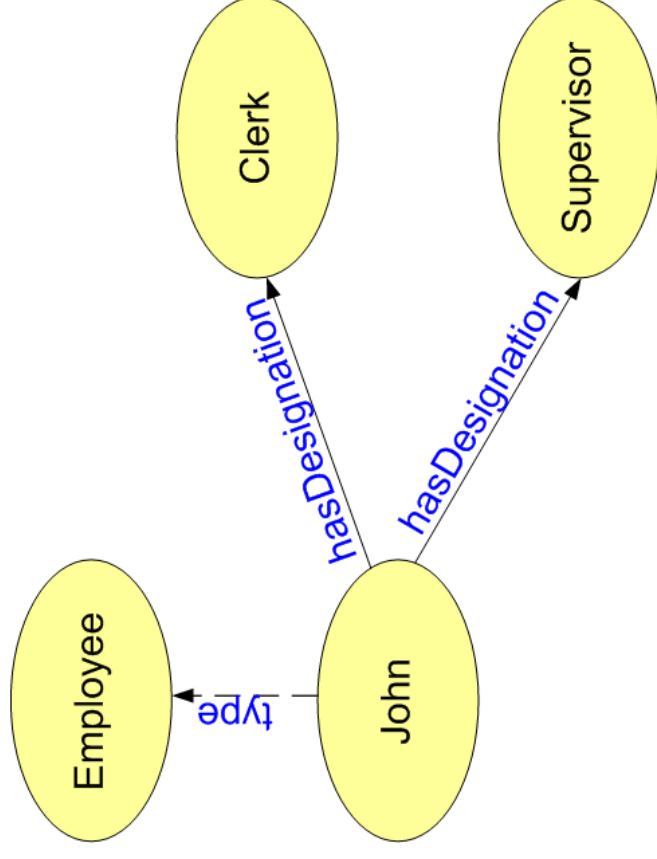
Class Assertion Conflicts

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



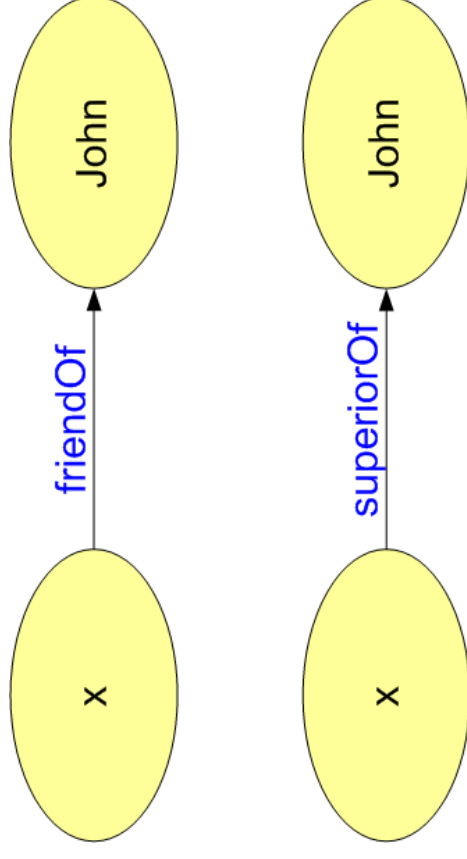
Class Assertion Conflicts

- Class 'Employee' has a restriction on a relation '*hasDesignation*' with a cardinality of '1' that conflicts with the class 'John' which has a restriction on the same relation '*hasDesignation*' with a cardinality of '2'.



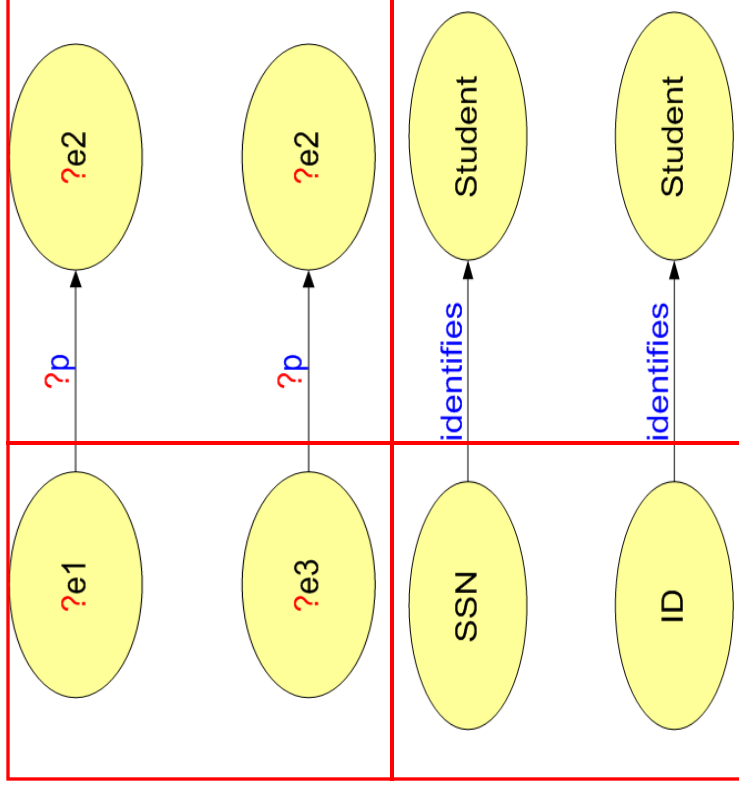
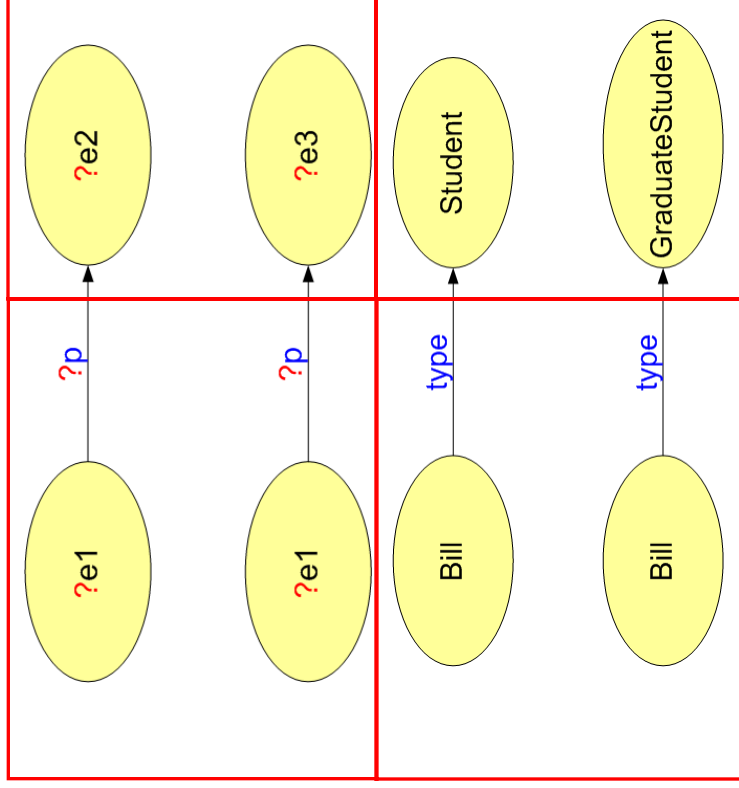
Statement Assertion Conflicts

- 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



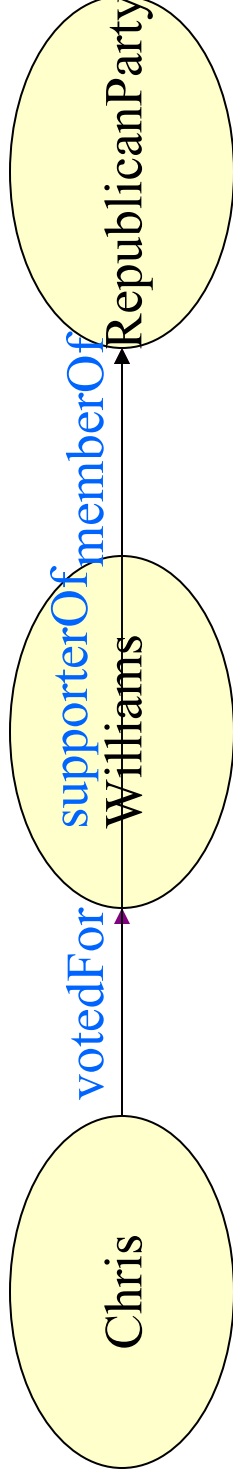
Conflict Definitions

- Two sets of triples T1 and T2 are said to be in *conflict* if their *simplifications* $S(T1) \rightarrow s1$ and $S(T2) \rightarrow s2$ 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 <i>simplification</i>
s	The result of <i>simplification</i> ($S(T) \rightarrow s$)
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,
 $E = \{e_1, e_2 \dots e_n\}$
 $P = \{p_1, p_2 \dots p_m\}$.

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,
 $C = \{(p_1, p_k), \dots, (p_a, p_b, p_c, \dots)\}$.
 $R = \{r_1, r_2 \dots r_n\}$, where $r_1, r_2 \dots r_n$ are results of the **composition**.

The triple $(e_i r_k e_j)$ is a **simplification if $r_k \in R$ and $e_i, e_j \in E$.**

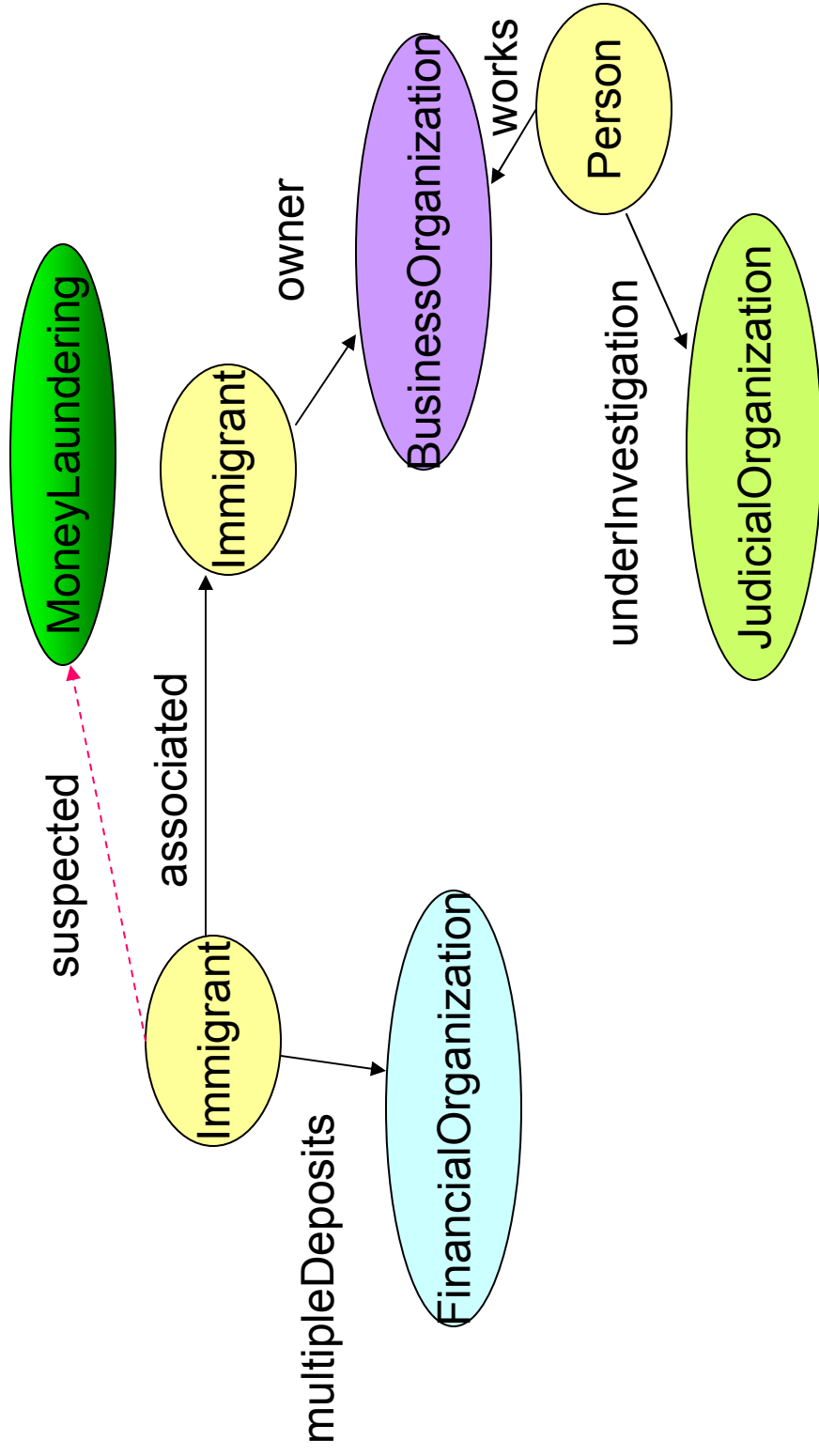
Statement Simplification

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

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

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

Statement Simplification



Defining Statement Simplification Rules

The screenshot shows the 'Summarization Rules Editor' window. The left pane contains the following rule definition:

```
IF
(1) [subject<var1> <eh:ae>]
  property<var1> > [http://idits.cs.uga.edu/proj/semdis/tesbed#supports]
  object<var1> > [http://idits.cs.uga.edu/proj/semdis/tesbed#religion]
(2) [subject<var2> <eh:be>]
  property<var2> > [http://idits.cs.uga.edu/proj/semdis/tesbed#opposes]
  object<var2> > [http://idits.cs.uga.edu/proj/semdis/tesbed#religion]
THEN
NewStatement[<eh:a>] [http://idits.cs.uga.edu/proj/semdis/tesbed#notSameIdeologyAs], <eh:b>]
```

The right pane shows the visual representation of the rule, divided into 'IF' and 'THEN' sections. Each section has three rows of controls: a text input field, a dropdown menu, and a button. The 'IF' section has 'b' in the Subject field, 'opposes' in the Property dropdown, and 'religion' in the Object field. The 'THEN' section has 'a' in the Subject field, 'notSameIdeologyAs' in the Property dropdown, and 'b' in the Object field. Below each section are 'Add', 'Clear', and 'Remove' buttons. A large orange arrow points from the 'IF' section to the 'THEN' section.

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.

if statement(x) and statement(y) and
subject(x,a) and relation($x,rel1$) and
object(x,b) and subject(y,a) and
relation($y,rel2$) and object(y,b) and
→ disjoint($rel1,rel2$)
then conflict(x,y)

Simplification Rules

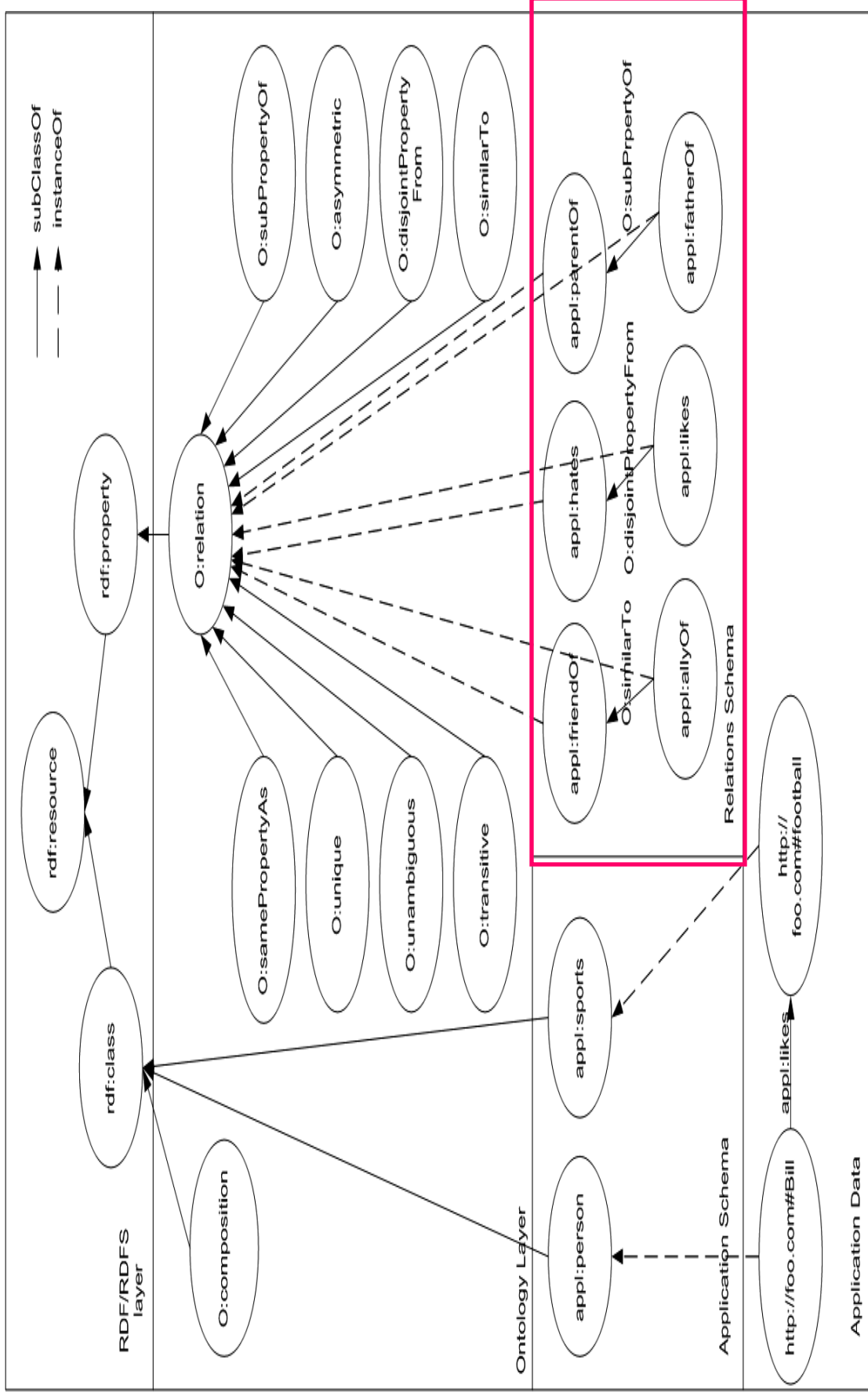
- Can be classified as Production Rules

```
if statement(x) and statement(y) and  
   subject(x,a) and relation(x,rel1) and  
   object(x,b) and subject(y,a) and  
   relation(y,rel2) and object(y,b)  
then newStatement(a, rel3, b)
```

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



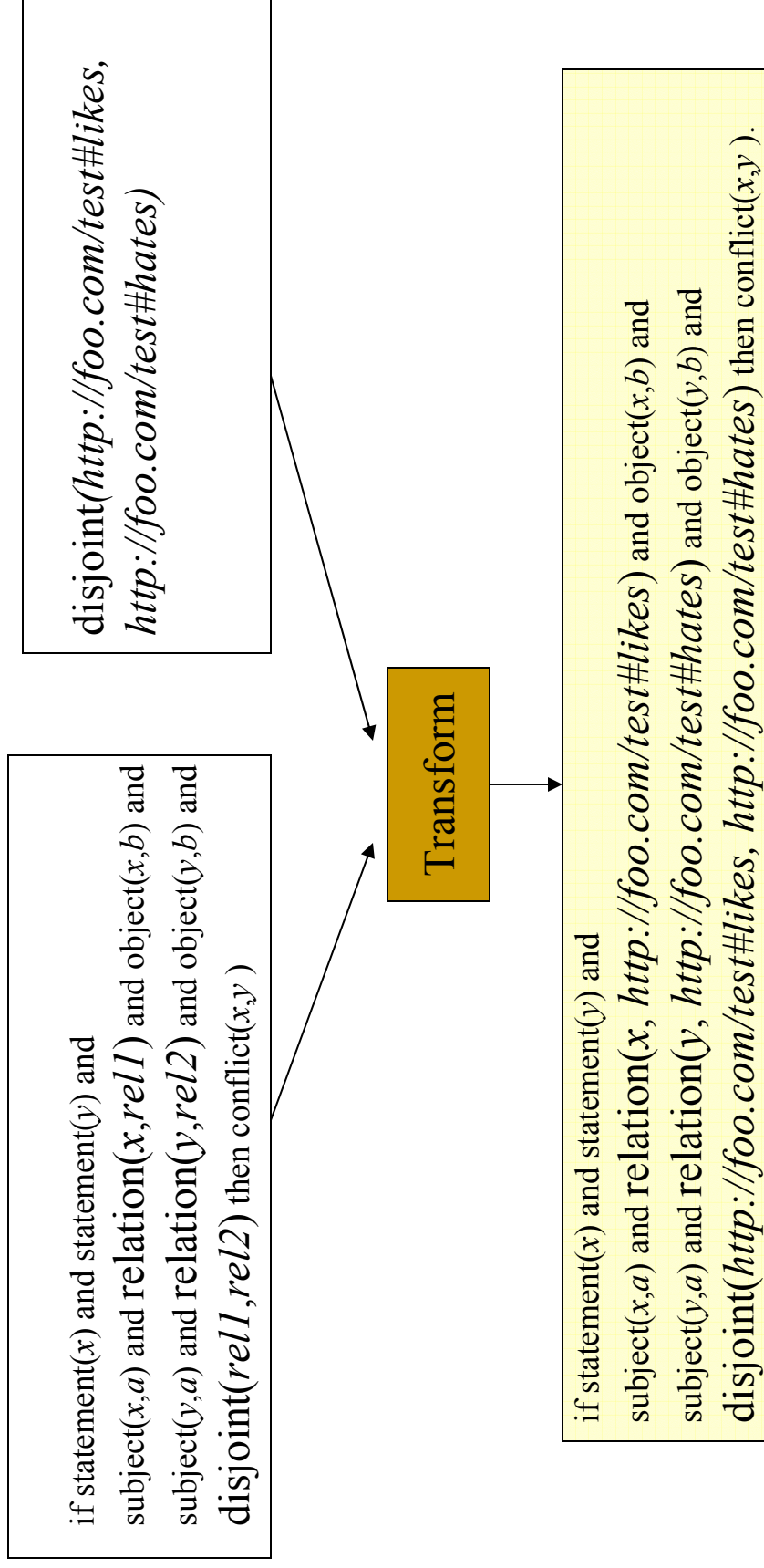
Editing/Populating the Relationship Ontology

The screenshot shows the 'RelationshipOntology Editor' window with three main sections:

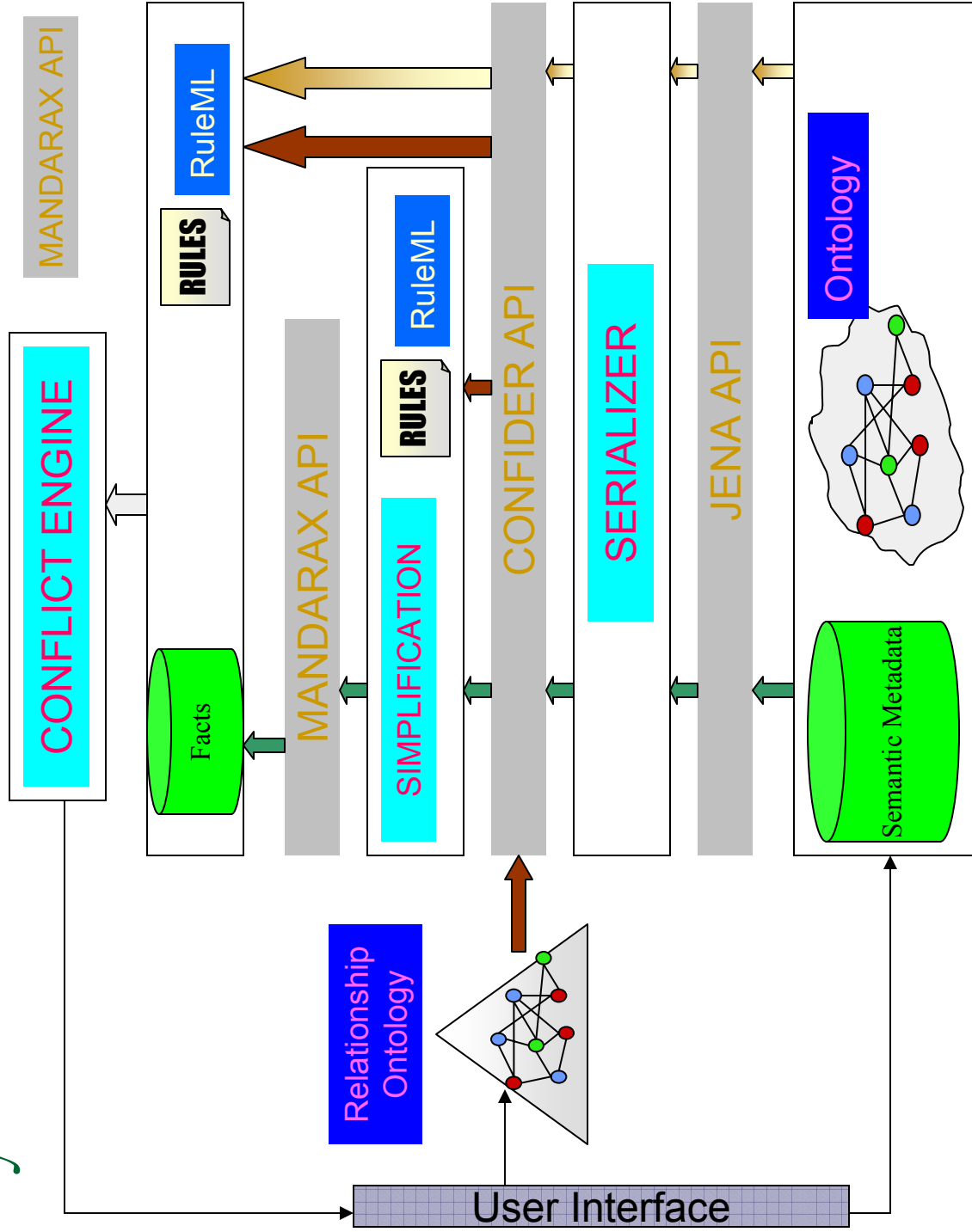
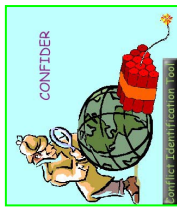
- Add Unary Relation:** Includes a text field for 'Enter the Relation:', a dropdown menu for 'Choose the constraint:' containing the value 'unique', and a text field containing the URI 'http://lsdis.cs.uga.edu/2003/conf#unique'. There are 'Add' and 'Clear' buttons.
- Add Binary Relation:** Includes text fields for 'Enter Relation1:', 'Enter Relation2:', and 'Choose the constraint:'. The dropdown menu contains the value 'samePropertyAs'. The text field contains the URI 'http://lsdis.cs.uga.edu/2003/conf#samePropertyAs'. There are 'Add' and 'Clear' buttons.
- Add Composition Relation:** Includes a text field for 'Result of Composition:' and a large text area for 'Composable Relations(comma delimited):'. There are 'Add', 'Clear', and 'Exit' buttons.

Three orange arrows point to the right of the interface, indicating a sequence of steps.

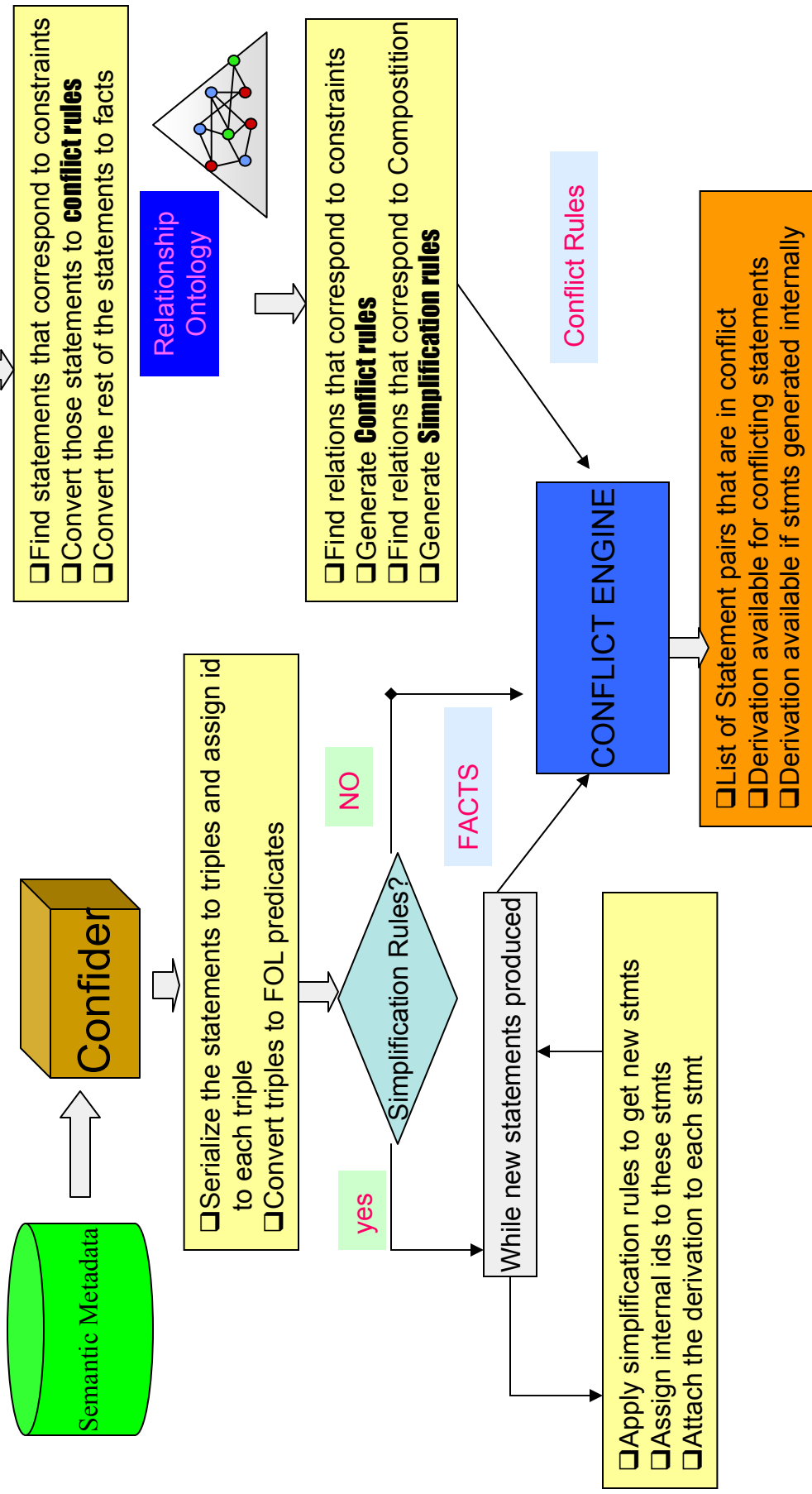
Template Based Rule Transformation



System Architecture



Algorithm



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

Confider-Conflict Identifier

Load the files

Knowledge Results

knowledge
conflict
IF statement(<stmt1 >, <subject>) AND property(<stmt1 >, <object(<stmt1 >, <object1 >, <object2 >) AND statement(<stmt2 >), A)

object
property
statement
subject

- subject(ID_10],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#Publication)
- subject(ID_24],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#Publication)
- subject(ID_49],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1666006)
- subject(ID_50],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1667893)
- subject(ID_51],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1667893)
- subject(ID_52],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1666006)
- subject(ID_53],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1666006)
- subject(ID_54],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1667893)
- subject(ID_55],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1667900)
- subject(ID_58],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1667900)
- subject(ID_59],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#Published_in)
- subject(ID_60],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1666006)
- subject(ID_61],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1667893)
- subject(ID_62],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#Published_in)
- subject(ID_63],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1667893)
- subject(ID_64],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1667900)
- subject(ID_78],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#Publication)
- subject(ID_97],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#Published_in)
- subject(ID_98],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1667900)
- subject(ID_99],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1667893)
- subject(ID_106],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#Publication)
- subject(ID_113],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1667900)
- subject(ID_114],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_1667893)
- subject(ID_129],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#Published_in)
- subject(ID_133],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#Scientific_Publication)
- subject(ID_134],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#Journal)
- subject(ID_137],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#Scientific_Publication)
- subject(ID_138],[http://lsdis.cs.uga.edu/proj/semdis/testbed/#Journal)

Conflict Identification Results

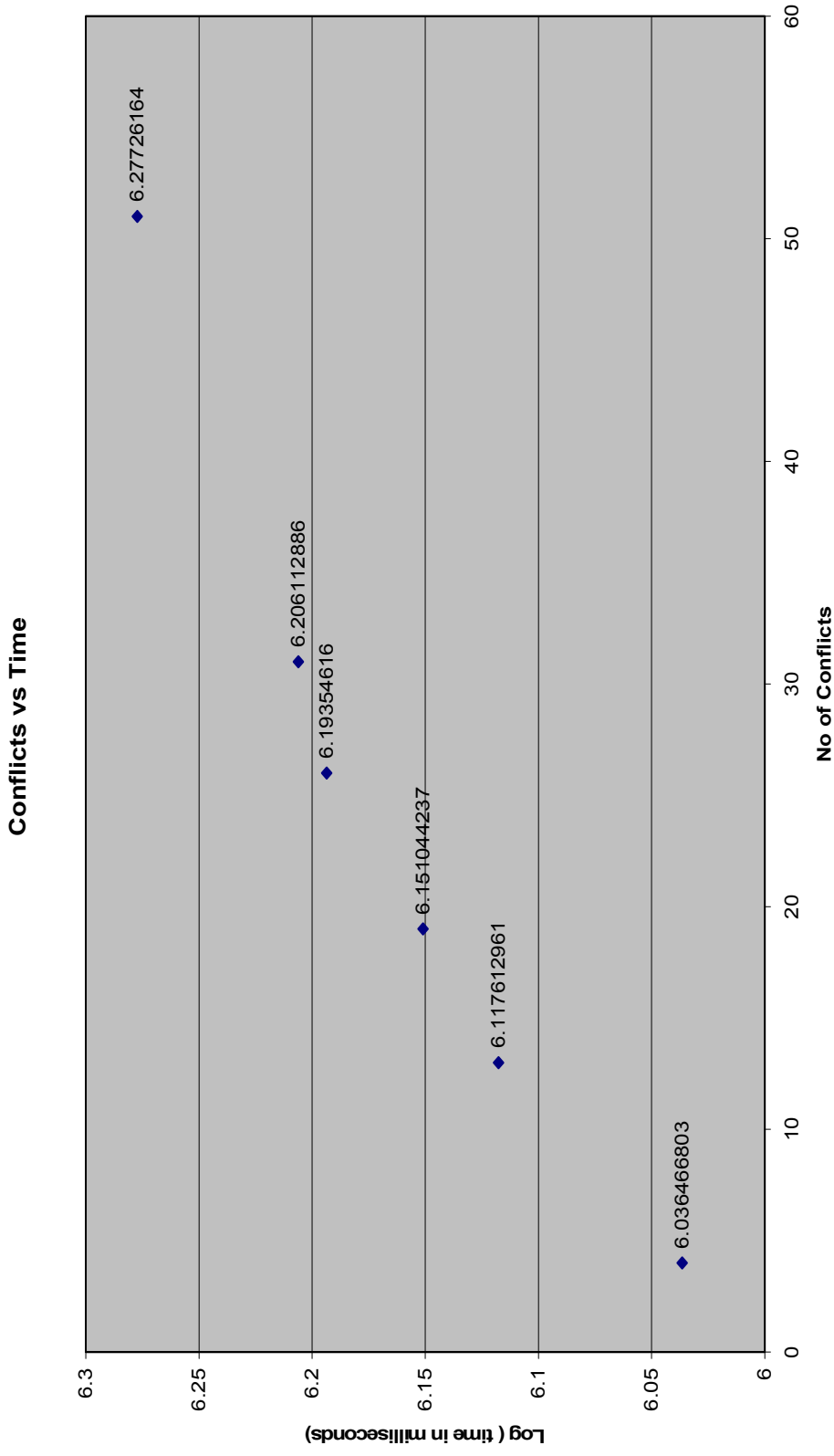
The screenshot displays the 'Confider-Conflict Identifier' application window. The interface is divided into two main panes. The left pane, titled 'Conflicts', shows a tree view with a folder icon and a file icon labeled '<ID_51_ID_55>'. An orange arrow points to this file. The right pane, titled 'Derivation', shows a hierarchical tree view of logical expressions. An orange arrow points to the root expression: 'IF statement(<stmt1> AND subject(<stmt1> <...> AND property(<stmt1> {http://isdls.cs.uga.edu/proj/semdis/testbed#Publish...})'. Below this, the tree expands to show 'statement(ID_51)', 'subject(ID_51){http://isdls.cs.uga.edu/proj/semdis/testbed#SWEET_1667893}', 'property(ID_51){http://isdls.cs.uga.edu/proj/semdis/testbed#Published_in}', 'object(ID_51){http://isdls.cs.uga.edu/proj/semdis/testbed#SWEET_1666006}', 'statement(ID_55)', 'subject(ID_55){http://isdls.cs.uga.edu/proj/semdis/testbed#SWEET_1667893}', 'property(ID_55){http://isdls.cs.uga.edu/proj/semdis/testbed#Published_in}', and 'object(ID_55){http://isdls.cs.uga.edu/proj/semdis/testbed#SWEET_1666007}'. The application window includes a standard Windows-style title bar with minimize, maximize, and close buttons, and a toolbar with icons for file operations and navigation.

Statement Provenance

The screenshot displays the 'Statement Provenance' tool interface. At the top, the title bar reads 'Statement Provenance'. Below the title bar, the 'Statement ID:' field contains 'INT_ID_95'. To the right of this field is a 'submit' button. Below the 'Statement ID:' field, there are four fields: 'Source:', 'Subject:', 'Property:', and 'Object:'. The 'Source:' field contains 'Internal'. The 'Subject:', 'Property:', and 'Object:' fields contain the same URL: 'http://orlando.drc.com/dam/ontology/Bibliographic3.1/Bibliogra...'. Below these fields is a large text area containing a provenance tree. The tree starts with 'Derivation' and branches into several nodes, including 'IF statement(=var1 =>_set(=)) AND property(=v...)', 'statement(ID_51())', 'subject(ID_51)[http://orlando.drc.com/dam/ontology/Bibliographic3.1/Bibliographic-ont#ABriefHistoryOTime]', 'property(ID_51)[http://orlando.drc.com/dam/ontology/Bibliographic3.1/Bibliographic-ont#opposeS]', 'object(ID_51)[http://orlando.drc.com/dam/ontology/Bibliographic3.1/Bibliographic-ont#religion()]', 'statement(ID_56())', 'subject(ID_56)[http://orlando.drc.com/dam/ontology/Bibliographic3.1/Bibliographic-ont#StoryC...', 'property(ID_56)[http://orlando.drc.com/dam/ontology/Bibliographic3.1/Bibliographic-ont#...', and 'object(ID_56)[http://orlando.drc.com/dam/ontology/Bibliographic3.1/Bibliographic-ont#...'. Four orange arrows point to specific elements: one points to the 'Statement ID:' field, one points to the 'Source:' field, one points to the 'Derivation' node, and one points to the 'IF statement(=var1 =>_set(=)) AND property(=v...' node.

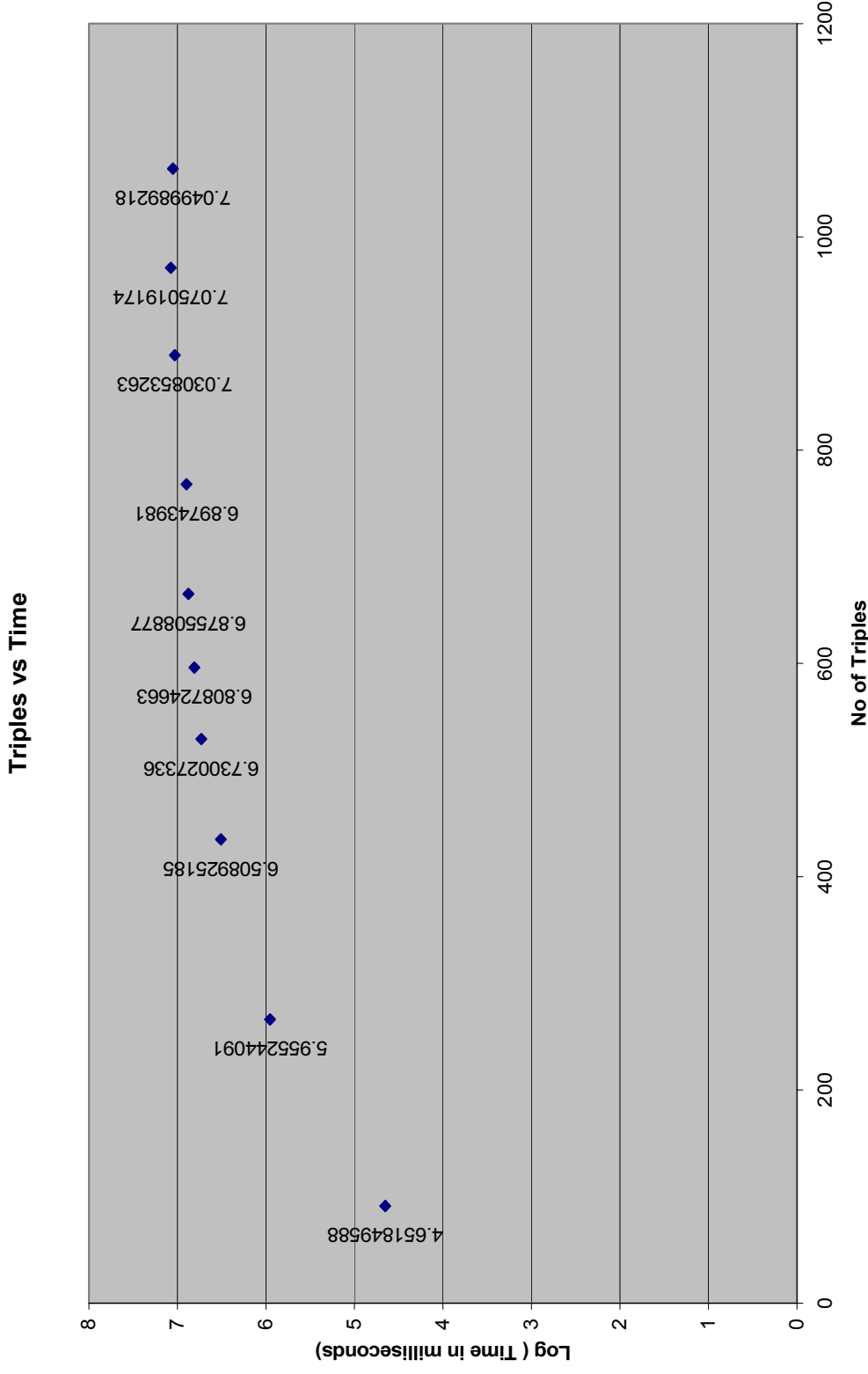
Performance Evaluation (1)

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:

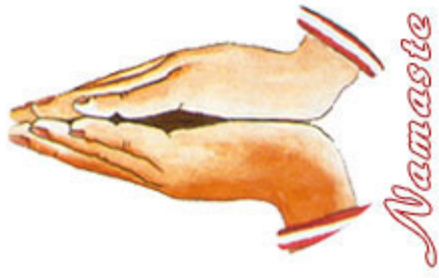
- Defined conflicts in semantic meta-data and classified them.
- Discussed a rule-based approach to identify the conflicts.
- Shown the use of relations between relations to simplify the triples and identify conflicts.
- Demonstrated the applicability of the approach over a limited data set using a prototype.

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