CHAPTER 4

ONTOLOGY MAPPING SYSTEMS – STATE OF THE ART

Ontology mapping has been a subject of discussion and research in the last decade. Many people have devoted much time in finding appropriate solutions to map ontologies which saw a spurt in their growth in the same period. However there seems to be no single point solution to address the problem of ontology mapping. In this chapter, a mix of existing matching systems (Euzenat & Shvaiko 2007) are presented illustrating salient features of each for a comparison and a final tabulation enlisting them.

4.1 PROMPT SYSTEM

PROTEGE is an ontology development application released by Stanford University. It uses ‘PROMPT’, an ontology mapping system, which is a semi-automatic tool, as a plug-in for its ontology mapping functionality (Natalya & Musen 2003). PROMPT maps ontology elements basically by determining string similarity and structure similarity and the ontology is analyzed inside the PROTEGE environment. It guides the developer to merge different ontologies. Then PROMPT suggests the possible mapping between ontologies.

The conflicts between the ontologies being tested are identified and solutions to avoid the conflicts are also provided by the tool “iPROMPT”, which is another interactive ontology merging tool. AnchorPROMT uses graph-based mappings to provide additional information to iPROMPT.
Ontology Extraction is possible with the PROMPTFactor. In the PROMPT ontology processing suite, the main mapping strategies are based on string similarity and structure. The PROMPT architecture working in the PROTEGE environment is indicated in Figure 4.1.

![Prote-2000 Project Browse](image)

**Figure 4.1 PROMPT as a plug-in for PROTEGE**

Input of PROMPT is two ontologies with represented either in OWL or OKBC. Output is the suggestion of mapping and a merging ontology, based on the user choice.

### 4.2 SAMBO SYSTEM

SAMBO (System for Aligning and Merging Biomedical Ontologies) developed at the Linkoping Universitet (Patrick Lambrix & He Tan 2006) is a system that assists the user in aligning and merging two biomedical ontologies. The user performs an alignment process with the help
of alignment suggestions proposed by the SAMBO. The system carries out the actual merging and derives the logical effects of the merge operations. SAMBO requires that the two ontologies under consideration be expressed in DAML+OIL or OWL. Output is based on user choices and is mostly the suggestions of mapping and merging ontology. The main strategies of SAMBO are including combinations of string similarity, synonyms based on WordNet and domain knowledge UMLS (Unified Medical Language Systems, found at http://umlsks.nlm.nih.gov) and structure-based strategies and algorithms based on machine learning. Figure 4.2 shows the SAMBO ontology alignment system.

![Figure 4.2 SAMBO Ontology Alignment System](image-url)
4.3 FCA-MERGE SYSTEM

FCA-Merge is from University of Karlsruhe, Germany (Gerd Stumme & Alexander Maedche 2001) is a method for merging ontologies based on mathematical techniques from Formal Concept Analysis (Bernhard Ganter & Rudolf Wille 1997). FCA-Merge is a bottom-up technique for merging ontologies based on a set of documents. It consists of three steps namely (i) instance extraction, (ii) concept lattice computation and (iii) the generation of the merged ontology based on the concept lattice as shown in Figure 4.3.

Input of FCA-Merge is two ontologies and a set of documents that are relevant to both ontologies. FCA_Merge compares the ontologies and maps elements and the final output is a merged ontology. The strategies of FCA-Merge are based on string similarity, FCA, instances and structure.

![Figure 4.3 FCA merge](image)

4.4 GLUE SYSTEM

GLUE is a system developed at the University of Washington (AnHai Doan et al 2003) and employs machine learning techniques to find mappings. Given two ontologies, for each concept in one ontology GLUE
finds the most similar concept in the other ontology. Figure 4.4 shows GLUE’s architecture. The Distribution Estimator takes as input two taxonomies and instances. Then it applies multiple machine learners and exploits information in concept instances and taxonomic structure of ontologies. It uses a probabilistic model to combine results of different learners. Next, GLUE feeds the above results into the Similarity Estimator that applies a user-supplied similarity function to compute a similarity value for each pair of concepts to generate similarity matrix. The Relaxation Labeler module then takes the similarity matrix combined with domain-specific constraints and heuristic knowledge, and finds mappings.

Figure 4.4 The GLUE architecture
4.5 OLA SYSTEM

OLA, abbreviated for “OWL-Lite Aligner” from INRIA Rhone-Alpes and University of Montreal, (Euzenat & Petko Valtchev 2004) is a system that is designed with the idea of balancing the contribution of each element of the ontology. It first transforms the input ontologies to graph structures and marks the relationships between entities. The similarity between nodes in the graph structures will depend on the category of nodes (e.g., class, property) considered and all the features of this category like the superclasses and properties.

Input of OLA is two OWL ontologies. OLA uses many elements (e.g., classes, properties, constraints, taxonomy, instances) in the ontologies. Output is one-to-many correspondences. The strategies of OLA are based on string similarity, synonyms, structure and instances. The OLA architecture is shown in Figure 4.5.

![Figure 4.5 The OLA architecture](image)

It provides a common conceptual basis, and hence, can be used for comparing (analytically) different existing ontology matching systems. It can help in designing a new matching system, or an elementary matcher, taking
advantages of state of the art solutions. It can help in designing systematic benchmarks, e.g., by discarding features one by one from ontologies, namely, what class of basic techniques deals with what feature.

4.6 IF-MAP SYSTEM

IF-Map (Information-Flow-based Map) developed at the University of Southampton and University of Edinburgh, by Yannis Kalfoglou and Marco Schorlemmer (2003) is an ontology mapping system based on the Barwise–Seligman theory of information flow (Jon Barwise & Jerry Seligman 1997). The basic principle of IF-Map is to match two local ontologies by looking at how these are related to a common reference ontology. The IF-Map matching process flow is illustrated in Figure 4.6. It is assumed that such a reference ontology represents an agreed understanding that facilitates the sharing of knowledge. This means that two local ontologies have significant fragments of them that conform to the reference ontology. It is also assumed that the local ontologies are populated with instances, while the reference ontology does not need to.

![Figure 4.6 Process of IF-Map](image-url)
When the mappings are not available, IF-Map generates candidate pairs of mappings (called info-morphism in information flow theory) and artificial instances. These are generated through the enforcement of constraints that are induced by the definition of the reference ontology and by heuristics based on string-based) and structure-based methods. IF-Map deals with ontologies expressed in KIF or RDF.

4.7 NOM AND QOM SYSTEMS

NOM for Naive Ontology Mapping and QOM for Quick Ontology Mapping are components of the FOAM frame work (Marc Ehrig & Steffen Staab 2004). These systems were developed by University of Karlsruhe. NOM adopts the idea of parallel composition of matchers from COMA. Some improvements with respect to the COMA are in the set of elementary matchers based on rules, exploiting explicitly codified knowledge in ontologies, such as information about super and sub concepts, super and sub properties, etc. For example, a rule states that if super concepts are the same then the actual concepts are similar to each other. These rules are based on various terminological and structural techniques. QOM is an approach that improves the efficiency of NOM. The idea is that the loss of quality, compared to a standard baseline, is marginal, but the improvement in efficiency can be significant that it allows for mapping large-size ontologies.

In an effort to make an efficient mapping algorithm, several measures are used in the processing. The various steps of the NOM-QOM system mapping process is shown in Figure 4.7. For example, in the second step, it uses heuristics to lower the number of candidate mappings; in the third step, it avoids the complete pair-wise comparison of trees in favour of top-down strategy; in the fourth step, it applies sigmoid function which emphasizes high individual similarities and de-emphasizes low individual
similarities; in the fifth step, it uses a threshold to discard spurious evidence of similarity.

![Diagram](image)

**Figure 4.7 The NOM-QOM mapping process**

Input of QOM is two OWL or RDFS ontologies. QOM uses many elements (e.g., classes, properties, instances) in the ontologies. Output is one-to-one or one-to-none correspondences. The strategies of QOM are based on string structure, similarity and instances.

### 4.8 S-MATCH SYSTEM

S-Match, developed at the University of Trento, implements the idea of semantic matching (Fausto Giunchiglia & Pavel Shvaiko 2003). The first version of the S-Match system was a rationalised re-implementation of “CtxMatch” with a few added functionalities (Fausto Giunchiglia, Pavel Shvaiko & Mikalai Yatskevich 2004). Later the system underwent several evolutions, including extensions of libraries of element- and structure-level matchers, adding alignment explanations as well as iterative semantic matching (Fausto Giunchiglia et al 2007). S-Match is limited to tree-like structures and does not consider properties or roles.

S-Match takes as input two graph-like structures, e.g., classifications, XML schemas, ontologies, and returns as output logic relations such as equivalence and subsumption that are supposed to hold
between the nodes of the graphs. The relations are determined by (i) expressing the entities of the ontologies as logical formulas, and (ii) reducing the matching problem to a propositional validity problem.

In particular, the entities are translated into propositional formulas which explicitly express the concept descriptions as encoded in the ontology structure and in external resources, such as WordNet. This allows for a translation of the matching problem into a propositional validity problem that can then be efficiently resolved using (sound and complete) state of the art propositional satisfiability solvers. The various stages of S-Match mapping system is shown in Figure 4.8 as functional blocks.

![Figure 4.8 The S-Match architecture](image)

4.9 MOMIS SYSTEM

MOMIS (Mediator Environment for Multiple Information Source) (Domenico Beneventano et al 2003) is an approach that creates a global virtual view (GVV) of the local sources. Figure 4.9 shows the process for building the GVV for a set of Web pages in five steps:
- Local source schemata extraction. Wrappers generate schemas for the local sources and translate them into the common language ODLI3 (extension of Object Definition language).

- Local source annotation with WordNet. The integration designer chooses a meaning for each element of a local source schema, according to the WordNet lexical ontology.

- Common thesaurus generation. It describes relationships of inter-schema and intra-schema knowledge about classes and attributes of the source schemata.

- GVV generation. It generates a global schema and sets of mappings with local schemata by using the common thesaurus and the local schema descriptions.

- GVV annotation. It semi-automatically generates mapping between local schemas and global schema by exploiting the annotated local schemas.

Figure 4.9 MOMIS system architecture
4.10 MAPPING SYSTEMS OVER VIEW SUMMARY

The panorama of systems considered has illustrated the diversity of basic techniques by the variety of strategies for combining them. Moreover, usually each individual system innovates on a particular aspect. However, there are several constant features that are shared by the majority of systems. In summary (Euzenat & Shvaiko 2007), the following can be observed concerning the presented systems:

Based on the number of systems considered we can conclude that schema-based matching solutions have been so far investigated more intensively than the instance-based solutions. We believe that this is an objective trend, since we have striven to cover state of the art systems without bias towards any particular kind of solutions.

Most of the systems under consideration focus on specific application domains, such as books and music, as well as on dealing with particular ontology types, such as DTDs, relational schemas and OWL ontologies. Only few systems aim at being general, i.e., suit various application domains, and generic, i.e., handle multiple types of ontologies.

- Most of the approaches take as input a pair of ontologies, while only few systems take as input multiple ontologies.
- Most of the approaches handle only tree-like structures, while only few systems handle graphs.
- Most of the systems focus on discovery of one-to-one alignments, while only few systems have tried to address the problem of discovering more complex correspondences, such as one-to-many and many-to-many.
## Table 4.1 Ontology Matchers and Strategies

<table>
<thead>
<tr>
<th>S.No</th>
<th>Systems</th>
<th>Element Level – Syntactic</th>
<th>External</th>
<th>Structure-level Syntactic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DIKE</td>
<td>String-based, Domain compatibility</td>
<td>WordNet</td>
<td>Matching of neighbourhood</td>
</tr>
<tr>
<td>2</td>
<td>SKAT</td>
<td>String-based</td>
<td>Auxiliary thesaurus, Corpus-based</td>
<td>Taxonomic structure, Matching of neighbours</td>
</tr>
<tr>
<td>3</td>
<td>H-Match</td>
<td>Domain compatibility, Language-based, Domains and ranges</td>
<td>Common thesaurus</td>
<td>Matching of neighbours, via thesaurus, Relations</td>
</tr>
<tr>
<td>4</td>
<td>Tess</td>
<td>String-based, Domain compatibility</td>
<td>-</td>
<td>Matching of neighbours</td>
</tr>
<tr>
<td>5</td>
<td>Anchor - Prompt</td>
<td>String-based, Domains and ranges</td>
<td>-</td>
<td>Bounded paths matching: (arbitrary links), Taxonomic structure</td>
</tr>
<tr>
<td>6</td>
<td>OntoBuilder</td>
<td>String-based, Language-based</td>
<td>Thesaurus look up</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>Cupid</td>
<td>String-based, Language-based, Datatypes, Key properties</td>
<td></td>
<td>Weighted by levels</td>
</tr>
<tr>
<td>8</td>
<td>COMA &amp; COMA++</td>
<td>String-based, Language-based, Datatypes,</td>
<td>Auxiliary thesauri, Alignment reuse, Repository of structures</td>
<td>DAG (tree) matching with a bias towards various structures, e.g., leaves</td>
</tr>
<tr>
<td>9</td>
<td>Similarity</td>
<td>String-based, Datatypes, Key properties</td>
<td>-</td>
<td>Iterative fixed point computation</td>
</tr>
<tr>
<td>10</td>
<td>ToMAS</td>
<td>-</td>
<td>External alignments</td>
<td>Preserving consistency, Structure comparison</td>
</tr>
<tr>
<td>11</td>
<td>MapOnto</td>
<td>-</td>
<td>External alignments</td>
<td>Structure comparison</td>
</tr>
<tr>
<td>12</td>
<td>OntoMerge</td>
<td>-</td>
<td>External alignments</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>MoA</td>
<td>Language-based</td>
<td>WordNet</td>
<td>-</td>
</tr>
<tr>
<td>14</td>
<td>BayesOWL</td>
<td>Text classifier</td>
<td>Google</td>
<td>Bayesian inference</td>
</tr>
<tr>
<td>15</td>
<td>IF-Map</td>
<td>String-based</td>
<td>-</td>
<td>Formal concept analysis</td>
</tr>
<tr>
<td>16</td>
<td>OLA</td>
<td>Language-based, Datatypes</td>
<td>WordNet</td>
<td>Iterative fixed point computation, Matching of neighbours, Taxonomic structure</td>
</tr>
</tbody>
</table>
Most of the systems focus on computing confidence measures in the [0 1] range, most often standing for the fact that the equivalence relation holds between ontology entities. Only few systems compute logical relations between ontology entities, such as equivalence, subsumption.

Table 4.1 summarises how the matching systems cover the solution space in terms of the classifications. For example, DIKE exploits string-based element-level matchers, external matchers based on WordNet, propositional satisfiability techniques, etc. OLA, in turn, exploits, besides string based element-level matchers, also a matcher based on WordNet, iterative fixed point computation, etc. Table 4.1 also testifies that ontology matching research so far was mainly focused on syntactic and external techniques. In fact, many systems rely on the same string-based techniques. Similar observation can be also made concerning the use of WordNet as an external resource of common knowledge. In turn, semantic techniques have rarely been exploited, this is only done by HMatch and OntoMerge. Concerning instance-based system, techniques based on naive Bayes classifier and common value patterns are the most prominent.