



ECIMF Toolkit

Andrzej Bialecki

ECIMF Project Chair

<ab@getopt.org>



Requirements

- Support ECIMF methodology
 - Business Context equivalence
 - Business Process mediation
 - Semantic translation
 - Syntax mapping
- Open Source Software
 - Suitable for extension and inclusion in commercial applications
- Based on standards where possible
- Platform-independent (or highly portable)



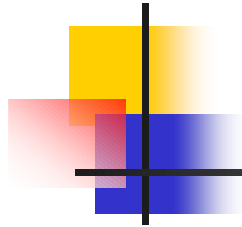
Additional requirements

- Ability to support standard models and definitions used in e-commerce
 - EDIFACT directories
 - DTD or XML Schema-based definitions
 - UML/XMI models
- Extensible
 - Through add-on modules
 - Through scripting languages
- Clear strategy from the mapping model to runtime implementation



Current implementation status

- Just the Semantic Translation module
- BUT highly extensible framework
- Based on a sophisticated knowledge-management framework (Protégé-2000)
 - Implemented in Java (min. JDK 1.2.2)
- Supports semantic translation through labeling (tagging)
- Fully scriptable in many popular languages
- All source code available under free and business-friendly licenses
 - Mozilla, BSD, Apache and Protégé (=Mozilla)
 - No GPL code, at most LGPL will be allowed if indispensable



Information model

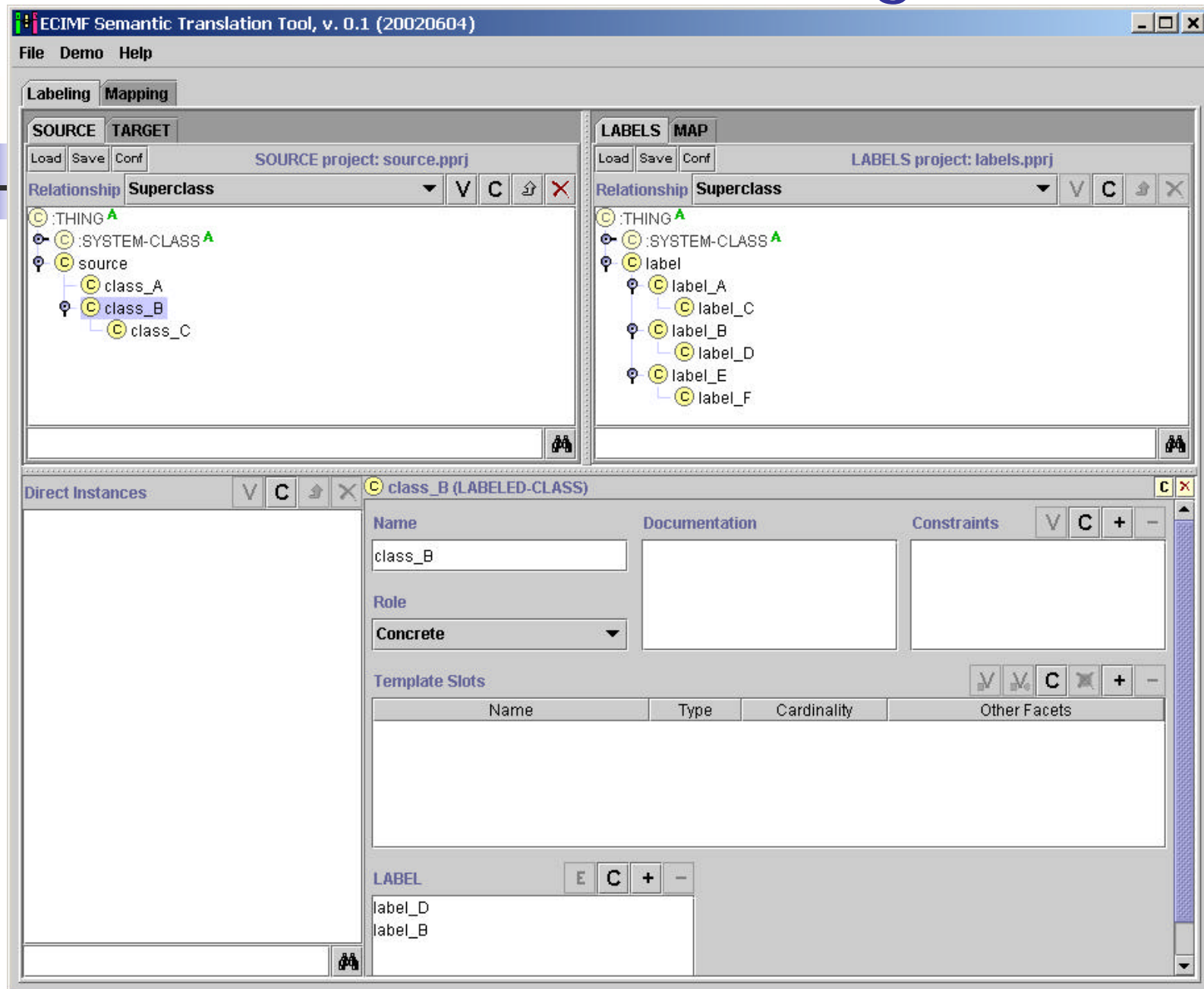
- Domain models are contained in frame-based knowledge bases
 - SOURCE – source model
 - TARGET – target model
 - LABELS – labels model
 - MAP – mapping model
- Concepts from LABELS model are used for SOURCE and TARGET concept tagging
- MAP contains formulas for translating between concepts from SOURCE and TARGET



Mapping formulas

- The tool supplies hints for finding corresponding concepts in SOURCE and TARGET
 - Hints are prepared based on several customizable algorithms
- The tool automatically:
 - Sets the execution context ("stack frame") of directly related concepts in all KBs
 - Generates skeleton of a translating formula
- User fine-tunes the formula using his favorite scripting language, and accessing external resources if needed
- A MAP::Formula instance is created, which defines:
 - The context
 - The translation formula ("scriptlet")
 - The direction of translation

Demonstration – labeling



The screenshot displays the ECIMF Semantic Translation Tool, v. 0.1 (20020604). The interface is divided into several panes:

- Labeling / Mapping:** The top section with tabs for 'Labeling' and 'Mapping'. It contains two project-specific panes: 'SOURCE project: source.pprj' and 'LABELS project: labels.pprj'. Both panes show a 'Relationship Superclass' hierarchy. In the source project, the hierarchy is: :THING A (green) -> :SYSTEM-CLASS A (green) -> source (yellow) -> class_A (yellow) -> class_B (blue) -> class_C (yellow). In the labels project, the hierarchy is: :THING A (green) -> :SYSTEM-CLASS A (green) -> label (yellow) -> label_A (yellow) -> label_C (yellow) -> label_B (yellow) -> label_D (yellow) -> label_E (yellow) -> label_F (yellow).
- Direct Instances:** The bottom section, currently showing 'class_B (Labeled-Class)'. It includes fields for 'Name' (class_B), 'Role' (Concrete), 'Documentation', and 'Constraints'. Below these is a 'Template Slots' table with columns for Name, Type, Cardinality, and Other Facets. At the bottom, there is a 'LABEL' section with a list of labels: label_D and label_B.

Demo – mapping

The screenshot displays the ECIMF Semantic Translation Tool, v. 0.1 (20020604). The interface is divided into several panes for managing semantic mappings.

Top Panel: Includes a menu bar (File, Demo, Help) and tabs for Labeling and Mapping. The Mapping tab is active, showing two project panes: SOURCE project: source.pprj and TARGET project: target.pprj.

SOURCE project: source.pprj: A tree view under the Relationship Superclass shows a hierarchy: :THING A, :SYSTEM-CLASS A, source, class_A, class_B, and class_C. The class_C node is highlighted with a red box.

TARGET project: target.pprj: A similar tree view shows: :THING A, :SYSTEM-CLASS A, target, class_A, class_B, class_C, and class_E.

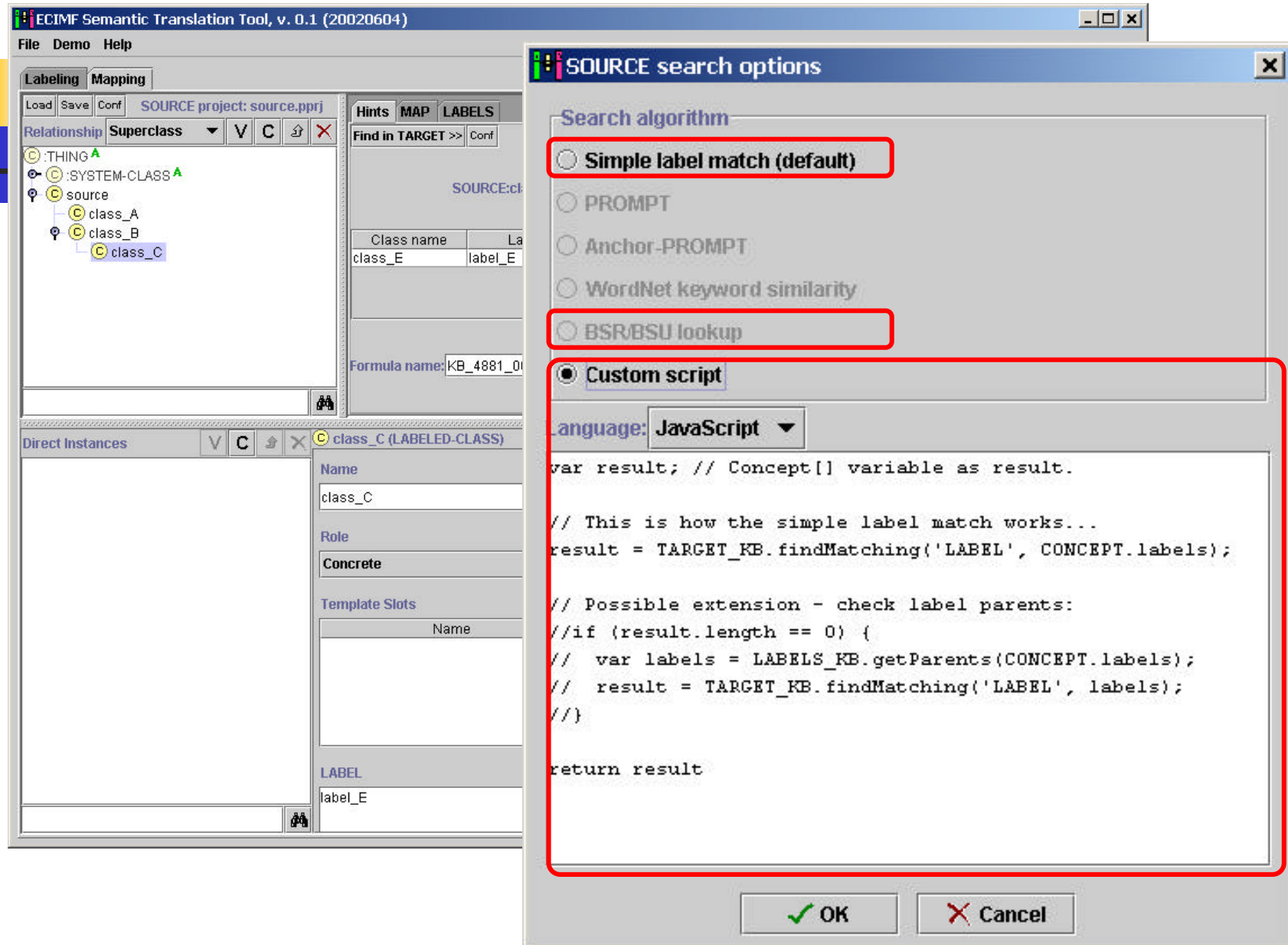
Central Mapping Panel: Features tabs for Hints, MAP, and LABELS. The MAP tab is selected, showing a "Find in TARGET >> Conf" button (highlighted with a red box and a red arrow with a question mark) and a "<< Find in SOURCE" button. Below this, a red-bordered box contains the text "SOURCE:class_C - 1 matches in TARGET" and a table:

Class name	Label	Relation	Use?
class_E	label_E	EQ	<input type="checkbox"/>

Below the table is a "Formula name:" field containing "KB_4881_00000" and a "Create" button.

Bottom Panel: Titled "Direct Instances", it shows details for "class_C (Labeled-CLASS)". It includes fields for Name (class_C), Role (Concrete), and Documentation. Below these are Template Slots and a LABEL field containing "label_E".

Demo – search script



The screenshot displays the ECIMF Semantic Translation Tool, v. 0.1 (20020604). The main window is divided into several panes. The top-left pane shows a project tree for 'SOURCE project: source.pprj' with a hierarchy: :THING A -> :SYSTEM-CLASS A -> source -> class_A -> class_B -> class_C. The top-right pane has tabs for Hints, MAP, and LABELS, with a 'Find in TARGET >>' button. The bottom-left pane shows 'Direct Instances' for 'class_C (Labeled-CLASS)' with fields for Name (class_C), Role (Concrete), and Template Slots. The bottom-right pane shows a 'LABEL' field with the value 'label_E'. A 'SOURCE search options' dialog box is open in the foreground, featuring a 'Search algorithm' section with radio buttons for 'Simple label match (default)', 'PROMPT', 'Anchor-PROMPT', 'WordNet keyword similarity', 'BSR/BSU lookup', and 'Custom script'. The 'Custom script' option is selected. Below this, a 'Language' dropdown is set to 'JavaScript'. A text area contains a JavaScript script for searching by label. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

ECIMF Semantic Translation Tool, v. 0.1 (20020604)

File Demo Help

Labeling Mapping

Load Save Conf SOURCE project: source.pprj

Relationship Superclass V C

Find in TARGET >> Conf

Class name La

class_E label_E

Formula name: KB_4881_0

Direct Instances V C X class_C (Labeled-CLASS)

Name

class_C

Role

Concrete

Template Slots

Name

LABEL

label_E

SOURCE search options

Search algorithm

☐ Simple label match (default)

☐ PROMPT

☐ Anchor-PROMPT

☐ WordNet keyword similarity

☐ BSR/BSU lookup

☒ Custom script

Language: JavaScript

```
var result; // Concept[] variable as result.

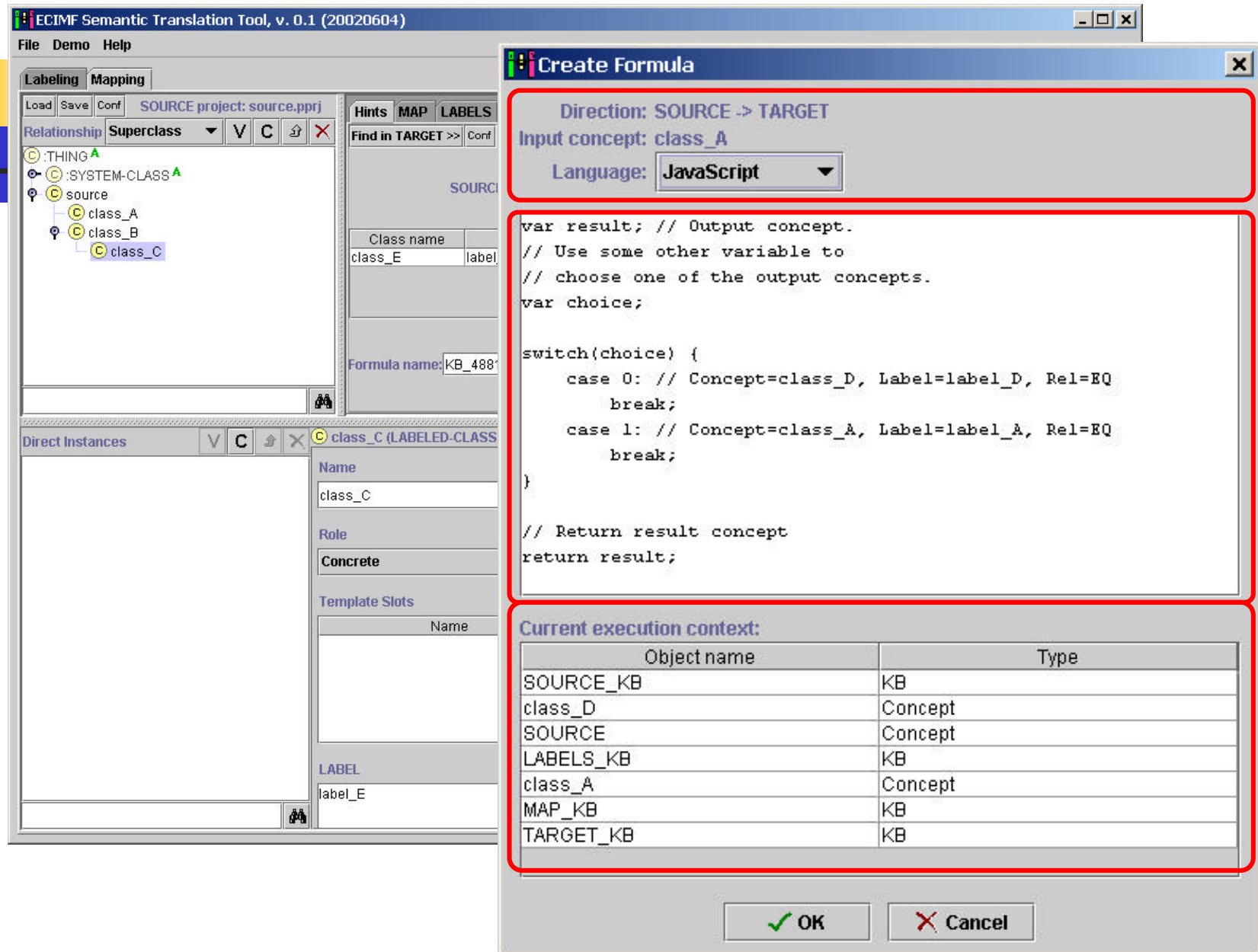
// This is how the simple label match works...
result = TARGET_KB.findMatching('LABEL', CONCEPT.labels);

// Possible extension - check label parents:
//if (result.length == 0) {
//  var labels = LABELS_KB.getParents(CONCEPT.labels);
//  result = TARGET_KB.findMatching('LABEL', labels);
//}

return result
```

OK Cancel

Demo – formula editor



The image displays the ECIMF Semantic Translation Tool, v. 0.1 (20020604). The main window shows a project tree on the left with a hierarchy: :THING A, :SYSTEM-CLASS A, source, class_A, class_B, and class_C. The right pane shows the 'LABELS' tab with a table for 'class_E' and a 'Formula name' field containing 'KB_488'. A 'Create Formula' dialog is open in the foreground, featuring a red border. The dialog has a title bar 'Create Formula' and a close button. It contains the following fields and content:

- Direction:** SOURCE -> TARGET
- Input concept:** class_A
- Language:** JavaScript (dropdown menu)
- Code editor:** Contains JavaScript code for a switch statement that returns a result concept based on a choice.
- Current execution context:** A table listing objects and their types.
- Buttons:** OK and Cancel at the bottom.

The JavaScript code in the editor is as follows:

```
var result; // Output concept.  
// Use some other variable to  
// choose one of the output concepts.  
var choice;  
  
switch(choice) {  
    case 0: // Concept=class_D, Label=label_D, Rel=EQ  
        break;  
    case 1: // Concept=class_A, Label=label_A, Rel=EQ  
        break;  
}  
  
// Return result concept  
return result;
```

The 'Current execution context' table is as follows:

Object name	Type
SOURCE_KB	KB
class_D	Concept
SOURCE	Concept
LABELS_KB	KB
class_A	Concept
MAP_KB	KB
TARGET_KB	KB

Demo – “Walk the KB”

The screenshot shows the ECIMF Semantic Translation Tool, v. 0.1 (20020604). The main window is titled "Walk the knowledge base - demo". It features a "Labeling" panel on the left with "Load" and "Save" buttons. The "Relationship" panel shows a hierarchical tree of concepts, including "THING", "SYSTEM-CLASS", "CLASS", "STANDARD-CLASS", "LABELED-CLASS", "SLOT", "STANDARD-SLOT", "FACET", "STANDARD-FACET", "CONSTRAINT", "PAL-CONSTRAINT", "ANNOTATION", and "INSTANCE-ANNOTATION". The "Run" button is highlighted. The "Code" panel displays the following Java code:

```
importPackage(java.lang);
importPackage(Packages.javax.swing);

// Create the status dialog
status = new JOptionPane(" ", JOptionPane.INFORMATION_MESSAGE);
dialog = status.createDialog(null, 'Status');
dialog.setModal(false);
dialog.show();

// Walk the tree
var concepts = SOURCE_KB.allConcepts;
var length = concepts.length;
for (i = 0; i < length; i++) {
    cls = concepts[i].cls;
    tree.setSelectedCls(cls);
    status.setMessage(i + ' of ' + length + ' : ' + cls.name);
    // Delay
    Thread.sleep(200);
}
status.setMessage('Finisher');
```

A "Status" dialog box is open, showing "6 of 17: SLOT" and an "OK" button. The "Concrete" panel at the bottom shows a table with columns "Name", "Type", "Cardinality", and "Other Facets". The "LABEL" panel shows a list of labels: "label_A" and "label_D".

Demo – integrated with Protégé

The screenshot displays the Protégé-2000 software interface. The top menu bar includes 'Project', 'Window', and 'Help'. Below it is a toolbar with icons for file operations and a row of tabs: 'Classes', 'Slots', 'Forms', 'Instances', 'Queries', 'SM Tab', 'XML', 'Ontoviz', 'Prompt', and 'WordNet'. The 'Mapping' window is active, showing a 'SOURCE project: source.pprj' on the left and a 'TARGET project: targ...' on the right. The 'Relationship' is set to 'Superclass'. The 'Find in TARGET >>' button is visible. The 'SOURCE: class_A - 2 matches in TARGET' section contains a table:

Class name	Label	Relation	Use?
class_D	label_D	EQ	<input type="checkbox"/>
class_A	label_A	EQ	<input type="checkbox"/>

Below the table is a 'Formula name:' field with the text 'KB_5221_00000' and a 'Create' button. The 'Direct Instances' window is also open, showing 'class_A (Labeled-Class)'. It has tabs for 'Name', 'Documentation', and 'Constraints'. The 'Name' tab is active, showing 'class_A' and 'Concrete' as the role. The 'Template Slots' section has a table with columns 'Name', 'Type', 'Cardinality', and 'Other Facets'. At the bottom, there is a 'LABEL' section with a table containing 'label_A' and 'label_D'.



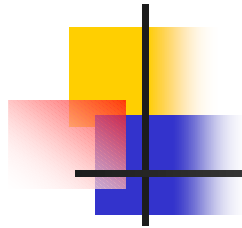
Summary – current status

- Implemented an extensible and scriptable framework
 - Supports 6 most popular scripting languages (JavaScript, VBScript, JScript, Python, Perl, Tcl)
 - Any model and data manipulations are possible, but most of them are now rather inconvenient
- Imported parts of EDIFACT D.01c into Protégé format
- Created parts of ebXML ontology as candidate labels (needs updating with ebTWG!)
- DTD and XSD import module in design phase



Next steps

- More import/export and resource modules
 - **BSR** – as an online service? Not yet available...
 - Excel ☹ - alas, quite popular
 - ebXML-RR, UBL, others ...
- Support for more mapping strategies (e.g. Anchor-PROMPT, WordNet...), and relationship qualifiers (from MULECO draft?)
- Generation of runtime translators
- Support for other parts of ECIMF framework



Tool availability

- The tool is available now for testing
 - Self-installing platform-independent package, containing full sources
 - At <http://www.ecimf.org>
- The distribution package contains example projects
- Unfortunately, no further documentation exists now ...
 - But a short README can be prepared soon... ☺



Questions?

- ECIMF project
 - <http://www.ecimf.org>
- CEN/ISSS WS-EC
 - <http://www.cenorm.be/iss>
- Author:
 - Andrzej Bialecki <ab@getopt.org>