Association for Information Systems
AIS Electronic Library (AISeL)

PACIS 2003 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

December 2003

Semantic Web Services Delivery: Lessons From the Information Systems World

Gerard McGrath
Victoria University

Recommended Citation

McGrath, Gerard, "Semantic Web Services Delivery: Lessons From the Information Systems World" (2003). *PACIS 2003 Proceedings*. 16.
http://aisel.aisnet.org/pacis2003/16

Semantic Web Services Delivery: Lessons From the Information Systems World

G. Michael McGrath

School of Information Systems Victoria University
PO Box 14428, Melbourne City, Australia, MC8001
michael.mcgrath@vu.edu.au

Abstract

The 'Semantic Web' promises a new generation of web services delivery applications enabled by semantic markup of web resources, shared ontologies and highly-intelligent agents. The vision is astounding and the potential benefits enormous. However, previous experience with information systems and database integration initiatives would suggest that a degree of caution is warranted. It is also argued that lessons might be gleaned from these intra-organizational integration projects that might be of considerable value in developing and implementing Semantic Web applications.

Keywords

Semantic Web, strategic information systems planning (SISP), information integration, change management.

1. Introduction

The Semantic Web has attracted a good deal of interest recently. Generally, it is described in terms such as: i) the Semantic Web is the existing web marked up with semantic information that adds meaning to the content of web pages; and ii) this will assist in the development of highly-intelligent agents, capable of automatically providing advanced web services - often in response to quite imprecise user requests (Berners-Lee, Hendler and Lassila, 2001). Fraunfelder (2001; p.52) definitely takes a big-picture perspective in declaring that the Semantic Web:

> "*--- will radically change the nature of the Web – from a place where information is merely displayed to one where it is interpreted, exchanged and processed*" and "*--- the ultimate goal of the Semantic Web is to give users near omniscience over the vast resources of the Internet, turning the millions of existing database islands into a single gigantic database Pangea*".

This latter assertion is awesome in its ambition. In a sense though, it is not new: i.e. many organizations have been down a similar path previously in attempting to address inconsistent and redundant information island problems *within* their own companies (Martin, 1982; pp.1-20). These organizations were often very large, with massive investments in their information systems and were suffering the consequences of a long-term, poorly-planned, piecemeal approach to their systems development activities. The Bell companies were fairly typical, with their plight summarised by Nolan, Puryear and Elron (1989; p.301) as:

> *"The Bell System's early start in automation, coupled with its massive investment in computer systems, would appear to be a major advantage for the Regional Holding Companies in their new strategies. But this is not the case: on the contrary, their information technology is more of a liability than an asset with respect to their desired transformations."*

In many instances, these organizations attempted to rectify their problems by embarking on top-down, strategic information systems planning (SISP) projects, with the ultimate aim of establishing a new generation of systems, built around a single, common set of enterprise-wide databases[1]. This approach was appealing and appeared to be logically sound and, as such, it was adopted by a great many organizations from the late-1970s through to the early-1990s (at least). The only problem was that it didn't work or, perhaps more precisely, it was not allowed to work! Lederer and Sethi (1992) are among many to have identified reasons for these failures. Given that Semantic Web applications can be viewed as a new generation of 1980s-style SISP projects, we argue here that the lessons of the past should not be forgotten. Furthermore, while SISP projects failed for a great many reasons, we focus particularly on the need for the use of specific design principles (namely abstraction, interfaces and non-determinism) in Semantic Web projects: our contention being that this will lead to robust applications, able to evolve in response to environmental change and other demands.

In the following section, we briefly introduce the Semantic Web and then, in Section 3, we revisit the data-centred SISP projects of a decade ago (and more). In Section 4, we compare and contrast these projects with Semantic Web applications and argue for our particular web services delivery design and development approach. We then detail some benefits of this approach and, finally, in Section 6 we present concluding comments.

2. The *Semantic Web*

Broadly speaking, the S*emantic Web* (Fensel et al., 2000) refers to a range of standards, languages, development frameworks and tool development initiatives aimed at annotating web pages with well-defined metadata, so that intelligent agents can reason more effectively about services offered at particular sites. Both agent development and website annotation are based on ontologies: an ontology being a consensual, shared and formal description of key concepts in a given domain. The ultimate (very ambitious) aim is to be able to handle user requests such as: "*Arrange a one-week holiday for me, somewhere in Victoria (Australia), during March*". An intelligent agent needs to determine an appropriate time slot, make airline and car hire bookings (if the user is based interstate or overseas), find a suitable hotel in an appropriate location and, possibly, book tours, sporting events, restaurants etc. The user's personal website needs to be accessed to obtain details of preferences: ranging from the general (such as her preferred airline is Qantas, her favourite Melbourne restaurant is Grossi Florentino, and she likes smaller, manual cars) through to detail specific to this trip (such as she would like to be based mainly near a surf beach, take in the Great Ocean Road and visit one of her favourite Melbourne restaurants). In addition to the user's personal website, the

---

[1] While there are other varieties of SISP (Hamilton, 1994), when employing the term we refer to the data-centred approach described briefly in the text and in detail by Martin (1982).

application will have to access web locations such as tourism board promotional sites (e.g. http://www.visitvictoria.com/), the websites of individual tourism operators plus, of course, the likes of airline and rental car company sites.

A number of Semantic Web markup languages have been developed in recent years. These have evolved in a layered fashion and, in general, later proposals have been designed to address perceived deficiencies in previous initiatives.

At the bottom of the pile, we have HTML and XML. Despite the almost complete lack of any explicit semantics in HTML markup, the reality today is that higher-level markup languages must be rendered in HTML or interface with HTML resources. It is highly likely this will remain the case for some considerable time. Arguably, XML is the most important mainstream web initiative of recent years. In itself though, XML lacks complex semantic specification capabilities. However, given that most higher-level Semantic Web language proposals are specified in XML terms, its role here should not be underestimated: indeed, in (Fensel et al., 2000; p.67), the Semantic Web is defined as *"an XML application"*.

At the next level, we have the Resource Definition Framework (RDF) (Candan, Liu and Suvarna, 2001). The RDF was developed by the World Wide Web Consortium (W3C) and it is specified in XML as a metadata standard, with the goal of adding a formal semantics to the web. The accompanying RDF Schema (RDFS) may be thought of as a minimal ontology modelling language. RDF is an important Semantic Web building block but it lacks expressive power. For example, according to McIlraith, Son and Zeng, 2001; p.84), "--- *one cannot define properties of properties, necessary and sufficient conditions for class membership, equivalence and disjointness of classes, and the only constraints permissible are domain/range constraints on properties"*.

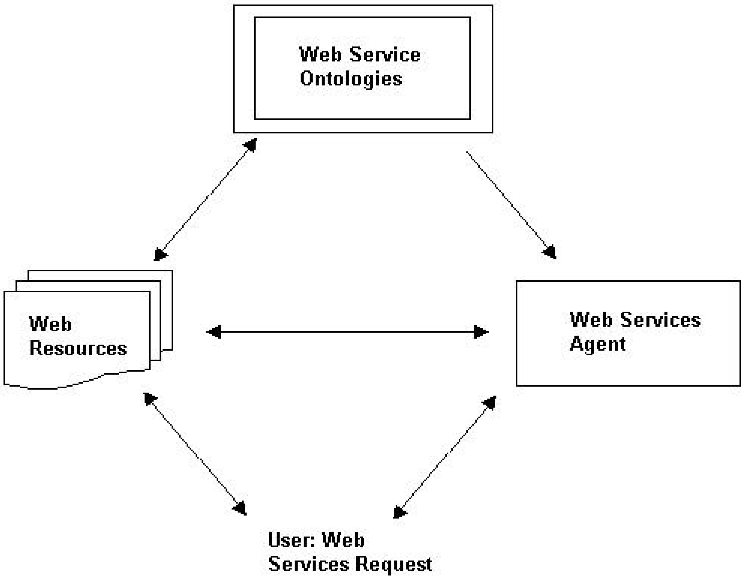


Figure 1. A framework for Semantic Web services delivery.

DAML (DARPA Agent Markup Language) and OIL (Ontology Inference Layer) sit immediately above RDF(S) in the hierarchy (Fensel et. al, 2001). With its roots in AI description logics, DAML+OIL surmounts many RDF(S) expression problems and is unambiguously computer-interpretable, thus making it amenable to intelligent agent interoperability and automated reasoning techniques. Thus, a combination of DAML+OIL markup and a logic-based agent technology seems to offer considerable promise as a framework for realizing the more ambitious goals of the Semantic Web. One development of this type that appears to have considerable promise as a tool for web services delivery has been detailed by McIlraith et al. (2001). Their architecture is illustrated in Figure 1.

From one perspective, the Semantic Web may be viewed as a (very ambitious) integration initiative; encompassing information, web services, backend enterprise systems and platforms. According to Candan et al. (2001; p.7): *"The long-term goal of RDF is to link different applications into a new global network, the Semantic Web"*, while McIlrath et al. (2001) envisage a situation where common, domain-specific ontologies are built as subclasses of more generic ontologies. Thus domain ontologies might be developed on an industry-by-industry basis and, within industry sectors at least, firms can communicate with each other because they use a common language.

This is a breathtaking vision. However, it is not entirely new: as noted, many companies have been down a similar path before and their experiences would suggest that a degree of caution is warranted. We turn our attention to this issue in the following section.

3. Lessons From the Information Systems World

Effective information systems and database integration has long been an issue of major concern for organizations. Indeed, as far back as the mid-1960s, there is evidence that forward-thinking organizations were embarking on major systems integration initiatives (Fenna, 1966). These projects were driven by concerns little different from those that occupy the minds of Chief Information Officers today: specifically, problems and costs associated with data redundancy and inconsistency, inaccurate and incomplete data, a plethora of 1:1 system interfaces, and massively excessive development and processing costs (resulting from multiple developments of the same functionality).

Over the years, methods and technologies proposed as a means of addressing the integration problem have included databases (Martin, 1975), information engineering (Finkelstein and Martin, 1981), strategic information systems planning (SISP) (QED, 1989), federated heterogeneous (meta)databases (Hsu, 1996), enterprise resource planning (ERP) systems (Markus, and Tanis, 2000) and web-based middleware based upon XML standards (Stal, 2002). Thus, from an information and process modelling perspective, the Semantic web may be viewed as among the latest in a long line of data and systems integration initiatives – none of which could be judged as an unequivocal success.

With its focus on web resources and applications, some may be tempted to see the Semantic Web as a completely new phenomenon, owing little to the integration approaches detailed in the previous paragraph. Also, with many Semantic Web projects being driven from within the AI community, this tendency is probably exacerbated. However, to those with experience in mainstream information systems integration projects, the Semantic Web is likely to invoke irresistible memories of top-down SISP projects, underpinned by shared databases and corporate data models (CDMs). In particular, we see distinct parallels between CDMs and Semantic Web ontologies.

To investigate this further, we need to go back over 25 years to the very influential work of James Martin (1975). Martin, in specifying the essential elements of a database, stated that all data must be consistent and conform to a single common definition of key entities, plus the relationships between them (i.e. an ontology). In a sense, Martin and other proponents of true-shared databases were seeking nirvana. Classically, a true database environment is established by: i) using formal conceptual modelling techniques (e.g. information engineering) to develop a CDM; ii) constructing a set of *Subject Databases (SDBs* – groupings of data sets, with close affinities, derived from and consistent with the CDM); and iii) building applications around the common set of SDBs. Theoretically, this approach makes a lot of sense and from the late 1970s through to the early 1990s (at least) it was the favoured approach of a great many companies trying to truly integrate their information systems.

As noted though, the problem with this approach is that it doesn't seem to work. Some local (Australian) examples[2] of projects attempting to integrate information systems through a common CDM, but which did not succeed, include the following:

- During the mid and late 1980s, a major bank embarked on a major initiative, which planned to substantially and radically replace its major product management systems with a new and integrated system based on data-centred concepts. After a billion-dollar investment, the project faltered in its implementation and was shelved.
- A major telecommunications company has struggled over many years with integration of its customer data, due in large part to the use of packaged software for its major operational systems (which did not integrate). These problems largely arose from a 'protection of turf' mentality (particularly related to billing) and, in part, to business mergers and reorganizations.
- In the early 1990s, an organization in the construction business was presented with the value and benefits of a Road Asset Information System which took a data-centred approach to integrating information about the history of road pavement. The project was estimated to have a Benefit/Cost Ratio of about 8. The project was not implemented and packaged system alternatives were evaluated. They were still being evaluated in 2002.

In addition, in a 1990 survey of 22 large Australian companies that had attempted to integrate their systems around a CDM, Doll Martin Associates (DMA, 1990) found that not one of these organizations had succeeded (although 16 of the 22 said they were committed to try again because they believed that integrated systems were essential if they were to realize their wider business objectives). Nor does the success rate outside Australia appear to be much better. For example, Lederer and Sethi (1992) surveyed 80 US companies, all of which had recently completed SISP projects and attempted to establish information architectures around shared databases. One very depressing result from this particular study was that, while an average of 2.1 years had elapsed since study completion, only 24% of projects identified in plans had actually been started (no project completion details were reported).

Reasons identified for the failure of these information system integration ventures (Lederer and Sethi, 1992) include technical obstacles, overly-optimistic cost and schedule estimates, a lack of senior management support, poor communication and change management, inappropriate IT Department structures and, possibly most important of all, a failure to adequately address people-related issues (especially power-political considerations). We should also note that we have restricted our discussion here to the integration of systems *within* organizations. With the Semantic Web, we are dealing with systems integration *between* organizations. One might, therefore, reasonably expect that the problems listed

---

[2] For these, we are indebted to Doll Martin Associates, a Sydney based management consultancy firm.

above might be magnified when developing Semantic Web ontologies and the web services applications that utilize them. Fortunately, we believe that the prognosis is not quite that grim. We elaborate on this in the following section.

4. A Way Forward

Effective change management has long been recognised as being absolutely critical to many aspects of organizational life, and nowhere more so than in the implementation and diffusion of high-technology innovations and policies. These types of initiatives represent fairly large-scale organization change, often regarded in terms of business transformation, especially where such change is enabled by IT. While there are problems with any major organizational change, there is a particularly high failure rate for change involving technology (Hammer and Stanton, 1995).

Moreover, while there is a growing body of literature in this field, it is deficient in a number of areas, including: a failure to differentiate between initiating change and successfully implementing such change (Watkins, Ellinger and Valentine, 1999); a real understanding of how change is well introduced and implemented (Buchanan and Badham, 1999); and neither has there been adequate addressing of the variable dimensions of depth, size, pervasiveness of change and the diverse perspectives from which such change is viewed and managed (Bolman and Deal, 1997). What is reflected in the literature is the need to take a holistic view, to address innovation (Chakravarthy, 1997) and to track actual longitudinal case studies.

Given such confusion in the research and literature, the well-tried Leonard-Barton (1988) change model is an appropriate change management tool; not only because of its clarity, but also because of its focus on change management in high-technology organizations. While her framework is by no means complete, over recent years we have found it to be an excellent starting point for assessing the probability of success of proposed organizational innovations. In Table 1, we employ the framework to contrast and compare a hypothetical Semantic Web (services delivery) application with the type of SISP-driven enterprise-wide database implementation discussed in the previous section.

	SISP Project	**Semantic Web Project**
Transferability:		
Preparedness	Low	Low
Communicability	Low	Medium
Complexity:		
Org. span	High	Very High
Org. scope	High	Very High
Divisibility:		
Modularization	Low	Low-High
Individualization	Low	Low-High
Chances for success	Low	Low-?

Table 1. SISP and Semantic Web projects – implementation characteristics.

The basis of Leonard-Barton's model is that: i) any technologically-based change initiative has *implementation characteristics;* and ii) that these characteristics set conditions within which change managers must operate and, in turn, dictate implementation tactics. The principal characteristics are:

- *Transferability,* which includes:
    - *Preparedness,* the extent to which the technology has been proven; and
    - *Communicability,* the extent to which the technology's features can be communicated to end-users and decision makers.
- *Implementation complexity,* which includes:
    - *Organizational span,* the number of people affected by the innovation; and
    - *Organizational scope,* the number of organization units affected.
- *Divisibility,* which includes the degree to which the innovation implementation can be partitioned according to:
    - *Modularization* (i.e. partition the innovation itself); and
    - *Individualization,* meaning implement the total innovation in one part of the organization at a time.

Transferability is low in SISP work because the technical challenges are formidable and because both executives and users fail to see much benefit in waiting around for years while a corporate-wide information architecture is established. With the Semantic Web, the technical challenges are no less daunting but communicability is rated medium because, as evidenced by recent history, executives (in particular) have demonstrated that they appreciate and are genuinely excited by the promised benefits of eBusiness (even allowing for some disillusionment of late).

The implementation complexity of both types of project is high: essentially because each is concerned with the establishment and acceptance of uniform standards (for data models, ontologies and the processes that interact with them). In each case, maximum project benefits will only be realized if *all* parties (organization units and individuals) agree to abide by whatever standards are designated. In addition, the implementation complexity of Semantic Web projects is rated as very high because, as noted, here we have the added burden of ensuring standards acceptance across organizations.

The final dimension, divisibility, is particularly interesting. There is evidence that some (and, quite possibly, many) SISP projects failed because they were unable to devise a means of migrating to their target set of databases (and associated applications) on a gradual basis (Hamilton, 1994). A major impediment was a key underlying premise of the data-centred approach: namely, that data is more stable than the processes that use it and, provided an organization takes the time to get its CDM right, derived databases will prove to be stable and resilient to environmental change (Finkelstein and Martin, 1981). That highly-optimistic view proved to be unfounded and many SISP projects floundered as implementation teams struggled to find effective means of minimizing disruption as data models evolved. Interfacing new applications with legacy systems also proved to be a major problem; something that SISP projects have in common with ERP implementations (Markus and Tanis, 2000).

Thus, we have rated the divisibility of SISP projects as low. Also, given the strong parallels between the CDMs used in SISP work and Semantic Web ontologies, it would therefore seem reasonable that we assign Semantic Web divisibility the same value. However, we believe the adoption of two important design principles can be used to improve the divisibility of Semantic Web projects and, thereby, greatly improve the prospects for successful implementations of these applications. The first of these principles – interface architectures built upon web-based middleware (Stal, 2002) – is a relatively recent development. The second – abstraction in data (and ontology) modelling – has been around for a long time (Feldman and Miller, 1986) but is rarely used to its full potential (especially in non-OO applications).

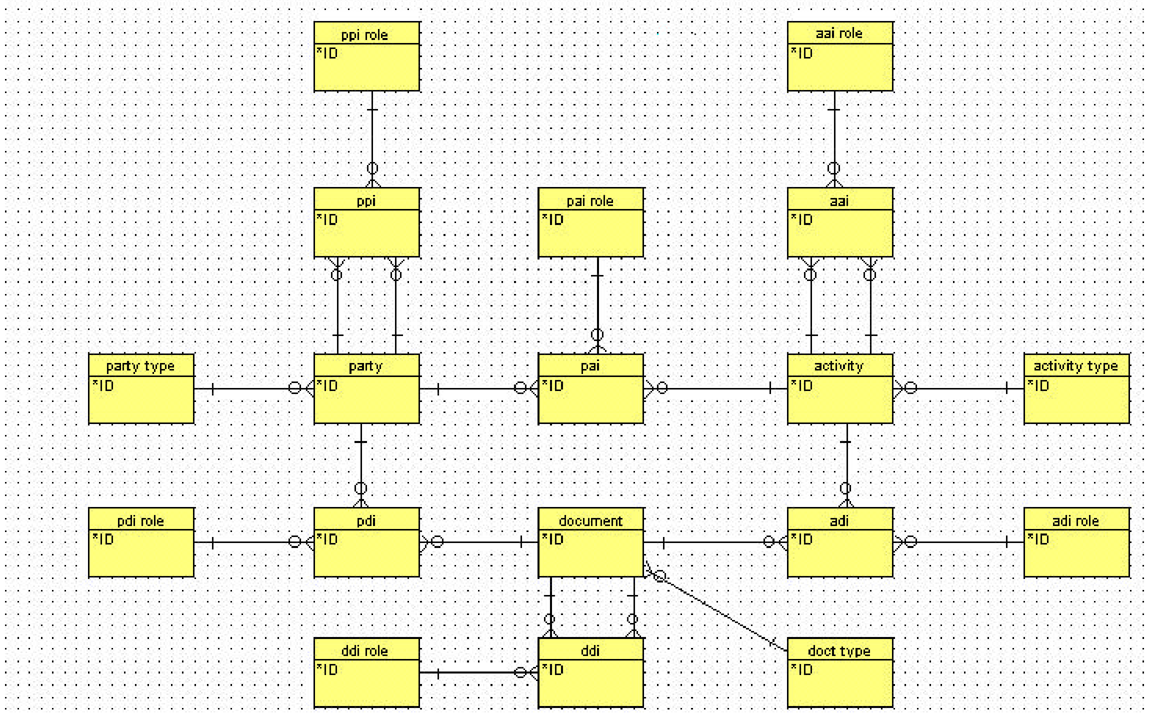


Figure 2. Abstracted ontology – ER form.

These two design principles, together with non-determinism (Kowalski, 1979), are central to the DAML/logic-based technology of McIlraith et al. (2001). We believe their technology represents one very promising starting point for an effective Semantic Web services delivery architecture, based as it is on design principles consistent with past experience. In the following section, we discuss the benefits of this particular approach. Our discussion is also informed by information modelling lessons learned during studies conducted over the previous 10 years (McGrath, 1994; McGrath, 1997; McGrath, Dampney and More, 1998). We shall illustrate our exposition through reference to a *travel management* system, designed to simplify travel planning, approval, travel services delivery and claims processing for personnel within organizations. Our abstracted ontology is presented in Figure 2. It is represented in entity-relationship (ER) form because of the simple, natural mapping from ER to logic. Mapping from our model to the OO form commonly used with Semantic Web ontology development tools (such as *OntoEdit*[3], *OntoMat*[4] and *Protege*[5]*)* is a relatively trivial

---

[3] Available from Ontoprise at www.ontoprise.de.

exercise and, indeed, the essential details were specified by Chen (1976) in his seminal ER paper.

5. Learning From Experience: The Technical Dimension

As noted earlier, we are staunch supporters of the view that a holistic approach should be taken of any IT change management exercise. In this section though, we focus mainly on the technical dimension and, in particular, the benefits of employing abstraction, interfaces and (limited) non-determinism in web services delivery applications.

5.1 Schema Integration

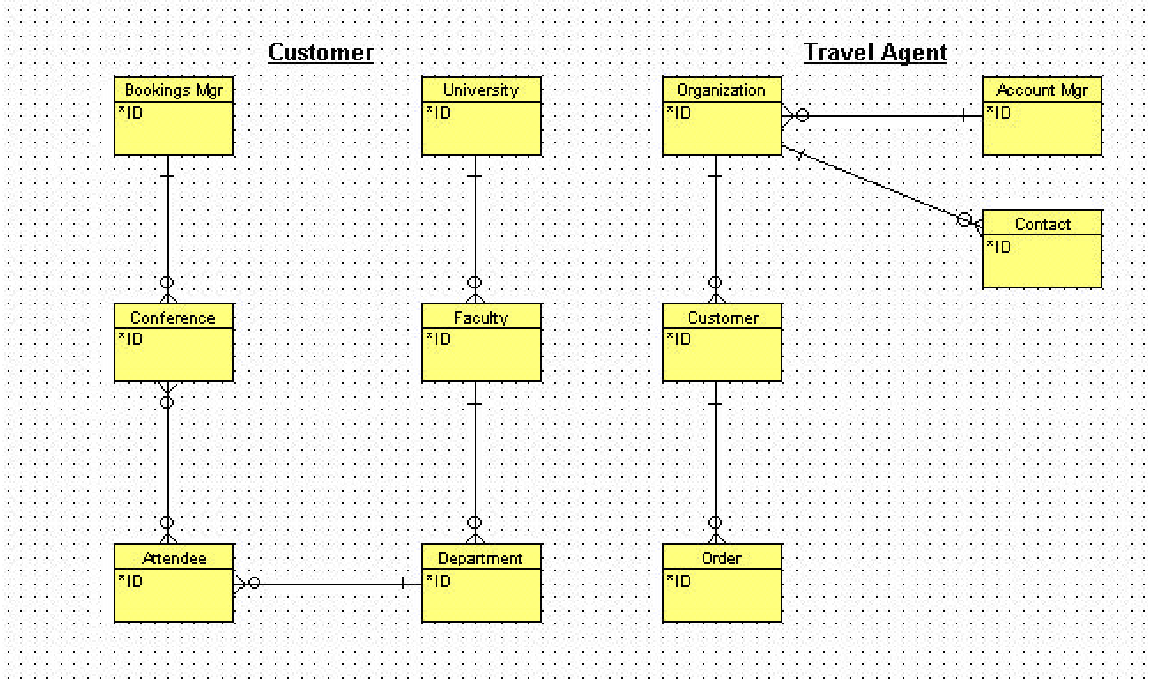


Figure 3. Customer and travel agent views of travel management application.

Firstly, we can integrate (apparently) widely-different views of the same concept by mapping to and from our common abstracted schema. Consider, for example, the two views of our travel management application, illustrated in Figure 3. From the customer's viewpoint:

1) *Bookings Mgr*, *Attendee*, *University*, *Faculty* and *Department* all become parties, with the first of these typed as a *position*, the second as a *person* and the final three as *org_units;*
2) *Conference* becomes an activity; and
3) The intersecting entity, *Conference_Attendee*, becomes a *pai (party-activity involvement)*, with an involvement role of *applicant.*

---

[4] Available from AIFB at http://annotation.semanticweb.org/.

[5] Available at http://protege.stanford.edu/index.html.

From the travel agent's viewpoint:

1) *Organization, Account Mgr, Customer* and *Contact* all become parties; and
2) *Order* becomes an activity.

We can easily facilitate implementation of these links using (for example) the mapping feature of OntoEdit. Alternatively, if we prefer to deal directly with user views, a web authoring and annotation tool such as OntoMat allows us to establish links between web resources and the common schema through simple drag-and-drop operations. Whatever mapping approach is employed, all reasoning about services and processes may now be expressed in terms of the abstracted information model. Thus, both customer and travel agent systems can evolve independently without (in most cases) any impact on our intelligent agent software. Changes are restricted to interfaces to the agent.

5.2 Business Environment Volatility

Secondly, our abstracted schema is less vulnerable to business environment changes than more concrete versions.

Again, using our example, consider a situation where the customer (a University) wishes to change its current structure *(University – Faculty – Department)* to a *University – Division – College – Department* setup. In our implementation, each instance of an organizational component is represented as an instance of the party relation – for example:

party(p102, 'Business and Law', faculty, ----)

where *p102* is the *PartyId, faculty* is the *PartyType* and *Business and Law* the *PartyName.* Relationships between parties are represented as instances of *ppi (part-party involvement)* relationships. An example is:

ppi(ppi1001, p101, p102, part_of)

where the *PPIRole (party-party involvement role)* indicates that *p101* is *part_of p102. p101* and *p102* identify parties typed as some kind of organizational unit and the first field is the relationship identifier.

With our structural change, essential maintenance is restricted to replacing tuples in tables (e.g. *p102* may now be typed as a college in the *party* relation). Provided all references to base data are expressed in terms of the abstracted schema, most agent code will not have to be touched.

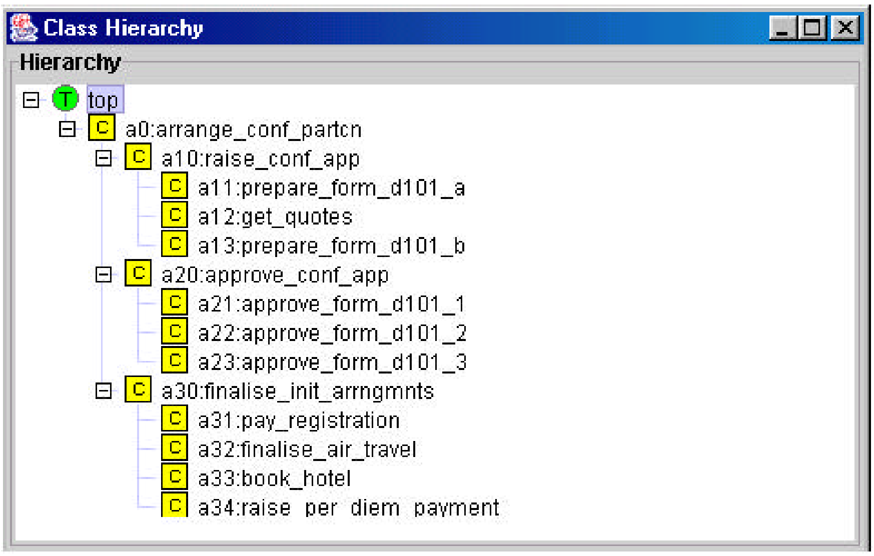
5.3 Development Costs

Thirdly, development costs may be reduced. Specifically, code common to abstracted entities and relationships (together with their subtypes and roles) is implemented once-only and in one place. It is true that, with a sound design, these benefits may be realized to a large extent with a more-concrete schema. However, our abstracted approach virtually forces the elimination of redundancy.

5.4 Agent Complexity

Finally, our logic-based approach and, in particular, Prolog's deductive inference mechanism provide a natural means of reducing agent complexity.

One of our aims is to develop an agent technology that automatically delivers web services by exploiting DAML markup. However, there is enormous variability in the way these services can be realized and the AI community is far from united on the merits of different approaches to reasoning about actions. The *Plan Domain Description Language* (Ghallab et al., 1998) aims to address this lack of consensus through expressing actions in terms of their *inputs, outputs* and *constraints.* We employ a similar approach. In addition, action invocation is guided by an activity tree where there is a *state* corresponding to each node in the tree. The tree for our travel management example is presented in Figure 4. Constraints (many of which are inherent in the logic-based implementation of our domain schemas) restrict the solution space (as represented by the tree), and these, coupled with Prolog backtracking, serve to deliver efficient non-deterministic solutions to web services requests.



- top
  - a0:arrange_conf_partcn
    - a10:raise_conf_app
      - a11:prepare_form_d101_a
      - a12:get_quotes
      - a13:prepare_form_d101_b
    - a20:approve_conf_app
      - a21:approve_form_d101_1
      - a22:approve_form_d101_2
      - a23:approve_form_d101_3
    - a30:finalise_init_arrngmnts
      - a31:pay_registration
      - a32:finalise_air_travel
      - a33:book_hotel
      - a34:raise per diem payment

Figure 4. Activity tree for travel management application.

An example of a constraint is:

> *pai(pai12, P, a12, travel_agent) :-*
>   *ppi(ppi120, P, p103, auth_travel_agent).*

which means that when obtaining a quote for travel (activity *a12),* only authorised travel agents can be used. This is an organizational-level constraint. Other constraints may apply at each level of the organization and individuals may have personal preferences, which they can express as constraints: for example, “I prefer to travel Qantas”. Ideally, these should be specified on personal websites and made available to web services applications and vendors as deemed appropriate by the site owner. This is consistent with the much-enhanced web services delivery vision (via personal websites) detailed by Watson et al. (2002).

We should note here that, notwithstanding the above, there are occasions where we may wish to explicitly specify sequencing information regarding activities. Fortunately, modern Prolog implementations – such as LPA’s *WinProlog* – provide a powerful range of meta-predicates that can be employed to implement control constructs such as *if-then-else, do-while* etc.

6. Conclusion

Rightly enough, the Semantic Web has generated a great deal of excitement. For its full potential to be realized though, many significant problems must be addressed. Within the IT industry, there is a tendency to ignore the lessons of the past. In this paper, we have argued that the integration issues to be faced in Semantic Web applications are very similar to those that caused the downfall of many data-centred SISP projects some 10-20 years ago. In addition, through a specific example (utilizing model abstraction, interfaces and non-determinism) we have demonstrated that past experiences might be utilized to benefit today's Semantic Web applications in a very real, practical sense.

Finally, we should note that the building of computer systems involves, at its core the still emerging disciplines of software and information/knowledge engineering as well as other more traditional areas of engineering and science, plus a number of non-technical management, social science and organizational disciplines. In essence, systems are not built in a vacuum, but within organizational environments where outcomes are heavily influenced by a myriad variety of internal and external socio-technical factors. The Leavitt Diamond (named after its originator, Harold, J. Leavitt) clearly demonstrates that in any major organizational change initiative the four dimensions of technology, structure, people and process are inextricably linked and that a change in any one of these will, almost invariably, have a significant impact on the remaining three (El Sawy, 2001). In this paper, we have focussed mainly on the technical dimension. This may be slightly misleading in that, while the technical challenges in Semantic Web work are certainly formidable, history (again) shows that these may pale into insignificance when compared to the structural, business process and people-related problems that will have to be addressed. This, we believe, is a critical area for future Semantic Web research.

References

Berners-Lee, T., Hendler, J. and Lassila, O. (2001). “The semantic Web”, *Scientific American,* Vol.284, No.5. pp. 34-43.

Bolman, L. and Deal, T. (1997). *Reframing Organizations*, Jossey-Bass Publishers: San Francisco, CA.

Buchanan, D. and Badham, R. (1999). *Power, Politics, and Organizational Change.* Sage: London.

Candan, K.S., Liu, H. and Suvarna, R. (2001). “Resource Description Framework: Metadata and its Applications”, *ACM SIGKDD Explorations Newsletter,* Vol.3, No.1, pp. 6-19.

Chakravarthy, B. (1997). “A New Strategy Gamework for Coping with Turbulence,” *Sloan Management Review,* Vol.38, No.2, pp. 69-82.

Chen, P.P.S. (1976). “The Entity-Relationship Model – Towards a Unified View of Data”, *ACM Transactions on Database Systems”,* Vol.1, No.1, pp. 9-36.

DMA (1990). *Data Centred Environment Study,* Doll Martin Associates, Sydney, Australia.

El Sawy, O.A. (2001). *Redesigning Enterprise Processes for e-Business,* McGraw-Hill: Boston.

Feldman, P. and Miller, D. (1986). “Entity Model Clustering: Structuring a Data Model by Abstraction”, *The Computer Journal,* Vol.29, No.4, pp. 348-360.

Fenna, D. (1966). “A System Design in the Australian Post Office”, *Proceedings of the 3rd Australian Computer Conference,* May, 1966, Australian Trade Publications, pp. 135-138.

Fensel, D., Lassila, O., van Harmelen, F., Horrocks, I., Hendler, J. and McGuinness, J. (2000). “The Semantic Web and its Languages”, *IEEE Intelligent Systems,* Nov/Dec 2000, pp. 67-73.

Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D.L. and Patel-Schneider, P.F. (2001). “OIL: An Ontology Infrastructure for the Semantic Web”, *IEEE Intelligent Systems,* Mar/Apr 2001, pp. 38-45.

Finkelstein, C. and Martin, J. (1981). *Information Engineering,* Savant Institute: Carnforth, Lancs, UK.

Fraunfelder, M. (2001). “A Smarter Web”, *Technology Review,* Vol.104, No.9, pp. 52-59.

Ghallab, M. et al. (1998). *The Planning Domain definition Language. Version 1.2,* Tech Report CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control.

Hamilton, D. (1994). “Unjustified Optimism in Information Systems Strategic Planning”, *Proceedings of the 5th Australasian Conference on Information Systems,* Melbourne, 27-29 September, pp. 661-673.

Hammer, M. and Stanton, S. (1995). *The Re-engineering Revolution,* Harper Collins: New York.

Hsu, C. (1996). *Enterprise Integration and Modeling: The MetaDatabase Approach,* Kluwer: Norwell, MA.

Kowalski, R.A. (1979). *Logic for Problem Solving,* North Holland: New York.

Lederer, A.L. and Sethi, V. (1992). “Meeting the Challenges of Information Systems Planning”, *Long Range Planning,* Vol.25, No.2, pp. 69-80.

Leonard-Barton, D. (1988). “Implementation Characteristics of Organizational Innovations”, *Communication Research,* Vol.15, No.5, pp. 603-631.

McGrath, G.M., (1994). “MP/L1: Towards an Automated Model of Organisational Power”, *Proceedings of the 2nd IEEE Australian and New Zealand Conference on Intelligent Information Systems,* Brisbane, 29 November – 2 December, pp. 487-491.

McGrath G.M., (1997). “An Implementation of a Data-Centred Information Systems Strategy: A Power/Political Perspective”, *International Journal of Failures and Lessons Learned in Information Technology Management,* Vol.1, No.1, pp. 3-17.

McGrath, G.M., Dampney, C.N.G. and More, E., (1998). “A Structured Approach to Conflict Prediction in Information Systems Strategy Implementation”, *International Journal of Intelligent Systems in Accounting, Finance and Management,* Vol.7, No.2, June, 1998, pp. 107-124.

McIlRaith, S.A., Son, T.C. and Zeng, H. (2001). “Mobilizing the Semantic Web with DAML-Enabled Web Services”, *Proceedings of the Second International Workshop on the Semantic Web,* Hong Kong, 1 May, 2001, pp. 82-87.

Markus, M.L. and Tanis, C. (2000). “The Enterprise System Experience - From Adoption to Success”, *Framing the Domains of IT Research: Glimpsing the Future Through the Past,* (R.W. Zmud ed.), Pinnaflex Educational Resources Inc: Cincinnati OH.

Martin, J. (1975). *Computer Data-Base Organization,* Prentice-Hall: NJ.

Martin, J. (1982). *Strategic Data-Planning Methodologies,* Prentice-Hall: Englewood Cliffs, NJ.

Nolan, R.L., Puryear, C.R. and Elron, D.H. (1989). “The Hidden Barriers to the Bell Operating Companies and their Regional Holding Companies’ Competitive Strategies”, *Future Competition in Telecommunications,* (M.M. Parker ed.), Harvard Business School Press: Boston, MA, pp. 301-327.

QED (1989). *Information Systems Planning for Competitive Advantage,* QED Information Sciences Inc: Wellesley MA.

Stal, M. (2002). “Web Services: Beyond Component-Based Computing”, *Communications of the ACM,* Vol.45, No.10, pp. 71-76.

Watkins, K., Ellinger, A. and Valentine, T. (1999). “Understanding Support for Innovation in a Large-Scale Change Effort,” *Human Resource Development Quarterly,* Vol.10, No.1, pp. 63-77.

Watson, R.T., Brohman, M.K., Parasuraman, A. and Piccoli, G. (2002). “Customer Service and Network Completeness”, *Proceedings of the 15th Bled Electronic Commerce Conference,* Bled, Slovenia, 17-19 June, pp. 229-243.