Semi-Automated Instance Migration between Evolving Ontologies

Vladimir VLADIMIROV
Richard SOHNIUS
Vadim ERMOLAYEV
Wolf-Ekkehard MATZKE

GlobalLogic Ukraine
Cadence Design Systems GmbH, Germany
Zaporozye National Uni, Ukraine
Cadence Design Systems GmbH, Germany

May 23, 2007, Kharkiv, Ukraine
Outline

• Why do we need instance migration between ontologies?
• Performance Simulation Initiative (PSI)
  – Why do we need instance migration in PSI?
• How did we do it before: PSI KB Instance Population Script
  – Weak and strong points …
• How do we do it now:
  – PSI Instance Migration – Proof-of-Concept Prototype
• Conclusions and Outlook

No deep technical details in 15 min. Focus on motivation and high-level presentation.
Why Instances migrate?

**Reason 1:** I found a “better” conceptualization than my own ...

**Mine:**

- Student
  - So-So: do not care
  - Smart: are rare
    - Lazybone
      - Vladimirov
    - Dummy
      - ... have to control ...

**College’s:**

- Student
  - Full-Time
  - Part-Time
    - CSStudent
      - StrongInTheory
      - StrongInCoding
      - WeakInPresenting

1) **Where** an Instance goes?
2) What is the change in its content?
Why Instances migrate?

**Reason 2**: I refined my conceptualization myself …

- Same problems between different versions of an evolving ontology
- Constraints are more stiff:
  - Loss of content in instances is not acceptable if not approved by evolution
- Mind distributed scenarios …
  - We collaboratively refined …
Performance Simulation Initiative (PSI)

- Internal Initiative of Cadence Design Systems, GmbH
- Research and Development in Engineering Design Performance Assessment and Management
- A horizontal framework for R&D cooperation
  - E.g., PRODUKTIV+ project (German Federal Ministry of Education and Research)
- Current PSI partners:
  - VCAD, Cadence Design Systems, GmbH
  - Dept of Cybernetics and Gerstner Lab, Czech Technical Uni
  - CERTICON Corp.
  - Intelligent Systems Research Group, Zaporozhye National Uni

- ZNU does knowledge modeling and management
Why do we need instance migration in PSI?

- We have an *evolving* Suite of Ontologies
- PSI Suite is developed *collaboratively*
  - PSI and PRODUKTIV+ projects
  - Several European partners: AMD, Bosch, Cadence, Infineon, IMS, FSU, OFFIS, ZNU
- PSI knowledge base (instances) is expanded and refined incrementally
Semi-Automated Instance Migration between Evolving Ontologies

TBox Transformation ...

High-level overview pictures are taken from Reference Specifications of v.1.4 and v.2.0
PSI KB Instance Population Program

- Does not migrate instances – populates ABox from CSV source
  - Bad if OWL KB is further updated
- Semantics is hard-wired in Perl script
  - Requires manual re-programming if TBox is changed
- Only changed parts need re-programming
- Later versions can load OWL TBox
- Contains KB validation code
Semi-Automated Instance Migration between Evolving Ontologies

Instance Migration Procedure at a Glance

1. Target TBox
2. Rough Instance Creator
3. Failure list
4. ABox Refinery
5. IM Scenario
6. Final KB
7. Add missing instances (tool support)
8. Generate relationships (automatic)
9. Validate target ABox (automatic)

Source ABox

Source TBox

Target ABox

OWL

Create mappings (tool support)
Create draft target ABox (automatic)
Refine mappings (tool support)
Generate relationships (automatic)
Add missing instances (tool support)
Validate target ABox (automatic)
 Instance Migration Execution Environment

• Functions
  – Read and save ontology files in OWL format
  – Manage ontology model
    • Concepts, instances, set data and object properties
  – Load mapping rules
  – Execute instance mapping algorithms coded in Python

• Components
  – Jena API – ontology management library
  – Jython – to bridge Jena API with Python scripts
  – Mapping patterns library – set (incl. custom ones) of instance to instance mapping algorithms for different ontology mapping cases
Instance Migration Scenario Script

- Very similar to IPP
- Differences:
  - Uses Jython and Jena API
  - Reads OWL, but not CSV
  - Scope is MUCH narrower – maps only a small part of ABox
- Complex Mapping Pattern
  - Used to code (and perform) mappings which can’t be coded in OntoMap language
- Runs in Instance Migration Execution Environment
- A programmatic interface for an “advanced” user
- May be re-used
- Collected in the Library
3-d Party Technologies: Mappings

Using Ontoprise OntoStudio with OntoMap plugin
(SEKT IP: http://sekt.semanticweb.org/)

- **OntoMap** – ontology mapping language
- 3 types of mapping rules:
  - CCMapping
  - AAMapping
  - RRMapping
- Only simple mappings
- Complex – to be coded manually
  - IM Scenario
Concluding Remarks

• Automated ontology instance migration
  – Is hard
  – Is must-have for many real-world intelligent applications
• So far we do it only semi-automatically
• We also provide tool support for manual steps
• Proof-of-Concept implementation is done
  – Tested on the pair of versions of PSI Ontology Suite (v.1.4 to v.1.5)
  – Result encourages future work
Future Work

• Further development of the tools for manual steps:
  – Refine mappings
  – Add missing instances
• Refinement of mapping language to cope with:
  – Property type, data pattern, length changes
  – Property value aggregations
• Incremental development of the Library of re-usable
  Instance Migration Scenario Scripts
• Performance issues
• Scalability issues
• Evaluation on “heterogeneous” TBoxes
Resources:
PSI: http://ermolayev.com/ISRG/ISRG-projects-PSI.htm

Questions please