Transitioning Applications to Ontologies

Kalina Bontcheva
on behalf of
the TAO consortium

http://www.tao-project.eu

Funded by: European Commission – 6th Framework
Project Reference: IST-2004-026460
Addressing the problem of transitioning legacy applications to ontologies

What is a legacy software system:
- “A large software system that is vital to [an] organisation, but resists modification and evolution to meet new and constantly changing business requirements”

Towards semantic-assisted software engineering
Legacy systems: Main Problems

- Built with languages and data models that are now **out-dated**
- Badly structured and **hard to maintain**
- **Badly** documented and **understood**
- **Difficult to integrate** with each other and with new systems
- Need for migration towards Web 2.0 applications & **services**
TAO: Towards a Low-Cost Migration Path

- Make transitioning to ontologies fast and effective
- Build a reusable transitioning process
- Minimize consulting time during migration and integration
- Minimize costs
- Reduce integration overheads and limit risk
Transitioning Process

- Semi-automatic learning of domain ontologies from software artefacts and legacy content
- Semantic augmentation of legacy content and web service definitions
- Heterogeneous & distributed semantic repositories
- Transitioning Infrastructure supporting IT providers
Partner Contributions

- Semantic annotation of text and multimedia content; **GATE** – widely used open-source infrastructure for text mining
- **Transitioning methodology**
- **Machine learning** and data mining
- **Transitioning infrastructure**; exploitation
- Most scalable & efficient semantic repository
- **Content and knowledge management**
- Aviation case study; aircraft manufacturer
Transitioning Problems

- From legacy databases to ontologies
- Towards semantic-based software engineering
- Transitioning web applications towards ontologies and services
Common Logistic Ontologies in OWL

Ontology Learning + Standards reuse

Distributed Knowledge Repositories

- A/C Parts
- M & O Tasks
- Consumables
- Tools
- Workloads
- Maintenance Plans
- Due lists
- ...

Semantic Content Augmentation

Aircraft Maintenance Documentation

Process Modelling

Service Registry

Receive Mo
Invoke S1
Invoke S2
Receive S3
Invoke S4
Reply So

Improving semantic interoperability in business processes through Semantic technologies
Example Learnt Ontology – 400k triples
Example Learnt Ontology (2)
A Semantic Tagging Example

Content Augmentation of Maintenance Manuals

Maintenance Data Repository

- Valve
  - Non-Return-Valve
    - Denominations
      - Full: Non Return Valve
      - Abbrev: NRV
      - valve-1888
        - Denominations
          - Full: PUMP1 Non Return Valve
      - valve-1838
        - Denominations
          - Full: HYDR Non Return Valve

Heterogeneous Knowledge Store

Semantic Content Augmentation

Annotation Interface

- Remove PUMP1 NR valve
- Check PUMP1 valve and clean it

Ontologies

- Non-Return-Valve
  - PUMP1 NRV
    - threshold: 3422 psi
Transitioning Problems

- From legacy databases to ontologies
- Towards semantic-based software engineering
- Transitioning web applications towards ontologies and services
Learning Ontologies from Software Artefacts

Software data sources

Structured data = networks

Unstructured data = textual documents

Document network

= a set of interlinked documents; each link has a type and a weight
Creating a Document Network

DocumentFormat.class

/** The format of Documents. Subclasses of DocumentFormat know about
 * particular MIME types and how to unpack the information in any
 * markup or formatting they contain into GATE annotations. Each MIME
 * type has its own subclass of DocumentFormat, e.g., MpegDocumentFormat.
 * AbstractLanguageResource. These classes register themselves
 * with a static index containing here when they are constructed. Static
 * getDocumentFormat methods can then be used to get the appropriate
 * format class for a particular document.
 * */
public abstract class DocumentFormat
    extends AbstractLanguageResource implements LanguageResource;

/** The MIME type of this format. */
private Mime_Type mime_Type = null;

/** Find a DocumentFormat implementation that deals with a particular
 * MIME type, given that type.
 * Splits an stateDocument this document will receive as a feature
 * the associated Mime_Type. The name of the feature is
 * Mime_Type and its value is in the format type/subtype
 * Splits mime_Type the name type that is given as input.
 * */
static public DocumentFormat getDocumentFormat(String stateDocument, Mime_Type mime_Type)

} // class DocumentFormat
Ontology Learnt from Software Code
Google-like Conceptual Search (1)
Question-Based Conceptual Search (2)

Answer is showing relations between identified concepts given in the query.

Search knowledge with CLOnE QL

Ontology: w/gate-ontology-instances.owl
Query: what are the parameters of annie pos tagger?

Result:

ANNIEANNIETaggerRulesURL --> [inverseProperty] hasInitTimeParameter --> ANNIE POS Tagger
ANNIEANNIETaggerEncoding --> [inverseProperty] hasInitTimeParameter --> ANNIE POS Tagger
lexiconURL --> [inverseProperty] hasInitTimeParameter --> ANNIE POS Tagger

baseSentenceAnnotationType --> [inverseProperty] hasRunTimeParameter --> ANNIE POS Tagger
outputAnnotationType --> [inverseProperty] hasRunTimeParameter --> ANNIE POS Tagger
ANNIEANNIETaggerInputASName --> [inverseProperty] hasRunTimeParameter --> ANNIE POS Tagger
ANNIEANNIETaggerOutputASName --> [inverseProperty] hasRunTimeParameter --> ANNIE POS Tagger
ANNIEANNIETaggerBaseTokenAnnotationType --> [inverseProperty] hasRunTimeParameter --> ANNIE POS Tagger
ANNIEANNIETaggerDocument --> [inverseProperty] hasRunTimeParameter --> ANNIE POS Tagger
Transitioning Problems

- From legacy databases to ontologies
- Towards semantic-based software engineering
- Transitioning web applications towards ontologies and semantic-web services
Transitioning Web Applications

- Legacy application:
  - database driven
  - no interoperability

- Ontologies + SOA:
  - Learn ontologies
  - Manage complex resources and knowledge links
  - Use Service Oriented Application to integrate added value services from other suppliers: cartography, translation, booking services...
Software Demonstrations

- Learning ontologies from software code:

- Supporting software developers with conceptual search:
Thank you! Questions?

This presentation + demos:
http://www.tao-project.eu/demos-dec07/

Kalina Bontcheva:
kalina@dcs.shef.ac.uk
Contact Information

For queries / further information, please contact the project co-ordinator:

Kalina Bontcheva
Department of Computer Science
University of Sheffield
Regent Court
211 Portobello Street
Sheffield S1 4DP

phone: (+44 - 114) 222 1930
fax: (+44 - 114) 222 1810
email: K.Bontcheva@dcs.shef.ac.uk