A-BOA: Basics, Applications, Theoretical Foundations, and Demonstration

**Tutors:** Vadim Ermolayev, Maxim Davidovsky
Dept of IT, Zaporozhye National Univ., Ukraine
Components, IPR, Sponsors

- **A-BOA Tutorial** has been developed at the Intelligent Systems Research Group affiliated at the Department of IT of Zaporozhye National University

- A Theoretical Framework for Agent Negotiations on Semantic Contexts and Propositional Substitutions has been developed in RACING project

- **Structural Difference Discovery Engine (SDDE)** agent-based software tool has been developed by Maxim Davidovsky as a part of his PhD Project

- **Instance Migration Engine (IME)** software tool have been developed in Performance Simulation Initiative (PSI) project funded by Cadence Design Systems GmbH
  - All rights with respect to IME are retained by Cadence Design Systems GmbH

- **A-BOA Wiki** containing support materials for A-BOA Tutorial – a Semantic MediaWiki based resource

- Questions and answers are supported using live contextual collaboration in LiveNetLife

- **A-BOA tutorial** at WIMS 2012 is sponsored in part by DataArt
Plan

- **Walkthrough Problem and Example**
  - Ontology Instance Migration Problem
  - Simple Biblio ontologies

- **Part 1: Motivation, Basics, and Applications**
  - What is ontology alignment? and
  - Why is the technology needed?

- **Part 2: Theoretical Foundations and Demonstration**
  - Use of agent-based approaches for building ontology alignments - answering "how" questions
  - Demo of Agent-Based solution for Ontology Instance Migration Problem

- Round the World in 80 ... min
  - Some important things will be just mentioned
  - Tasties are left for individual exploration
Workflow

**What:**
- Problem Statement and Classification

**Why:**
- Applications and Requirements

**How:**
- A-BOA Tutorial
  - June 13, 2012
  - A-BOA Tutorial
  - June 13, 2012

**Demo**
- Requirements Solved by Agent Orientation
- Walkthrough Example and Problem
- A-BOA

1. Problem Statement and Classification
2. Applications and Requirements
3. A-BOA Tutorial
Support and Questions

• A-BOA Wiki
  – http://isrg.kit.znu.edu.ua/a-boa/
  – Wiki articles to follow the Tutorial
  – Sections in printer friendly form (PDF) – not yet there
  – Tutorial slides corresponding to Wiki articles – not yet there

• Questions and answers anytime
  – Focused: LiveNetLife chat
    – No connection 😞
  – Broader: Oral or Wiki discussion pages
  – After the Tutorial: @Wiki discussion pages answered by e-mail
A Walkthrough
Problem and Example

A-BOA Wiki: Walkthrough Problem and Example
Ontology Instance Migration

- When is IM needed:
  - OE – new version developed
  - Schema Changed
  - Instances to be transformed ...
  - System
    Interoperability/Integration

- Simplification:
  - We have the result – for teaching purposes

- How?
  - Different techniques
  - We will show one in the Demo
Example: Biblio Schemas

- **Biblio ontologies** – a VERY simple example
  - Different knowledge representations for the same body of knowledge about conference papers
- Real ontologies are:
  - MORE complex (schema)
  - MUCH MORE bulky (instances)
  - E.g. [10]
- Imagine:
  - Biblio-2 is for a conference management system
  - Biblio-1 is the model for a paper repository at a publisher
  - Papers accepted for a conference have to appear in the publisher’s paper repository
  - Publisher’s information about the page limits has to be communicated to the conference management system
- **Biblio-1 and Biblio-2 have to be aligned**

Example: Biblio Instances

a) The instances of Biblio-1

b) The instances of Biblio-2

The result we have to achieve
Part 1: Motivation, Basics, and Applications

A-BOA Wiki: Motivation, Basics, and Applications
Part 1: Structure

- **Ontology Alignment** in general and at a relatively basic level:
  - Outlines the **motivation** to study OA
  - **WHAT:** Denotes OA and puts the problem into the context of the other knowledge harmonization and integration problems
  - **WHY:** Analyses the **applications** that require aligning knowledge representations, summarizes requirements
Section 1.1: Motivation to Study Ontology Alignment

“I find it critical to remember that every ontology is a treaty – a social agreement – among people with some common motive in sharing.”

Are Interpretations the Same?

• In row 1?
• In row 20?
Motivation - Abstract

• The World is multi-faceted and **polysemic**
  => Many different views or interpretations by different individuals or groups

• Reflected in different **knowledge representations of the same** reality

• We do many things across several facets or even across **subject domains**
  => Several knowledge representations (ontologies) have to be harmonized or aligned
  – To enable proper **communication, coordination** or information processing
Motivation - Utility

- An **alignment** is essentially:
  - A **result** of applying a set of formal transformations to a knowledge representation – to its schema and individuals

- An alignment allows:
  - Interpreting knowledge that is external to the interpreter
  - In the same way the interpreter views his own knowledge schema and assertions

- E.g., given that a bi-directional alignment of Biblio-2 to Biblio-1 exists:
  - A publisher (Biblio-1) – seamlessly imports the assertions about the accepted papers to its production repository
  - A conference organizer (Biblio-2) – gets publisher’s information about publication constraints, like page limits
  - Common motive in sharing is satisfied

- Many kinds of important applications require OA
Section 1.2: Basics of Ontology Alignment

A-BOA Wiki: Basics of Ontology Alignment
Section 1.2: Structure

- **Basic Definitions and Generic Problem Statement**
  - Denotes an *ontology*, *ontology schema*, *assertional part*, *mapping*, and *ontology matching process*
  - Based on these a definition of *ontology alignment* is given

- **Classification of Ontology Alignment Problems**
  - Several features of participating ontologies
  - The span of the aligned ontology elements across ontologies

- **Ontology Instance Migration Problem**
  - A *walkthrough problem* with a little bit of more formal detail

- **Ontology Alignment Metrics**
  - Not all of them, but those important for solving Ontology Instance Migration Problem – structural and assertional
Section 1.2.1: Basic Definitions and Generic Problem Statement
Basic Definitions: Ontology

- **Ontology** (c.f. [22]) – a tuple:
  \[ O = (C, P, I, T, V, \leq, \perp, \in, =) \]

  where the sets \( C, P, I, T, V \) are pair-wise disjoint and:
  - \( C \) – set of **concepts** (or classes)
  - \( P \) – set of **properties** (object and datatype properties)
  - \( I \) – set of **individuals** (or instances)
  - \( T \) – set of **datatypes**
  - \( V \) – set of **values**
  - \( \leq \) – reflexive, anti-symmetric, and transitive relation on \((C \times C) \cup (P \times P) \cup (T \times T)\) called **specialization** (subsumption) that form partial orders on:
    - \( C \) – concept hierarchy; and
    - \( P \) – property hierarchy
  - \( \perp \) – irreflexive and symmetric relation on \((C \times C) \cup (P \times P) \cup (T \times T)\) called **exclusion**
  - \( \in \) – relation over \((I \times C) \cup (I \times V)\) called **instantiation**
  - \( = \) – relation over \( I \cup P \cup (I \times V)\) called **assignment**

Basic Definitions: TBox and ABox

- An **ontology** (c.f. [36]) \( O \) comprises:
  - Schema \( S \) and its assertional part \( A \)
  - \( O = (S, A) \); \( S = (C, P, T, ...) \); \( A = (I, V, ...) \)

- **Ontology schema** \( S \) (or a terminological component, TBox) contains statements describing:
  - The concepts from \( C \) of \( O \)
  - The properties from \( P \) of those \( C \)
  - The datatypes \( T \) for the elements of \( P \)
  - The axioms over the elements of \( C, P, T \)

- The **set of individuals** \( A \) (or assertional component, ABox) contains:
  - Ground statements about the instances of \( O \)
  - Attribution of the instances of \( O \) to the schema

New York, NY, USA
A **Mapping** (or a **Mapping Rule**, c.f. [22]) is a tuple

\[ m = (e, e', R, n), \]

where:
- \( e, e' \) are the elements of \( C, P, I, T, V \) of the respective ontologies \( O \) and \( O' \)
- \( R \) is a set of relations
- \( n \) is a confidence value (typically in the range of \([0, 1]\))

---


---

**TBox: Biblio v.1**

- **Person**
  - **name**: String
- **Paper**
  - **title**: String
- **RegularPaper**
  - **volumeTitle**: String
  - **maxNoPages**: Integer = 15
- **Proceeding**
  - **volumeTitle**: String

\[ m = (\text{Paper} \in C, \text{ProceedingsPaper} \in C', \leftrightarrow, 1) \]

---

**TBox: Biblio v.2**

- **Author**
  - **name**: String
  - **affiliation**: String
- **ProceedingsPaper**
  - **title**: String
  - **startPage**: Integer
  - **endPage**: Integer
- **FullPaper**
  - **volumeTitle**: String
  - **maxPageNo**: Integer = 15

---

**A-BOA Wiki: Basic Definitions ...**
Basic Definitions: Mapping

- A more complex Mapping: 
  \[ m = (\langle \text{PaN1.strtPage}=179\rangle \in V, \langle \text{PrP1.strtPage}\rangle \in V', \text{migrate}, 1) \]

\[ \text{...} \]

\[ \text{...} \]

\[ \text{...} \]

\[ \text{...} \]

\[ \text{...} \]

\[ \text{...} \]
### Basic Defs: Ontology Matching

- **Ontology matching** (c.f. [22])
  
  - a *process* of *discovering* the *mappings* between the elements $e$ and $e'$ of different ontologies $O$ and $O'$

- A generic ontology matching process
  
  - **Discover Mappings**

---

BasicDefs: Ontology Alignment

- **Ontology Alignment**
  - the **result** of applying the discovered set of **mappings** to the respective **ontologies**

- **A Generic Ontology Alignment Problem**
  - Build alignments following a **Generic Ontology Alignment process**
    - Discover Mappings
    - Apply Mappings
    - Could be interweaved
    - Result: **Alignment** – shaded gray

- Several kinds of OA problems ...
Section 1.2.2: Classification of Ontology Alignment Problems

A-BOA Wiki: Classification of Ontology Alignment Problems
Classification: Dimensions

- Let:
  - \( O = (C, P, I, T, V, ...), \ e \ \text{belongs to} \ O 
  - \( O' = (C', P', I', T', V', ...), \ e' \ \text{belongs to} \ O' 

- **Ontology Alignment Problems** are classified based on:
  - The features of participating ontologies \( O, O' \); and
  - The span of \( e, e' \) across \( C, P, I, T, V - s \) of \( O, O' \)

- **Classification dimensions:**
  - **Span** – Complete, Structural, or Assertional alignment
  - **Dynamicty** – Static versus Dynamic aligned ontologies
  - **Direction** – Bi-directional versus Uni-directional alignment
  - **Distribution** – Fully Distributed settings versus the use of a Central referee ontology

- Additionally we differentiate:
  - **One-Shot** versus **Iterative** Alignment approaches
Classification: Span

- By the **span** of aligned elements Ontology Alignment Problems are classified as:
  - **Complete** - if alignments span across TBox-es and ABox-es of $O, O'$
  - **Structural** - if alignments cover only the TBox-es of $O, O'$
  - **Assertional** - if alignments cover only the ABox-es of $O, O'$
Classification: Dynamicity

- **Wrt dynamicity** of aligned elements Ontology Alignment Problems are classified as:

  - **Static** – $e, e'$ of $O, O'$ are considered unchanged
    - At least for the time of alignment

  - **Dynamic** – $e$ and $e'$ may be changed while DM or AM phase is executed
    - Potential invalidity of mappings and alignments
    - Additional revision may be required
Classification: Direction

- By **direction** of alignments, Ontology Alignment Problems are classified as:
  - **Bi-directional** – $e$ and $e'$ of both ontologies ($O$ and $O'$) are aligned
  - **Uni-directional** – alignments are applied to only one ontology – either $O$ or $O'$
Classification: Distribution

- By the degree of **distribution** in their settings Ontology Alignment Problems are classified as:
  - **Centralized** – rely on a central **Referee Ontology** $O^r$ as a bridge for constructing correct mappings
    - Not always possible
      - E.g. competitors
      - E.g. appropriate referee ontology is not available
  - **Distributed** – without a central referee
One-Shot vs Iterative

- **One-Shot** techniques – align $e$, $e'$ of $O$, $O'$ in one iteration
  - Shortcomings:
    - **Dynamicty**: $e$, $e'$ may change – invalid alignment
    - **Bad quality** revealed in subsequent evaluation

- **Iterative** approaches
  - Add **evaluation** step in the loop
  - Iterate in the **refinement loop** until the quality of alignment is not sufficient
Ontology Instance Migration

• Let:
  - \( O_s = (S^s, A^s) \) - a source ontology
  - \( O_t = (S^t, A^t) \) - a target ontology
  - \( O_s, O_t \) conceptualize the semantics of the same Universe of Discourse \( U \)
    - E.g. the same Bibliography domain
  - \( U \) regarded as a collection of ground facts: \( U = \{ f \} \)
  - Essentially, \( O_s \) and \( O_t \) are the interpretations of \( U \)
    - E.g. Marylin vs Albert

• \( O_s \) and \( O_t \) would be considered identical iff:
  - \( \forall f \in U \) \( \text{int}^{s}_I (f) \equiv \text{int}^{t}_I (f) \)
  - E.g. Either Marylin OR Albert
    - \( \text{int}_I (f) \) is the interpretation of the fact \( f \) by the individuals from \( I \) of ontology \( O \)
Ontology Instance Migration

• Let $idiff(U, O^s, O^t)$:
  
  − An **abstract metric of interpretation difference**
  
  − $idiff = 0$ for identical ontologies
  
  − $idiff$ increases monotonically to $1$ with the increase of the number of $f \in U$ such that
    
    \[ \neg (\text{int}_{I^s}(f) \equiv \text{int}_{I^t}(f)) \]
  
  − $idiff = 1$ iff $\forall f \in U (\neg (\text{int}_{I^s}(f) \equiv \text{int}_{I^t}(f)))$

• $(1 - idiff)$ may further be interpreted as **balanced F-measure** in evaluation of semantic distance
Ontology Instance Migration

- $O^s$ and $O^t$ are **structurally different** if $S^s \neq S^t$
  - Structural difference – a transformation $T: S^s \rightarrow S^t$
  - $T$ may be sought in the form of a set of nested transformation rules

- Let:
  - $\text{ABox}$ of $O^s$ contains individuals ($I^s \neq \emptyset$), while
  - $\text{ABox}$ of $O^t$ is empty ($I^t = \emptyset$)

- The problem of minimizing $\text{idiff}(U, O^s, O^t)$ by:
  - (1) Taking the individuals from $I^s$
  - (2) Transforming them correspondingly to the structural difference between $O^s$ and $O^t$ using $T$; and
  - (3) Adding them to $I^t$

- is denoted as **Ontology Instance Migration** problem
  - Classified as **ASUD** Ontology Alignment Problem
Ontology Instance Migration

- Theoretically can be solved in one shot
- In practice each of the sub-tasks (1-3) may result in the loss of assertions[10]
  - **Iterative refinement** could yield results with a lower resulting \( idiff \) value

- An iterative solution:
  - Develops a sequence of \( O^s \) states \( O^s_{st_i} \) to minimize the \( idiff(U, O^s_{st_i}, O^t) \) in a way that:

\[
idiff(U, O^s_{st_i}, O^t) < idiff(U, O^s_{st_j}, O^t) \Rightarrow i < j
\]

where: \( O^s_{st_i} \) is \( O^s \) in the state after accomplishing iteration \( i \)
Section 1.2.4: Ontology Alignment Metrics

“... I would contend that analysts frequently should not seek a single measure and will never find a perfect measure. ... It is time to stop acting embarrassed about the supposed surplus of measures and instead make fullest possible use of their diversity.”

Ontology Alignment Metrics

- **OA problem** – minimizing **semantic difference** between \( e \) and \( e' \) of \( O \) and \( O' \)
  - **Metrics** – for measuring this semantic difference
  - \( O = (S, A) \)
  - Have to cope with the semantic differences between:
    - \( S \) and \( S' \) - the **metrics for Structural Difference**
    - \( A \) and \( A' \) - the **metrics for Assertional Difference**
Metrics for Structural Difference

- Based on assessing the semantic distance between the structural elements, comparing:
  - **Structural Elements** themselves
  - The semantic context of the Structural Components
  - The **individuals** relevant to the Structural Components

- A good overview - in [20]
  - Not all discussed here

Instance(-Based Structural) Similarity

- **Rationale:** similar structural elements (e.g. concepts) have similar instances

- Let:
  - $D$ a domain
  - $A$ and $B$ – the concepts in $D$

- $A$ is (somewhat) similar to $B$ if $I_A \cap I_B \neq \emptyset$
  - $I_A$ and $I_B$ are the sets of individual assertions about $D$; and
  - $I_A = \{i_k\}: \forall k, \text{instance}_\text{of}(i_k, A)$

- Instance Similarity is often measured \cite{doan2003learning} by a symmetric Jaccard coefficient:

$$\text{Sim}_I(A, B) = \frac{P(I_A \cap I_B)}{P(I_A \cup I_B)}$$

  - $P(I)$ is the probability that a randomly chosen instance of $D$ belongs to $I$

**Instance (Based Structural) Similarity**

\[
\text{Sim}_I(A, B) = \frac{P(I_A \cap I_B)}{P(I_A \cup I_B)}
\]

- **Concepts**
  - \(A=\text{Biblio-1:RegularPaper}\); and
  - \(B=\text{Biblio-2:FullPaper}\)

- **So:**
  - \(I_A \cap I_B = I_A \cup I_B\); and
  - It is very probable that \(P(I_A \cap I_B) = P(I_A \cup I_B)\)

- Hence: \(\text{Sim}_I(A, B)\) is close to 1.0
Contextual or Feature Similarity

- **Rationale**: similar structural elements (e.g. Concepts) have similar structural contexts [17, 18]

- Contexts may be understood as **feature sets**

- Hence, Contextual Similarity may be measured using **Tversky metrics** [48]
  - A feature set has to be defined
  - E.g. the set of similarity measures of the object properties and related concepts $s_j$
  
  $Sim_C = \frac{1}{m} \sum s_j$

  - In that sense contextual similarity may be regarded as an integrative metric for a pair of concepts

---


---

A-BOA Wiki: **Ontology Alignment Metrics**
Datatype and Measurement Similarity

- **Rationale**: similar structural elements (e.g. Concepts) have similar properties
- **Shortcoming**: the problem of determining similarity among properties has the same complexity as measuring the similarity of concepts
- **Hint**: the set of properties of a concept is the part of its feature set
  - Measure property similarity using a contextual similarity metric
- **Complication**: different types of properties, e.g.:
  - **Datatype properties**
    - Reflect that a concept has a characteristic which:
      - Has a particular type (a color, a weight, a string, ...)
      - Is measured using specific units (a year, an integer, ...)
      - May have constraints on its values expressed as logical formulas, e.g.: \((weight \leq 90) \land (age \leq 30)\)
  - **Referential (object) properties**
    - Reflect a relationship to another concept (property)

A-BOA Wiki: **Ontology Alignment Metrics**
Datatype and Measurement Similarity

- **Biblio example:**
  - $A = \text{Biblio-1:RegularPaper}$
    - Has a datatype property $a = \text{maxNoPages}$
    - Measured in integers
  - $B = \text{Biblio-2:FullPaper}$
    - Has a datatype property $b = \text{maxPageNo}$
    - Measured in integers
  - Properties $a$ and $b$ may be considered similar
    - Hypothesis $\text{similar_to} (A, B)$
    - $\text{Sim}_M$ between $A$ and $B$ increased
  - $\text{Sim}_M$ will be even higher if $a$ and $b$ have the same constraints/values:
    - $(a = 16)$ and $(b = 16)$
Lexical Similarity

- **Rationale**: similar structural elements have similar identifiers
  - E.g. lexical roots are the same
  - May of course lead to confusion
  - However, a good hint for supposing similarity

- **Lexical heuristics** work if supported by other evidence:
  - E.g. instance or feature similarity for a pair of concepts is high

- The following lexical metric $Sim_L$ is often used
  - Let $R_A$, $R_B$ be the sets of roots of the words which constitute the names of concepts $A$ and $B$ respectively, then:
    \[
    Sim_L = \frac{|(R_A \cap R_B)|}{|(R_A \cup R_B)|}
    \]

- **Biblio example**:
  - $A = \text{Biblio-1:RegularPaper}$
  - $B = \text{Biblio-2:FullPaper}$
  - $Sim_L (A, B) = 0.33$
Metrics for Assertional Difference

- Have a slightly different nature
- Are often based on measuring the fraction of aligned individuals in terms of:
  - Recall
  - Precision, or
  - A combination of those
    - E.g. balanced F-measure

- For the **ontology instance migration** problem:
  - $\text{Precision} (P)$ is the fraction of migrated individuals that are relevant
  - $\text{Recall} (R)$ is the fraction of relevant individuals that are migrated
For the **ontology instance migration** problem:

- **Precision** \((P)\) is the fraction of migrated individuals that are relevant
  \[P = \frac{tp}{tp + fp};\]
- **Recall** \((R)\) is the fraction of relevant individuals that are migrated
  \[R = \frac{tp}{tp + fn}\]
- Additionally - **Accuracy** \((A)\)
  \[A = \frac{(tp+tn)}{(tp + fp + tn + fn)}\]

An ideal migration outcome corresponds to \(P = R = 1\)

Neither \(P\) nor \(R\) separately fully reflects the correctness of migration results

**F-measure** \((F)\) could be of interest as it brings \(P\) into correlation with \(R\) as a harmonic mean

\[
F = \frac{1}{\alpha / P + (1 - \alpha) / R} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \\
\beta^2 = (1 - \alpha) / \alpha
\]

Balanced F-measure equally weights \(P\) and \(R\):

\[\alpha = 1/2 \Rightarrow \beta = 1\]
Section 1.3: Applications of Ontology Alignment

A-BOA Wiki: Applications of Ontology Alignment
Section 1.3: Structure

- A few (1 😊) selected categories of applications
  - A broader spectrum is surveyed in [11]
- Focus on the requirements to ontology alignment that are posed by the applications in a category
  - A particular **ontology alignment problem**
  - Why is an **agent-based solution** appropriate?
- Categories of applications:
  - **Distributed Information Retrieval**
  - Human-Machine Dialogues
  - Ontology Evolution, Versioning, Refinement
  - Service Composition
- The **requirements** to ontology alignment technology are finally summarized

Distributed Info Retrieval

- **DIR** applications assist retrieving and fusing information from heterogeneous, distributed, and independent IR

- **Ontologies** in DIR are used for:
  - Transforming user queries and system responses
  - Representing Structures and Semantics of IR

- Ontology alignments are required:
  - At Query Transformation step
  - At Result Fusion step

We provide IR-s annotated in terms suitable for us. Normally, we do not care about the others.

I have a query to all of you in terms (and in language) that I understand.
Distributed Info Retrieval

- Ontology alignments are required:
  - At Query Transformation (QT) – for:
    - Correlating query structure and semantics with different information resource schemas
    - Building respective partial queries
  - At Result Fusion (RF) – for:
    - Transforming and putting together the retrieved information instances
- QT Requirement: a solution for a Structural Static Uni-directional Distributed (SSUD) OA problem
- RF Requirements:
  - A solution for an Assertional Static Uni-directional Distributed (ASUD) OA problem
  - High recall – important not to miss any potentially relevant information; irrelevant can be filtered out using other techniques
- General requirement: scalability wrt the complexity and number of aligned ontologies

A-BOA Wiki: Distributed Info Retrieval
Agents in DIR

- E.g. SEWASIE project [13]:
  - A multi-agent system for querying heterogeneous data sources integrated using ontologies
  - [http://www.sewasie.org/](http://www.sewasie.org/)

## Requirements to Onto Alignment

### Application Category

<table>
<thead>
<tr>
<th>Application Category</th>
<th>Requirements</th>
<th>Ontology Alignment Problems</th>
<th>Method</th>
<th>Agent Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Span</td>
<td>Dynamics</td>
<td>Direction</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>Precision</td>
<td>Run-time Solution</td>
<td>Semantic contexts</td>
</tr>
<tr>
<td>Distributed Information Retrieval</td>
<td>+</td>
<td>~</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Human-Machine Dialogues</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Ontology Engineering and Management</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Service Composition</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

**Legend:** + = minimal requirement/basic solution; ~ = desired; + = required.

- **Instance Migration** *(ASUD, Iterative):*
  - **Required:**
    - Distributed Information Retrieval
    - Ontology Engineering and Management *(Evolution, Versioning, Refinement)*
  - **Good to have:**
    - Service Composition *(ASBD)*
Part 2: Theoretical Foundations and Demonstration
Part 2: Structure

- Answers the "how" group of questions
  - A more advanced material
  - More focused on agent-based approaches for building ontology alignments

- Overviews selected **agent-based frameworks** for ontology alignment:
  - Information Flow Theory based approaches
  - Argumentation based frameworks
  - Semantic Contexts and Propositional Substitutions

- Offers, as a **practical reinforcement** for the overview
  - Demonstration of the Agent-Based Software Prototype
    - A very brief one – showing the results, not the process
  - The tool for solving one particular problem of Ontology Alignment
    - Ontology Instance Migration Problem
Theoretical Foundations

- Agent-based approach for solving a **generic ontology alignment problem**
- **Discover Mappings**
  - $W$ and $W'$ are the wrapper agents for ontologies $O$ and $O'$
  - Agent $R$ wraps the central referee ontology $O^r$ and helps $W$ and $W'$ finding the proper mappings $M$ and $M'$ using $O^r$ (a *matchmaker* function)
  - $W$ and $W'$ produce their own sets of mappings $M$ and $M'$:
    - In collaboration with each other (a fully distributed problem setting); or
    - Also in collaboration with $R$ (the problem setting with a central referee ontology)
- **Apply Mappings**
  - $M$ and $M'$ are autonomously applied by $W$ and $W'$ to $O$ and $O'$
- Problem: **How** do the agents collaborate and develop these mappings?
Theoretical Foundations

- Substantial attention in the literature

- **Mainstream**: use of (different flavors of) negotiation techniques as the most natural and well-proven mechanism for reaching agreements

- Several fundamental theoretical approaches with different expressive power
  - Most widely used formalism is the Dung’s *Argumentation Framework or its derivatives*
  - The formalism used in our software (demo):
    - Negotiations on propositional substitutions in semantic contexts
    - Based on the Type Theory
Section 2.1: Information Flow Theory Based Approaches
Information Flow Theory

- A formal foundation by Schorlemmer et al. [42]
  - **Ontology Alignment** – a product of *meaning negotiation* between software agents
  - Focus: introduction of **general alignment interaction models**

- The approach
  - Is grounded on Barwise and Seligman’s **theory of information** [2]
  - Uses their notion of **information flow** (IF) as a basic formalism

- Alignment is:
  - Defined as a system of **classifications** and **infomorphisms**
  - Obtained via **meaning coordination** between agents $Ag_1$ and $Ag_2$ through the information channel:
    - $C$ is the classification determined by the meaning coordination done before
    - $A_1, A_2$ – respective classifications
    - $f_1, f_2$ – respective infomorphisms

---


The **IF**-based approach has been implemented as the **IF-Map** method for automated ontology mapping [29]


http://www.aktors.org/technologies/ifmap/
Section 2.2: Argumentation Based Frameworks
• Abstract **Argumentation Framework (AF)** introduced by Dung [14] as a pair:

\[
AF = \langle AR, \text{attacks} \rangle
\]

- \(AR\) – a set of arguments
- \(\text{attacks}\) – a binary relation on \(AR\); and
- \(\text{attacks}(A, B)\) signifies that argument \(A\) attacks argument \(B\)

• **Different flavors of AF** used for ontology alignment by agents to determine acceptable mappings in negotiations

Argumentation Frameworks

• **Different flavors of AF** used for ontology alignment:

  - **Value-Based Argumentation Framework (VAF)** by Bench-Capon [3]
    
    \[ AF = \langle AR, attacks, V, val, P \rangle \]
    
    - \( V \) – a non-empty set of values
    - \( val \) – a function which maps the elements of \( AR \) to the elements of \( V \)
    - \( P \) – the set of possible audiences

  - **Voting-based VAF (V-VAF)** and a **Strength VAF (S-VAF)** by Isaac et al. [28]
    
    - **S-VAF** extends **VAF** with a strength function \( S : AR \rightarrow [0, 1] \)
    - **V-VAF** is defined by adding a notion of support
      - A reflexive binary relation over \( AR \) disjoint to \( attacks \)
      - Allows counting arguments as **defenders** (or **co-attackers**) within a particular attack


Section 2.3: Semantic Contexts and Propositional Substitutions

A-BOA Wiki: Propositional Substitutions
Propositional Substitutions

- Given $O_s$ and $O_t$
- Choose a **center of gravity**
  - A pair of “central” concepts with high similarity
- Discover mappings for a **structural context** by
  - Exchanging **hypotheses** (propositions)
  - Trying **substitutions** of own statements by received propositions
  - Measuring **similarity improvement** (several metrics)
- Accepting good propositions (conceding)
- Exclude the pair from the negotiation set
Propositional Substitutions

• Center of Gravity: Paper ↔ ProceedingsPaper

Publisher

Conf MS

TBox: Biblio v.1 ($O^s$)

TBox: Biblio v.2 ($O^t$)
Section 2.4: Demonstration of A-BOA Solution for Instance Migration

A-BOA Wiki: Demonstration of the Agent-Based Software Prototype
Workflow and Tools

- (I) SDiff Discovery Engine (SDDE)
  - Compared to manual
- (II) SDiff Discovery Engine (SDDE)
- (III) Instance Migration Engine (IME)
- (IV) Knowledge Engineer
(I) Discover Structural Changes

- Structural Diff Discovery Engine

- Done manually
(II) Generate Transformation Rules

- Generated by the Structural Diff Discovery Engine
- Imported by the Instance Migration Engine
(III) Migrate Instances

- Instance Migration Engine [10]
  - Generates Migration Log

(IV) Evaluate Migration Log

- Manual – by a Knowledge Engineer
- Decision to be made about a need to refine ...

Migration Log

Migrating instances of class: <http://www.somewhere.com/ontologies/2011/1/BibliographicOntology_1.owl#Person>
To
<http://www.somewhere.com/ontologies/2011/1/BibliographicOntology_1.owl#PCMember>
The initial set of instances:
Migrated instances:

New Condition to Transformation Rule
(added manually)

<concept concept_name="http://www.somewhere.com/ontologies/2011/1/BibliographicOntology_1.owl#Person">
    <condition>
        <supplementaryCondition type="concept" name="PCMember"/>
    </condition>
    <rename>PCMember</rename>
    <removeProperty>name</removeProperty>
    <addProperty datatype="integer">id</addProperty>
    <addProperty datatype="string">expertIn</addProperty>
</concept>
Example: Biblio Instances

a) The instances of Biblio-1

b) The instances of Biblio-2

The result we have to achieve
Final Questions Please

http://isrg.kit.znu.edu.ua/a-boa/