



# Shuttle: Current State and Next Generations

Presented by Kao, Chen-yi



# Outline

- **Current Shuttle**
  - Motivation
  - Related research
  - Concept synchronization
  - Linking model elements and concepts
  - Text boundary detection
  - GMF integration
- Short-term plan (within a year)
- Long-term plan



# Current Shuttle

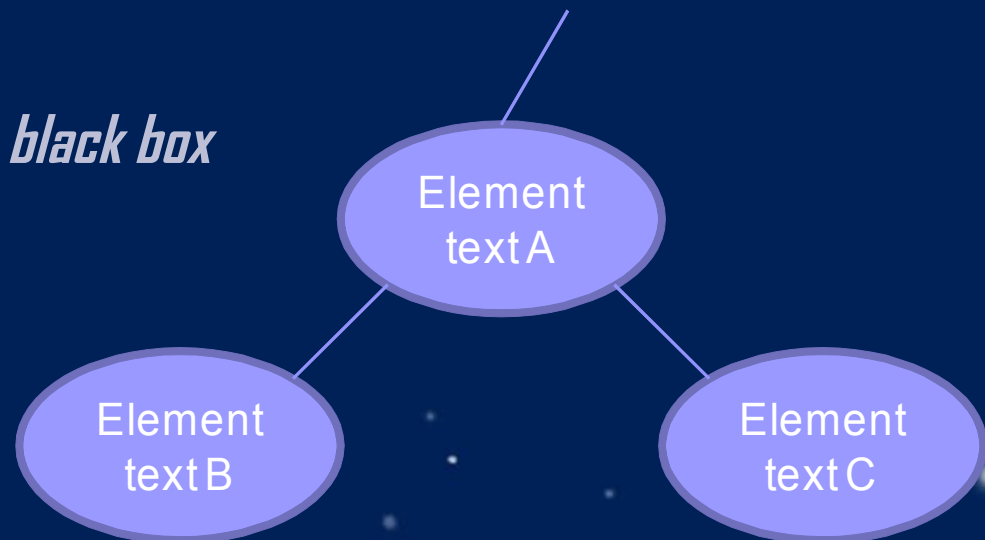
- An Instant Model Synchronization Assistant for GMF Editors Based on Concept Synchronization
- Motivation: modern model synchronization still needs...
  - Instant (better automatic and intuitive) round-trip engineering *anytime everywhere*
    - ★ NOT ONLY in synchronizing GUI/Java or UML/Java, etc., but generally ALL kinds of models
  - Refactoring beyond exact string matching
    - ★ E.g. renaming variable – in variable name and comment there may be different occurrences of the equivalent term.



# Related research

- Graph/Model Transformation
  - AGG, CODEX, UMLX, ATL, etc.
  - Another (transformation) language to learn
  - Did not *automatically* transform graphs/models according to element text

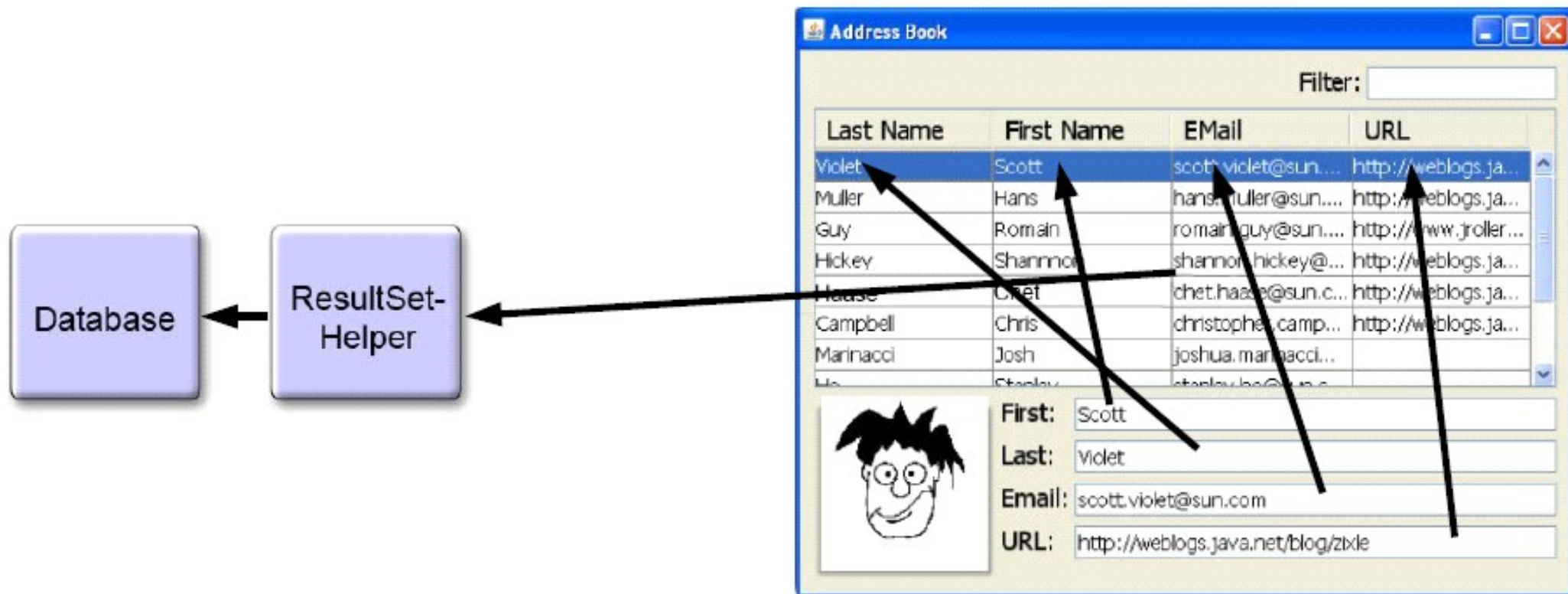
★ *Treating element text as black box*





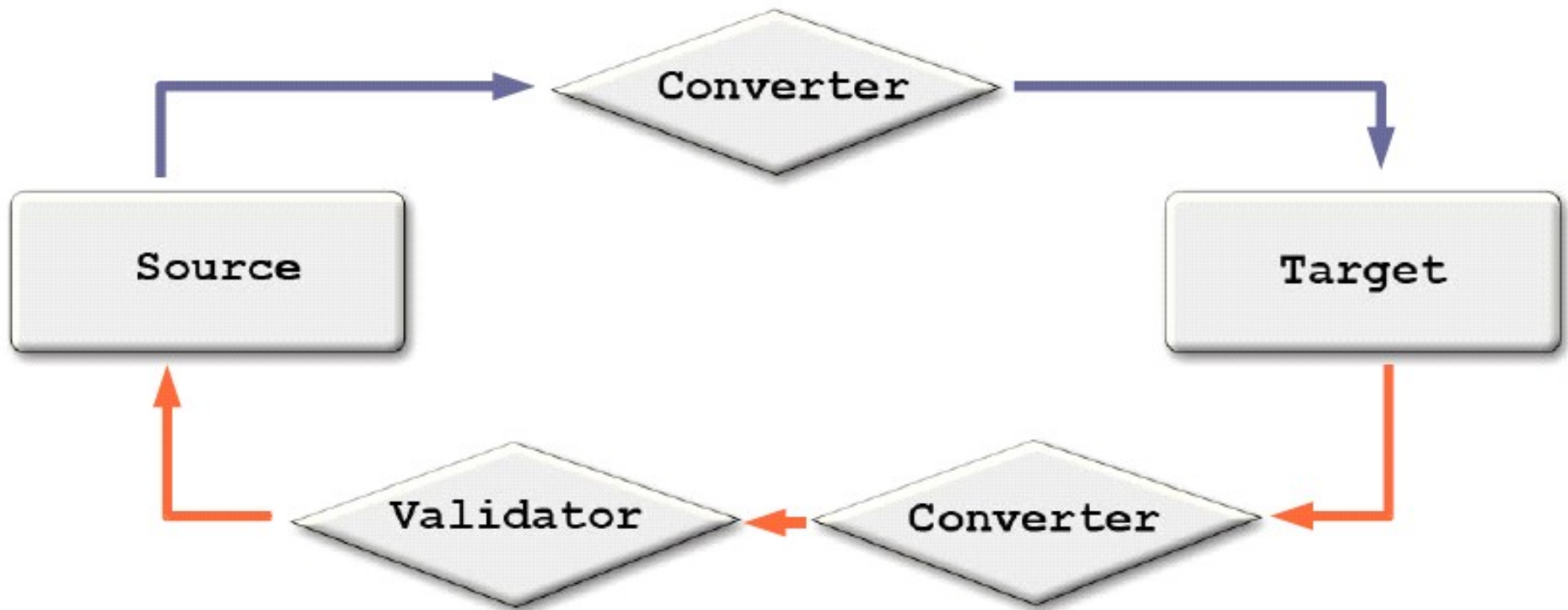
## Related research (cont.)

- Data Binding
- Beans Binding (JSR 295), JFace Data Binding





- Assign binding, then synchronization is automatic





# Concept synchronization

- **Ontology**
  - ...is often conceived as a set of classes (*concepts*), relations, functions, axioms and instances (Gruber, T.R., 1993)
- **Information Retrieval**
  - *Concept search* – beyond keyword search
  - **Concept** – semantically related keywords





## Concept synchronization (cont.)

# WordNet

WordNet Search - 3.0 - [WordNet home page](#) - [Glossary](#) - [Help](#)

Word to search for:

### Verb

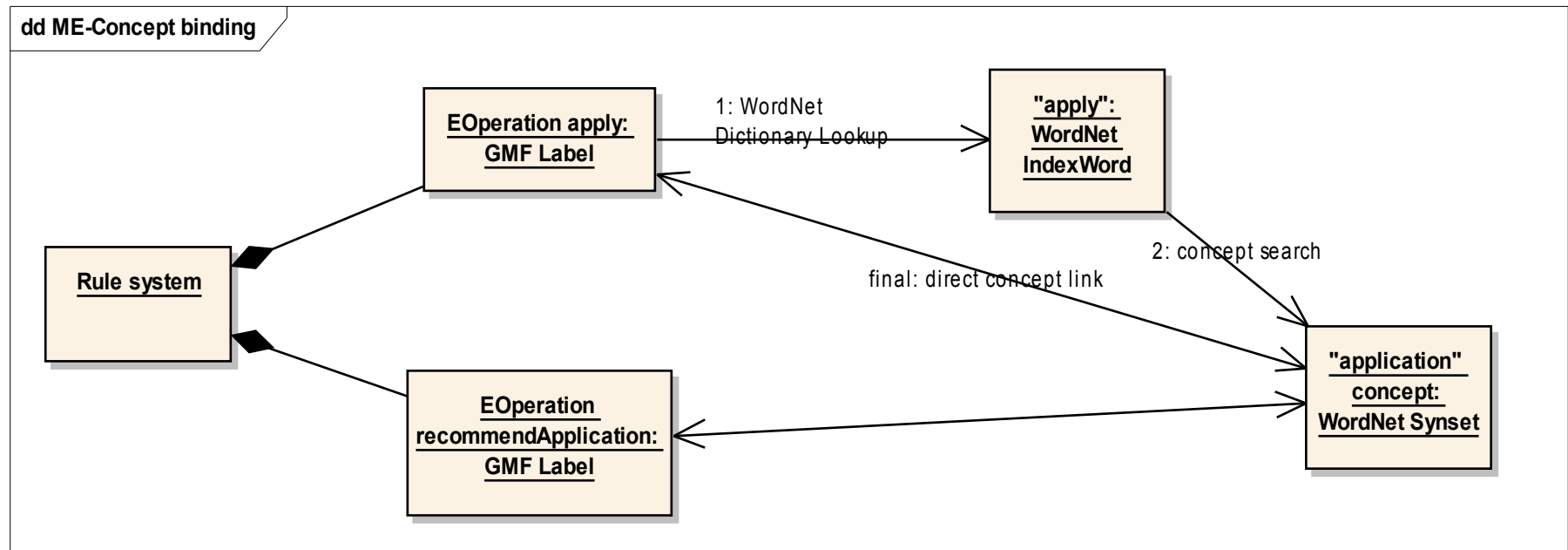
- **S: (v)** **use**, **utilize**, **utilise**, **apply**, **employ** (put into service; make work or employ for a particular purpose or for its inherent or natural purpose) *"use your head!"*; *"we only use Spanish at home"*; *"I can't use this tool"*; *"Apply a magnetic field here"*; *"This thinking was applied to many projects"*; *"How do you utilize this tool?"*; *"I apply this rule to get good results"*; *"use the plastic bags to store the food"*; *"He doesn't know how to use a computer"*
- **S: (v)** **apply**, **hold**, **go for** (be pertinent or relevant or applicable) *"The same laws apply to you!"*; *"This theory holds for all irrational numbers"*; *"The same rules go for everyone"*
- **S: (v)** **apply** (ask (for something)) *"He applied for a leave of absence"*; *"She applied for college"*; *"apply for a job"*
  - *direct troponym / full troponym*
  - *direct hypernym / inherited hypernym / sister term*
  - *derivationally related form*
    - **W: (n)** **application** [Related to: **apply**] (a verbal or written request for assistance or employment or admission to a school) *"December 31 is the deadline for applications"*
    - **W: (n)** **applicant** [Related to: **apply**] (a person who requests or seeks something such as assistance or employment or admission)
    - **W: (n)** **applier** [Related to: **apply**] (a person who requests or seeks something such as assistance or employment or admission)
  - *sentence frame*
- **S: (v)** **put on**, **apply** (apply to a surface) *"She applied paint to the back of the house"*; *"Put on make-up!"*





## Concept synchronization (cont.)

- By concept search, Web pages of the same concept can be found together, and how about *model elements*?
- Shuttle: Concept based on *WordNet Synset*





# Linking model elements and concepts

- MIT Java WordNet Interface (JWI)

- From String to Synset

- 1. WordNet Dictionary Lookup

- \* `edu.mit.jwi.dict.IDictionary`

- \* `IIndexWord getIndexWord(java.lang.String lemma, PartOfSpeech pos);`

- 2.1 Concept Search - *Basic Concept Search*

- 1. `edu.mit.jwi.item.IIndexWord`

- \* `IWordID[] getWordIDs();`

- 2. `edu.mit.jwi.item.IWordID`

- \* `ISynsetID getSynsetID();`



## From String to Synset (cont.)

### • 2.2. Extended Concept Search

#### • `edu.mit.jwi.item.IWord`

1. `IWordID[] getRelatedWords()` ;

★ Words in other part-of-speeches (POS, one of NOUN, VERB, ADJECTIVE and ADVERB)

#### 2. And like Basic Concept Search

#### • `edu.mit.jwi.item.ISynset`

★ `ISynsetID[] getRelatedSynsets()` ;

★ Hypernym (泛義／上位詞) , Hyponym (特義／下位詞) , Meronym (成分詞) , Holonym (合成詞) , ... Synsets under the same POS

→ To resolve `ISynset` from `ISynsetID`:

→ `ISynset edu.mit.jwi.dict.IDictionary.getSynset(ISynsetID)`



# Text boundary detection

## • `java.text.BreakIterator`

### 1. Sentence boundary analysis

```
1. BreakIterator BreakIterator.getSentenceInstance()  
2. void BreakIterator.setText(String newText)  
3. int BreakIterator.next()  
4. String String.substring(int beginIndex, int  
    endIndex)
```

### 2. Word boundary analysis

```
1. BreakIterator BreakIterator.getWordInstance()  
2. ...
```

|Your|balance|is|\$1,234.56|...|I|think|.

This period is part of the number  
because it has a digit on either side.

This period is not. It only has a digit  
on one side.



## Text boundary detection (cont.)

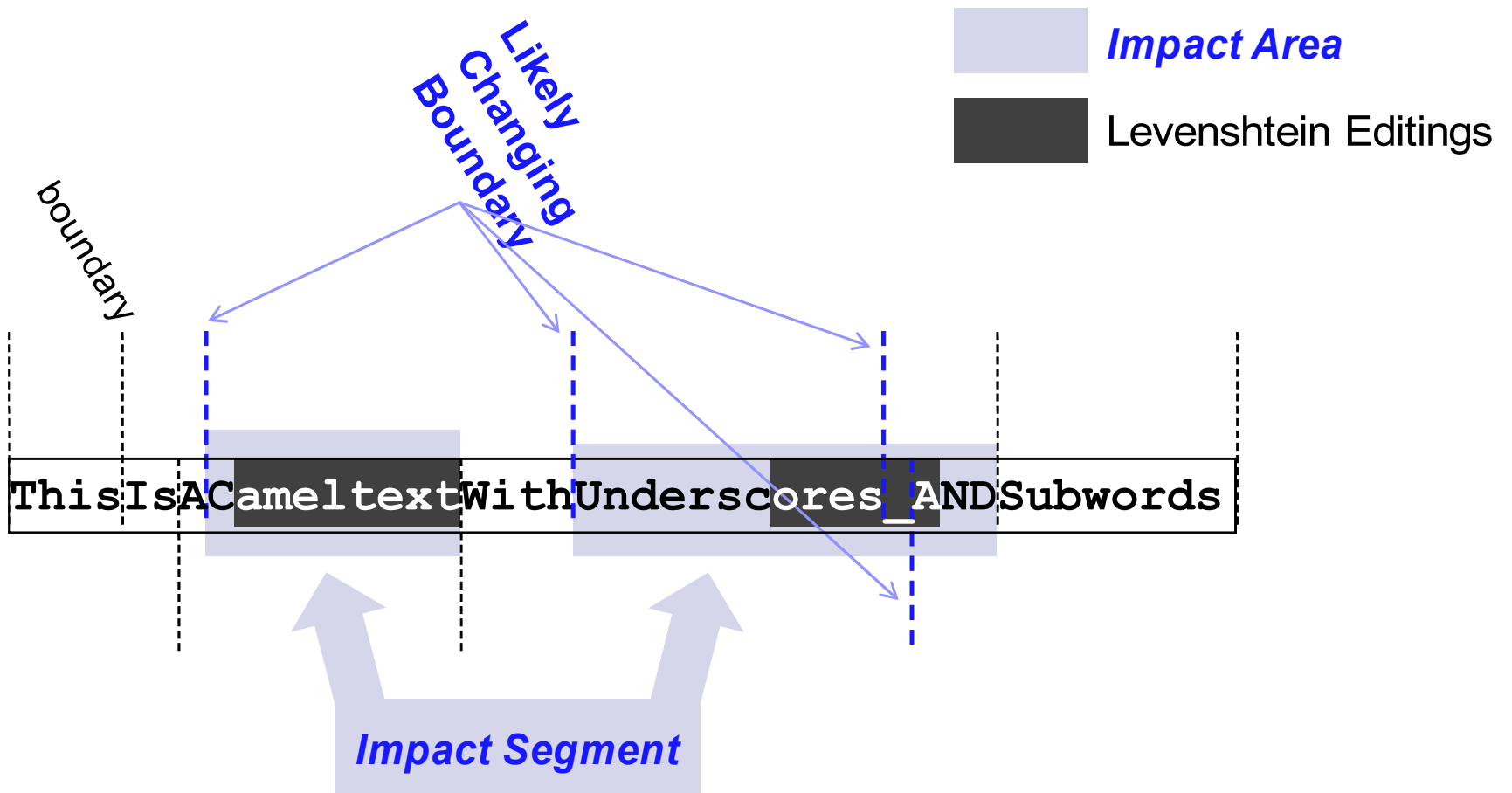
- To handle compound words with **CamelCase/underscore**
- **We design** CompoundWordBreakIterator
  - ★ **Sub-classing** BreakIterator



```
camelCaseLooksLikeThis  
lowerCamelCaseLooksTheSame  
UpperCamelCaseLooksLikeThis
```



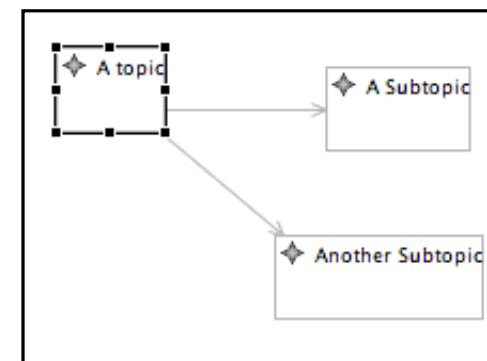
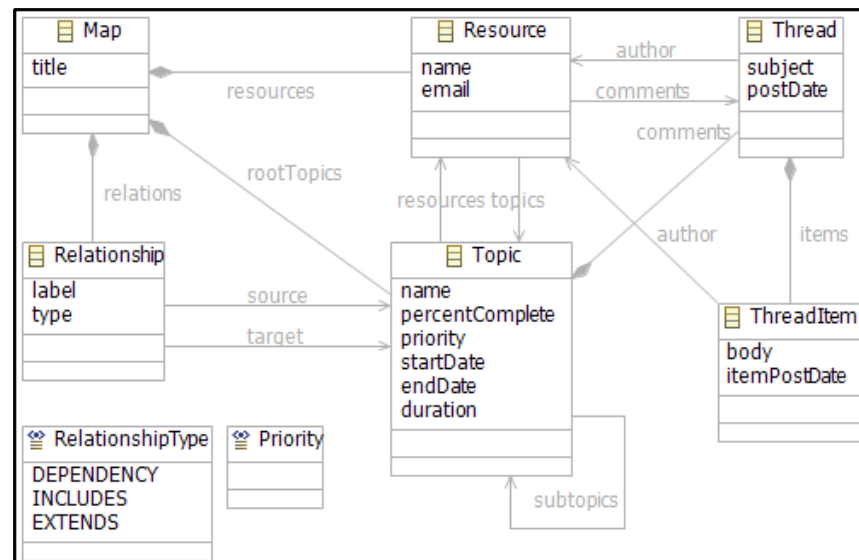
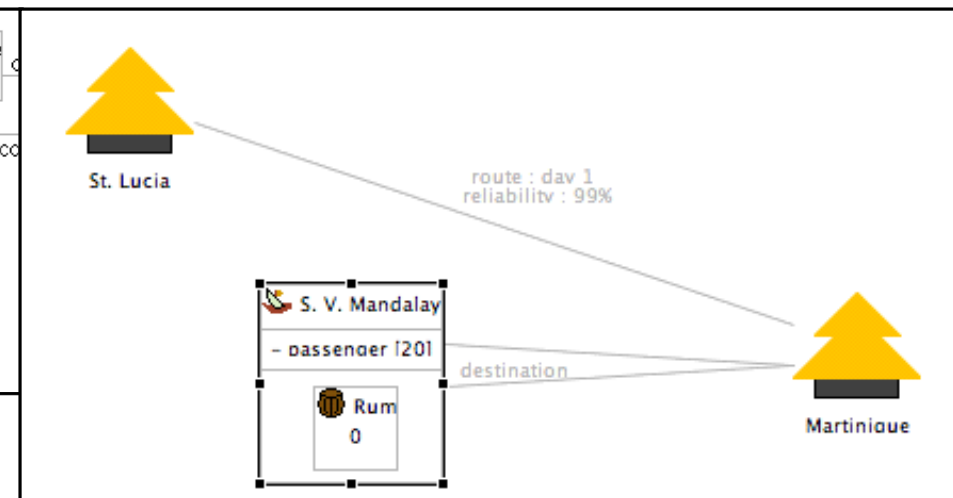
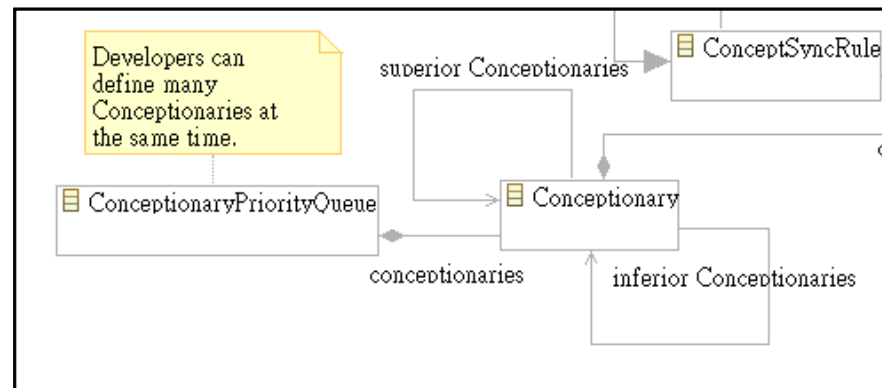
# When model element text is modified...





# GMF integration

## ● GMF Labels...



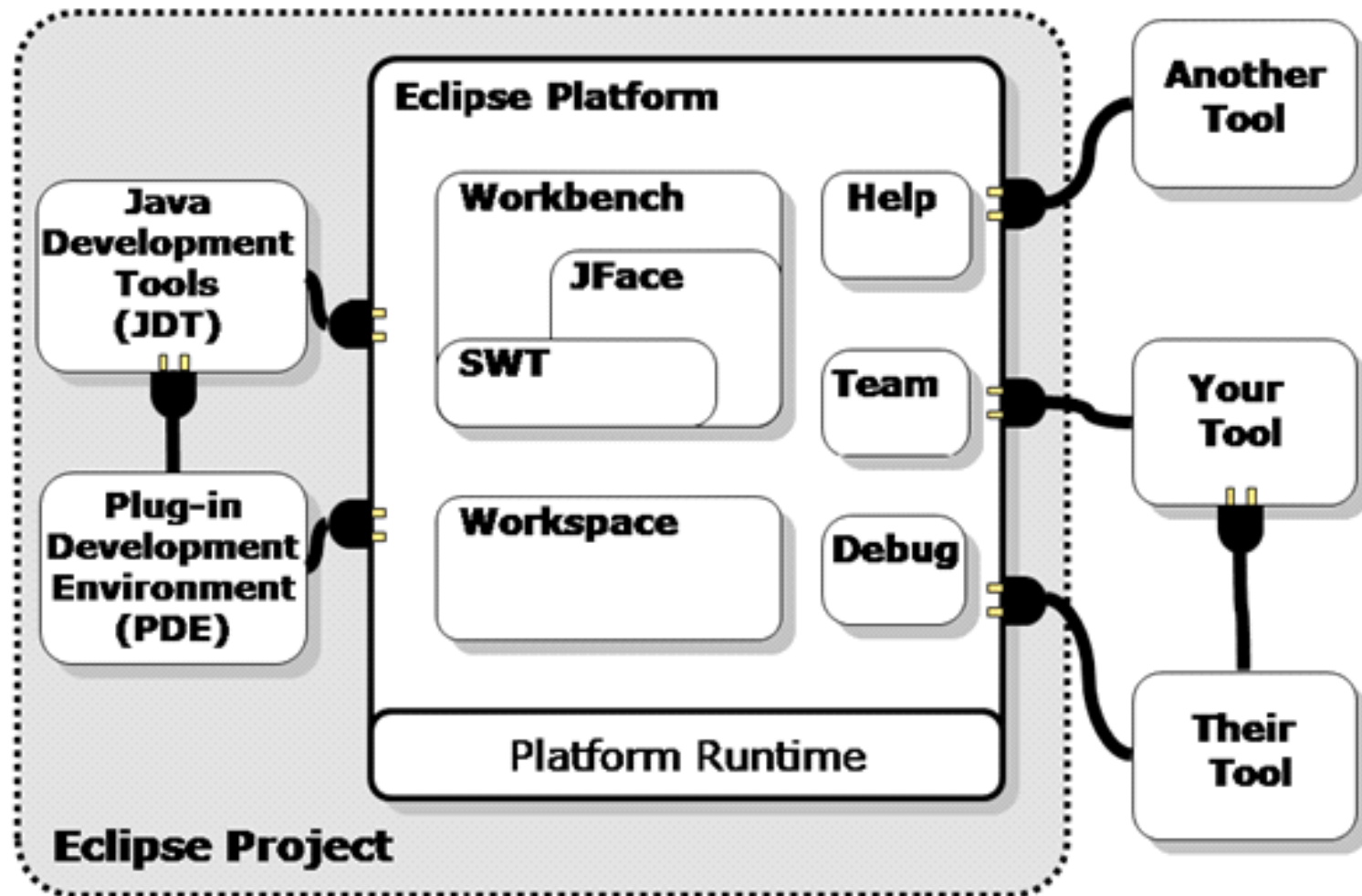




GMF integration (cont.)

# Eclipse Plug-ins and Extension Points

- GMF: plug-in of Eclipse; Shuttle: plug-in of GMF





# Outline

- Current Shuttle
- Short-term plan (within a year)
  - Related works
  - Client-side
    - ★ (Structural) rule inference result recommendation (RIRR) visualization
  - Server-side
    - ★ Light-weight rule-base
- Long-term plan

# Related works

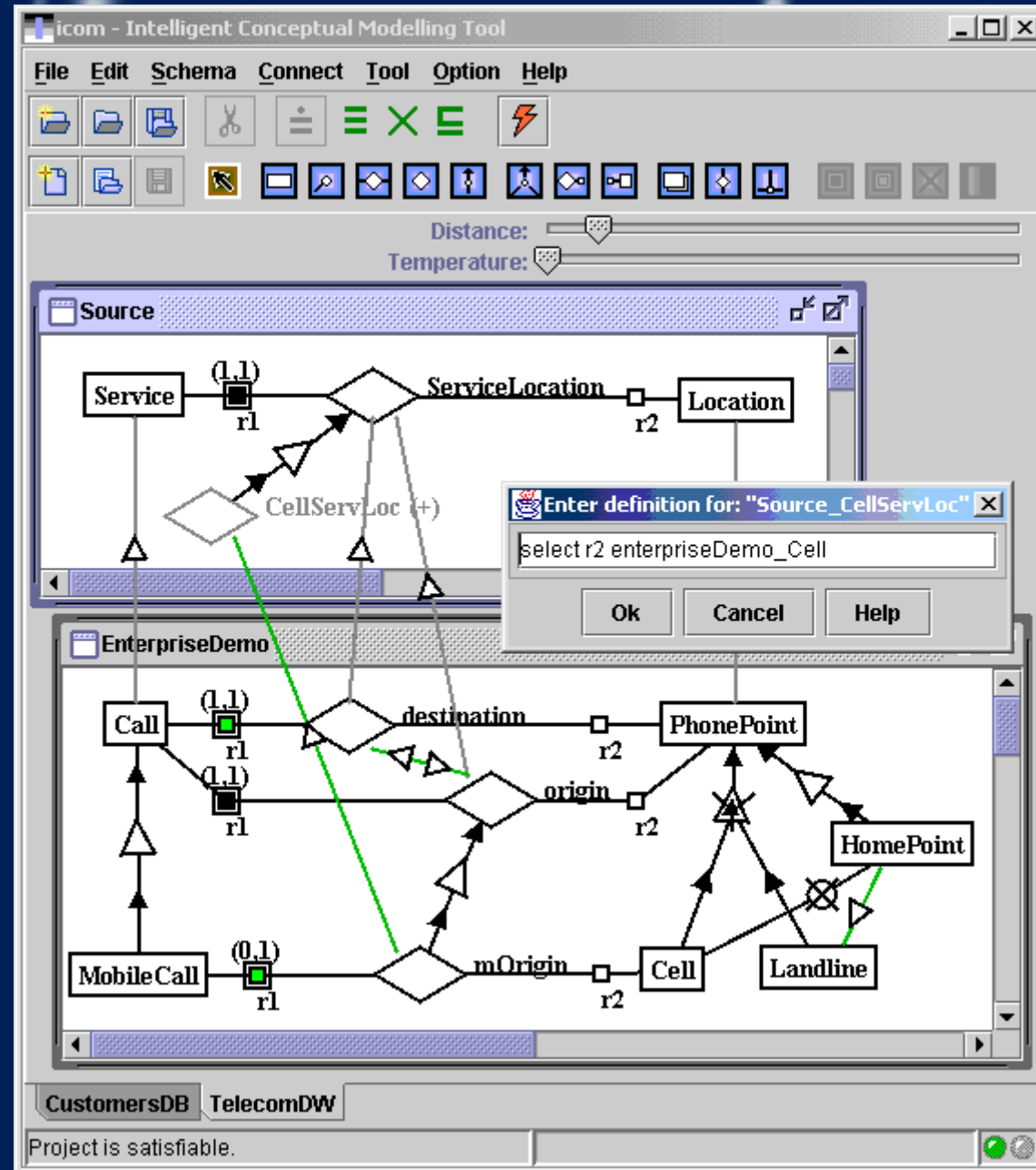
- LaSSIE (Devanbu et al., 1991)

- Low "semantic noise" but *Manually* connecting domain and code models

- ICOM (

<http://www.inf.unibz.it/~franconi/icom/>)

- Complete logical reasoning to *verify the specification, infer implicit facts, devise stricter constraints, and manifest any inconsistency.*





# Client-side: RIRR visualization

- Reduce semantic noise
- Extended from concept linking...

1. OWL ontology linking

2. SWRL rule triggering

★ `hasParent(?x1,?x2) ∧ hasBrother(?x2,?x3) ⇒ hasUncle(?x1,?x3)`

3. SWRL rule inference

★ Direct inference

★ Indirect inference by KAON2(<http://kaon2.semanticweb.org/>) reasoner

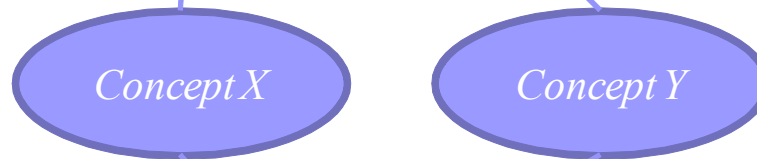
4. RIRR visualization

★ Case by case...



# Link Models and Ontologies

```
*/  
class CompoundWordBreakIterator extends BreakIterator {  
    final private BreakIterator WORD_BREAK_ITERATOR = BreakIterator.get  
    private String subjectText = null;  
}
```



```
<rdf:subClassOf rdf:resource="#Individual" />  
</owl:Class>  
<owl:Class rdf:ID="CompoundWord">  
    <rdf:label xml:lang="en">compound words</rdf:label>  
    <rdf:comment>NOTE: THIS TERM AND ASSERTIONS THAT USE IT ARE NOT STRICTLY PART OF CYC  
    SENSE KNOWLEDGE. RATHER, THEY ARE PART OF CYC'S NL SYSTEM, WHICH IS USED TO RELATE  
    LANGUAGE TO CYC'S COMMON-SENSE KNOWLEDGE. SEE #$$SharedNoteOnOpenCycNLConstants F  
    INFORMATION.</rdf:comment>  
    <guid>bf879415-9c29-11b1-9dad-c379636f7270</guid>  
    <rdf:type rdf:resource="#NLMorphologyCollection" />  
    <rdf:type rdf:resource="#LexicalWordType" />  
    <rdf:type rdf:resource="#NLWordType" />  
</owl:Class>
```



# SWRL rule consequent may be...

- atom ::= description '(' i-object ')'
  - | dataRange '(' d-object ')'
  - | individualvaluedPropertyID '(' i-object i-object ')'
  - | datavaluedPropertyID '(' i-object d-object ')'
  - | sameAs '(' i-object i-object ')'
  - | differentFrom '(' i-object i-object ')'
  - | builtIn '(' builtinID { d-object } ')'



# Server-side: light-weight rule-base

- For rule persistence and reuse
- Based on Protégé (<http://protege.stanford.edu/>)
- Indexed by concept to accelerate ontology linking





# Outline

- **Current Shuttle**
- **Short-term plan (within a year)**
- **Long-term plan**



# Long-term plan

- Handling *acronym* and *custom concepts*
- Handling *multi-lingual* models/ontology
- Handling *non-textual (structural)* model elements
- Using model-ontology linkage in *program rediscovery / mining*
- Integrate design patterns and aspects, etc. into software rules
- Integrate the inference in *program/document validation*