Achieving interoperability in e-government services with two modes of semantic bridging: SRS and SWRL

Saravanan Muthaiyah[1] and Larry Kerschberg[2]

[1] George Mason University, E-Center for E-Business, smuthaiy@gmu.edu
[2] George Mason University, E-Center for E-Business, kersch@gmu.edu

Abstract

Data heterogeneity in the public sector is a serious problem and remains to be a key issue as different naming conventions are used to represent similar data labels. The e-government effort in many countries has provided a platform for government entities and their business partners to exchange data through Information Communication Technologies (ICT) and standards such as RosettaNet (B2B data exchange standard), EDIFACT (Electronic Data Interchange for Administration, Commerce, and Transport), XML (Extensible Mark-up Language) and EDI (Electronic Data Interchange). However, e-government efforts have not really resolved data heterogeneity problems significantly due to limitation of these standards. One such limitation is the inability of data inheritance. In order to solve this problem with emphasis on Service Oriented Architectures (SOA) and Web Services, a semantically enriched web service for the public sector is needed. Thus we propose an ontology-based solution which allows data inheritance and polymorphism. This goal of this paper is to show how heterogeneous e-government documents can be semantically matched. We propose a shared hierarchical knowledge repository approach and a detailed process methodology for semantic mediation. A two-part semantic mediation approach using SRS (Semantic Relatedness Scores) and SWRL (Semantic Web Rule Language) is highlighted. Both measures are complimentary and provide the semantics necessary for resolving schema heterogeneity. Our approach incorporates a rule-based engine that reads and executes SWRL rules (i.e. RacerPro). We also adopted several tools for proof-of-concept such as Protégé (i.e. ontology editor) and JESS (Java Expert Shell System).

Key-Words*:* Semantic, Syntactic, Semantic Web, EDI, XML, Web Services, Ontology, Mediation, Reasoning

1 Introduction

The stovepipe phenomenon creates a barrier for government bodies to freely exchange data. In most e-government initiatives today the main goal would be for government entities to seamlessly exchange data via parsing mechanisms such as EDI and XML. Data from different domains such as health records and national registration would require manual data transformations in order to achieve data interoperability. SOAP (Simple Object Access Protocol), UDDI (Universal Discovery Description Interchange) and WSDL (Web Services Description Language) standards only provide syntactical interoperability but semantic mediation provides semantic interoperability which is more reliable. Without semantics, a shared conceptualisation would not be possible. Semantics i.e. meanings, provide machine-understandable data that enable machines to reason data, draw inferences and perform semantic mediation on-the-fly. Currently only syntactic matching and data parsing is carried out for public agencies to share and exchange data. Without semantics in place, a human domain expert has to mix and match services between agencies manually [12][15].

With a multitude of e-service requests between government agencies, a human being, processing these service requests has to understand the implicit semantics and choreograph them to establish syntactical interoperability, which evidently causes delay, unavoidable human errors and most importantly inaccuracy due to the lack of semantic processing. Semantic mediation is based on meanings unlike syntactic matching that is just based on string, prefix and suffix matching of schemas [14]. Thus, a less labour-intensive approach for schema integration is needed. The Semantic Web provides several components that are necessary for the creation of this vision, mainly ontologies, Web Ontology Language (OWL), resource description framework (RDF), description logic and reasoning capabilities. It includes the provision of metadata and data interchange formats such as N3 (Notation 3), Turtle (Terse RDF Triple Language) and N-Triples. Ontologies support formal logics which allow data inference and are more powerful than XML schemas. They are defined as a shared conceptualization of a domain of knowledge [5].

Currently, e-government public services utilize very specialized applications that are only available to certain agencies and not all agencies participating in the consortium. To ensure interoperability some countries have implemented XML schemas with Web Service interfaces e.g. Danish e-government [20]. The effort is similar to that of maintaining a shared repository to ensure interoperability for all government systems by using the same schema language to avoid reusability problems of syntax specific definitions [3][4][7]. Schema interoperability guidelines are issued for this purpose and in most cases made mandatory to ensure all government agencies adhere to the same naming conventions. For example in England the schema had to be registered with UK GovTalk [21]. Examples of these schemas for addresses are CorrespondenceAddress, HomeAddress, BusinessAddress and ElectoralAddress. In some cases interoperability is only limited to a boundary within central control such as the UN/CEFACT initiative [22].

This creates a barrier for inter-organizational services between public agencies of different domains outside that boundary. The lack of semantics causes data exchange to be impossible. For example figure 1 illustrates a data heterogeneity problem for inter-organizational services between public agencies. The scenario here is that a customer wants to renew his driver's license online. He first logs into the DMV (Department of Motor Vehicle) portal and selects the type of service. Then he provides essential information such as full_name, DOB (10-10-1965), DMV customer ID (A33-05-7156) and address (1234 Oakton Circle Rd Arlington VA 22202). These details are passed on to a license renewal inspector where details provided by customer are verified. At the same time a mode of payment is selected which is verified by the records inspector before licence renewal is performed. Data is then passed on to the DMV License Renewal server which validates the data and passes on details to the license renewal clerk who updates and verifies renewal data. When payment is validated with updates from the DMV License Renewal server, the records inspector receives this information with updates exchanged from DMV Records server. The bank is then notified for the charges and the customer's account is debited. An option for printing a receipt is also provided. The customer then waits for his renewed licence to arrive in the mail.

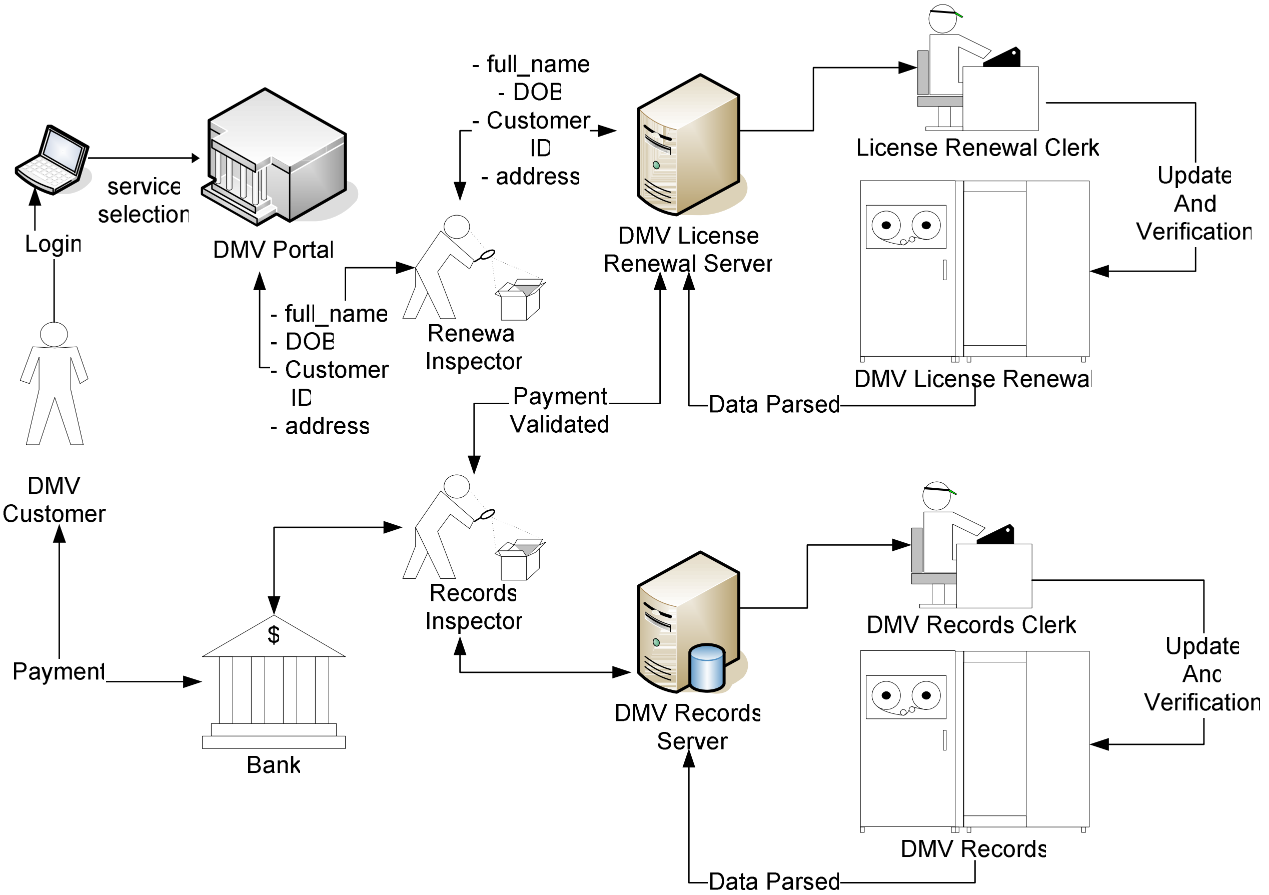


Figure 1 DMV licence renewal process

Based on the process above, let's analyze the reasons for data heterogeneity problems in this environment. Data heterogeneity is caused by the separate data definitions or naming conventions maintained by both the DMV agencies depicted above i.e. DMV Licence Renewal and DMV Records for their customer records. For example, DMV Records maintains first_name, middle_name and last_name however DMV Licence Renewal maintains a complex string called full_name that literally combines first_name, middle_name and last_name. Also address in the DMV Licence Renewal is treated as a complex string and in DMV Records it is divided into street name, city, state and zipcode. For example ("Oakton Circle Rd Arlington VA 22202") compared to ("Oakton Circle Rd", "Arlington", "VA", "22202"). Since public agencies develop their own systems independently from each other, the granularity of how information is expressed can differ greatly. As we mentioned earlier having all agencies to adhere to one type of naming convention and making it mandatory is not practical. This would be analogous to developing a global ontology with a global schema [1][3]. A more practical approach would be to focus on creating a semantic bridge between domain specific local ontologies [13][14]. [15][16]. This provides the semantic interoperability for inter-agency data exchanges.

2 Shared Hierarchical Ontology Structures

We propose that domain specific local ontologies in public agencies e.g. DMV Licence Renewal and DMV Records maintain their own naming conventions for their customer records and at the same time borrow general concepts from an upper ontology. This helps facilitate knowledge reuse and allows domain experts to express their knowledge, even when they don't completely agree with each other. This type of structure is referred to as a shared hierarchical ontology [2]. In this structure, knowledge is organized in different levels, each inheriting knowledge from upper level or parent ontologies. For multiple inheritance relationships it is important for knowledge inherited in local ontologies to be consistent with upper ontologies with no naming clashes. Ontologies are hand crafted by domain experts in their respective public agencies and it will be impossible to find a perfect ontology that covers all aspects of a shared domain of knowledge i.e. public services for the DMV. In order to maintain rich definitions, a shared hierarchical structure is crucial [2][19].

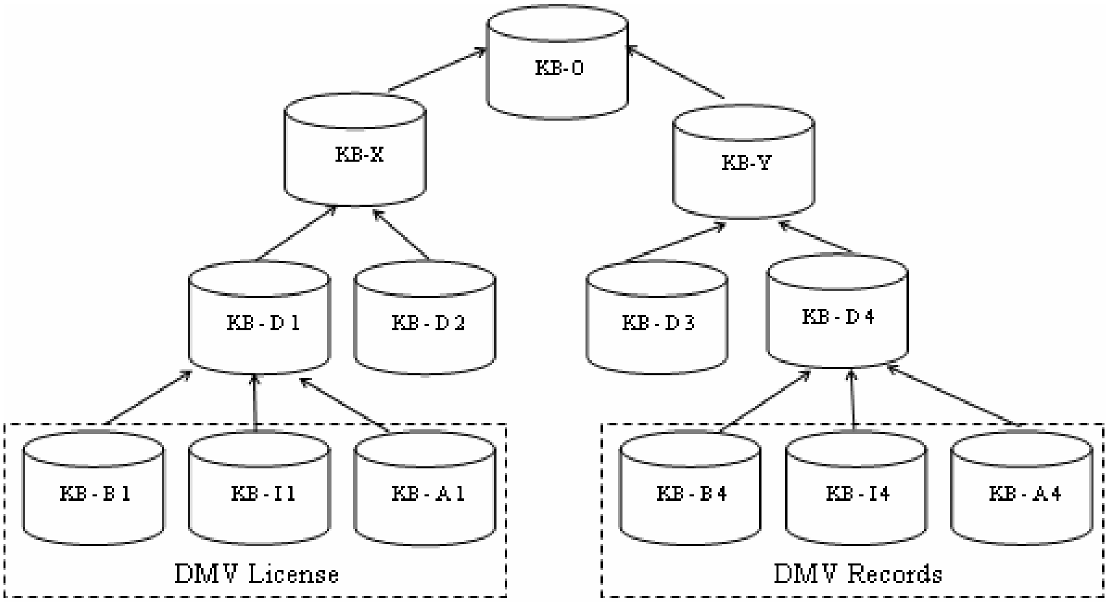


Figure 2 Shared hierarchical ontology knowledge repository

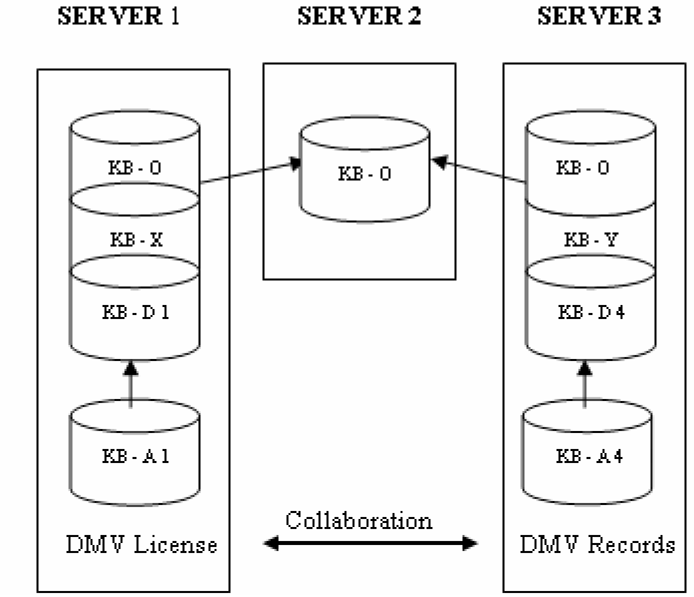


Figure 3 Distributed shared hierarchical ontology knowledge repository

Figure 2 illustrates a shared hierarchical knowledge repository that begins with knowledgebase (KB-O) on the top. Two public agencies i.e. DMV Licence Renewal and DMV Records would have their own local knowledgebases that house their local ontologies e.g. KB-B1, KB-I1, KB-A1 KB-B4, KB-I4 and KB-A4. DMV licence renewal contains KB-B1, KB-I1 and KB-A1 and would inherit knowledge from KB-D1 which inherits data from KB-X. DMV Records contains and manages ontologies in KB-B4, KB-I4 and KB-A4. These ontologies inherit data from KB-D4, which then inherits from KB-Y. Both KB-X and KB-Y inherit data from KB-O. DMV Licence Renewal and DMV Records are aware that their inherited knowledge is common to KB-O (see figure 3) when they collaborate. Knowledge reuse becomes easier in this way. Particularly when creating a new ontology, we can determine which parent ontology to inherit from and start adding the new knowledge to what already exists.

Although the hierarchical structure helps knowledge reuse, it will not be realistic to assume that all developed ontologies will be under one central control and available at all times [18]. As such, a distributed model of a hierarchical repository is more appropriate (see figure 3). There are three servers (i.e. server 1, server 2 and server 3) which are distributed. The servers could be maintained by other agencies within the DMV consortium or even other agencies like the National Registration Department (NRD) or the Internal Revenue Service (IRS). To solve the availability problem, when a new ontology inherits knowledge from another upper ontology, a copy of the inherited knowledge is made available locally. The reason for this is that the parent ontology does not have to be available online at all times in order to have all their children ontologies functioning properly.

3 Semantic Bridging Process Methodology

In this section we present the process methodology for semantic bridging (see figure 4) [12][14]. The first step is called ontology development. It involves creating or selecting the source ontology (SO) and target ontology (TO). As mentioned earlier DMV Licence Renewal could maintain its own set of definitions that is different from DMV Records. This is really a very important step for the ontologist as the domain specific local ontologies of public agencies e.g. DMV Licence Renewal and DMV Records have to be determined before mediation can be done. If DMV Licence Renewal is selected to be SO then TO would be DMV Records.

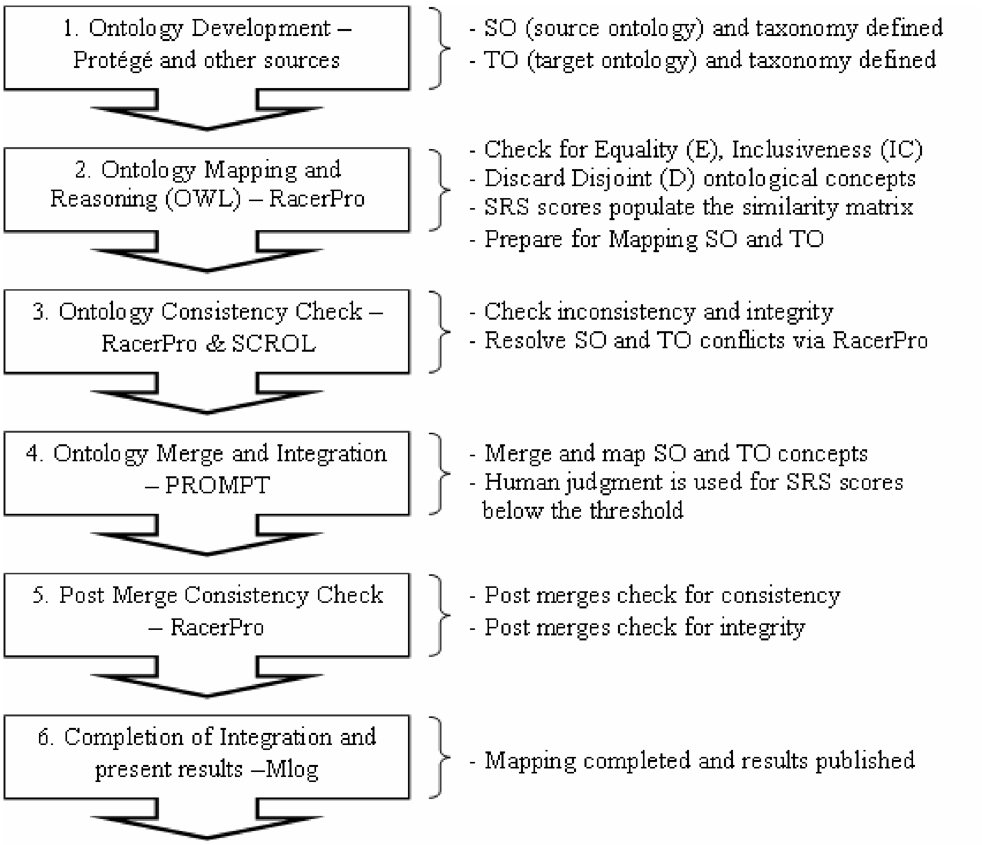

1. Ontology Development – Protégé and other sources
- SO (source ontology) and taxonomy defined
- TO (target ontology) and taxonomy defined
2. Ontology Mapping and Reasoning (OWL) – RacerPro
- Check for Equality (E), Inclusiveness (IC)
- Discard Disjoint (D) ontological concepts
- SRS scores populate the similarity matrix
- Prepare for Mapping SO and TO
3. Ontology Consistency Check – RacerPro & SCROL
- Check inconsistency and integrity
- Resolve SO and TO conflicts via RacerPro
4. Ontology Merge and Integration – PROMPT
- Merge and map SO and TO concepts
- Human judgment is used for SRS scores below the threshold
5. Post Merge Consistency Check – RacerPro
- Post merges check for consistency
- Post merges check for integrity
6. Completion of Integration and present results –Mlog
- Mapping completed and results published

Figure 4 Semantic Bridging Process Methodology

In the second step, ontologies are checked for equality (E), inclusiveness (IC) and all disjoint (D) concepts are negated. We use the following symbols: 1) C for concept or class, 2) c for attributes or slots and 3) O for ontology, for simplicity [14]. The detailed definition is as follows:

- **Equality (E)**

  Two classes (C) of DMV Licence Renewal and DMV Records are **equal** if, they: 1) have semantically equivalent data labels, 2) are synonyms or 3) have the same slots or attribute names. For example: 1) *C_1=Customer & C_2=Customer*, 2) *C_1=Customer & C_2=Client* and 3) C_1 and C_2 have same slots (c) names, e.g. *c_1= <CustID, name, address, DOB>* and *c_2= < CustID, name, address, DOB >.*

- **Inclusiveness (IC)**

  Two classes (C) of DMV Licence Renewal and DMV Records are **inclusive** if, the attribute (c) of one is inclusive in the other. For example if c_i = StreetAddress and c_j = Address, then c_i is a type of c_j. In other words *StreetAddress* is inclusive in *Address* c_i ($c_i \geq c_j$). This is applicable to *hyponyms.*

- **Disjoint (D)**

  Two classes (C) of DMV Licence Renewal and DMV Records are **disjoint** if, their attributes (c), c_i and c_j have nothing in common between them. For example *charge* and *name* are not equal or inclusive and have nothing in common which results in an empty set ø.

Semantic Relatedness Score (SRS) is determined based on a hybrid matching technique that combines syntactic and semantic matching. The scores are used to populate a similarity matrix which is used as a basis to match the different schemas that the public agencies have defined in their ontologies. In the third step is called the respective ontologies are tested for consistency. This is to ensure that the concepts that have been mapped are in fact consistent and that there are no conflicting concepts. We use a reasoning engine (i.e. RacerPro) to check for these inconsistencies. If inconsistencies are discovered then they are resolved immediately. Consistency is defined as follows:

- **Consistency (CN)**

  Two classes of DMV Licence Renewal and DMV Records are **consistent** if, all the attributes for C_1 (O_1) i.e. *CustID, name, address* and *DOB* have nothing in common (c_1~*CustID* ≠ c_2~*name* ≠ c_3~*address* ≠ c_4 ~*DOB*) s.t. $c_1 \cap c_2 = \{\}$. All slots (c_1~*CustID*, c_2~*name*, c_3~*address*, c_4 ~*DOB*) must be subsets of C_1 (O_1).This can be configured with RacerPro.

In step four, SO and TO are merged and integrated. At this point to ensure that schemas are perfectly matched the ontologist performing the merge selects the data labels which have higher scores than the threshold score. SRS produces scores between 0 and 1. A score of 0 means that data labels have no match and 1 indicates a perfect match. For instance if DMV Licence Renewal had the following schema, </Address> and at the same time if DMV Records, had the following schema </Address> it would result in a score of 1. For data labels that are in the range of 0 to 1, the ontologist is presented with the scores above the threshold of 0.5 (e.g. t>0.5). All scores below the threshold are maintained in a log for the ontologist to refer to at a later point [14]. In step five, we use the same reasoning engine (i.e. RacerPro) to check for post matching inconsistencies. This ensures consistency is maintained even after data labels are matched. It also ensures integrity of matched data labels. Lastly in step six, a log report is produced and results are published. This data is also annotated to document all changes that have been updated. We do this to provide other ontologists to trace the lineage of data that had been used to make any changes during the mapping process. In the next section we describe how SRS is determined.

4 Semantic Relatedness Score (SRS)

SRS is a hybrid measure that comprises syntactic matching (SYN) and semantic matching (SEM) [14]. SYN uses approximate string matching to integrate data labels based on a number of deletions, insertions and substitutions to match a source string with a target string. Suppose we had two concepts i.e.</client> and </customer>. If "client" is chosen to be the source string, "customer" would be the target string. SYN would give on a scale of 0 to 1 the syntactic distance (*d*) of these two strings [19]. We convert distance (d) to similarity (s) by using the inverse of distance (i.e.1-d). SYN does not measure similarity based on meanings and does not consider the ontological structure or taxonomy of concepts being matched. However it gives a gross match. This is why we combine it with SEM. SEM uses representation of meaning to measure similarity by considering word senses that use linguistic and cognitive measures. Although there are many linguistic algorithms, we use Lin, Gloss Vector, WordNet Vector and LSA (Latent Semantic Analysis) to determine SRS. Scores from these measures are aggregated and normalized to produce SRS. Our experiments have proven that the combination of these four measures provides higher reliability and precision [14]. Figure 5 below show that SRS had higher precision and relevance scores when matched against actual feedback received from human experts, i.e. human cognitive responses (HCR). The idea was to see how accurate SRS was with a human expert's judgement. This was how we validated our findings that SRS had a better match compared to just using SYN matching for the ontology mediation process. An experiment was conducted with 30 word-pairs (see figure 6 and appendix 1) based on a study done at Princeton [6] to test the accuracy of SRS. In terms of relevance, SRS had a 96.67% match with scores obtained from human subjects and a 40% match in terms of precision. These were higher compared to how pure SYN scores fared with human responses [14]. As we can see in figure 5, SYN scores have lower relevance (73.33%) and precision (16.67%). Figure 6 shows the correlation of SRS scores with HCR which was positive r = 0.919 (91.9%). The SRS scores also had a smaller variance when matched against HCR scores compared to pure SYN scores. A matching agent will run matches based on SRS and provide results to the ontologist this makes ontology matching less laborious if otherwise would have to be done manually by the ontologist.

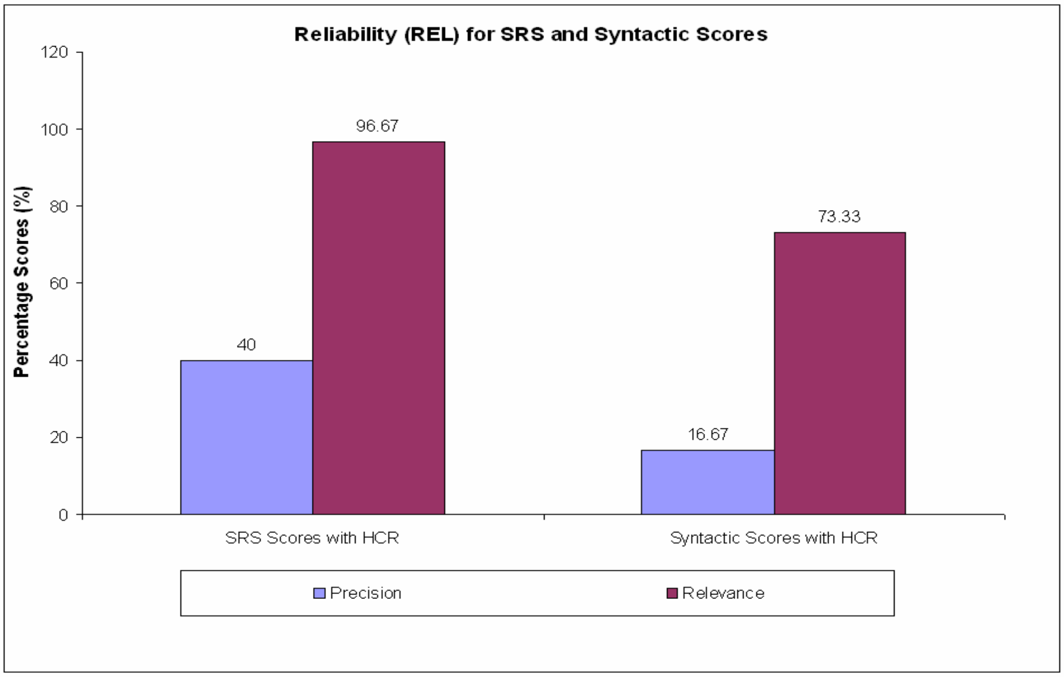


Figure 5 SRS precision and relevance scores compared to HCR scores

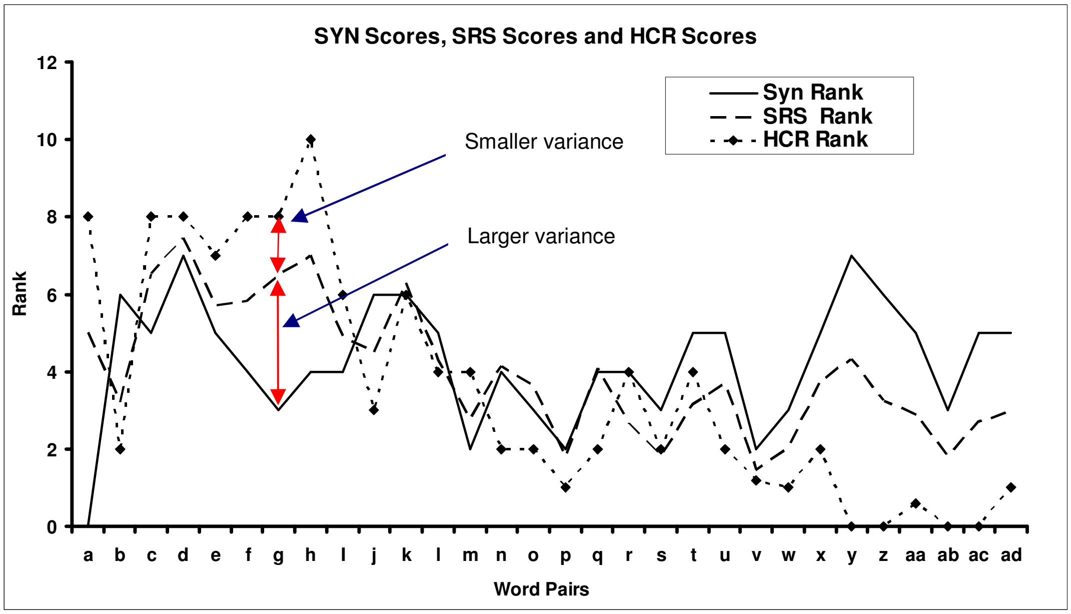
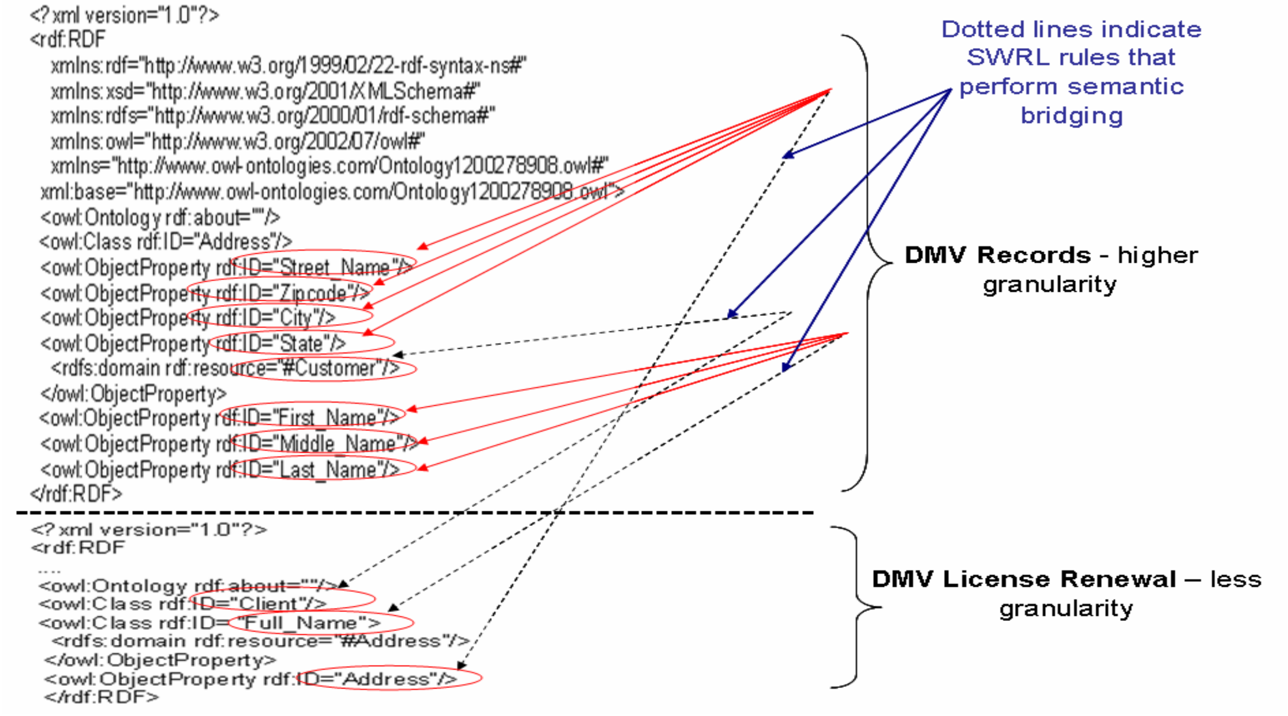


Figure 6 SRS scores variance with HCR scores

5 Semantic Bridging via Reasoning with SWRL

In this section we present how semantic bridging with SRS can be further extended by using rules. SRS provides the ontologist all schemas that are likely to be matched with high reliability and precision [14]. Rules on the other hand are cardinality constraints that can be used for matching data labels, schemas and concepts. Rules can be predefined ahead of time so that frequently appearing schemas can be matched automatically [10]. To match schemas on names for instance we can write a rule that would match *</first_name>*, *</middle_name>* and *</last_name>* with *</full_name>*.If we had schemas for example *</street_name>*, *</city>, </state>* and *</zipcode>* to define an address in one domain ontology and defined as just *</address>* in another, a simple rule can be executed to match them on-the-fly. As mentioned earlier in section one and two, DMV Licence Renewal and DMV Records may have domain specific schema definitions that are different. Since establishing services between them will be an ongoing task rules could provide an automatic solution for creating homogeneity amongst heterogeneous schemas used by them [10] [17].

In view of the reasoning aspects that are possible in the Semantic Web, that supports web services, we propose an approach where rules can be used to match concepts. Reasoning is an approach when agents in a knowledge system perform tasks by inference [8][9]. Given the following statement "If X has a son Y, and X has a brother Z, given that X is a male" the agent is able to then infer that Y has an uncle, Z. The agent does not need to be explicitly told about the relationship between Y and Q. As long as uncle is defined earlier an agent will be able to infer this quickly. SWRL is based on OWL and RuleML (Rule Markup Language). It enables OWL axioms to include Horn-logic that can be used to execute rules in a knowledgebase like the ones that public agencies will need to share (see section 2). SWRL rules show the implication between an antecedent (body) and consequent (head). In other words if the antecedent holds true, then the consequent must hold true also. In our example earlier if antecedents "X has a son Y, and X has a brother Z", X is a male" is all true then the consequent must also hold true which is "Y has an uncle, Z". With rules in place we can easily automate matching of schemas on-the-fly, which otherwise would be very labour intensive. As such SWRL rules and SRS would be complementary efforts towards semantic bridging.



```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.owl-ontologies.com/Ontology1200278908.owl#"
 xml:base="http://www.owl-ontologies.com/Ontology1200278908.owl">
 <owl:Ontology rdf:about=""/>
 <owl:Class rdf:ID="Address"/>
 <owl:ObjectProperty rdf:ID="Street_Name"/>
 <owl:ObjectProperty rdf:ID="Zipcode"/>
 <owl:ObjectProperty rdf:ID="City"/>
 <owl:ObjectProperty rdf:ID="State"/>
  <rdfs:domain rdf:resource="#Customer"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="First_Name"/>
 <owl:ObjectProperty rdf:ID="Middle_Name"/>
 <owl:ObjectProperty rdf:ID="Last_Name"/>
</rdf:RDF>
```

```
<?xml version="1.0"?>
<rdf:RDF
 ....
 <owl:Ontology rdf:about=""/>
 <owl:Class rdf:ID="Client"/>
 <owl:Class rdf:ID="Full_Name">
  <rdfs:domain rdf:resource="#Address"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="Address"/>
 </rdf:RDF>
```

Dotted lines indicate SWRL rules that perform semantic bridging
DMV Records - higher granularity
DMV License Renewal – less granularity

Figure 7 Semantically Bridging DMV Records and DMV Licence Renewal ontologies via SWRL rules

Based on the example given in section 1, DMV Records maintains first_name, middle_name and last_name however DMV Licence Renewal maintains a complex string called full_name. The address in the DMV Licence Renewal is treated as a complex string and in DMV Records it is divided into street_name, city, state and zipcode. Figure 7, depicts customer ontology for DMV Records where customer name and address is expressed with greater granularity. It also shows the client ontology for DMV Licence Renewal where customer name and address is expressed with less granularity. Dotted lines indicate semantic bridging that is done via SWRL rules. We will demonstrate how the rules are written in the next section.

5.1 Writing rules in SWRL

Based on the process of semantically bridging the ontologies in figure 7 earlier, in this section we show how SWRL rules are written to accomplish that task. The first rule would be to associate *</Street_Name>*, *</City>*, *</State>* and *</Zipcode>* from DMV Records to *</Address>* in DMV Licence Renewal. The following is how the rule 1 is expressed:

Rule 1: hasStreet_Name(?x1,?x2) ^ hasZipcode(?x2,?x3) →hasStreetZipAddress (?x1,?x3)

This implies that the (Antecedent (hasStreet_Name (I-variable(x1) I-variable(x2)), hasZipcode (I-variable(x2) I-variable(x3))) and thus the consequent would be (hasStreetZipAddress (I-variable(x1) I-variable(x3)))).

The following is how rule 2 is expressed:

Rule 2: hasCity (?x1,?x2) ^ hasState(?x2,?x3) →hasCityStateAddress (?x1,?x3)

This implies that the (Antecedent (hasCity (I-variable(x1) I-variable(x2)), hasState (I-variable(x2) I-variable(x3))) hasCityStateAddress (I-variable(x1) I-variable(x3))) and thus the consequent would be (hasRecipient (I-variable(x1) I-variable(x3)))).

Rule 3 combines rule 1 and 2 to determine address and is expressed as:

Rule 3: hasStreetZipAddress ^ hasCityStateAddress→hasAddress

This implies that the antecedent hasStreetZipAddress and hasCityStateAddress determine the consequent hasAddress. This rule concludes the matching of the addresses of both ontologies. A simple rule (i.e. rule 4) can be executed to map *</customer>* of DMV Records to *</client>* of DMV License Renewal which is expressed as <owl:equivalentClass> or owl:sameAs. The last rule would be to associate *</First_Name>*, *</Middle_Name>* and *</Last_Name>* from DMV Records to *</Full_Name>* in DMV Licence Renewal. The last rule, rule 5 is expressed as:

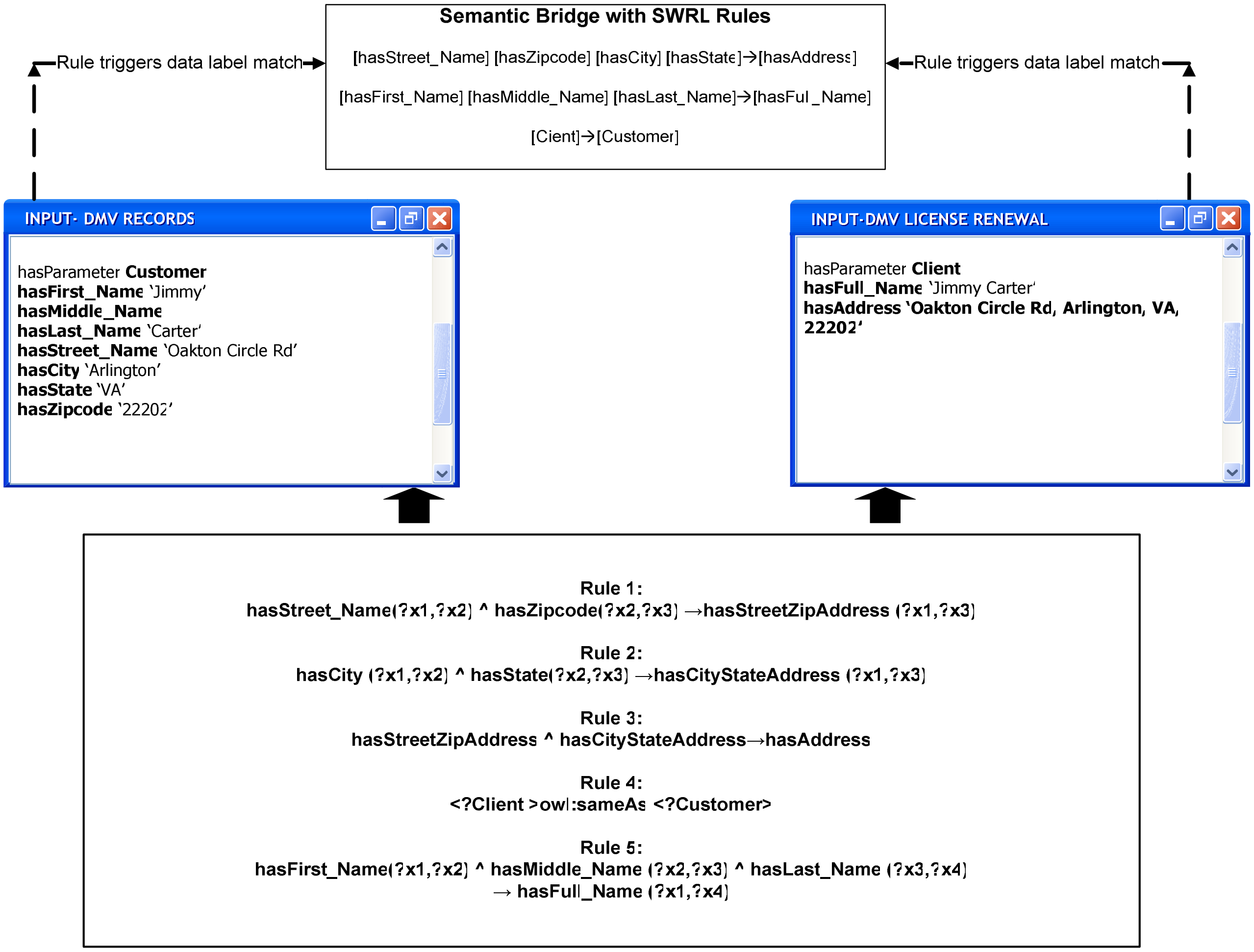
Rule 5: hasFirst_Name(?x1,?x2) ^ hasMiddle_Name (?x2,?x3) ^ hasLast_Name (?x3,?x4)
→ hasFull_Name (?x1,?x4)

This implies (Antecedent (hasFirst_Name (I-variable(x1) I-variable(x2)), hasMiddle_Name (I-variable(x2) I-variable(x3))), hasLast_Name (I-variable(x3) I-variable(x4))) and thus the consequent would be (hasFull_Name (I-variable(x1) I-variable(x4)))). These predefined rules will successfully match all the schemas and make resolve future heterogeneity issues. In the next section we describe how SWRL rules are implemented and its benefits.

5.2 Implementing SWRL rules and its benefits

Figure 8 extends the expression of the dotted lines shown figure 7. It highlights the semantic bridge that performs data label match for inputs from DMV Records and DMV License Renewal. The interfaces show the parameters that exist and Customer is shown in DMV Records on the left and Client is shown on the right for DMV License Renewal. All the highlighted items in bold are schemas that are actually being mapped. We summarize all five rules that have been triggered at the bottom of the diagram. The main benefits of SWRL rules are:

- It overcomes the limitations of standards and ICT technology such as EDI, XML, and RosettaNet which do not allow polymorphism and inheritance. For example what we have demonstrated in rule 5.
- Has greater reliability and precision when used together with SRS.
- Allows not only binary (1:1) but also multiple mappings (1: m, m: 1, m: n).
- Once the definition are agreed among public agencies like those done in rules 1 through 5, then future data exchange is automatically triggered as it would be predefined allowing data exchanges and knowledge inheritance to happen on-the-fly.
- Improves overall scalability and efficiency in the context of Web Service compositions among public agencies.


Semantic Bridge with SWRL Rules
[hasStreet_Name] [hasZipcode] [hasCity] [hasState]→[hasAddress]
[hasFirst_Name] [hasMiddle_Name] [hasLast_Name]→[hasFul _Name]
[Cient]→[Customer]
Rule triggers data label match
Rule triggers data label match
INPUT- DMV RECORDS
INPUT-DMV LICENSE RENEWAL

hasParameter **Customer**
hasFirst_Name 'Jimmy'
hasMiddle_Name
hasLast_Name 'Carter'
hasStreet_Name 'Oakton Circle Rd'
hasCity 'Arlington'
hasState 'VA'
hasZipcode '22202'

hasParameter **Client**
hasFull_Name 'Jimmy Carter'
hasAddress 'Oakton Circle Rd, Arlington, VA, 22202'

Rule 1:
hasStreet_Name(?x1,?x2) ^ hasZipcode(?x2,?x3) →hasStreetZipAddress (?x1,?x3)

Rule 2:
hasCity (?x1,?x2) ^ hasState(?x2,?x3) →hasCityStateAddress (?x1,?x3)

Rule 3:
hasStreetZipAddress ^ hasCityStateAddress→hasAddress

Rule 4:
<?Client >owl:sameAs <?Customer>

Rule 5:
hasFirst_Name(?x1,?x2) ^ hasMiddle_Name (?x2,?x3) ^ hasLast_Name (?x3,?x4)
→ hasFull_Name (?x1,?x4)

Figure 8 Implementing SWRL Rules

6 Conclusion

In summary we have presented an end-to-end framework to achieve data interoperability in e-government services via semantic mediation. Our contention is that all public agencies will need to share definitions of their respective domain specific ontologies in order to seamlessly exchange services. This is due to the fact that there cannot be a single perfect ontology for all agencies to comply with such as the federated approach or global ontology approach. This is why we propose that there should be a hierarchical knowledge repository where general concepts can be borrowed from upper ontologies and more specific ones are maintained locally. In this way we can obtain richer definitions of data. The limitation of the federated approach is that it does not produce semantically rich and does not also support inter-ontology mappings of local schemas.

Our approach provides a two part solution for this. Firstly, we introduce a detailed process methodology as to how semantic bridging should be done. We highlight the importance of establishing equality (E), inclusiveness (IC) negating disjoint (D) data and maintaining data consistency (CN). This is supported by an empirically tested SRS measure which is a hybrid measure that combines the best of both worlds i.e. linguistic properties and syntactic properties. Linguistic properties have been proven to enhance semantic matching of data in literature. Secondly, we provide semantic bridging with a two-part strategy, first with SRS and then with SWRL rules. The former is a semi-automated method that an ontologist would refer to before selecting schemas to be mapped and the latter matches schemas that are regularly used on-the-fly. Both semantic bridging strategies are complimentary and are to be used together. The methodology and framework are flexible and caters to binary and extends to complex mapping as well.

References

[1] Ahmed-Macer, M., "Solving Interoperability Problems on a Federation of Software Process Systems, *Proceedings of the 6th International Conference on Enterprise Information Systems*, Porto, Portugal, April 14-17, 2004, pp. 591-594

[2] Barbulescu M., Balan G., Boicu M., Tecuci G., "*Rapid Development of Large Knowledge Bases"* in Proceedings of the 2003 IEEE International Conference on Systems, Man & Cybernetics, Volume: 3, pp. 2169 - 2174, Washington, DC, October 5-8, 2003

[3] Castillo, J. R., Silbescu, A., Caragea, D., Pathak, J., and Honavar, V.G., "Information Extraction and Integration from Heterogeneous, Distributed, Autonomous Information Sources- A Federated -Ontology-Driven Query-Centric Approach", *IEEE International Conference on Information Reuse and Integration*, 2003, pp. 183-191

[4] Giunchiglia, F. and Zaihrayeu, I. "Making peer databases interact - a vision for an architecture supporting data coordination." *Proceedings of the Conference on Information Agents*, Madrid, September 2002

[5] Gruber, T.R., *Toward Principles of Design Ontologies Used for Knoweledge Sharing*, 1993

[6] G.Charles, G. A. M. a. W. (1991). Contextual correlates of semantic similarity *Language and Cognitive Processes, 6*(1), 1-28.

[7] H. Wache, T. Voegele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Huebner. "Ontology-based integration of information - a survey of existing approaches". *Proceedings. of IJCAI,* August 2001

[8] H. Stuckenschmidt, U. Visser and H. Wache, ISWC-Tutorial, Sundial Resort, Sanibel Island, Florida, USA,October 20, 2003

[9] Jerry, F.,, Brad, P., Marian, N., and Bruce, B., Agent-Based Semantic Interoperability in InfoSleuth, *SIGMOD Record* 28:1, March, 1999, pp. 60-67

[10] Kurgan, L, Swiercz, W and Cios, K "Semantic Mapping of XML Tags using Inductive Machine Learning", Department of Computer Science and Engineering, University of Colorado at Denver.

[11] Missikoff, M., Schiappelli, F., and Taglino, F., "A Controlled Language for Semantic Annotation and Interoperability in e-Business Applications", ", *Proceedings of the Second International Semantic Web Conference (ISWC-03),Sanibel Island, Florida,* 2003, pp. 1-6

[12] Muthaiyah, S and Kerschberg, L "Dynamic Integration and Semantic Security Policy Ontology Mapping for Semantic Web Services (SWS)", IEEE, *First International Conference on Digital Information Management (ICDIM'$_{06}$),* Bangalore, India, pp. 116-120

[13] Muthaiyah, S and Kerschberg, L "Virtual Organization Security Policies: An Ontology-based Mapping and Integration Approach", *Information Systems Frontiers (ISF*), *A Journal of Research and Innovation,* Springer, USA, Special Issue on Secure Knowledge Management*,* 2007, pp. 505-515

[14] Muthaiyah, S and Kerschberg, L "A Hybrid Ontology Mediation Approach for the Semantic Web", *Special Issue on Decision Technologies in E-Business, International Journal of E-Business Research (IJEBR)*, Idea Group Publishing Inc, U.S.A. ISSN: 1548-1131, EISSN: 1548-114X (forthcoming)

[15] Noy, N.F.a.M.A.M. *PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment*. in *Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*. 2000. Austin,Texas: The MIT Press

[16] Obrst, L., "Meditation and Data Sharing: Ontologies for Semantically Interoperable Systems", *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, November, 2003, pp. 366-369

[17] Park, J. and Ram, S., "Information Systems Interoperability: What Lies Beneath?" *ACM Transactions on Information Systems*, Vol. 22, No. 4, October 2004, pp. 595-632

[18] Peter Haase, F.H., Zhisheng Huang, Heiner Stuckernschmidt and York Sure, *A Framework for Handling Inconsistency in Changing Ontologies* Springer-Verlang Berlin Heidelberg, 2005: p. 353-367

[19] Tecuci G., Disciple: A Theory, Methodology and System for Learning Expert Knowledge, Thèse de Docteur en Science, University of Paris-South, France, 1988

[20] http://standarder.oio.dk/English/Guidelines/

[21] http://www.govtalk.gov.uk/

[22] http://www.unece.org/cefact/

[23] http://www.racer-systems.com/

Appendix 1

Symbol	Word Pairs	SYN Rank	SRS Rank	HCR Rank
a	car - automobile	0	5	8
b	gem - jewel	6	3	2
c	journey - voyage	5	7	8
d	boy - lad	7	7	8
e	coast - shore	5	6	7
f	asylum - madhouse	4	6	8
g	magician - wizard	3	7	8
h	midday - noon	4	7	10
I	furnace - stove	4	5	6
j	food - fruit	6	4	3
k	bird - cock	6	6	6
l	bird - crane	5	4	4
m	tool - implement	2	3	4
n	brother - monk	4	4	2
o	lad - brother	3	4	2
p	crane - implement	2	2	1
q	journey - car	4	4	2
r	monk - oracle	4	3	4
s	cemetery - woodland	3	2	2
t	food - rooster	5	3	4
u	coast - hill	5	4	2
v	forest - graveyard	2	1	1.2
w	shore - woodland	3	2	1
x	monk - slave	5	4	2
y	coast - forest	7	4	0
z	lad - wizard	6	3	0
aa	chord - smile	5	3	0.6
ab	glass - magician	3	2	0
ac	rooster - voyage	5	3	0
ad	noon - string	5	3	1