Enabling Adaptive Semantic Interoperability for Pervasive Computing

Declan O'Sullivan, Dave Lewis, Vincent Wade
Knowledge and Data Engineering Group
Trinity College, Dublin
Ireland
+353 1 6081765
Firstname.lastname@cs.tcd.ie

Abstract

Embedding processors, sensors and actuators in everyday products implies an explosion in the number and type of organisations that need to be involved in achieving the seamless interoperability implied by the pervasive computing vision. Many of the network interoperability problems can be addressed by Internet protocols and XML data encoding. However, the potential for debilitating heterogeneity of semantics in application level interoperability remains. In this paper we examine the potential for using ontology-based semantics for supporting run-time semantic interoperability between pervasive computing applications. Automatic semantic interoperability between ontologies representing arbitary, separately developed models is beyond the abilites of current semantic reasoning techniques. Thus, we present a software and process framework (called OISIN) that guides the design time identification of candidate mapping information for a set of ontologies and enables runtime determination of the actual mappings based on the context of the interoperability required. We demonstrate its operation with a "visiting lecturer" university case study.

Keywords

Ontologies, pervasive computing.

1. Introduction

Pervasive computing describes situations where processors, sensors, actuators and displays are integrated into the physical fabric of everyday life, e.g. personal computing devices, household appliances, furniture, buildings, transportation, all linked though a ubiquitous mixture of wired and wireless networks [weiser]. Significant progress has been made towards the vision of an environment that adapts to the user rather than the user to the environment, supported through the provision of adaptive, context-aware devices and services. However, the source of the most serious challenges to deploying the pervasive computing vision are not technological but structural. Embedding processors, sensors and actuators in everyday products implies an explosion in the number and type of organisations that need to be involved in achieving the seamless interoperability implied by the pervasive computing vision. Many of the network interoperability problems can be addressed by Internet protocols and XML data encoding. However, the potential for debilitating heterogeneity of semantics in application level interoperability remains. Consider the complexity involved in reaching agreements on and enforcing conformance to interoperability standards when the players involved expand from the likes of Microsoft, IBM and Cisco to all the potential organisations with applications embedded in their products, e.g. Kellogs, Nike, GAP, Yale, Ford, Pizza Hut, Pentel to name a few implied by pervasive computing scenarios. It is therefore clear that the pervasive computing vision implies a massive increase in scale of the application interoperability problem.

The work presented here is motivated by the observation that we cannot, therefore, rely on shared common interoperability standards alone to solve the application interoperability problems on the scale needed for pervasive computing. Instead, application software must somehow adapt at runtime to dynamically interoperate with other application software.

In this paper we examine the potential for using ontology-based semantics for supporting run-time semantic interoperability between pervasive computing applications. Automatic semantic interoperability between ontologies representing arbitary, separately developed models is currently beyond the abilites of current semantic reasoning techniques. Thus, we present a software and process framework (called OISIN) that guides the design time identification of candidate mapping information for a set of ontologies and enables runtime determination of the actual mappings based on the context of the interoperability required.

Dividing the framework across design time and run time activities we believe has a number of benefits for applications in the pervasive computing environment:

- Guides the knowledge engineering of the ontologies to maximise automated semantic interoperability;
- Minimises manual ontology mapping which is very important as it is envisaged that the framework is applied when the need for application interoperability is identified unexpectedly and at short notice;
- Enables context aware determination of mappings at runtime;
- Eliminates the need for software to be incorporated into the visiting application and minimises the impact on the visited applications.

The paper first introduces semantic and ontology interoperability, and briefly the visiting lecturer case study. The OISIN framework is then described and applied to the case study. Finally some remarks and conclusions are provided.

2. Semantic Interoperability and Ontologies

Interoperability problems can be classified as: *system*, i.e. related to hardware and operating systems; *syntactic,* related to representation languages and data formats; *structural*, related to model representation; and *semantic,* related to different meaning given to terms [sheth]. Ceri and Widom identify four categories of semantic conflicts [ceri]: naming conflicts (homonyms and synonyms); domain conflicts due to different reference systems, e.g. imperial vs. metric; structural conflict where differently structured data represents the same concept; and meta data conflicts, e.g. where a entity is a class in one system and an instance in another.

In the past, such semantic conflicts have typically been dealt with at "design time": through careful schema design in distributed database solutions; through the hand crafting of interoperability gateways (with system integrator solutions); or by forcing each system to conform to a standard mechanism for interchange (e.g. ebXML). Although these traditional approaches have been successful in well understood/static interchange environments, each of these approaches are inappropriate for systems that want to interchange in dynamic environments [cui]: the schema design solution fails due to the rapidly changing nature of the interchanges required; the handcrafting of gateways solution fails as it does not scale to large numbers of information systems; and the standards solution fails due to the lack of certainty as to whether there is a common interpretation of the standard.

Ontologies can be used to describe the semantics of information sources and make the content explicit, and thus can be used to discover semantic equivalence between information concepts. The use of ontologies as a possible solution to the semantic interoperability problem has been studied over the last six or seven years. Wache et al reviewed and categorized twenty five approaches that have been proposed over this period and concluded that reasonable results for integration can be achieved through ontology based approaches [wache]. Like the traditional approaches most of the approaches proposed are "design time" solutions towards integration. However as ontologies represent semantics, then semantic interoperability can be achieved by runtime comparison of and inference about ontological information.

Of course in a dynamic environment such as the one that we envisage, people, organizations, device and application vendors will have freedom of choice in terms of how they structure their semantics and the representation language that they use to do so. We will term this the ontology heterogeneity problem. Visser et al focuses on ontology heterogeneity and classifies mismatches that may occur under headings of Conceptualization (on how concepts are classified) and Explication (on how concepts are specified). They then go on to discuss how easy/hard it is to deal with each type [visser]. More recently Klein has undertaken a similar analysis of the mismatch problems that can occur and examines how various projects propose solutions to these problems [klein]. Several papers propose particular approaches or architectures for resolution of ontological heterogeneity, mainly at design time. For example Corcho and Gomez-Perez propose a system that will semi-automatically integrate ontologies of different types in the e-business domain [corcho].

In summary, whereas the use of ontologies has been shown to overcome the semantic interoperability problem between parties that know they want to interact a priori, the use of ontologies to achieve semantically interoperable interactions between parties at runtime in a very dynamic situation is still an area for research.

3. "Visiting Lecturer" Case Study

In order to illustrate the potential usage of the framework we present the case of a "visiting lecturer" to a university for a six month term and our aim is to enable the visiting lecturer's personal agent software (which has been configured for the home university) to interoperate seamlessly with the visited university's pervasive computing services such that the lecturer can interact with the environment as if he/she was still in his/her own university. For example, to book lecture rooms, to get access to budget controlled resources (e.g. video conferencing) etc. This requires the visited environment to adapt to the concepts/terminology of the personal agent software. The setup of the adaptation needs to be achieved with minimum administrative overhead.

In a pervasive computing world where every device and application vendor might commit to a limited set of ontologies (such as those proposed by in the SOUPA initiative [soupa]) then the adaptation could just select the appropriate mapping from a library of preexisting mappings that would could be selected and used. However due to the natural diversity of organizations and people involved in the provision (and even more so with respect to provision of operational management) of pervasive computing environments, it is highly unlikely that such a set of preexisting mappings will always exist.

Thus it is envisaged that the main administrative task in adapting to the applications of our visiting lecturer is undertaking a "sufficient" mapping of the referenced home ontologies of the lecturer to the equivalent range of ontologies within the visited domain. What is considered "sufficient" is highly dependent on the context of the expected interoperability. For example, the mapping required between the two universities to allow sharing of exam results would need to be deeper than our example of a once off visiting lecturer case.

4. OISIN framework overview

In our group we have been investigating ontology processing techniques to enable interoperation of two different applications which we assume have been developed using different ontologies. In [osullivan] we have shown how a dynamic bridge can be automatically generated given a set of preexisting ontology mappings. More recently we have extended this work by focusing on: (1) how the mapping information between the ontologies can be derived and (2) how the adaptation of the bridge can be made context aware.

Ideally a dynamically generated context aware bridge could be achieved through a runtime system that would automatically derive a mapping between selected parts of the relevant ontologies on the fly based on the context of the application interoperability been sought. However, automatically deriving ontology mapping information at runtime without the involvement of a human is generally considered impossible [klein] and so the challenge in our work has been to identify an integrated software and process framework which will minimise the amount of design time work involved and devolve as much decision making as possible to a runtime algorithm. Minimising design time work and devolving as much as possible to runtime processing is crucial for the uptake of this approach in pervasive computing environments. Equally important is maximising the applicability of human generated ontology mappings by ensuring it is sufficient to maximise chances of a successful runtime mapping between information conforming to concepts from the two ontologies concerned.

The resultant OISIN (**O**ntology **I**nteroperability for **S**emantic **IN**teroperability) framework is overviewed in Figure 1. It consists of a series of design tools which guides the development of candidate mappings between the ontologies, and a runtime Semantic Matching Utility (SMU) service which is used by the bridge to determine the actual mapping in a particular context.

Providing full details on the software and process elements of the OISIN framework would not be feasible in this paper but rather in the next section we focus on how the OISIN framework has been applied to support interoperability for our case study.

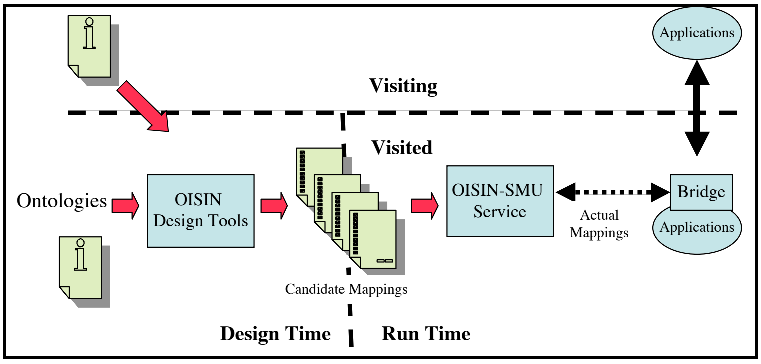

Figure 1: Overview of the OISIN Framework

5. The OISIN framework applied

In our case study the ontologies used by the two universities have been developed independently, namely one from the university of Manchester [ka.owl] and one from the university of Aberdeen [abdn.owl]. As will be seen later in the section although these ontologies on the surface are similar in some respects, there are some significant differences which need to be handled.

The first phase of the design time framework (shown in Figure 2) transforms each ontology (whether in Ontology Web Language (OWL) format, relational database format etc.) into a canonical form for ease of processing. The software tools in the second phase characterise:

- the quality[1] of the ontology, so that poorly defined ontologies can be filtered out;
- the lexical patterns used in the ontology, to determine if it is suitable for use by the runtime SMU service;
- the dimension of the ontology, which will be used in determining the extent of candidate mappings possible.

---

[1] through referring to ranking services such as SWOOGLE [swoogle.umbc.edu] and/or through interaction with peers

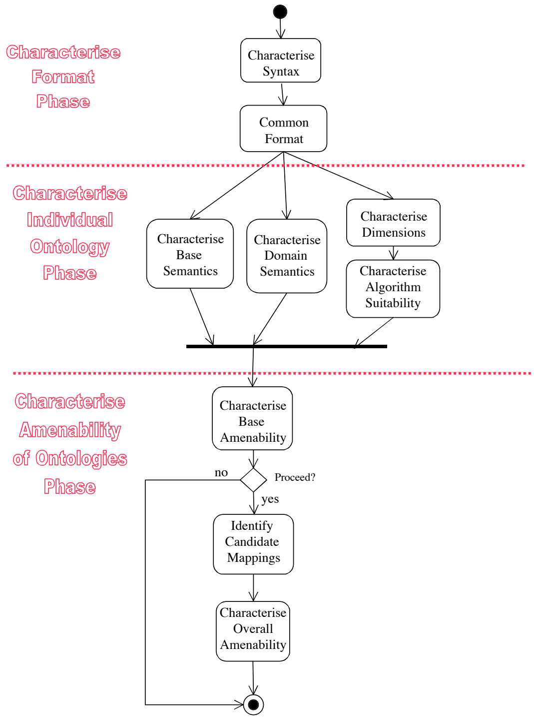

Figure 2: OISIN Design Process Overview

In the third phase class and property names of the ontologies are lexically compared (with support of WordNet and an encoded university domain specific thesauris) to identify potential matches (through exact or synonym matches) of ontology classes and their properties. Together with the information generated in phase 2, a summary of the base amenability for mapping is provided to the user in the form of an Excel spreadsheet (excerpt in Figure 3). This provides an early indication of the potential difficulty in undertaking the mapping. For example it can easily be seen if there is a fundamental modeling mismatch if there is a high number of classes in one ontology which potentially match to properties in the other ontology. Depending on the circumstances examination of the spreadsheet may determine whether to proceed or not with the candidate mapping activity.

	abdn.owl	ka.owl
Syntax Analysis		
Validates?	yes	yes
Hyphenated Term Combinations	94	0
Underscored Term Combinations	0	3
MiddleCapital Term Combinations	0	89
Base Semantics Analysis		
WORDNET		
Terms known directly	28	77
ombinations whose constituent parts are known	94	92
Unknown Terms	0	19
Domain Semantics Analysis		
ONTOLOGY		
Ontologies Imported	0	1
Ontologies Referenced	6	3
SWOOGLE		
Rank	2.1	1.5
Ontologies which import	1	1
Ontologies which reference	2	1
Ontologies which extend	3	2
Ontologies which map	0	0
Dimensions Analysis		
Concepts	56	96
Unique Attributes	20	32
Unique Relationships	46	60
Overall Tree Shape		
Level	Width	Width
0	7	6
1	14	28
2	15	47
3	14	13
4	9	1
5	9	0
6	6	0
7	0	0
8	0	0
9	0	0
10	0	0
Candidate Full Lexical Matches Analysis		
Class to Class	13	13
DatatypeProperty to DatatypeProperty	0	0
ObjectProperty to ObjectProperty	0	0
Property to Class	0	11
Class to Property	8	0
DatatypeProperty to ObjectProperty	0	0
ObjectProperty to DatatypeProperty	0	0
Candidate Partial Lexical Matches Analysis		
Class to Class	9	13
DatatypeProperty to DatatypeProperty	15	0
ObjectProperty to ObjectProperty	14	0
Property to Class	13	7
Class to Property	4	0
DatatypeProperty to ObjectProperty	1	0
ObjectProperty to DatatypeProperty	7	0

Figure 3: Excerpt from Mapping Amenability Spreadsheet

The potential match analysis is presented to the user in a graphical manner at the class level and a textual analysis at the property level once a class is selected. In Figure 4 for example the **M** identifies exact lexical matches (e.g Researcher) and the **P** identify partial matches on a lexical or synonym basis (e.g. Working-Person partially matches to Person).

The user then identifies the “anchors” which correspond to key partial mappings between the ontologies. This involves examining the two ontologies to try to identify equivalent concepts. Once an anchor is chosen it is annotated with an **E** (e.g. Lecturer in the figure). During this examination typically the properties of the concepts are examined to identify equivalence as well. In our example the family-name property

property of the Lecturer-In-Academia class can be seen to be equivalent to the lastName property concept of the Lecturer class. In addition, transformation code can be associated with a mapping in order to provide the ability to translate from one value range to another.

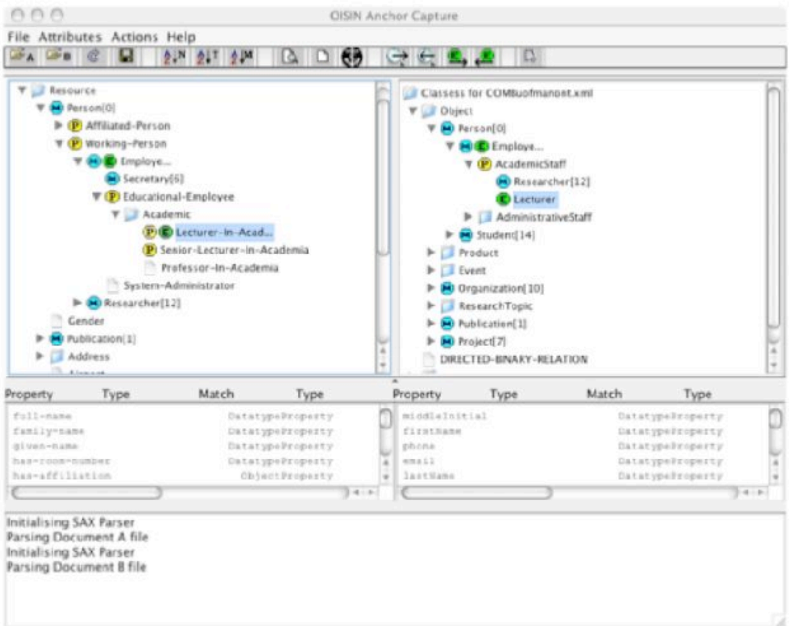

Figure 4: Example of OISIN anchor capture tool

During this process "anchor paths" are also identified. The concept of "anchor paths" was first introduced by [noy01]. The idea is that if two anchors are specified in isa hierarchy of Ontology A the chances are that the classes which appear in the intervening path have a good chance of correspondence with those on the path of the corresponding anchors in Ontology B. The use of anchor paths obviates the need for enumeration of every possible candidate mapping by the tool user, whilst providing the runtime SMU service with well bounded search spaces. The user is supported by the tool in identifying invalid anchor paths. For our example in our case study if the Researcher concepts were also identified by the user as an anchor pair then the tool would indicate that Employee-Researcher would be an invalid path in the right hand ontology shown in Figure 4, as there is no corresponding path in the left hand ontology[2].

[2] as there is a mismatch between the two ontologies as to how Researcher is modeled

The output of the design phase is a set of candidate anchor mappings (expressed using the owl:equivalentClass and owl:equivalentProperty XML elements), the lexical matching information for class and properties, the candidate anchor path mapping information (Figure 5 illustrates a snippet) and the equivalence transformation information. This set of information is then used by the SMU at runtime to transform concepts used in the user profile of visiting lecturer's university into terms used in the visited university.

```
<valid_path>
        <anchor_path>Employee:Lecturer-In-Academia
        </anchor_path>
        <matches>Employee::Lecturer
        </matches>
        <length>4</length>
        <nd>
          <name>Employee</name>
          <match>
            <kind>complete</kind>
            <types>
              <type>Class-Class</type>
            </types>
          </match>
        </nd>
        <nd>
          <name>Educational-Employee</name>
          <match>
            <kind>partial</kind>
            <types>
              <type>Class-Class</type>
            </types>
          </match>
        </nd>
        <nd>
          <name>Academic</name>
          <match>
            <kind>none</kind>
          </match>
        </nd>
        <nd>
          <name>Lecturer-In-Academia</name>
          <match>
            <kind>anchor</kind>
          </match>
        </nd>
      </valid_path>
```

Figure 5: Candidate Anchor Path Mapping Information snippet

A key differentiation of our approach from others is that the anchor capture tool identifies candidate mappings and the determination of what is an actual mapping is left to the runtime SMU service given the context of the interoperability being attempted. In contrast other tools such as PROMPT/Anchor [noy00] require actual mappings to be chosen at design time. For example the user may have annotated that the concept "Lecturer" as a candidate mapping to "Lecturer-In-Academia". From the point of view of a room booking application of the visited university this may be sufficient, but for an international conference organization application which requires a property called “organiserOrChairOf” (which exists in the visited ontology only) then the candidate mapping would be considered insufficient.

6. Conclusion and Further Work

Previous approaches to semantic interoperability and ontology interoperability have focused on determining what are actual mappings between schemas purely at design time, as it is considered that determination of such mappings at runtime without the involvement of a human is considered beyond the current semantic reasoning capabilities. This paper has described a hybrid approach where candidate mappings are outlined at design time but actual mappings are determined at runtime in the context of the application interoperability that is required. The process and software framework outlined (OISIN) has been designed with the view to minimise the human effort involved at design stage and minimising the impact on the applications involved.

Initial experimentation has shown that this approach holds promise in situations where semantic interoperability is required between domains unexpectedly and at short notice, such as those situations envisaged in pervasive computing environments. Further experimentation and evaluation is being undertaken in the coming period.

Acknowledgements

This work was partially funded by the Irish higher education authority under the M-Zones programme.

References

[abdn.owl] http://www.csd.abdn.ac.uk/~cmckenzi/playpen/rdf/abdn_ontology_LITE.owl

[ceri] Ceri and Widom (1993), Ceri S., Widom J., "Managing Semantic Heterogeneity with Production Rules and Persistent Queues", Proceedings of the 19th VLDB Conference, Dublin, Ireland, pp. 108-119.

[corcho] Corcho and Gomez-Perez (2001), Corcho, O. and Gomez-Perez, A., "Solving Integration Problems of E-commerce Standards and Initiatives through Ontological Mappings", In: Proceedings of the Workshop on E-Business and Intelligent Web at the 17th International Joint Conference on Artificial Intelligence (IJCAI2001) , Seattle, USA, August 5, 2001.

[cui] Cui Z., Jones D. and O'Brien P., "Semantic B2B integration: issues in ontology-based approaches", ACM SIGMOD Record, vol 31 no 1, pages 43—48, 2002, ACM Press

[ka.owl] http://www.cs.man.ac.uk/~horrocks/OWL/Ontologies/ka.owl
[klein] Michel Klein, "Combining and relating ontologies: an analysis of problems and solutions". In Workshop on Ontologies and Information Sharing, IJCAI'01, Seattle, USA, August 4-5, 2001
[lewis04a] Lewis, D., Conlan, O., O'Sullivan, D., Wade, V. "Managing Adaptive Pervasive Computing using Knowledge-based Service Integration and Rule-based Behavior", in Proc. of 2004 IEEE/IFIP Network Operations and Management Systems, pp 901-902
[noy00] Noy, N.F., Musen, M.A., "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment". Seventeenth National Conference on Artificial Intelligence (AAAI-2000), Austin, TX, . 2000.
[noy01] Noy, N.F., Musen, M.A., "Anchor-PROMPT: Using Non-Local Context for Semantic Matching". Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, WA, . 2001.
[osullivan] O'Sullivan, D., Lewis, D., "Semantically Driven Service Interoperability for Pervasive Computing", Proceedings of the 3rd ACM International Workshop on Data Engineering for Wireless and Mobile Access, San Diego, CA, USA, 19th September 2003, pp 17-24
[owl] W3C (2003) Ontology Web Language, http://www.w3.org/2001/sw/, Visited Apr 2003
[sheth] Sheth A.P. , "Changing Focus on lnteroperability in Information Systems: From System. Syntax, Structure to Semantics", M. F. Goodchild, M. J. Egenhofer, R. Fegeas, and C. A. Kottman (eds.) Interoperating Geographic Information Systems, Kluwer
[soupa] http://pervasive.semanticweb.org/soupa-2004-06.html
[visser] Visser, P.R.S., Jones, D.M., Bench-Capon, T.J.M., and Shave, M.J.R., "An analysis of ontology mismatches; heterogeneity versus interoperability." AAAI 1997 Spring Symposium on Ontology Engineering, 1997.
[wache] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner, "Ontology-based Integration of Information - A Survey of Existing Approaches," In: Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA, 2001, Vol. pp. 108-117.
[weiser] Weiser, M. (1991), "The Computer of the 21st Century", Scientific American, vol. 265, no.3, September 1991, pp 66-75.