Semantic technologies: from niche to the mainstream of Web 3? A comprehensive framework for web Information modelling and semantic annotation

Eftychia Fefie Dotsika

School of Electronics and Computer Science

SEMANTIC TECHNOLOGIES:

FROM NICHE TO THE MAINSTREAM OF WEB 3?

A COMPREHENSIVE FRAMEWORK FOR WEB INFORMATION MODELLING AND SEMANTIC ANNOTATION

Eftychia Fefie Dotsika

A thesis submitted in partial fulfilment of the
requirements of the University of Westminster
for the degree of Doctor of Philosophy

November 2012

Semantic technologies:
from niche to the mainstream of Web 3.0?
A comprehensive framework for web
information modelling and semantic annotation

Abstract

Context: Web information technologies developed and applied in the last decade have considerably changed the way web applications operate and have revolutionised information management and knowledge discovery. Social technologies, user-generated classification schemes and formal semantics have a far-reaching sphere of influence. They promote collective intelligence, support interoperability, enhance sustainability and instigate innovation.

Contribution: The research carried out and consequent publications follow the various paradigms of semantic technologies, assess each approach, evaluate its efficiency, identify the challenges involved and propose a comprehensive framework for web information modelling and semantic annotation, which is the thesis' original contribution to knowledge. The proposed framework assists web information modelling, facilitates semantic annotation and information retrieval, enables system interoperability and enhances information quality.

Implications: Semantic technologies coupled with social media and end-user involvement can instigate innovative influence with wide organisational implications that can benefit a considerable range of industries. The scalable and sustainable business models of social computing and the collective intelligence of organisational social media can be resourcefully paired with internal research and knowledge from interoperable information repositories, back-end databases and legacy systems. Semantified information assets can free human resources so that they can be used to better serve business development, support innovation and increase productivity.

Table of Contents

1. Introduction..…… 1

    1.1. Aims and objectives…… 2

    1.2. Structure of the dissertation…… 2

2. Publications and their relevance…… 4

    2.1. Background…… 4

    2.2. The grouping of the publications…….... 7

    2.3. Contribution to joint papers ……..... 9

    2.4. Methodology……..... 9

3. Group I: Interoperability, customisation, reusability ……... 13

    3.1. The articles……..13

    3.2. Methodology…….. 17

    3.3. Contributions and discussion……..18

4. Group II: The impact of Social Media, Web 2.0 and Semantic Web……. 20

    4.1. The articles…… 20

    4.2. Methodology……..... 24

    4.3. Contributions and discussion…… 25

5. Group III: Web info modelling…… 27

    5.1. The articles…… 27

    5.2. Methodology…… 32

    5.3. Contributions and discussion…… 33

6. Contributions and framework……..35

    6.1. Framework overview…..…… 36

    6.2. The knowledge capture framework…….. 37

7. Conclusions, implications and future work…….. 43

7.1. Implications for research and practice…… 43

7.2. Conclusions…….. 44

REFERENCES…… 46

LIST OF PUBLICATIONS……..54

LIST OF FIGURES……55

LIST OF TABLES……...55

Appendix 1 – Publications

Appendix 2 – GISMoE Code

1 Introduction

The web was originally designed as a text and image repository for human use, while information modelling was mainly left to back-end databases and middleware systems. Its unprecedented expansion has triggered a significant increase in the expectations for effective information retrieval, knowledge sharing and collaborative working and has resulted in the development of diverse enabling technologies. Web-based information systems have become increasingly important and are largely considered to be the answer to most our information and knowledge requirements. Information modelling for these systems needs to be effective in representing the underlying information complexity and successful in reflecting their frequently multifaceted functionality.

Web information technologies developed and applied in the last 10 years have considerably changed the way web applications operate and have revolutionised information management and knowledge discovery (Fred *et al.*, 2011; Buckland 2011). Starting with the first applications that used XML encoding for the interchanging of data and going through the constant evolution of information modelling languages and their supporting technological frameworks, web information management has grown way beyond the hypertext linkage that Web 1.0 introduced (Deependra & Jai, 2005; Garcia-Molina, 2008; Virgilio *et al*., 2010). Social technologies, user-generated classification schemes and formal semantics have a far-reaching sphere of influence. They promote collective intelligence, support interoperability, enhance sustainability, and instigate innovation with wide organisational implications that can benefit a considerable range of industries (Hayman 2007; Enders *et al.*, 2008; Kim *et al.*, 2008; Li & Bernoff, 2008; Fernández *et al.*, 2011).

Another aspect is system interoperability, a crucial factor for information access, discovery and retrieval. The use of the web as a platform connecting multiple and diverse information repositories is based upon the communication between distributed information sources that adhere to different formats and information types by means of a variety of applications that run on disparate infrastructures and conform to a broad array of standards (Aberer *et al.*, 2003; Wang *et al.,* 2004; Tursi *et al.,* 2009).

Information retrieval facilities, however, often fail to obtain the information required. While search engine technology is maturing, it is still relatively young compared to, say, database technology. Search engines are habitually limited by poor indexing,

ranking of pages according to inappropriate metrics, the absence of keywords on relevant pages and inaccessibility to distributed information repositories of different formats, while search engine indices have become too large, with every search producing an enormous amount of results. At the end of every query the searchers and knowledge workers are inundated with a great amount of links that they need to go through in order to gather the knowledge sought (Chowdhury & Chowdhury 2003; Craswell & Hawking, 2009; Croft *et al*., 2009)

Enhancements to current practices come from a variety of sources. Adaptive methods for personalisation of search, advances in natural language processing technologies, collaborative filtering and information relevance measuring metrics are some such techniques (Schafer *et al.,* 2007; Berberich *et al.,* 2010; Weikum & Theobald 2010; Steichen *et al*., 2012). The common factor and prerequisite for the working of all these methods however, is high-quality information modelling (Knight & Burn, 2005; Barini & Scannapieco, 2006; Madnick *et al*., 2009; Spaniol *et al.,* 2009). The enhancement of Web information modelling constitutes the focal area of the research carried out and presented in this thesis.

1.1 Aims and objectives

The aim of this thesis is to develop a comprehensive framework that facilitates web information modelling and its retrieval by means of quality semantic enrichment.

This will be done by achieving the following three objectives:

(a) support all stages of web information modelling by informing on appropriate methods of semantic enrichment,

(b) enhance information quality by providing methods that facilitate the handling of semantic conflicts, and

(c) improve semantic interoperability among heterogeneous information repositories by supporting appropriate standardised formats for information modelling.

The thesis will provide the narrative that binds together a number of articles already published on this field of research.

1.2 Structure of the dissertation

The research carried out and presented in this dissertation covers a period from 2003 to 2012. It follows the methods, changes and advancements realised in the field of web information modelling during this time, evaluates their efficiency, analyses their drawbacks and addresses their shortcomings.

The research resulted in nine publications which are thematically related in terms of their area of study and their objectives, and which are listed in Appendix 1. The publications can be divided into three groups, based on the area they focus on and the contributions they make. The dissertation provides the narrative that binds this body of research together, bridges the research findings and contributions and proposes a framework which enhances quality semantic enrichment and facilitates web information modelling.

The rest of the thesis is organised as follows:

- Chapter 2 gives an overview of the publications used, groups them into the three distinct clusters mentioned above and discusses the overall methodology followed.
- Chapters 3, 4 and 5 give a comprehensive outline for each group of publications. Each chapter discusses the issues raised, the methodology followed and the results reached. For each group the results are divided into two categories, direct and indirect contributions. Direct contributions are the tangible, explicit outcomes of the publications. Indirect contributions are implied outcomes that, although elusive to substantiate, can be identified as trend forecasting and emergent technology impact predictions that proved timely and accurate.
- Chapter 6 proposes the framework and presents the thesis' original contributions.
- Chapter 7 draws the conclusions and highlights the implications for research and practice.
- There are two appendices at the end of this thesis. The first contains the publications and the second one is the listing of the code for the prototype system that will be discussed in Chapter 1.

2 Publications and their relevance

A list of the publications can be found in page 54. The actual publications are provided in Appendix 1. They are clustered into three groups based on the area they cover and their objectives. As it happens, by following the timeline of technical changes and evolvement of new paradigms, the groups also cluster in time. In chronological order, the first three publications address system interoperability, customisation and reusability and form the first group. The following three articles follow the rise and impact of Web 2.0 and social media and form the second cluster, while the last group is formed by the remaining publications which specifically focus on web information modelling practices and methods.

This chapter sets the background and context for the publications considered, introduces the publications and discusses the general methodology.

2.1 Background

Current applications supporting knowledge sharing and interoperability between incompatible knowledge repositories rely on annotating data and maintaining a syntactic consistency. This process adds structure and semantics to an otherwise unstructured or semi-structured mass of text-based information which, when in great quantity, becomes almost impossible to retrieve. The *Web 2.0* and the *Semantic Web* are two distinct angles of web information technologies which, while stemming from the same needs, have come to satisfy certain requirements and represent two different but equally prominent trends.

Web 2.0 (O'Reilly 2005) was coined in 2005 by Tim O'Reilly and is a selection of technologies and applications rather than an architecture. Web 2.0 focuses on social interaction, end-user involvement and information sharing. The content is user-generated and the information modelling is informal, carried out bottom-up by means of user-generated tag systems. Data and information are seen as the driving forces. Paired with the relevant business practices, Web 2.0 brought about Enterprise 2.0, a term that describes the set of Web 2.0 technologies enabling access to collective intelligence within organisations. These core technologies enable innovation through websites/sources of collective content with functionality that gets enriched as more people use them.

Compared to the traditional static web pages, Web 2.0 content can be dynamically generated by means of blogs, wikis, Ajax applications and RSS feeds. Organisational blogs are particularly widespread in both the private and public sectors (Kim *et al.*,

2008) and have a considerable effect on employee engagement, communication and collaboration. Integrated tools called mashups, combine data from more than one source and are used as situational applications that solve immediate business problems (Jhingran, 2006). Rigid content management systems are successfully aided or even replaced by collaborative wikis (Melhrose *et al.*, 2009). Information sharing and syndication are enabled by aggregators and RSS feeds, a widely adopted family of formats used to publish frequently updated content that improves organisational communication by streamlining smart information within employees' communities of practice, on their desktops, mobile devices or through their email clients.

The heart of Web 2.0 is social. Social computing has transformed digital economics with business models that are scalable, have low barriers for entry and are sustainable in the long term. Harnessing the power of social computing has created the need for organisational strategies that reflect the shift in online culture (Shuen 2008, Li & Bernoff 2008). In the case of organisations with digital presence, user interactions in social networks, paired with effective communication govern the revenue models. Increasing the member base becomes crucial when the revenue model is advertising, willingness to pay is the prominent driver for a subscription model and trust is of paramount importance for revenue based on transactions (Enders *et al.*, 2008).

Web 2.0 information modelling is done by means of user-generated tags known as folksonomies (Smith 2008). Folksonomies are collaborative metadata, created bottom-up in an analytical synthetic way. They are successful in organising corporate (Patrick & Dotsika 2007) information and enable innovation (Hayman 2007).

Web 2.0 deploys web services which are applications requested and executed remotely and which interface with one another providing a standard means of interoperating between different software applications. Web services share business logic, data and processes and promote interoperability and re-use. Web services' composition creates business processes and complex workflows and is regulated by standards such as orchestration and choreography (Busi *et al.*, 2006). Adoption of web services is on the increase due to the fact that organisations associate competitive advantage with a process of ongoing adaptation through flexible business processes and web services are proven to be a key determinant on business process flexibility (Deependra & Jay 2005).

Quality of information is at the centre of the disadvantages cited about Web 2.0 (Antiqueira *et al.*, 2007). Information modelling with folksonomies presents a number of further quality issues (Dotsika 2009). Other organisation-centred problems include technology dependence, security concerns, information overload and difficulties in finding relevant context. Ethical and legal issues such as privacy, anonymity, reputation, intellectual property rights, copyright violations, monetary function and trust are other often-quoted concerns. On the web services front, adoption is affected by low performance, basic forms of service invocation and service discovery issues (Wang *et al.*, 2004). While business adoption increases, organisations are reluctant to establish service registries, repositories and service level objectives.

Tim Berners-Lee introduced the Semantic Web (SW) in 2001 (Berners-Lee 2001) as a form of web content where knowledge representation is standardised and relies on languages expressing information in a machine process-able form, by means of a framework based on RDF (Resource Description Framework) and ontologies. The information modelling is predominantly top-down and it is done formally, without the participation of end-users.

The organisational impact of the Semantic Web is based on system interoperability and adaptive, personalised information access. Interoperability addresses heterogeneity issues present in data and business processes and it ensures information integration across systems, a process too costly for any organisation. Interchange, distribution and creative reuse are a Semantic Web inherited standard, while scalability is dependent upon increasingly powerful implementations (Ankolekar *et al.*, 2007). Adaptive technologies facilitate the tailoring of information access according to given user profiles. Intelligent information integration and agents such as information brokers, filters, personalised search agents and knowledge management services are examples of innovative applications.

The SW framework consists of XHTML, XML, the Resource Description Framework (RDF) and the Web Ontology Language (OWL). The Resource Description Framework (Beckett 2004) is an XML-based, standardised semantic annotation method, and, as such, interoperable. The RDF Schema (RDFS) adds basic ontology description power to plain RDF and many of its components are included in OWL. Together with RDF they form Semantic Web's RDF layer which adds semantics to web content and enhances machine process-ability. The model is scalable and searches are improved as the information can be processed in relation to the modelled relationships between data and/or resources.

The top part of the SW framework is occupied by ontologies, sets of shared, explicit and formal concepts used to organise and classify content. From an organisational point of view, ontologies are used to model enterprise information and processes accurately and consistently, enabling automatic reasoning, concept-based searches, process composition and knowledge discovery by means of intelligent agents (Hendler 2001). The Web Ontology Language OWL (Smith *et al.*, 2004) is a family of languages built using XML/RDF syntax.

The problems with the Semantic Web are mostly of a technical nature and come as a consequence of the complexity that is associated with its technologies. RDF in particular is difficult to publish. Any development of RDF/RDFS or OWL requires specialised expertise and this has prevented widespread adoption. Its formality makes it difficult to master and limits its popularity.

Large ontologies come with quality issues. The main problem is semantic uncertainty, which can be divided into ambiguity, randomness, inconsistency, incompleteness and vagueness (W3C 2008B). Handling semantic uncertainty plays an important role in ontology languages for the Semantic Web.

All this makes organisational adoption expensive and cumbersome. While large companies and high budget projects embrace the Semantic Web readily in order to take better advantage of intellectual assets, enhance productivity and increase competitiveness, smaller companies with web presence have remained reluctant to do the same.

2.2 The grouping of the publications

The publications referenced at the end of the thesis follow various paradigms of semantic technologies, evaluate information modelling techniques, assess their role and impact on system interoperability, identify challenges involved and investigate the quality issues of the information networks they generate. In more detail and order of publication:

The three articles on medical informatics (Dotsika 2003, Dotsika & Watkins 2003a, Dotsika & Watkins 2003b) apply technological advances in information technology in order to influence and improve healthcare practice by enabling the flexible modelling, direct representation and adaptable use of medical knowledge. They aim at resolving a number of difficulties encountered by information repositories of the domain, such as costly customisation, lack of reusability, high maintenance and poor information modelling. The result is the design and development of a prototype consisting of a

multimedia-enhanced version of the functional database language FDL, and a web-based, two-way translator interface between the application's native language and XML. While these papers concentrate on the field of medical informatics, this focus is rather superficial. The systems introduced are in fact generic and fit any application area. Medical informatics was chosen due to the novelty of the field at the time of publication, but the research can be adapted to fit a wide application area.

The three publications that follow (Dotsika & Patrick 2006, Dotsika 2006, Patrick & Dotsika 2007) focus on web knowledge management and in particular the use, contribution and impact of Web 2.0 and social media in knowledge capture, distribution and support of end-user involvement.

The article on the new generation of web knowledge (Dotsika & Patrick 2006) reviews the emerging trends and patterns of web use and explores the future and potential of web-based knowledge management. It investigates the main requirements for the support of KM in the next web generation, looks into existing developments and solutions and provides an independent framework for the capturing, accessing and distributing of web knowledge.

The article on the Communities of Practice (Dotsika 2006) poses a number of questions about the value of existing systems that assist CoPs, assesses the maturity of the different products and evaluates their effectiveness. The research reinforces the indication that while online communities benefit from technology, knowledge manipulation still poses a significant and often decisive obstacle to the flow of knowledge inside these communities.

The knowledge sharing publication (Patrick & Dotsika 2007) identifies collaboration and knowledge sharing as the core aspects for providing added-value to services and products and explores the ways in which this process can be improved. The paper highlights the impact of Web 2.0 technologies, the importance and contribution of social software in bottom-up modelling and end-user empowerment, and the need for bridging the socio-technical gap.

The final three publications (Dotsika 2009, Dotsika 2010, Dotsika 2012) further develop the findings and contributions of the previous groups and their particular focus is the support of the semantic enrichment of web content and the different methods followed to that extent.

The article on the reconciliation of ontologies with folksonomies (Dotsika 2009) explores the basics of web information classification engineering, identifies the strengths and weaknesses of the existing methodologies, assesses their

effectiveness, investigates key quality issues and proposes a common framework for reconciliation of the two classification approaches and quality assurance.

The Semantic APIs publication (Dotsika 2010) investigates the different methods deployed that add semantics to web content: semantic tagging and semantic APIs. The research proposes a framework for the evaluation of semantic tagging based on the main requirements for information discovery and recommends a number of comparative assessments, ranging from basic product information and requirements' analysis to the evaluation of the APIs information modelling functionality.

The third article of this group (Dotsika 2012) investigates the organisational perspective of the next generation of web technologies, often referred to as Web 3.0 and assesses their effect on organisational change. The research investigates the challenges of combining the two web paradigms to form Web 3.0, the effectiveness of the next generation of web technologies in supporting innovative solutions and the impact that Web 3.0 will have on the social organisation.

2.3 Contribution to joint papers

My contribution in the joint papers is as follows:

Publications with Keith Patrick (x 2): they are to be considered on a 50-50 basis. My research covers the more technical parts and especially anything that has to do with information modelling, semantics and the technologies involved.

Publications with Andrew Watkins (x 2): mainly mine. Andrew's kind contribution was limited to the hosting of the prototype software and its dissemination to interested parties. At the time I needed a server that the university (ISLS: Information Systems and Library Services) would not provide, so I opted for using a server at Birkbeck College.

2.4 Methodology

The research is in the area of information management with a special focus on web information modelling and retrieval. As a branch of information science, information management can be considered part of the social sciences, drawing from disciplines such as software engineering, computer science, management science and economics (Buckland 2001). The publications share a strong social and organisational context and the elements of participation and observation were important to maintain. The research is founded on social constructs such as development methods and business processes.

Quantitative research was deemed unsuitable, mainly due to the way it addresses organisational parameters related to users, their information needs and supporting systems, regarding them as static, independent and objective rather than dynamic, interacting constructs (Kaplan & Duchon, 1988). Qualitative research was chosen instead, as the method traditionally applied to social sciences when there is a strong aspect of social and institutional perspective and the resulting need for context-dependent research. Within this methodology it is assumed and acknowledged that organisational constructs, their meanings and development methods may change over time and be defined differently depending on the view of participants and their dynamics (Kaplan & Maxwell, 1994).

For similar reasons, the underlying epistemology guiding the research is interpretive (Orlikowski & Baroudi, 1991). Positivist studies assume an objective physical and social world existing independently of humans and do not allow for the flux created by participant intervention and subjective meanings. Positivist research methods were consequently considered unsuitable. The interpretive perspective on the other hand acknowledges the social aspect of knowledge and its dependency upon the action, interaction and participation of the members of a given social group. This standpoint was particularly relevant to the research undertaken.

From the various qualitative methods, the ones suitable for information management are grounded theory, case study, action research and ethnography (Myers 2009). When deciding upon the particular research methods for data collection, neither the grounded theory, nor the case study were considered appropriate. Grounded theory, while very useful for developing context-based, process-oriented descriptions, is extensively detailed and time-consuming and therefore unsuitable for research aiming at scaling up to larger concepts and looking at the bigger picture. Similarly, case studies tend to be too focused and therefore not suitable for generalising findings, especially when the aim of the research is to develop a framework.

Action research was the obvious choice because of its suitability in dealing with the multifaceted and complex character of information management processes, its collaborative and competencies-enhancing nature and its understanding of change processes in social systems (Hult & Lennung 1980).

Action research is a research method used in social and medical sciences that grew in popularity for use in formal investigations of information systems towards the end of the 90s. The method is grounded in practical action and aimed at solving an immediate problem situation while carefully informing theory. Action research is in

fact a family of research approaches that share common characteristics and involve the practitioners as subjects as well as co-researchers (Baskerville, 1999). The cyclical process of this specific methodology (briefly outlined as: identify problem, plan action, take action, evaluate, specify learning and back to problem identification) can successfully link theory to practice. It is particularly relevant to the information management community (Wood-Harper 1985) and therefore appropriate for web information management research.

The articles apply a combination of action research methods with emphasis on participatory observation, process consultation, prototyping and Soft Systems Methodology. In more detail:

- Information systems prototyping was used in the three papers on medical informatics (Dotsika 2003, Dotsika & Watkins 2003a, Dotsika & Watkins 2003b). The building of a model of the system was deemed the appropriate method in all three cases. Prototyping is particularly effective in the cases where the researcher works together with the stakeholders and facilitates the development of a system satisfying their collaborative requirements.
- Participant observation and process consultation were employed in (Dotsika & Patrick 2006, Patrick & Dotsika 2007 and Dotsika 2006). These two forms of action research draw upon surveys, interviews, document analysis and observations.
- A combination of participant observation, process consultation and Soft Systems Methodology (SSM) was followed in (Dotsika 2009, Dotsika 2010, Dotsika 2012). SSM is a systemic method for tackling management problem situations using a systems engineering approach, and it is pertinent when handling complex organisational issues that need to be dealt with in an organised manner.

Apart from action research, certain aspects of ethnography (Myers, 1999) were also applied, particularly in places where multiple perspectives needed to be incorporated in systems design (Holzblatt and Beyer, 1993) or plain study of the development of information systems (Hughes et. al., 1992). Ethnography applied to information management can be especially effective in revealing the actual, as opposed to the assumed, organisational culture. However, proper application of the method requires very long and serious engagement, which was considered counterproductive. As a consequence, only certain aspects of the method were employed.

The methodology will be revisited in more detail and the particular methods employed will be discussed separately for every group of publications in chapters 3 to 5.

The table below summarises the publications and related methodologies employed.

<table>
<tr><th>Publications</th><th colspan="2">Methodology</th><th colspan="2">References</th></tr>
<tr><td>Dotsika, 2003
Dotsika & Watkins, 2003a
Dotsika & Watkins, 2003b</td><td rowspan="3">Action research; in particular:</td><td>Information systems prototyping</td><td rowspan="3">Baskerville, 1999</td><td>McKay & Marshall, 2001
Wood-Harper, 1985
Carey & Mason, 1983</td></tr>
<tr><td>Dotsika & Patrick, 2006
Dotsika, 2006
Patrick & Dotsika, 2007</td><td>Participant observation
process consultation</td><td>Jepsen et al. 1989
Schein, 1969
Avison et al., 2001</td></tr>
<tr><td>Dotsika, 2009
Dotsika, 2010
Dotsika, 2012</td><td>Participant observation
process consultation
Soft Systems Methodology (SSM)</td><td>Jepsen et al., 1989
Schein, 1969
Checkland & Holwell, 1997
Baskerville & Wood-Harper, 1998</td></tr>
</table>

Table 2.1. Methodology summary by group of publications

3 Group I: Interoperability, customisation, reusability

The first three articles were published in 2003 and were the culmination of research focused on the development of a web-based information modelling system, designed for interactive information capturing and targeted at naive users. The system supports automatic database schema generation and is interoperable by means of an interface that translates the triple-store of the underlying native database application into XML. The publications introduced in this chapter are, in order:

- Dotsika F., Watkins A., (2003a) An interoperable, graphical environment for the capturing of medical information, International Journal of Health Care Engineering, Technology and Health Care, Vol. 11, No 5
- Dotsika F., (2003) From data to knowledge in e-health applications: An integrated system for medical information modelling and retrieval, International Journal of Medical Informatics and the Internet in Medicine vol 28, issue 4, pp 231-251
- Dotsika F., Watkins A., (2003b) GISMoE: a Graph-based Information System Modelling Environment, Proceedings of the Conference on Internet and Multimedia

The joint publications with Andrew Watkins are entirely mine. Andrew's contribution was the hosting of the prototype software.

In this chapter we will introduce each of the publications, discuss the methodology followed and outline the findings and contributions. The code for the prototype implementation is provided in Appendix 2.

3.1 The articles

An interoperable, graphical environment for the capturing of medical information (Dotsika & Watkins 2003a)

The first article set the basics for this research, proposing a graphical electronic healthcare application for the capturing and management of medical knowledge aimed at end-users. The tool's modelling flexibility can hide technical complexity from the end-user group (typically consisting of healthcare administrators with basic IT application skills but no technical background) while enabling the more sophisticated type of practitioner to model information explicitly, via a number of advanced modelling implements.

Originally named MedISD, the system was the result of a series of interviews with NHS practitioners interested in the electronic capturing of medical information and faced with the challenge of choosing the right healthcare application that would enable them to capture, store, retrieve and use the relevant information at the right time. The main requirement was the development of a graphical user interface front-end for information modelling that would involve no technical knowledge or database expertise, apart from basic desktop environment skills.

The tool makes use of the conceptually easy to grasp entity-relationship model (Chen 1976) and captures information in the form of directed graphs and automatically generates tailor-made medical database schemas based on the functional data model.

MedISD was designed to be modular and comprises the following components:

a. the front end is the *model visualisation panel*, a graphical user interface where the users can edit the primary entities and relationships

b. the middle module, or *information capture component*, where the edited schema is translated into the entities and binary relations of the underlying database

c. the *data dictionaries* provide the list of entities already in the system and

d. the back module, or *schema generator*, which updates the back-end database to correspond to the running session.

A prototype implementation was carried out using Java 2 Platform, Standard Edition (J2SE), version 1.4.1 on both Solaris 9 and Windows 2000 Operating Systems. Java was chosen because it is architecture independent, provides portable user interface, and can enable loading on demand of the application front end as an applet over the web. The database schema generated complied with functional data model (Shipman 1981).

The key contribution of the publication is the design and development of a prototype information modelling system which enables conceptual modelling based on the entity-relationship model and generates data dictionaries and database schemas.

GISMoE: a Graph-based Information System Modelling Environment (Dotsika & Watkins 2003b)

The second publication from this group took the effort one step further and developed the Graph-based Information System Modelling Environment (GISMoE).

The focus was primarily the enhancement of the functionality of the first prototype and its use as a tool that models information quickly and effortlessly, generates the database schemas and allows for interoperability and communication with other information repositories. The requirements that took precedence this time were the need for frequent re-modelling of the information that comes from varied sources, a duty usually undertaken by database experts and never entrusted to end-users, as it requires careful planning and database development. The developed system bypasses these concerns by allowing the user to model information by means of directed graphs and automatically generates the database schema that corresponds to the designed diagram. It further simplifies modelling by supporting complex objects, sub-schemas and user views.

The functional data model (Shipman 1981) was again employed for the information modelling as it is conceptually easy to understand and can be adopted by end-users who are aware of the basic binary relational schema concepts (as opposed to the n-ary relations, generally concerning the relational model). The automatically generated schema was compatible with the functional database FDL (King & Poulovassilis 1988) and the prototype implementation of GISMoE ran both as an applet and as a stand-alone application.

The system is once again modular and supports the same main features as its predecessor, i.e. the graphical information modelling environment and the automatic database schema generation. The graphical information modelling environment is enhanced with extra functionality for schema editing and there are three extra modules:

(a) the *sub-schema generator* for the development of user views,

(b) the (explicit) *schema manipulator*, a component that allows the addition of extensionally defined functions that might be added by the more sophisticated user or administrator and,

(c) the XML *schema translator* that allows for system integration and interoperability with other databases and medical knowledge repositories. The translator generates a data definition document (DTD) based on the functional schema

The implementation carried out using Java 2 Platform, Standard Edition (J2SE), version 1.4.1 on both Solaris 9 and Windows 2000 Operating Systems (as before).

The key contribution of the article is the enhancement of the information modelling environment developed in the previous publication. The new system is web-based

and has extra functionality that facilitates knowledge capture with edit facilities, sub-schemas and user views, user-defined functions and FDL/XML output. The simplicity of the modelling can influence existing practices by allowing the direct involvement of end users.

From data to knowledge in e-health applications: An integrated system for medical information modelling and retrieval (Dotsika 2003)

The last paper of the group provides the theoretical background behind the system developed. It researches the suitability of the different data modelling paradigms for the design and development of a system that reduces the complexity of developing medical information systems and focuses on improving healthcare practice by enabling custom schema modelling, direct representation and flexible use of medical knowledge.

The relational model (Codd, 1970) and the available web-based relational systems at the time were found lacking in terms of costly development (lack of code reuse, expensive customisation) and modelling flexibility (non-existent end-user development, lack of support for complex objects and multimedia). While this was the case at the time, the relational database market has provided users with a number of rapid web application development tools since, from proprietary solutions such as the Oracle Application Express (Oracle APEX) (Zaharieva & Billen, 2009), to open source content management software platforms like Drupal (Byron *et al.*, 2008).

The object-oriented model (Won, 1990) and object-oriented database market was similarly found lacking in a number of ways associated with performance issues and flexibility in information modelling (similarly non-existent end-user development, system-wide repercussions of schema changes) and retrieval (lack of ad-hoc querying). As a result there were no object-oriented products aimed at traditional processing applications requiring high performance and scalability. Not much has changed since. The object database market has remained in the background with products such as Intersystems Caché (Intersystems 1996; Tanaka *et al.*, 2003) and open source systems such as Db4o (Versant 2000; Paterson *et al.*, 2006) and Perst (Mc Object 2009).

The functional architecture was chosen as the best solution for the given requirements and in particular due to its conceptual modelling simplicity, support for complex data structures, incomplete information and multimedia content, cost-effective customisation, code re-use and low maintenance. Besides facilitating

information modelling, the model allows for the collaboration of the developer, the practitioner and the end user in the modelling process.

Interoperability was based on the use of XML for connecting heterogeneous knowledge repositories and databases by means of a translator interface that gives the developer the choice, deployment, and merging of different models to fit particular circumstances. At the time this group of papers was published, XML and its related standards were not yet as widely adopted as they are today. Definition Type Documents (DTD) were used for validation and the GNOME project (GNOME, 2000) then XML query specification for database integration.

The key contribution of the article is the investigation of the suitability of different data modelling paradigms for web information and knowledge capture. It focuses on enabling end-user involvement in custom schema development, emphasises the need for system interoperability and highlights the appropriateness of the labelled directed graphs (triples) and XML.

3.2 Methodology

The methodology used in all three articles was information systems prototyping, which is a form of action research, a methodology well suited for information systems where there is constant interaction between humans, information and technology (Baskerville, 1999; McKay & Marshall, 2001). The prototyping methodology follows a cyclical process which can be summarised as:

- identification of basic requirements,
- development of the initial prototype,
- review and evaluation of interim solution,
- revision and enhancement of the prototype implementation.

The methodology links theory to practice and is particularly relevant to the information systems community (Wood-Harper 1985) and therefore appropriate for web information systems research. Its main advantages are that it improves the system's functional requirements and logic and enhances accessibility, user satisfaction and evolution requirements (Carey & Mason 1983).

However prototyping is not without its critics. The drawbacks are dependent upon the prototype's fidelity (Rudd *et al.*, 1996). Low-fidelity prototypes are characterised by limited functionality and they are generally developed in order to illustrate concepts. They do not model user interaction and do not demonstrate how the end product will

operate. In the case of low-fidelity prototyping, disadvantages include possibility of insufficient analysis, limited error checking, flow limitations and limited usefulness for usability checks. High-fidelity prototypes on the other hand are fully interactive and demonstrate the core functionality of the end-system. The disadvantages here are related to development times, inefficiency for proof-of-concept designs and high costs. Whilst the prototype developed was of high-fidelity, its development within an academic environment bypassed most of the disadvantages mentioned.

As the framework required was meant as a business solution, the *usability* prototype category of the Dynamic Systems Development Method (DSDM) was followed (DSDM, 1994). The system developed was a *horizontal* prototype, providing a broad view of the system and focused on end-user development and interaction (rather than the low-level system functionality expected in *vertical* prototyping). It was built as a display version to demonstrate the scope and functionality of the system, as well as verify and confirm the user requirements. These requirements were collected through interviews with professionals working for the Camden Primary Care Trust and unofficial interim requirements reports related to the Care Records Service project.

3.3 Contributions and discussion

The group's direct (*a* and *b*) and indirect (*c* and *d*) contributions are:

(a) The design and implementation of a web-based interactive prototype for information modelling aiming to improve knowledge capture by means of enabling conceptual modelling and to influence existing practices by allowing the direct involvement of end users. The system proposes solutions for a number of related problem issues such as costly customisation, reusability, high maintenance and poor end-user involvement.

(b) The investigation of the suitability of different data modelling paradigms for web information and knowledge capture. The research focused on enabling end-user involvement in custom schema development, emphasised the need for system interoperability and highlighted the appropriateness of the labelled directed graphs (triples) and XML.

(c) The first indirect contribution relates to the use of conceptual information modelling and the entity-relationship model's "triples". The claim of the model's suitability for web information modelling discussed in the publications proved correct. This was to become the de-facto standard for interoperable web

information modelling supported by the World Wide Web Consortium standards for the Resource Description Framework (RDF) (W3C RDF, 2004) and Web Ontology Language (OWL), (W3C OWL, 2004).

(d) The second indirect contribution relates to the choice of XML for the support of cross-system interoperability. Although by the time the publications appeared XML was already positioned at the for-front of information modelling, the choice of its use in the particular research had been made much earlier (2000).

Although the application area is electronic health care, the research carried out and the system implemented are of a more general nature. The prototype can easily be modified to fit information requirements of different fields. As a result, the modelling environment can be adapted so that it captures information from a variety of application areas, such as government (local and e-government), education and services.

The reason for focusing on e-health was strategic. At that time, NHS were looking at the creation of electronic care records and it was in December 2003 that the then Secretary of State for Health, John Reid, announced the plans for a national NHS Care Records Service and the development of the Summary Care Record (SCR), which would contain clinical information, such as summary medical history, prescriptions, possible allergies, operations and procedures (Powell & Thompson 2010). It was at the back of this that the interviews with healthcare professionals took place.

4 Group II: The impact of Social Media, Web 2.0 and Semantic Web

The following three articles were published between 2006 and 2007 and continue the research on web information modelling. Their particular focus is the support of web knowledge management as well as the use, contribution and impact of Web 2.0 and social media in knowledge capture and distribution, bottom-up modelling and end-user empowerment. The publications introduced in this chapter are:

- Fefie Dotsika, Keith Patrick, (2006) Towards the New Generation of Web Knowledge Search and Share, VINE: The Journal of Information and Knowledge Management Systems Vol. 36 No. 4, pp 406-422
- Dotsika F., (2006), An IT Perspective on Supporting Communities of Practice, Encyclopaedia of Communities of Practice in Information and Knowledge Management, Coakes, E., & Clarke, S., (Eds), 2006, Idea Group Inc, pp 257-263
- Keith Patrick, & Fefie Dotsika, (2007) Knowledge Sharing: Developing from Within, The Learning Organization: The International Journal of Knowledge and Organizational Learning Management, Vol. 14, No 5, pp 395-406

My contribution in the joint papers with Keith Patrick is on a 50-50 basis. My research covers the more technical parts and especially anything that has to do with information modelling, semantics and the technologies involved.

In this chapter we will introduce each of the three Web 2.0 publications, discuss the methodology followed and outline the findings and contributions.

4.1 The articles

Towards the New Generation of Web Knowledge Search and Share (Dotsika & Patrick, 2006)

The article on the new generation of web knowledge reviews the emerging trends and patterns of web use and explores the future and potential of web-based knowledge management. In order to do so it focuses on knowledge retrieval by investigating two search paradigms, the cognitive method and the automated approach.

(a) In the case of the cognitive method the end-user searches across pages either through hyperlinks, subject directories, or search engine results. This approach (Navarro-Prieto *et al.*, 1999; McEneaney, 2001; Wang and Zaıane, 2002) is found to be cheaper and more efficient when dealing with open domains and

community-based applications. However it is inappropriate when one is dealing with time constraints and a potentially overwhelming volume of results returned by the search engine.

(b) The automated approach implies the use of search engines and/or intelligent agents. The method (McIlraith *et al.*, 2001; Fensel, 2001; Benjamins *et al.*, (2004) fares best with closed domains of knowledge and when highly precise information needs to be retrieved automatically. However its results can be disappointing when conventional web mark-up is involved, which provides syntax but lacks semantics, a fact that severely limits the task of intelligent agents.

The enabling web technologies and emerging trends were reviewed and considered in order to identify the ones most pertinent in knowledge search and assess their overall impact in web-based knowledge management. The technologies associated with Web 2.0 and the increasingly popular social media platforms were found particularly influential in collaborative knowledge capture and sharing. This acknowledges the fact that knowledge-based systems are shared, dynamic, evolving resources whose underlying knowledge model requires careful management due to its constant changing. Semantic mark-up and web ontologies in particular were deemed equally indispensable in information and knowledge retrieval.

The article's key contribution is the proposal of an independent framework for the capturing, accessing and distributing of web knowledge. The framework promotes the pairing of collaborative technologies associated with Web 2.0 and social media platforms with the use of semantic mark-up and proposes the deployment of web ontologies for structuring organisational knowledge and semantic text processing for the extraction of knowledge from websites.

An IT Perspective on Supporting Communities of Practice (Dotsika 2006)

The paper's premise is the increasing awareness among organisations that encouraging and maintaining communities of professionals with common interests can reduce costs and increase profits. Communities of Practice (CoP) are often viewed as a catalyst to the success of a particular organisation's knowledge management system. The paper poses a number of questions about the value of existing systems that assist those communities. The importance of emerging technologies such as social media is identified and analysed by means of the four groups of social actions framework (Ngwenyama & Lyytinen, 1997). Web-based collaborative technologies are recognised as highly influential for instrumental,

communicative, discursive and strategic actions and therefore advantageous to all four groups of the framework.

The research carried out goes on to assess the maturity of the different products of collaborative technologies used in the support of knowledge management and evaluates their effectiveness. The enabling software platforms are divided into two categories: platforms aimed specifically at assisting CoP (particular focus on the communication layer) and platforms designed to support knowledge management in general (particular focus on content management), but which meet the requirements for CoP support as well (Domingue *et al.*, 2001; Motta *et al.*, 2000). Both communities-dedicated and general knowledge management support systems were evaluated. With a large and constantly increasing number of available platforms in each category, the list of products appraised was representative of the range of services available, but was by no means exhaustive. The conclusions reached are consistent. Social media and collaborative technologies enable, facilitate and enhance the work of CoP.

The research further reinforces the indication that while online communities benefit from technology, knowledge manipulation still poses a significant and often decisive obstacle to the flow of knowledge inside these communities. It establishes that the emergence of the Semantic Web seems to tackle a number of these problems, enhancing the sharing of a common understanding of a domain among the members of the community, analysing and reusing domain knowledge and making explicit any domain assumptions. Another Semantic Web application is used to identify CoP within an organisation (Alani *et al.*, 2002), by examining the connectivity of instances in a knowledge base with regard to their type, weight and density, a process presently done by means of structured interviews. Despite the evidence suggesting that a Semantic Web platform would benefit any organisation with active CoPs, the process of migration was found to be rather cumbersome, requiring specialist knowledge of the technologies involved and would therefore prove costly.

The key contribution of the article is the review, evaluation and assessment of collaborative knowledge management systems used for the support of communities of practice. The products were found lacking in semantic expressiveness and end-user involvement.

Knowledge Sharing: Developing from Within (Patrick & Dotsika 2007)

The knowledge sharing publication identifies collaboration and knowledge sharing as the core aspects for providing added-value to services and products, explores the

ways in which this process can be improved and proposes an approach that encourages knowledge sharing through the development of systems from within.

The paper highlights the impact of Web 2.0 technologies, the importance and contribution of social software in bottom-up modelling and end-user empowerment, and the need for bridging the socio-technical gap. It goes on to demonstrate how "developing from within" provides an effective solution to the problem of knowledge sharing by means of the combination of the social and technical systems. The research identified four problem areas of knowledge sharing following this particular development.

(a) Knowledge modelling and interoperability issues. At the time of the publication, the popular opinion was that open dynamic environments did not benefit from traditional semantic reconciliation techniques that depend upon shared vocabularies and global ontologies (Aberer *et al.*, 2004). A methodology that merged successfully formal semantics and bottom up design claimed the adoption of emergent semantics as a possible solution when based on the adoption of new heuristics founded on a domain's emerging properties and locally agreed semantics (Aberer *et al.*, 2003, Cudre´-Mauroux and Aberer, 2004).

(b) Standardisation issues were discussed within the premise that successful knowledge sharing relies on a common meaning, syntax, definition and delivery mechanism, so that, standardising on information interchange increases the ability to share data throughout organisations. A proposed solution (Dodds, 2006) paired bottom-up development (key in knowledge sharing) with formal modelling (key in knowledge retrieval).

(c) Security issues and in particular risks inherent in certain Web 2.0 technologies were discussed and in particular problems linked to cross-site scripting, code correctness issues, object model violations, insecure randomness and poor error handling.

(d) Maintenance indicated the need to frame local solutions in a wider organisational context and a strategy of collaborative activities extending beyond individuals, workgroups or departments. Closely related to maintenance, scalability issues were discussed, both technical (network effects in the case of particularly popular applications) and financial (economic effects when the revenue does not scale with the application usage). Open source software was identified as a possible solution that can potentially minimise the financial burden of scaling/changing

applications and platforms, it has the drawback of lack of technical support, which dictates the need for in-house technical expertise.

The paper concluded that “developing from within” provides an effective solution to the problem of knowledge sharing by means of the combination of the social and technical systems. This solution is facilitated by the social phenomena that underpin emerging web trends and hindered by identified potential weaknesses. From the four areas identified as problem areas, issues related to knowledge modelling and standardisation were further researched in the papers presented in the next section.

The key contribution of the article was the confirmation of the influence of social media and Web 2.0 technologies in organisational knowledge modelling and identification of interoperability, standardisation, security and maintenance as the main problem areas for organisational knowledge sharing.

4.2 Methodology

The research methodology applied in this group of publications is action research (Baskerville, 1999). In order to evaluate the organisational impact of the adoption of knowledge systems based on social media and emerging web technologies, multiple different stakeholder views and value conflicts had to be taken into consideration. The publications drew upon interviews and informal surveys with practitioners, consultants and knowledge workers, as well as document analysis and participant observations. Not one single organisation was used; instead a number of different sources were consulted during the period of a year.

Two particular forms of action research were employed, participant observation (Jepsen *et al.*, 1989) and process consultation (Schein, 1969). Participant observation was deemed predominantly suitable as it focuses on gaining familiarity with knowledge practitioners and their practices. Process consultation was adopted as the method to influence, develop and enhance the practitioners’ ability to anticipate and solve future related issues.

The control aspects related to the initiating procedure, authority within the project and degree of formalisation (Avison *et al.*, 2001) were identified. The possible variations are as follows:

(a) The initiation can be activated by the “researcher”, the “practitioner” or can be classified as “collaborative”, depending on whether the project is research-driven (the researcher is looking for appropriate settings to apply a specific theoretical

approach), problem-driven (the practitioner is confronted with a particular problem that requires solution), or somewhere in between.

(b) The determination of authority can be classified as “client-driven”, “identity” (the team of researchers are the initiating practitioners), or “staged” (there is a power migration during the project).

(c) The formalisation aspect registers “formal” for specific written contracts and intellectual property agreements, “informal” for lack of control structures, or “evolved” in the case of a change.

The control parameters for each one of the publications were recorded as follows:

Publication	**Initiation**	**Authority**	**Formalisation**
(Dotsika & Patrick, 2006)	Collaborative	Staged	Informal
(Dotsika, 2006)	Researcher	Identity	Informal
(Patrick & Dotsika, 2006)	Collaborative	Staged	Informal

Table 4.1. Control parameters

Apart from action research, certain aspects of ethnography (Myers, 1999) were employed. Ethnography applied to IS research can be especially effective in revealing the actual, as opposed to the assumed, organisational culture. It provides information systems researchers with a good grasp of the social and organisational aspects of information systems’ development. However, proper application of the method requires very long and serious engagement, which under the time constraints was considered counterproductive. As a consequence, only certain features of the method were employed. In particular, the research carried out focuses on context which in ethnography is considered crucial and is not regarded as noise in the data. In all three publications the organisational context was especially relevant.

4.3 Contributions and discussion

The group researches web-based information and knowledge management with a particular focus on the use, contribution and impact of Web 2.0 and social media in organisational knowledge capture, bottom-up modelling and end-user empowerment The group’s direct (*a* to *c*) and indirect (*d*) contributions are:

(a) Review and assessment of the maturity of existing knowledge management products applying collaborative technologies, used for the support of communities

of practice and evaluation of their effectiveness. Most were found lacking in semantic expressiveness and end-user involvement.

(b) Confirmation of the influence of social media and Web 2.0 technologies in organisational knowledge modelling and strong evidence of their leading role in future developments. Identification of interoperability, standardisation, security and maintenance as the main problem areas for organisational knowledge sharing.

(c) Proposal of a framework for the capturing, accessing and distributing of web knowledge which promotes the pairing of collaborative technologies and social media platforms with the use of semantic mark-up and the deployment of web ontologies for structuring organisational knowledge and semantic text processing for the extraction of knowledge from websites. This last contribution was identified as the theme of further future research (Chapter 5).

(d) The indirect contribution of the group is related to the early recognition of the impact that social media and Web 2.0 technologies would be having in future knowledge practices of the extended organisation. Web 2.0 was coined by Tim O'Reilly in late 2004 (although it appears in blogs as early as 2002), so that, at the time the research was taking place, the trend was very new indeed. The Semantic Web framework, although slightly older, had been less popular in terms of acceptance and adoption. The proposed pairing of Web 2.0's collaborative strength with the Semantic Web's standardised semantic markup was, at that time (2006), highly innovative.

5 Group III: Web info modelling

The final papers were published between 2009 and 2012 and continue the research on web information modelling. They build upon the findings and contributions of the previous groups and their particular focus is the support of the semantic enrichment of web content and the different methods followed to that extend. Diverse schemes of web content classification are reviewed and their role and functionality analysed. The various methods of automatic and semi-automatic semantification of web content are investigated and the findings are assessed and evaluated. The publications presented in this chapter are:

- Fefie Dotsika, (2009) Uniting formal and informal descriptive power: Reconciling ontologies with folksonomies, International Journal of Information Management, Volume 29, Issue 5, October 2009, pp 407-415
- Fefie Dotsika, (2010) Semantic APIs: scaling up towards the Semantic Web, International Journal of Information Management, Volume 30, Issue 4, August 2010, pp 335-342
- Fefie Dotsika (2012) The next generation of the web: an organisational perspective, WBS Working Paper Series in Business and Management, 12-1, March 2012

The first two articles have been published in 2* journals (Association of Business Schools ranking of journals). All articles in this section are single author papers.

In this chapter we will introduce each of the publications, discuss the methodology followed and outline the findings and contributions.

5.1 The articles

Uniting formal and informal descriptive power: Reconciling ontologies with folksonomies (Dotsika 2009)

The paper looks into web information modelling by examining the two prevailing classification schemes (folksonomies and ontologies) and their detailed characteristics. Folksonomies are collaborative user-generated metadata, created bottom-up and represent the prominent classification tagging scheme for Web 2.0. Ontologies are explicit and formal, created top-down and are part of the Semantic Web framework. The research is based on the premise that the two can be integrated in a way that reconciles their differences while preserving their advantages,

maintaining thus the collaboratively engineered content while ensuring a platform for automated search, intelligent agents and system interoperability.

Six methods for integrating the different paradigms by bringing together the bottom-up approach of folksonomies with the traditional top-down design of ontologies were reviewed and evaluated separately and against one another. The comparison criteria ranged from the methods' modelling power (outlined as relation building, attribute support, complexity handling and ontology mapping) to quality, automation, application and metrics (Macgregor & McCulloch 2006; Abbott 2004; Mai 2004; Hess *et al.*, 2008). The advantages and disadvantages of each method were identified and a requirements' framework was proposed for integrating ontologies and folksonomies, which highlights the list of criteria as follows:

(a) Quality issues

(b) Semantic enrichment

(c) Mapping completeness

(d) Trust and ethics.

Quality issues were found to be a multifaceted issue (Colomb & Weber 1998; Rector *et al.*, 2001; Kashyap 2003) and were further investigated. Bringing together the existing quality assurance methodologies, design processes and best practice guidelines the following quality criteria framework was introduced:

(a) Quality assurance criteria should be established during the original design

(b) The relationship between an organization's semiotic system to the information system that describes it should be proportional.

(c) Multiple inheritance should be avoided

(d) Balance between logical theory and reality should be maintained

(e) Use of a basic taxonomic structure should be supported

(f) Avoidance of ambiguity and inconsistency should be ensured

(g) Discrepancies in granularity should be prevented

(h) Issues of trust should be addressed

The resulting proposed framework of reconciliation for ontologies and folksonomies provides a flexible, yet rigorously regulated interface between the two parts that allows the adoption of the dual approach of bottom-up population and top-down standardisation.

The key contributions of the article are:

1) *a framework of integration requirements for semantic enrichment, pertaining to quality, semantic enhancement, mapping completeness and trust/ethics and focusing on quality issues which are identified and distinguished from concerns about mapping and semantic clustering, and*
2) *an evaluation matrix for semi-automatic semantic enrichment methods which provides a dashboard of potential requirements (relation analysis, tag cleaning and quality control, ontology mapping, attributes and complexity, multiple resource service, automation, developer support, evaluation and metrics), highlights availability or effectiveness and helps determine possible shortcomings.*

Semantic APIs: scaling up towards the Semantic Web (Dotsika 2010)

The second paper takes the research further by investigating the different methods used to add semantics to web content. The research carried out investigates existing systems that enable machine readability and automatic interpretation of web content, outlining their primary features and functionality.

Semantics can be added either by applying semantic markup or by means of semantic application programming interfaces (APIs). Semantic markup adds semantics by tagging web content through methods such as microformats, topic maps, ontologies and versions of the resource description framework family (RDF, Notation 3, RDFa). Semantic APIs take unstructured web pages as input and return the content's contextual framework. The semantic APIs explored were Dapper, OpenCalais, SemanticHacker, SemanticCloud, Zemanta and Ontos.

In order to assess the assorted methods, the different approaches to semantic tagging were compared. The basic requisites for traditional information retrieval were adapted to web information retrieval requirements (Cleverdon 1966; Agosti & Melucci 2001, Pokorny 2004) and the following categories for comparison were identified:

(a) Coverage: system interoperability & standardisation.

(b) Precision: issues of information modelling (completeness, granularity) and quality.

(c) Presentation: issues of usability, navigation; divided into:

    a. browser support,

    b. use of XHTML attributes for semantic tagging.

(d) Cost: pricing the solution and user-effort; divided into:

a. simplicity of solution, expertise requirements, issues of code integration and maintenance

b. issues of custom-made vs. off-the-self, open-source vs. bespoke.

In order to assess the semantic APIs, they were compared against each other in terms of a set of criteria based on practitioner requirements, consultants input and participant observation results. This comparison was divided into basic product information and requirement-based decision planning. The latter concentrates on information modelling and this led to a third category based on the APIs' common ground of enhancing web information retrieval and discovery. The list of criteria is as follows:

(a) General characteristics (these include developer, availability of extra tools, Web service information, online user support, cost and performance)

(b) User requirements (these include key concepts and categories, relevance scores, new format creation, content presentation, content expansion and content findability)

(c) Input and output supported formats

The result and the article's key contribution is a framework for the evaluation and comparative assessment of semantic APIs which assists the choice of the best suited interface for adoption. Decision-making is based on a step-by-step guide relating to:

1) content requirements and the systems that support them ;
2) product specifics such as developer, availability of extra tools, Web service information, online user support, cost and performance;
3) information modelling input/output format requirements;
4) input libraries and custom taxonomies

The next generation of the web: an organisational perspective

The third paper of this group investigates the next generation of web technologies referred to as Web 3.0 and assesses their influence over organisational change. This investigation brings together and bridges over previous research results while taking the effort further to consider the challenges of combining the two web paradigms to form Web 3.0, the effectiveness of the next generation of web technologies in supporting innovative solutions and the impact that Web 3.0 will have on the social organisation.

The use and role of Web 2.0 in the organisation were analysed and compared to the traditional static web content. The social aspect of the applications was paired with the support of web services. While social media has transformed digital economics with business models that are scalable, have low barriers for entry and are sustainable in the long term, web services brought the advent of cloud computing with applications that share business logic, data and processes and promote interoperability and re-use. Adoption of social media and web services is on the increase due to the fact that organisations associate competitive advantage with a process of ongoing adaptation through flexible business processes and web services are proven to be a key determinant on business process flexibility (Deependra & Jay 2005).

The organisational use and role of the Semantic Web were also analysed and firmly placed in the area of system interoperability and adaptive, personalised information access. Interoperability addresses heterogeneity issues present in data and business processes and ensures information integration across systems, a process too costly for any organisation. Interchange, distribution and creative reuse are a Semantic Web inherited standard, while scalability is dependent upon increasingly powerful implementations (Ankolekar *et al.*, 2007). Echoing the work carried out in Web 2.0 applications, Semantic Web adaptive technologies facilitate the tailoring of information access according to given user profiles. Intelligent information integration and agents such as information brokers, filters and personalised search agents are examples of innovative applications.

Outlining the advantages of integrating Web 2.0 with Semantic Web technologies, the article examines the requirements, challenges and organisational implications of the methods available. Quality of information was analysed across the spectrum of web paradigms (Web 1.0, 2.0, SW and Web 3.0) using the four-category quality model that comprises representation, accessibility, contextual and intrinsic data quality (Wang *et al.*, 1997; Zhu & Wang 2010). Other impact aspects analysed are content generation, distribution, retrieval and deployment, as well as the social side as a networking enabler. The conclusion is that organisations can truly benefit from low-cost organisational adoption of semantic enrichment that is easy to implement and flexible to update.

The key contribution of the article is a dashboard for tracking and assessing organisational information quality in relation to the employed web model and a framework for evaluating the organisational impact of the adoption of semantic web

technologies in terms of content, from generation, distribution and re-use to retrieval and deployment. Four impact aspects were identified:

1) *Innovation focusing on technologies and related applications supporting semantic content innovation with organisational implications;*
2) *content-driven impact focusing on content generation, distribution, retrieval and deployment;*
3) *information quality focusing on contextual attributes, representation, accessibility/access security and intrinsic data quality;*
4) *organisational change focusing on the impact that new technologies bring to organisational processes, functions, values and power.*

5.2 Methodology

The research methodology applied in this group of publications is once again action research. As before, due to the multivariate social setting, multiple different stakeholder views and value conflicts had to be taken into consideration. The articles drew upon interviews and informal surveys with practitioners, consultants and knowledge workers, as well as document analysis and participant observations. Contrary to the previous group, a large fraction of the input came from the companies providing semantic technologies and especially their developers, online forums, blogs, and users.

A combination of participant observation (Jepsen *et al.*, 1989), process consultation (Schein, 1969) and Soft Systems Methodology (Checkland & Holwell 1997) was followed.

The collaborative aspect of participant observation and process consultation and their commitment to improve practice were particularly relevant for this group of publications. The control clauses of initiation and authority lay with the researcher.

Contrary to restricting the research process to a mere understanding and supporting of practice, it is essential to extend it further and achieve triangulation (Mathiassen, 2002). The three activities and their goals have as follows:

(a) Practice engagement in order to understand systems development (through interpretation)

(b) Practice support in order to build new knowledge (through design) and

(c) Social and technical intervention in order to improve practice (through intervention)

Other aspects of the methods employed that were especially pertinent to the research carried out were the model of the process, structure, role of the researcher and the primary goal (Baskerville & Wood-Harper, 1998).

The process model of participant observation is categorised as reflective, in the sense that it focuses on the differences between the methodology employed and the one promoted. The participants reflect on their practice, promoting thus understanding. Iteration is implied and structure is generally fluid (as opposed to rigorous). The objective is knowledge gain and the researchers' role is based on expertise.

Process consultation is categorised as implicitly linear, following the route engagement-diagnosis-planning-action. The structure is rigorous and determined by the consultation framework. The objective is organisational development and the researchers' role is based on expertise.

Soft Systems Methodology is a systemic method for tackling management problem situations using a systems engineering approach, and it is pertinent when handling complex organisational issues that need to be dealt with in an organised manner and/or problem situations that lack a formal problem definition. As such it was considered suitable for dealing with finding a way to evaluate and compare the semantic APIs. The method is categorised as iterative (in this case the iteration was implied), with fluid structure. The objective is organisational development and the researchers' role is collaborative.

5.3 Contributions and discussion

The group researches the various paradigms of semantic technologies used to model web information and focuses on the semantic enrichment of web content and the different methods employed to that extend. The group's direct (*a* to *e*) and indirect (*f* and *g*) contributions are:

(a) *A framework of integration requirements for semantic enrichment, pertaining to quality, semantic enhancement, mapping completeness and trust/ethics.*

(b) *Evaluation matrix for semi-automatic semantic enrichment methods which provides a dashboard of potential requirements (relation analysis, tag cleaning and quality control, ontology mapping, attributes and complexity, multiple resource service, automation, developer support, evaluation and metrics).*

(c) *A framework for the evaluation and comparative assessment of semantic APIs which assists the choice of the best suited interface for adoption. Decision-*

making is based on a step-by-step guide relating to content requirements, product specifics, information modelling, input/output format requirements, input libraries and custom taxonomies.

(d) *A dashboard for tracking and assessing organisational information quality in relation to the employed web model.*

(e) *A framework for evaluating the organisational impact of the adoption of semantic web technologies in terms of content, in terms of innovation, content-driven impact, information quality and organisational change.*

(f) *Identification of folksonomies and ontologies as the main web information classification schemes.*

(g) *Awareness that the reconciliation of the two approaches will give web applications the edge needed for the retrieval of information.*

The last two contributions follow the theme of the previous group, which proposed the innovative pairing of Web 2.0 collaborative strength with the Semantic Web's standardised semantic markup. However, the formal and robust variety of the Semantic Web comes at a high cost that makes organisational adoption problematic, while the alternative of an automated user-friendly approach, easier to implement and therefore better suited for organisational adoption, is not yet available.

6 Contributions and framework

In the previous chapters we separately examined the three groups of papers and discussed their direct results and key contributions. The contribution of each individual publication can be found at the end of each article (sections 3.1, 4.1 and 5.1). The contribution of each group as a whole can be found in sections 3.3, 4.3, 5.3. Looking at them as a whole, the first group set the scene for web information modelling by means of formal semantic notation and demonstrated the importance of automated development. The second group introduced the aspect of social media, confirmed the benefits and challenges of their adoption and verified the need for imported standardisation to assist knowledge management and interoperability. The last group built upon the previous findings, researched existing technologies and methods for web information modelling and content standardisation, and developed a requirements-driven framework for web information modelling and semantic enrichment.

Revisiting the initial objectives, we consider what has been achieved:

(a) Support for all stages of web information modelling by informing on appropriate methods of semantic enrichment.
(b) Enhancement of information quality by providing methods that facilitate the handling of semantic conflicts.
(c) Improvement of semantic interoperability among heterogeneous information repositories by supporting appropriate standardised formats for information modelling.

This chapter re-visits the aim and research findings of the publications and brings everything together to present the original contribution to knowledge of the research undertaken.

6.1 Framework overview

The aim and original contribution of this thesis has been to develop a comprehensive framework that facilitates web information modelling and retrieval by means of quality semantic enrichment.

The knowledge capture framework is based on two basic aspects governing decision making in organisational change and relates to the perennial questions *why* and *how* (Pettigrew, 1990; Quatrone & Hopper 2001; Tondem By 2005). The first aspect presents the case for change and addresses the expected impact on the organisation (*why?*), while the second informs and facilitates the choice of method (*how?*). The framework is schematically represented by a tree structure. The “change” here signifies the organisational adoption of semantified web content and corresponds to the root node of the tree, while the two children nodes correspond to the organisational impact and method determination.

The rest of the tree structure is dictated by our research findings as follows:

The impact of semantic enrichment was found to be four-fold, having a direct effect on change and sustainability, information quality, innovation and, of course, the content itself. As a result the *Organisational impact* root has four children, one for each of these areas of influence.

The choice of method is determined by the actual form of semantic tagging and three lists of requirements addressing system design, format and integration issues. The *Method determination* root therefore branches out into four nodes that correspond to these requirements. The *Semantic tagging* root is subsequently divided into the *Semi-automatic* approach and the *Semantic APIs* root, which is further developed to provide a decision making framework informing the API adoption selection.

An overview of the proposed framework can be seen in Figure 6.1 below.

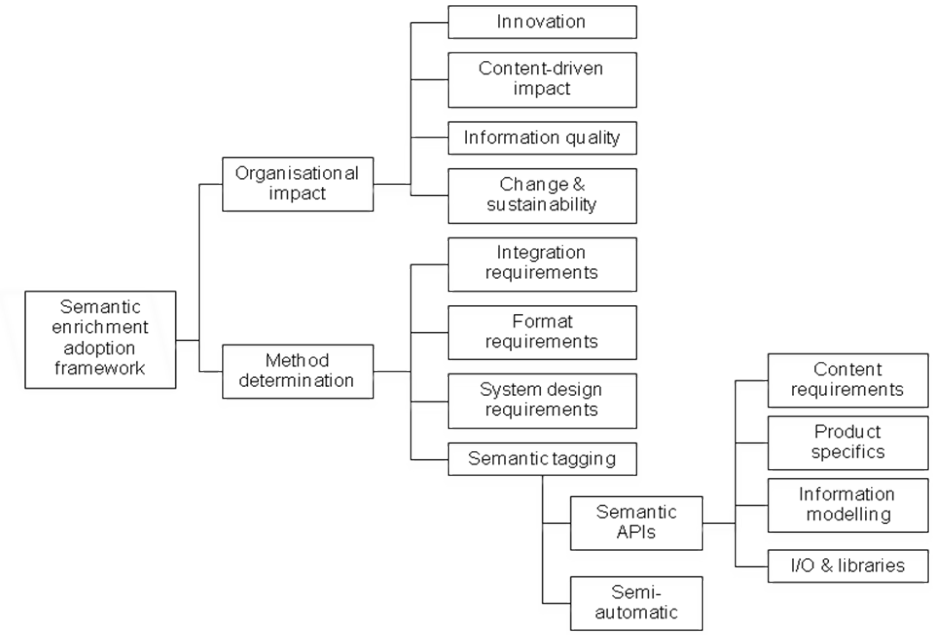


Figure 6.1 Framework for semantic enrichment.

6.2 The knowledge capture framework

The knowledge capture framework seeks to preserve semantic formality and enable interoperability, while harnessing end-user knowledge and organic annotation richness. As we saw, the tree structure represents the organisational adoption of semantified web content (tree root on the left of Figure 6.1). The first (top) branch corresponds to the impact of such adoption and the second (bottom branch) to the requirements-based method determination for information modelling and semantic enrichment. The twelve leaf nodes correspond to findings presented as tables in the body of the publications. These we will now visit in more detail.

The figure below expands the *Organisational impact* root into the four tables for *Innovation, Content-driven impact, Information quality* and *Change & sustainability*. The numbers next to each node correspond to an original table provided in the body of publications. The format is of the form section.group.table (e.g. “Innovation 5.3.2” corresponds to table 2 of the 3rd publication in section 5).

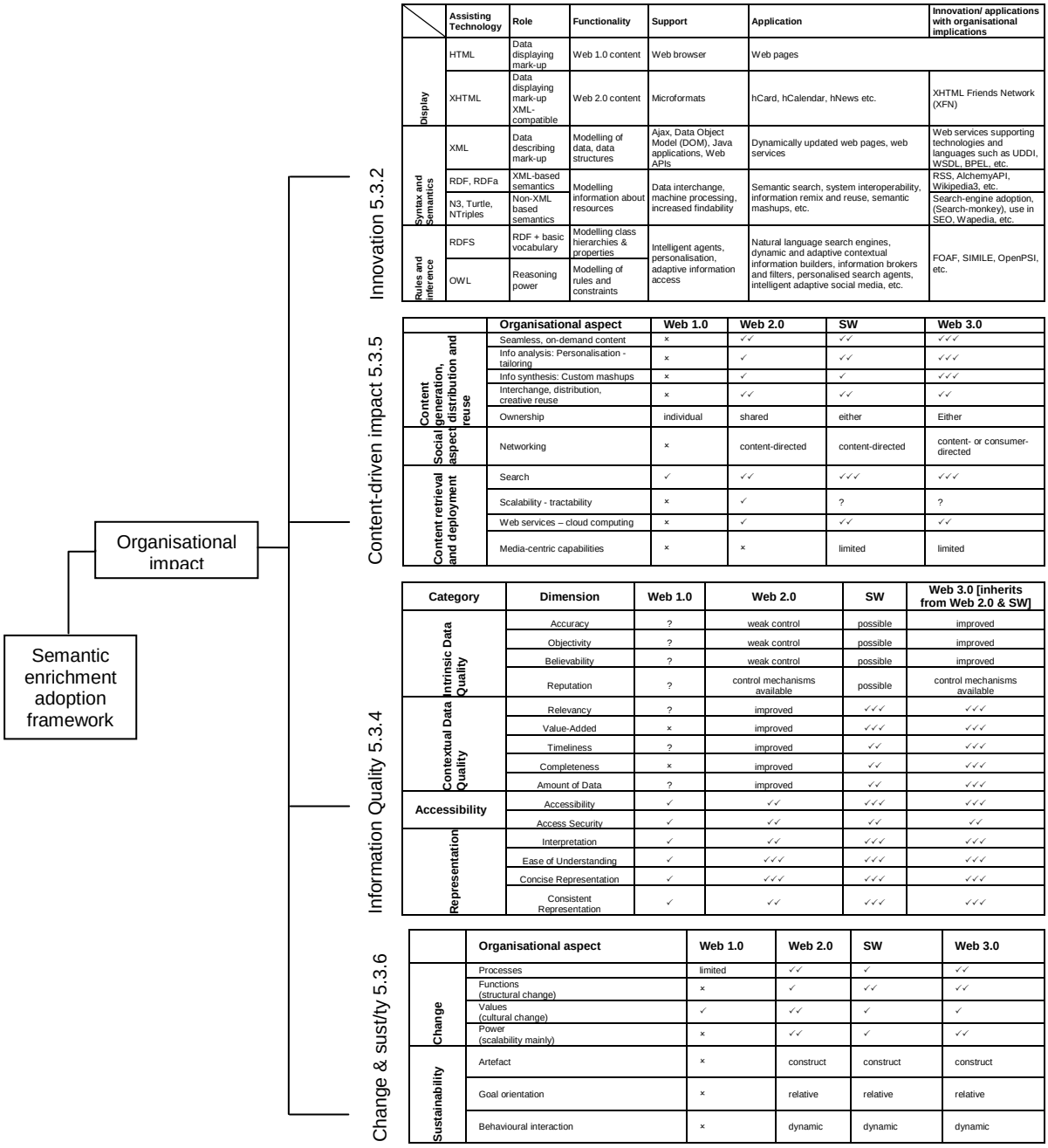


Figure 6.2. Root *Organisational impact*

The diagram provides a knowledge base that informs of the various impact aspects, supports the transition process and assists decision making. In particular:

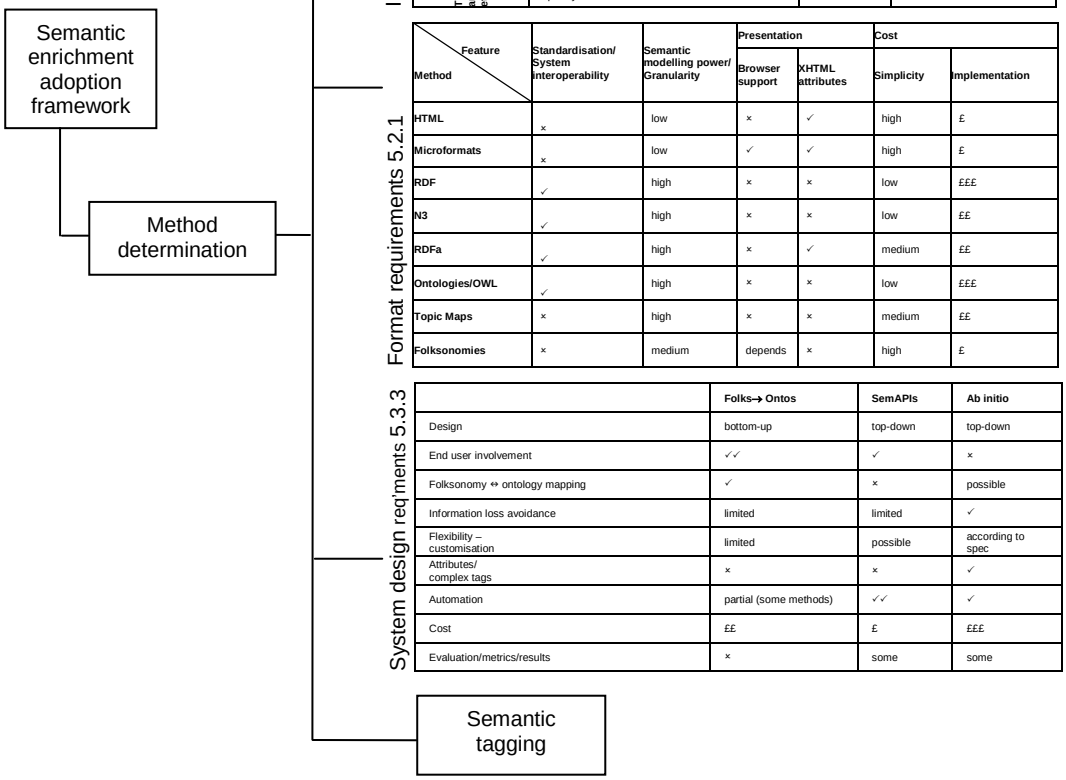
- The aspect of innovation focuses on technologies and related applications supporting semantic content innovation with organisational implications. Semantified content models enterprise information and processes with accuracy and consistency, enabling automatic reasoning, concept-based searches, process composition and knowledge discovery.

- Content-driven impact focuses on content generation, distribution, retrieval and deployment. Content generation displays considerably enhanced performance with distribution lagging behind and advanced automation enabling networking to be content- as well as consumer-directed. Cloud computing is clearly aided while media-centric capabilities remain limited.
- The information quality category has a direct impact on organisational success and profitability and focuses on contextual attributes (relevancy, value-added, timeliness, completeness and volume), representation (interpretation, ease of understanding, concise and consistent representation), accessibility/access security, and intrinsic data qualities (accuracy, objectivity and reliability).
- Organisational change focuses on the impact new technologies bring to organisational processes, functions, values and power and is found to be mostly dependent upon the use of web services and cloud computing. Sustainability is assessed following the underpinning aspects that analyse its conceptual developments (artefact, goal orientation and behavioural interaction). There is no evidence that semantic enrichment makes organisations more or less sustainable.

The *Method determination* branch is expanded in Figure 6.3. The table comprises three leaf nodes and a forth composite one (Semantic tagging) that is further expanded in Figure 6.4.

The method determination assists web content semantic enrichment by means of a modular design of requirements-based tools. Each table addresses a different aspect of the decision-making process, as follows:

- The integration requirements (categorised as pertaining to quality, semantic enhancement, mapping completeness and trust/ethics) focus on the identification of specific issues, highlighting the relevant domain and potential problems. Information modelling quality issues in particular are identified and distinguished from concerns about mapping and semantic clustering.
- The format requirements (requirements-based semantic format determination) address the question of semantic format adoption, offering a dashboard covering standardisation, modelling power, presentation particulars and cost constraints related to product and migration.
- The system design requirements act as a decision-making tool based on the design architecture, end-user involvement, automation, cost, evaluation process and issues of information loss, customisation requirements and granularity.



Requirement	Nature/Specifics	Domain/Focus	Potential problems
Quality issues	Concatenated tags (bankAccount) Variations (bank, banks, banking) Polysemes (the bank and to bank) Homonyms (bank and river bank) Synonyms (bank, deposit, pay-in)	Folksonomy	Lack of automated tools Ambiguity and inconsistency issues Intended vs. accidental granularity Non-proportionality between an organisation's semiotic system and the information system that describes it
Quality issues	Completeness Use of a basic taxonomic structure Overloading of the is-a relation Domain experts' guidelines	Ontology	
Semantic enrichment	Analysis and clustering Relation identification Attributes and properties	Folksonomy	Broad vs. narrow folks. Lack of sophisticated automated tools
Mapping completeness	Novel terminology Instances Multilingual tags	Folksonomy	Inability to map through
Mapping completeness	Specialised expert knowledge	Ontology	
Trust and ethics	Harvesting information across systems implicitly or explicitly connected	Social networks	Information used against the wishes of stakeholders.

Method \ Feature	Standardisation/ System interoperability	Semantic modelling power/ Granularity	Presentation		Cost	
			Browser support	XHTML attributes	Simplicity	Implementation
HTML	×	low	×	✓	high	£
Microformats	×	low	✓	✓	high	£
RDF	✓	high	×	×	low	£££
N3	✓	high	×	×	low	££
RDFa	✓	high	×	✓	medium	££
Ontologies/OWL	✓	high	×	×	low	£££
Topic Maps	×	high	×	×	medium	££
Folksonomies	×	medium	depends	×	high	£

	Folks→ Ontos	SemAPIs	Ab initio
Design	bottom-up	top-down	top-down
End user involvement	✓✓	✓	×
Folksonomy ↔ ontology mapping	✓	×	possible
Information loss avoidance	limited	limited	✓
Flexibility – customisation	limited	possible	according to spec
Attributes/ complex tags	×	×	✓
Automation	partial (some methods)	✓✓	✓
Cost	££	£	£££
Evaluation/metrics/results	×	some	some

Figure 6.3. Root *Method determination*

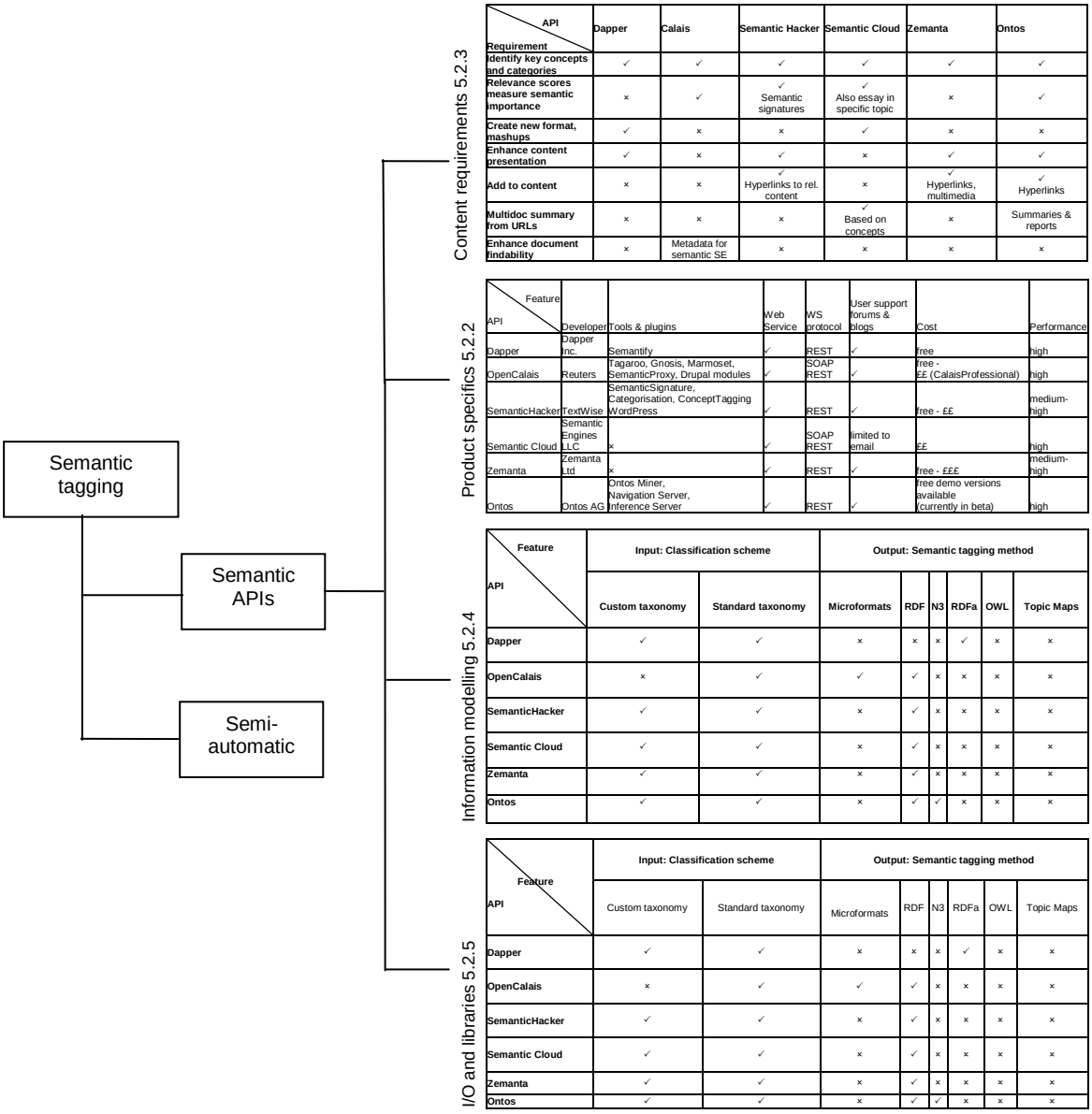
The *Semantic tagging* node is divided into two sub-branches.

The APIs branch assists the choice of the best suited interface for adoption. Decision-making is based on requirements relating to content, product specifics, information modelling, and associated formats. In particular:

- The first matrix assists requirements-based decision-making by listing content requirements opposite the systems that support them.
- The semantic APIs specifics' table provides product information and general characteristics such as developer, availability of extra tools, Web service information, online user support, cost and performance.
- Information modelling is based on the (existing/intended) API input and (required) output. The table facilitates the sorting of possible solutions and aids decision-making especially when a pre-specified output format is a system requirement.

- The last matrix presents a comprehensive listing of available input libraries along with the corresponding output formats that can be used in decision-making if a custom taxonomy is desirable but unavailable.

Figure 6.4 below expands the Semantic APIs sub-branch into its four leaf tables.



Requirement \ API	Dapper	Calais	Semantic Hacker	Semantic Cloud	Zemanta	Ontos
Identify key concepts and categories	✓	✓	✓	✓	✓	✓
Relevance scores measure semantic importance	×	✓	✓ Semantic signatures	✓ Also essay in specific topic	×	✓
Create new format, mashups	✓	×	×	✓	×	×
Enhance content presentation	✓	×	✓	×	✓	✓
Add to content	×	×	✓ Hyperlinks to rel. content	×	✓ Hyperlinks, multimedia	✓ Hyperlinks
Multidoc summary from URLs	×	×	×	✓ Based on concepts	×	Summaries & reports
Enhance document findability	×	Metadata for semantic SE	×	×	×	×

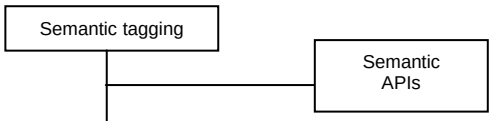
API \ Feature	Developer	Tools & plugins	Web Service	WS protocol	User support forums & blogs	Cost	Performance
Dapper	Dapper Inc.	Semantify	✓	REST	✓	free	high
OpenCalais	Reuters	Tagaroo, Gnosis, Marmoset, SemanticProxy, Drupal modules	✓	SOAP REST	✓	free - ££ (CalaisProfessional)	high
SemanticHacker	TextWise	SemanticSignature, Categorisation, ConceptTagging WordPress	✓	REST	✓	free - ££	medium-high
Semantic Cloud	Semantic Engines LLC	×	✓	SOAP REST	limited to email	££	high
Zemanta	Zemanta Ltd	×	✓	REST	✓	free - £££	medium-high
Ontos	Ontos AG	Ontos Miner, Navigation Server, Inference Server	✓	REST	✓	free demo versions available (currently in beta)	high

API \ Feature	Input: Classification scheme		Output: Semantic tagging method					
	Custom taxonomy	Standard taxonomy	Microformats	RDF	N3	RDFa	OWL	Topic Maps
Dapper	✓	✓	×	×	×	✓	×	×
OpenCalais	×	✓	✓	✓	×	×	×	×
SemanticHacker	✓	✓	×	✓	×	×	×	×
Semantic Cloud	✓	✓	×	✓	×	×	×	×
Zemanta	✓	✓	×	✓	×	×	×	×
Ontos	✓	✓	×	✓	✓	×	×	×

API \ Feature	Input: Classification scheme		Output: Semantic tagging method					
	Custom taxonomy	Standard taxonomy	Microformats	RDF	N3	RDFa	OWL	Topic Maps
Dapper	✓	✓	×	×	×	✓	×	×
OpenCalais	×	✓	✓	✓	×	×	×	×
SemanticHacker	✓	✓	×	✓	×	×	×	×
Semantic Cloud	✓	✓	×	✓	×	×	×	×
Zemanta	✓	✓	×	✓	×	×	×	×
Ontos	✓	✓	×	✓	✓	×	×	×

Figure 6.4. Root *Semantic APIs*

The *Semi-automatic* leaf node corresponds to the table in Figure 6.5. The table is an evaluation matrix for semi-automatic semantic enrichment methods. It provides a dashboard of potential requirements (relation analysis, tag cleaning and quality control, ontology mapping, attributes and complexity, multiple resource service, automation, developer support, evaluation and metrics), highlights availability or effectiveness and helps determine possible shortcomings.



Semi-automatic methods 5.1.1

Method / Feature	Van Damme et al.	Specia & Motta	Angeletou et al.	Kapetanios & Schaal	Lux & Dosinger	Baldoni et al.
Analysis to determine relations	✓	Partial	Partial (is-a)	N/A	✓	✓
Tag cleaning/quality control	Partial	×	×	N/A	✓	×
Ontology mapping techniques	✓	✓	✓	✓	✓	✓
Handling of attributes/ complex tags	×	×	×	N/A	×	×
Use of multiple resources	✓	×	Partial	✓	✓	×
Automation	Partial	×	Partial	✓	Partial	Partial
Worked examples/ Case studies	×	×	×	×	×	✓
Evaluation/metrics/ results	×	×	Some	×	Some	Some

Figure 6.5. Root *Semantic Tagging*

7 Conclusions, implications and future work

Web information technologies developed and applied in the last 10 years have considerably changed the way web applications operate and have revolutionised information management and knowledge discovery. Starting with the first applications that used XML encoding for the interchanging of data and going through the constant evolution of information modelling languages and their supporting technological frameworks, web information management has grown way beyond the hypertext linkage that Web 1.0 introduced.

Amid this, social technologies, user-generated classification schemes and formal semantics have a far-reaching sphere of influence. They promote collective intelligence, support interoperability, enhance sustainability, and instigate innovation with wide organisational implications that can benefit a considerable range of industries.

7.1 Implications for research and practice.

The thesis has implications both for researchers intending to further this work and practitioners planning to use the proposed framework.

Research implications. From a theoretical perspective, the research contributes to the enhancement of web information modelling and the overall understanding of the nature and significance of the semantic enrichment of web content. As it stands, the framework covers the semi-automatic methods and semantic APIs existing at the time of publication. Although it is unlikely that research in the area of converting folksonomies to ontologies in a semi-automatic way will realise any new methods, semantic APIs are becoming more popular and there will probably be new and/or enhanced products appearing. Whatever the advancement, the modularity of the framework makes adjustments possible and the modules can be easily updated with the particulars of the new methods/products. The relevant matrices and diagrams have to be followed for every fresh product evaluation and the outcome will provide immediate comparison results.

The APIs' performance is a grey area. It is mentioned in the *Product specifics 5.2.2* of *Figure 6.4.* and refers to the speed of semantic enrichment, as opposed to the speed of information retrieval. Performance has been difficult to establish and compare due to lack of consistent information. Company indicators do not compare like with like and performance measurements are relative and approximate. Queries to the developers did not yield any further detail. Another matter that makes

performance questions difficult is the differences in the actual functionality of the APIs, which makes comparisons on equal terms problematic.

One way of doing this is to create a unified access portal to the semantic APIs. This will be employed for testing the systems' responsiveness and reliability under a number of workloads and against a set of performance requirements so that the results can be used for analysis and comparison. This is an area for future research.

Practice implications. The implications for practice are centred on and represented by the proposed framework which formalises the adoption of semantified web content and aids decision making. The support for practitioners is two-fold.

The first part of the framework assists practitioners with evaluating the expected impact of the change on the organisation. The four aspects of the organisational impact are related to organisational information and knowledge, its quality, issues regarding change and sustainability, and the effect of adopting innovation-enabling technologies. In relation to organisational knowledge assets in particular, the method can be used to assess the influence of semantic enrichment on content generation, distribution, retrieval and re-use.

The second part of the framework assists practitioners in deciding the specific method of semantification to be followed. This choice is determined by a comprehensive list of requirements, from system design to format and integration issues, entry and exit states, tailoring and personalisation. Depending on the choice of method made further support is provided so that the optimum product can be selected.

Semantic technologies coupled with social media and end-user involvement can instigate innovative influence with wide organisational implications. The scalable and sustainable business models of social computing and the collective intelligence of organisational social media can be resourcefully paired with internal research and knowledge from interoperable information repositories, accounting systems, back-end databases and legacy systems. Semantified information assets can free human resources so that they can be used to better serve business development, support innovation and increase productivity.

7.2 Conclusions

The research carried out and consequent publications followed the various paradigms of semantic technologies that model information and the information

networks they generate. They assessed each approach, evaluated its efficiency and identified the challenges involved.

The resulting framework for web information modelling and semantic annotation can assist decision-making from the ground up and inform all stages of information assets transitioning to semantically enriched content. Alternatively it can be used as a best-match method between an entry state and a required exit outcome, especially when organisational requirements and existing constraints dictate a specific course of action.

The research presented aspires to make an impact not only by adding to the body of knowledge but by informing practice and contributing to successful evidence-based problem solving and decision making. It facilitates the development of closer links between the researcher and the practitioner and provides a bridge to integrating academia with work practice.

There is enough evidence to suggest that the next web generation, so-called Web 3.0, will be a hybrid mix of Web 2.0 technologies reinforced with semantic markup. Whether this markup is the formal, robust variety of the Semantic Web, or an automated, user-friendly approach that is easier to implement and better suited for organisational adoption, is yet to be seen. In either case, organisational transition to semantically enriched information, which is standardised to meet certain interoperability requirements, necessitates a framework that will facilitate decision making, support the changeover, assist the implementation and manage the impact.

This is the contribution this thesis makes.

REFERENCES

Abbott, R. (2004), ‘‘Subjectivity as a concern for information science: a Popperian perspective’’, Journal of Information Science, Vol. 30 No. 1, pp. 95–106.

Aberer, K., Cudre´-Mauroux, P. and Hauswirth, M. (2003), “Start making sense: the chatty Web approach for global semantic agreements”, Journal of Web Semantics, Vol. 1 No. 1, pp. 89-114.

Aberer, K., Cudre´-Mauroux, P., Ouksel, A.M., Catarci, T., Hacid,M.S., Illarramendi, A., Kashyap, V.,Mecella, M., Mena, E., Neuhold, E.J., Troyer, O.D., Risse, T., Scannapieco, M., Saltor, F.,deSantis, L., Spaccapietra, S., Staab, S. and Studer, R. (2004), “Emergent semantics principles and issues”, Database Systems for Advanced Applications, Springer Lecture Notes in Computer Science, 2004, Volume 2973/2004, 25-38

Allsop J. (2007), Microformats: Empowering Your Markup for Web 2.0, Friends of Ed, NY

Agosti, M., Melucci, M., (2001), Information Retrieval on the Web, *Lectures on Information Retrieval: Third European Summer School* (ESSIR 2000), M. Agosti, F. Crestani, and G. Pasi, eds., Springer, 2001, pp. 242–285.

Alani Harith, O'Hara Kieron, Shadbolt Nigel (2002) ONTOCOPI: Methods and Tools for Identifying Communities of Practice . In *Proceedings Intelligent Information Processing 2002*, Montreal - Canada

Antiqueira, L.; Graças, M.; Nunesm V.; Oliveira O. N.; Da F. Costa, L. (2007), “Strong correlations between text quality and complex networks features” Physica. A, 373, 2007. pp. 811-820

Ankolekar, M. Krötzsch, T. Tran and D. Vrandecic, (2007), The Two Cultures, Mashing up Web 2.0 and the Semantic Web, Proceedings of the 16th International Conference on World Wide Web Banff, Alberta, Canada (May 2007), pp. 825–834.

Avison, D.E., Baskerville, R. & Myers, M. (2001) Controlling action research projects. *Information Technology and People*, 14, 28–45.

Barini,C., and Scannapieco, M., (2006), “Data Quality – Concepts, Methodologies and Techniques”, Springer Verlag Berlin-Heidelberg, 2006.

Baskerville, R., Wood-Harper,A.T. (1998) Diversity In Information Systems Action Research Methods, European Journal of Information Systems, 7, 1998

Baskerville, R. (1999) Investigating information systems with action research, Communications of the AIS, 2, 1–32.

Beckett D. (ed) (2004) [online] RDF/XML Syntax Specification, http://www.w3.org/TR/rdf-syntax-grammar/ , accessed 24 July 2010

Benjamins, V.R., Contreras, J., Corcho, O. and Gomez-Perez, A. (2004), “Six challenges for the semantic web”, SIGSEMIS Bulletin, Vol. 1 No. 1, April, p. 2004

Berberich,K., Bedathur, S., Alonso,O., and G. Weikum. (2010) A Language Modeling Approach for Temporal Information Needs. In Proceedings of the 32nd European Conference on Information Retrieval Research (ECIR ’10), pages 13–25

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. Scientific American. May 2001.

Buckland, Michael (2011). What kind of science can information science be? Journal of the American Society for Information Science and Technology, Vol. 63 No.1, pp1-7.

N. Busi, R. Gorrieri, C. Guidi, R. Lucchi, and G. Zavattaro. Choreography and orchestration conformance for system design. In COORDINATION, volume 4038 of LNCS, pages 63.81, 2006.

Byron A., Berry A., Haug N., Eaton J., Walker J. and Robbins J., (2008) Using Drupal, O'Reilly Media, December 2008. ISBN-0-596-51580-4

Carey, T .T., and Mason, RE.A., "Information System Prototyping: Techniques,Tools and Methodologies, "*INFOR,* Volume 21, No . 3, August 1983, pp . 177-187 .

Checkland, Peter and Sue Holwell. 1998. “Action research: Its nature and validity.” Systemic Practice and Action Research, 11 (1), 9-21

Chen Pin-Shan,. (1976) The Entity-Relationship Model – toward a unified view of data, ACM TODS 1, No 1, March 1976

Chowdhury, G. G., & Chowdhury, S. (2003). Introduction to digital libraries. London: Facet Pub.

Cleverdon, C.W., Mills, J., and Keen, E.M. (1966). An inquiry in testing of information retrieval systems. (2 vols.). Cranfileld, U.K.: Aslib Cranfield Research Project, College of Aeronautics.

Codd, E.F., (1970), A relational model of data for large shared data banks, Commun. ACM 13, 6, June 1970, pp. 377-387

Colomb R.M. and Weber R. 1998, Proceedings of the *International Conference on Formal Ontology in Information Systems* (FOIS'98) Trento, Italy, 6-8 June, 1998. In N. Croft, W.B., Metzler,D., and Strohman, T., (2009), Search Engines: Information Retrieval in Practice. Addison-Wesley Publishing

Guarino (ed.) *Formal Ontology in Information Systems* IOS-Press (Amsterdam) pp. 207-217

Craswell, N., Hawking, D. 2009. Web information retrieval. In Information Retrieval: Searching in the 21st Century. Wiley, UK, 85–101.

Cudre´-Mauroux, P. and Aberer, K. (2004), "A necessary condition for semantic interoperability in the large", CoopIS/DOA/ODBASE, (2), pp. 859-872

Deependra Moitra , Jai Ganesh, Web services and flexible business processes: towards the adaptive enterprise, Information and Management, v.42 n.7, p.921-933, October 2005

Dodds, L. (2006), "SPARQLing services", Proceedings of the XTech 2006 Conference, Amsterdam

Domingue J. *et al.*, (2001) Supporting ontology-driven document enrichment within communities of practice. In *Proceedings 1st International Conference on Knowledge Capture (K-Cap 2001)*, Victoria, BC, Canada

DSDM, (1994), [online], Dynamic Systems Development Method Consortium, http://www.dsdm.org/ , accessed 20 July 2011

Enders, A., Hungenberg, H., Denker, H., Mauch, S., (2008), The Long Tail of Social Networking: Revenue Models of Social Networking Sites, European Management Journal, Volume 26 (3), June 2008, p. 199-211. 8 15.579

Fensel, D. (2001), Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce, Springer-Verlag, Berlin

Fernández,M., Cantador,I., López,V., Vallet,D., Castells,P., Enrico Motta, (2011) Semantically enhanced Information Retrieval: An ontology-based approach, Web Semantics: Science, Services and Agents on the World Wide Web, Volume 9, Issue 4, December 2011, Pages 434-452

Fred, A., Dietz, J.L.G, Liu, K., Filipe, J.,(Eds.) (2011), Knowledge Discovery, Knowledge Engineering and Knowledge Management, Communications in Computer and Information Science, Vol 128, 1st Edition, Springer.

GNOME (2000), The GNOME Project Community, http://www.gnome.org/about/ accessed 20 July 2011

Garcia-Molina, H. (2008) Web Information Management: Past, Present, Future. In ACM WSDM, Palo Alto, CA, 2008.

Hayman, Sarah (2007), Folksonomies and Tagging, New Developments in Social Bookmarking, Ark Group Conference: Developing and Improving Classification Schemes, Sydney June 2007

Hess, A., Maass, C. Dierick, F. (2008) From Web 2.0 to Semantic Web: a Semi-Automated approach, ESWC 2008 Workshop on Collective Semantics: Collective Intelligence and the Semantic Web (CISWeb 2008), Tenerife, Spain, 2008

Holzblatt, K. and Beyer, H. (1993), “Making customer-centered design work for teams”, Communications of the ACM, Vol. 36 No. 10, pp. 93-103.

Hughes, J.A., Randall, D. and Shapiro, D. (1992), “Faltering from ethnography to design”, Proceedings of the Conference on Computer-supported Co-operative Work: Sharing Perspectives (CSCW ’92), ACM Press, New York, NY, pp. 115-23.

Hult, M. and S. Lennung. (1980) “Towards A Definition of Action Research: A Note and Bibliography,” *Journal of Management Studies*, (17), pp. 241-250.

Intersystems (1996), [online], Intersystems Caché, http://www.intersystems.com/cache/index.html accessed 20 July 2011

Jepsen, L., L. Mathiassen and P. Nielsen (1989), "Back To The Thinking Mode: Diaries for The Management of Information Systems Development Projects,," *Behaviour and Information Technology*, (8) 3, pp. 207-217.

Jhingran, A., (2006), Enterprise information mashups: integrating information, simply, Proceedings of the 32nd international conference on Very Large Data Bases, Seoul, Korea 2006

Kaplan, B., Duchon, D., 1988. Combining qualitative and quantitative methods in information systems research: a case study.MIS Q. 12 _4., 571–586.

Kaplan, B. and Maxwell, J.A. "Qualitative Research Methods for Evaluating Computer Information Systems," in *Evaluating Health Care Information Systems: Methods and Applications,* J.G. Anderson, C.E. Aydin and S.J. Jay (eds.), Sage, Thousand Oaks, CA, 1994, pp. 45-68.

Kashyap V. 2003,*Trust and quality for Information Integration: The Data-Metadata-Ontology Continuum*, Workshop on Data Quality, Dagstuhl, Germany, September 2003

Kim, J., Lee, SH., Shin, M. S., (2008), Current usage of organisational blogs in the public sector, International Journal of Information Technology and Management 2008 - Vol. 7, No.2 pp. 201 – 216

King P.J.H, Poulovassilis A. (1988) FDL: A Language which integrates Databases and Functional Programming Actes du Congres INFORSID 88 pp 167-181

Knight S., J. Burn (2005). Developing a framework for assessing information quality on the world wide web. Informing Science, 8:159-172.

Li, Charlene; Bernoff, Josh (2008). Groundswell: Winning in a World Transformed by Social Technologies. Boston: Harvard Business Press

Macgregor, George, McCulloch, Emma (2006), Collaborative tagging as a knowledge organisation and resource discovery tool, Library Review, Vol. 55, Issue 5, pp 291-300

Madnick,S.E., Wang,R., Zhu H.,(2009). Overview and framework for data and information quality research. Journal of Data and Information Quality,1(1):1-22

Mai, J.E. (2004), ‘‘Classification in context: relativity, reality, and representation’’, Knowledge Organization, Vol. 31 No. 1, pp. 39–48

Mathiassen,L., (2002) Collaborative practice research, Information Technology & People, 15(4), 2002

McEneaney, J.E. (2001), “Graphic and numerical methods to assess navigation in hypertext”, International Journal of Human-Computer Studies, Vol. 55 No. 5, pp. 761-86

McIlraith, S.A., Son, T.C. and Zeng, H. (2001), “Mobilizing the semantic web with DAML-enabled web services”, Proceedings of the 2nd International Workshop on the Semantic Web,Hong Kong.

McKay, J. & Marshall, P. (2001) The dual imperatives of action research. *Information Technology and People*, 14, 46–59

McObject (2009), [online], Perst Product Website, http://www.mcobject.com/perst accessed 20 July 2011

Meloche, J. A., Hasan, H. M., Willis, D., Pfaff, C. & Qi, Y. (2009). Co-creating Corporate Knowledge with a Wiki. International Journal of Knowledge Management, 5 (2), 33-50.

Motta E., Buckingham-Shum, S. and Domingue, J. (2000). Ontology-Driven Document Enrichment: Principles, Tools and Applications. International Journal of Human-Computer Studies, 52, 1071-1109

Myers, M. D. (1999). Investigating information systems with ethnographic research. Communications of the AIS, 2(23), 1–20.

Myers, M.D. (2009) *Qualitative Research in Business & Management*. Sage Publications, London, 2009

Navarro-Prieto, R., Scaife, M. and Rogers, Y. (1999), “Cognitive strategies in web searching”, Proceedings of 5th Conference on Human Factors and the Web, July 1999

Ngwenyama, O. K., & Lyytinen, K. (1997). Groupware environments as action constitutive resources: A social action framework for analyzing groupware technologies. Computer Supported Cooperative Work: *The Journal of Collaborative Computing, 6*(1), pp. 71-93.

O’Reilly, T., 2005. [Online] What is Web 2.0? <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html> Accessed 15/10/10.

Orlikowski, W.J. & Baroudi, J.J. "Studying Information Technology in Organizations: Research Approaches and Assumptions", Information Systems Research (2) 1991, pp. 1-28

Paterson, J., Edlich, S., H¨orning, H., H¨orning, R.: The Definitive Guide to db4o. Apress, Berkely (2006)

Pettigrew A.M., (1990) Longitudinal Field Research on Change: Theory and Practice, Organization Science August 1990 vol. 1 no. 3 267-292

Pokorny, J., (2004) Web searching and information retrieval, Computing in Science and Engineering, Vol. 06, Issue 4, July-Aug. 2004 pp. 43-48.

Powell, T. Thompson, G., (2010), Electronic Patient Records: the roll-out of the Summary Care Records, House of Commons Library, Standard Note SN/SP/5601, June 2010

Quattrone, Paolo, Hopper, Trevor (2001), What does organizational change mean? Speculations on a taken for granted category, Management Accounting Research, Volume 12, Issue 4, December 2001, Pages 403-435

Rector A.L., Wroe C., Rogers J., Roberts A. 2001 *Untangling taxonomies and relationships: personal and practical problems in loosely coupled development of large ontologies*. K-CAP 2001: 139-146

Rudd, J., K. Stern, and S. Isensee. The Low vs. High-Fidelity Prototyping Debate. *Interactions, 3* (1): 76-85, 1996.

Schafer, J. B., Frankowski, D., Herlocker, J., Sen, S. (2007). Collaborative filtering recommender systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl (Eds.), The adaptive web: methods and strategies of web personalization, LNCS , Vol. 4321, pp. 291–324

Schein, E. (1969) Process Consultation: *Its Role in Organizational Development*, Reading, MA: Addison-Wesley.

Shipman, David, The functional data model and the data language DAPLEX, ACM Transactions on Database Systems (TODS), March 1981, Vol 6, Issue 1, pp. 140-173

Shuen, A., (2008) Web 2.0: A Strategy Guide Business thinking and strategies behind successful Web 2.0 implementations, O'Reilly Media, Inc

Smith, M.K., Welty, C., McGuinness, D.L. (2004) [online], OWL Web Ontology Language, http://www.w3.org/TR/owl-guide/ accesses 24 July 2010

Spaniol, M., Denev, D., Mazeika, A., Weikum, G., and P. Senellart., (2009) Data Quality in Web Archiving. In Proceedings of the 3rd Workshop on Information Credibility on the Web (WICOW '09), pages 19–26.

Steichen, B., Ashman,H., Wade, V., (2012) A comparative survey of Personalised Information Retrieval and Adaptive Hypermedia techniques, Information Processing & Management, Volume 48, Issue 4, Pages 698-724

Tanaka, K., Sato, J., Guo, J., Takada, A., andYoshihara, H., Cost accounting by diagnosis in a Japanese University Hospital. J. Med. Syst. 28(5):437–445, 2004.

Todnem By, Rune (2005), Organisational change management: A critical review, Journal of Change Management, Vol. 5, Iss. 4, 2005

Tursi,A., Panetto H.,Morel,G., M. Dassisti, (2009) Ontological approach for products-centric information system interoperability in networked manufacturing enterprises, Annual Reviews in Control, Volume 33, Issue 2, December 2009, Pages 238-245

Versant Corp. (2000), [online], Db4objects by Versant, Product Website, http://www.db4o.com accessed 20 July 2011

Virgilio, R. D., Giunchiglia, F., & Tanca, L. (2010). Semantic Web Information Management Springer, 2010.

Wang, W. and Zaıane, O.R. (2002), "Clustering web sessions by sequence alignment", Proceedings of DEXA Workshops, IEEE Computer Society, Los Alamitos, CA, pp. 394-8.

Wang, H., Huang, J. Z., Qu, Y., & Xie, J. (2004), Web services: Problems and Future Directions. Journal of Web Semantics, 1, 309-320

Weikum G.and Theobald M., (2010) From Information to Knowledge: Harvesting Entities and Relationships from Web Sources. In Proceedings of the 29th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems of Data (PODS' 10), pages 65–76, 2010.

Won Kim, (1990), Introduction to Object-Oriented Databases, Computer Systems, MIT Press, Cambridge, MA, 1990

Wood-Harper, T. (1985) "Research Methods in Information Systems: Using Action Research." in E. Mumford *et al.*, (eds.) Research Methods in Information Systems, Amsterdam: North-Holland, pp. 169-191

W3C 2008B, [online] , Uncertainty reasoning for the WWW, W3C Incubator Group report, http://www.w3.org/2005/Incubator/urw3/XGR-urw3-20080331/, accessed 15/10/2010

W3C RDF (2004), "Rescource Description Framework", http://www.w3.org/RDF/ accessed 20 Jul 2011.

W3C OWL (2004), "OWL web ontology language overview", www.w3.org/TR/owl-features/ accessed 20 Jul 2011.

Zaharieva, Z., Billen, R., (2009) "Rapid Development of Database Interfaces with Oracle APEX, used for the Controls Systems at CERN", ICALEPCS'09, Kobe, Japan, Oct-2009, THP108.

LIST OF PUBLICATIONS

Dotsika F., (2003) From data to knowledge in e-health applications: An integrated system for medical information modelling and retrieval, International Journal of Medical Informatics and the Internet in Medicine vol 28, issue 4, pp 231-251

Dotsika F., Watkins A., (2003a) An interoperable, graphical environment for the capturing of medical information, International Journal of Health Care Engineering, Technology and Health Care, Vol. 11, No 5

Dotsika F., Watkins A., (2003b) GISMoE: a Graph-based Information System Modelling Environment, Proceedings of the Conference on Internet and Multimedia.

Dotsika, F., Patrick, K., (2006) Towards the New Generation of Web Knowledge Search and Share, VINE: The Journal of Information and Knowledge Management Systems Vol. 36 No. 4, pp 406-422

Dotsika F., (2006), An IT Perspective on Supporting Communities of Practice, Encyclopaedia of Communities of Practice in Information and Knowledge Management, Coakes, E., & Clarke, S., (Eds), 2006, Idea Group Inc, pp 257-263

Patrick, K., Dotsika, F., (2007) Knowledge Sharing: Developing from Within, The Learning Organization: The International Journal of Knowledge and Organizational Learning Management, Vol 14, No 5, pp 395-406

Dotsika F., (2009) Uniting formal and informal descriptive power: Reconciling ontologies with folksonomies, International Journal of Information Management, Volume 29, Issue 5, October 2009, pp 407-415 *[2* in ABS list]*

Dotsika F., (2010) Semantic APIs: scaling up towards the Semantic Web, International Journal of Information Management, Volume 30, Issue 4, August 2010, pp 335-342 *[2* in ABS list]*

Dotsika F. (2012) The next generation of the web: an organisational perspective, Working Paper Series in Business and Management, 12-1, March 2012 (ISBN ONLINE: 978-1-908440-07-5)

LIST OF FIGURES

6.1 Framework for semantic enrichment…….…. 37
6.2 Root Organisational impact…….. 38
6.3 Root Method determination……... 40
6.4 Root Semantic APIs…….. 41
6.5 Root Semantic Tagging…….. 42

LIST OF TABLES

2.1 Methodology summary by group of publications……..…….... 12
4.1 Control parameters……..…….. 25

APPENDIX 1

LIST OF PUBLICATIONS

GROUP 1

Dotsika F., (2003) From data to knowledge in e-health applications: An integrated system for medical information modelling and retrieval, International Journal of Medical Informatics and the Internet in Medicine vol 28, issue 4, pp 231-251

Dotsika F., Watkins A., (2003a) An interoperable, graphical environment for the capturing of medical information, International Journal of Health Care Engineering, Technology and Health Care, Vol. 11, No 5

Dotsika F., Watkins A., (2003b) GISMoE: a Graph-based Information System Modelling Environment, Proceedings of the Conference on Internet and Multimedia.

GROUP 2

Dotsika, F., Patrick, K., (2006) Towards the New Generation of Web Knowledge Search and Share, VINE: The Journal of Information and Knowledge Management Systems Vol. 36 No. 4, pp 406-422

Dotsika F., (2006), An IT Perspective on Supporting Communities of Practice, Encyclopaedia of Communities of Practice in Information and Knowledge Management, Coakes, E., & Clarke, S., (Eds), 2006, Idea Group Inc, pp 257-263

Patrick, K., Dotsika, F., (2007) Knowledge Sharing: Developing from Within, The Learning Organization: The International Journal of Knowledge and Organizational Learning Management, Vol 14, No 5, pp 395-406

GROUP 3

Dotsika F., (2009) Uniting formal and informal descriptive power: Reconciling ontologies with folksonomies, International Journal of Information Management, Volume 29, Issue 5, October 2009, pp 407-415 *[2* in ABS list]*

Dotsika F., (2010) Semantic APIs: scaling up towards the Semantic Web, International Journal of Information Management, Volume 30, Issue 4, August 2010, pp 335-342 *[2* in ABS list]*

Dotsika F. (2012) The next generation of the web: an organisational perspective, Working Paper Series in Business and Management, 12-1, March 2012 (ISBN ONLINE: 978-1-908440-07-5)

MED. INFORM. (DECEMBER 2003) VOL. 28, NO. 4, 231-251

From data to knowledge in e-health applications: an integrated system for medical information modelling and retrieval

FEFIE DOTSIKA*

Department of Business Information Management and Operations,
Westminster Business School, University of Westminster
35 Marylebone Road, London NW1 5LS

Abstract. The system described in this paper uses the technological advances in information technology in order to influence and improve healthcare practice by enabling the flexible modelling, direct representation and adaptable use of medical knowledge. It aims at resolving a number of difficulties encountered by current information repositories, such as costly customisation, reusability, high maintenance and poor information modelling, by employing the architecture of the functional data model (FDM), while maintaining full interoperability with existing systems by means of XML. On the information-modelling front the system supports a variety of modelling techniques that are especially relevant to medical applications, such as complex objects, incomplete or missing information, partially structured data and multimedia content. A prototype implementation of the system has been developed which consists of a multimedia-enhanced version of the functional database language FDL, and a web-based, two-way translator interface between the application's native language and XML. This interface provides full interoperability with other, heterogeneous systems over the web, thus, significantly reducing the complexity of developing distributed healthcare systems and e-health applications.

Keywords: XML; data modelling; system interoperability

1. Introduction

Electronic applications such as e-medicine, e-commerce, e-education etc. can be thought of as backed up by three support categories: (a) *People*, including practitioners, customers and participating organisations, (b) *Public policy* such as legal issues, standards and regulations and (c) *Use and Distribution* including management and logistics. The infrastructure of these support groups can be divided into a general part, which deals with information distribution and the underlying network framework, and a second part, which deals with the knowledge infrastructure. Figure 1 below pictures a possible representation of this framework.

Among the above modules, our research focuses upon the knowledge infrastructure, and more specifically the back-end database management system, a software component that stores and manages the information relevant to the application and constitutes the heart of the operational knowledge for every e-application. Especially in the case of e-health and e-medicine, the knowledge repositories come in a great variety of shapes and packages, since the actual information is stored not only as traditional database records, but also as images, plain text, semi-structured or partially structured data.

Medical operational knowledge can be grouped by its *type* and *source*. The type of medical information varies from simple numeric and string-based data residing in traditional database

*Author for correspondence; Tel 020 7911 5000 ext 3027. E-mail: f.e.dotsika@westminster.ac.uk

Medical Informatics & The Internet in Medicine ISSN 1463-9238 print/ISSN 1464-5238 online © 2003 Taylor and Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/1463923031000161783 2

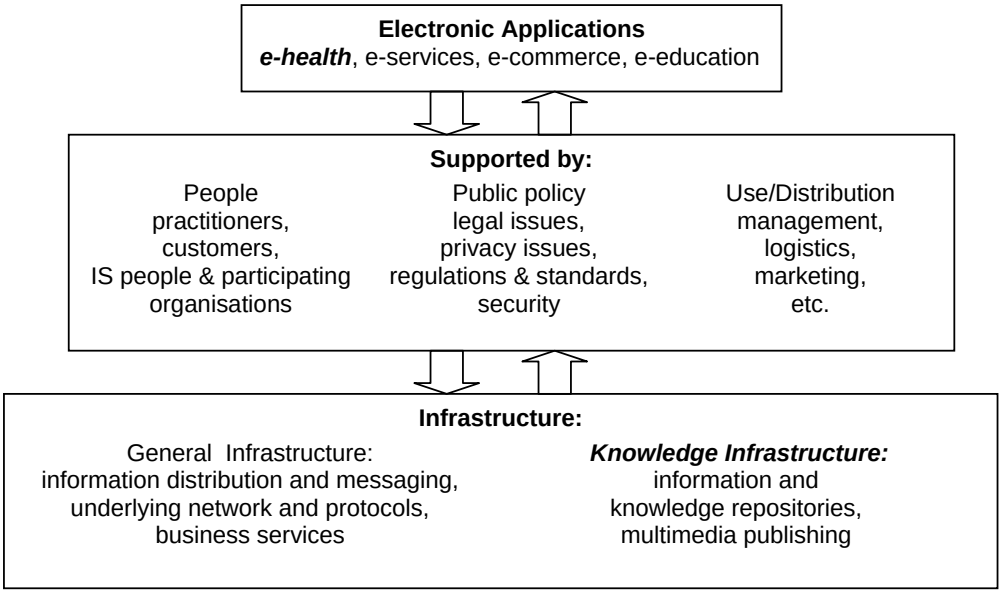


Figure 1: Electronic Applications Framework

systems, to text, graphics and image data. The source of this information can be as widespread as the web itself. More specifically, Internet based medical applications include electronic patient records, databases of clinical practice and literature, health portals, distance-learning type applications, decision-making tools for diagnosis and optimal treatment selection etc. Patients' Internet support groups and education packages revolutionise the traditional patient support, while terms such as *telemedicine* and *teleconsulting* (but also *cyberchondria)* find their way into our everyday lives. All of the above applications rely on the fact that it is easier and cheaper to move data than people and/or other resources.

Amid the different DB systems available, today's market is dominated by the aggregation based *relational* databases. Despite their commercial success however, conventional relational database systems lack the richness of conceptual models and cannot satisfy the special requirements of non-traditional, non-business-oriented database applications [1], such as medical applications and electronic healthcare.

Aiming at overcoming the problems and limitations of relational DBMS's, our approach follows the Functional Data Model (FDM) [2] for the modelling of medical information. The resulting system is persistent, by means of a back-end functional database. On the client interface front the system is designed to be fully XML compatible, adhering to XML's principal features of structure, extensibility and validation.

The rest of the paper is organised as follows. Section 2 describes the modelling of information and looks into various models highlighting their appropriateness for the task at hand, concentrating on the proposed model. Section 3 addresses issues of interoperability and content management by means of XML. Section 4 presents the implementation of the system, and finally, section 5, draws our conclusions and imparts future work.

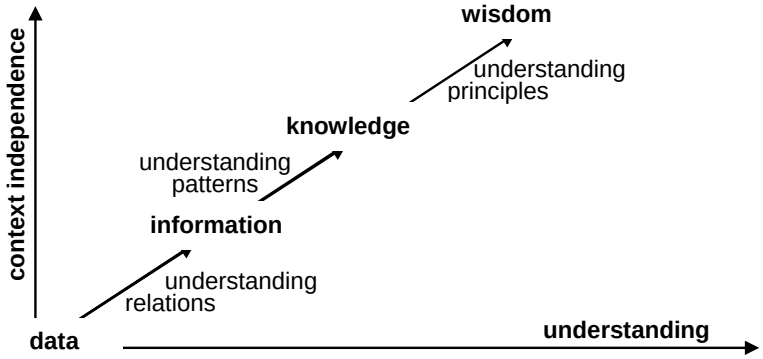


Figure 2: From Data to Wisdom

2. The modelling of information

It is an unfortunate fact that system developers often use the words data, information and knowledge interchangeably. Figure 2 provides a context representing the transition from data, to information, to knowledge and wisdom, according to G. Bellinger [3].

When working with the modelling of information, the challenge for the practitioner is to be able to go from knowledge to data and back with no information loss. The heart of such an undertaking is the model adopted.

Traditional database management systems (DBMSs) store data and information. Data is stored physically whereas information is modelled as relations among data. In this highly popular but conventional frame, knowledge is provided by means of techniques such as intelligent data analysis and data mining. Knowledge bases on the other hand store knowledge mainly in the form of keyword mark-up text. Although efficient in both capturing and conveying codified knowledge, knowledge bases can neither hold nor retrieve efficiently the sheer volume of information held in a traditional DBMS. In medical and e-health applications a combination of both systems is often deemed appropriate.

2.1.The relational model

The relational model appeared in 1970 [4] and stores data in (what is perceived by the user as) tables, each holding data about a particular theme. The rows represent instances and the columns represent attributes. Within each table the rows are uniquely identified by one special column, or a combination of columns, known as the *primary key*. The tables below depict the relational approach of a *Patient* database:

patient

pid	pname	GP	...
123	U.N.Well	54	...
345	C.S.Poorly	3	...

doctor

did	dname	...
12	Dr Who	...
54	Dr No	...

Although relational database products account for the lion's share in the market, they have specific inherent drawbacks that make their use outside the traditional business-oriented applications problematic:

- More often than not new relational applications are implemented from scratch, due to minimal reusability of code. Development tools that allow for re-use of program designs demand higher levels of skill and training and are not very widespread. Furthermore, changes in the schema of an already existing database - unless minor - result in the need to develop a new set of programs, whose development is time consuming and costly.
- Despite the essential need for detailed customisation, relational healthcare applications tend to be tailored to meet the needs of large numbers of users. Complex parameterisation results in crude application tuning and expensive upgrades.
- Non-business applications, such as electronic patient records need modelling based on complex objects such as component hierarchies, image data and structured texts. E-health applications should be able to represent complex objects directly and implement them effectively. Relational databases simulate complex object by joining relations, an approach that complicates modelling and results in performance problems.
- Healthcare applications use text, graphics and image archives. Therefore they should be able to model, store and manipulate extensive multimedia data efficiently, while still operating at a reasonable speed. Although relational design has no inherent impediment in supporting multimedia types, image, video and audio data structures are different from standard data and cannot be easily searched on a content base.
- Web-based medical applications require information retrieval from various sources, not necessarily based in the same location. Even when the modelling of information is done in the same way, merging data from different databases is proved to be impractical more often than not. When trying to merge data from two different tables for instance, chances are that there are differences in the structure of the tables to be merged, such as number of columns, data types etc.
- Linking to external knowledge bases has to be done as a separate, non-standardised task and seamless implementation is problematic. Moreover, interoperability between the two systems is achieved at the expense of performance.

2.2. The object oriented model

Object-oriented databases [5] were a result of the evolution of object-orientation during the 80's. Information is kept in the custody of an object, and cannot be directly accessed. Every object is an instance of a class. Retrieving or updating data is done by sending a message to the object involved and consequently invoking one of the object's methods. Figure 3 below shows the object-oriented version of the *Patient* database.

The object model of data was originally developed to provide persistent storage for CAD programs, and has proved to be enduringly popular for this type of applications [6]. Moreover, object-oriented approaches were especially promising for use in database technology, as objects can support complex objects directly and can represent behavioural knowledge by means of methods.

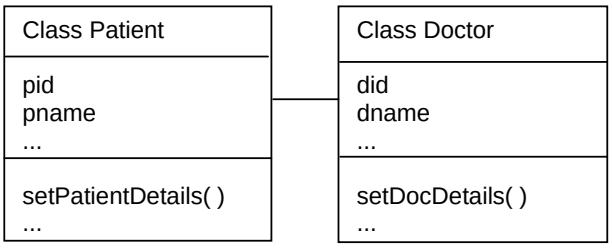


Figure 3. An O-O patient database.

However, the object database market never really took off despite the high commercial expectations. Some of the characteristics of the model proved to be drawbacks in real life applications, especially over the Internet.

- In certain aspects of query and transaction processing the OO approach never proved a match to the relational model. The concept of encapsulation in particular creates a processing overhead when populating or querying the database. The reason for this is that each and every retrieval is preceded by an object method invocation, a process that makes updates and transactions cumbersome.
- Due to encapsulation, there is less granularity in an object database than in a relational db. Small numbers of large objects are more efficiently stored than large numbers of small ones, a framework that far from favours the idiosyncrasies of medical applications.
- Web-based applications tend to require lightweight technologies whose components can be distributed across the Internet and can function equally well on PC's, servers, network computers etc. However, in trying to adapt and also keep up with the relational model, the object approach has become complex and heavy.
- Linking to external knowledge bases creates similar problems as before.

2.3. The object-relational model

The popularity of object orientation led the relational database manufacturers to consider a hybrid model, which would bring together the best qualities of both systems. The resulting products were described as *object/relational,* though they do not really represent a new model, but are in fact relational databases with borrowed object-oriented features.

As a result of their relational nature, the object-relational databases typically support SQL. They also support complex objects, as a result of their object-oriented nature. Nonetheless they still fall short of important features when it comes to information modelling:

- With a disparity of concepts from different paradigms, the resulting model is a mismatch rather than a genuine model.
- Most object-relational products lack a conceptual model. With the exception of the model of Date and Darwen [7], object/oriented proposals implement conflicting modelling techniques, which are neither conceptual nor consistent.

- ✓ Regardless of its conceptual model, Darwen's specification does away with SQL, a language whose popularity is relational products' best selling agent.

2.4. The functional model

The functional approach emerged as one of the different flavours of the *semantic* data models [8], from a requirement for more conceptual information modelling. *Semantic nets* emphasize semantics and have been widely used in AI for representing meaning. They are the logical forms that state relationships between persons, things, attributes and events. The concept nodes represent entities, attributes, states and events. The relation nodes show how the concepts are interconnected.

Unlike the relational model which is record founded, FDM is based on graphs, and as such it provides a finer semantic granularity, which facilitates data modelling. Unlike relational databases there is no need for normalisation, as the schema is normalised by default.

According to the functional data model functions can be used to define the aggregation of attributes used to form an entity. A binary relationship R(A,B) defines the functions

F: A -> Set (B)

G: B -> Set (A)

However in practice most binary relationships are one-to-many rather than many-to-many so that the two functions can be viewed as F and its inverse G:

F: A -> B

G: B -> Set (A)

Entities are identified by an entity identifier, generated by the system. Non-identifiable entities have only values that can be of type string, integer, float, date and boolean. Identifiable entities are divided into *IB-* and *OB-* elements. IBentities (IB for *In-Base*) are traditional functional database elements, whereas OBentities (OB for *Out-Base*) are external elements that are managed by the DB system by means of their identifiers. They can be ASCII or binary files, text based knowledge sources, image data and other multimedia types.

Entity semantics are also associated with *classification*. We first define an IBentity called *patient* and then populate our database by creating three new instances of patients *p1*, *p2* and *p3*. By *patient* we now mean both the entity and the set of patients. The following query:

All_patient;

returns the list of all patients held currently in the database [$p1,$p2, $p3].

Generalisation can be supported explicitly, by creating super entities and modelling the generalisation hierarchy via *isa* relations.

Figure 4 below pictures the schema of a patient database. As the database is used for demonstration purposes, many simplifying assumptions were made in the presentation of the diagram depicting the schema.

Square shapes correspond to IBentities, record-shaped forms depict OBentities and oval shapes represent base type entities. One-to-one relationships are pictured as single-head arrows

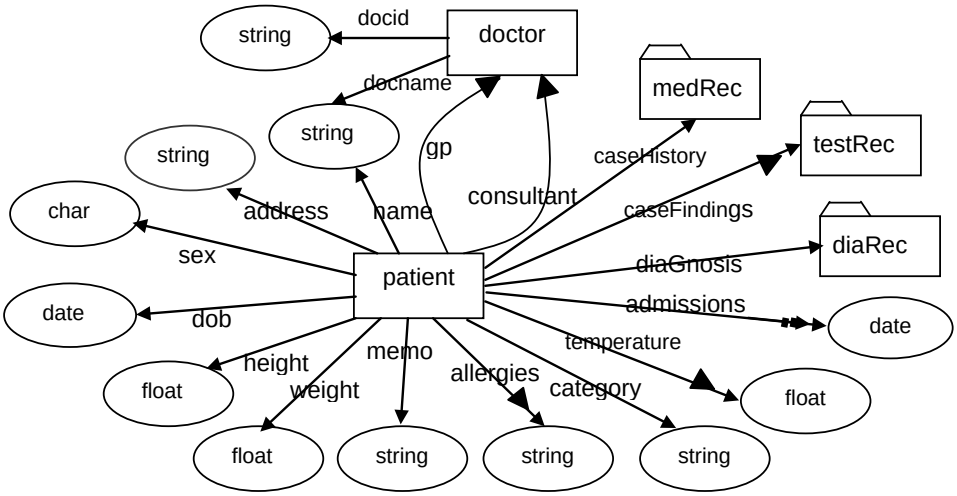


Figure 4: The Patient database schema

one-to-many with double-head arrows and there is one-to-many 2-tuple function that models patient admission information. For simplicity, relations are modelled using nouns rather than verbs. Therefore the relation *called* linking a *patient* to their *name* is modelled as the function *name* that takes a *patient* entity as its argument and returns a *string* (which is in fact the name of the patient). The following table provides the schema definition and the generation of an instance for patient *C.S.Poorly*.

The database holds *data* of type *integer* (such the range of the function *age*), *string* (such as the range of the function *name*) etc. Data however, provides no information. The decimal numbers 37.5, 38.7 etc. for instance have no actual meaning until we define the *temperature* relation between a *patient* and a list of *floats*, modelled here as an one-to-many function. Information such as *name*, *age*, *height* etc. is modelled using functions whose domain is typically an IBentity (such as patient, doctor) and whose range is either a base-entity, or another IBentity.

2.5. The FDM appropriateness for the task at hand

Taking up the various shortcomings of the models we mentioned earlier, we can now give a list of the functional model's advantages, and highlight its simplicity of use and appropriateness for e-medical applications.

- Modelling information is done in a simple, conceptual way, with no need for expensive normalisation procedures and expert care. Although coding the database schema requires expertise due to the required coding-language fluency, the end user can still participate actively in the model specification.
- Complex data structures are supported, allowing the use and manipulation of complex objects and multimedia content, the two most important elements of medical data. In the above example the OBentities *medRec*, *testRec* and *diaRec* are such examples. The data types *medRec*, *testRec* and *diaRec* are of multimedia content. The particular patient has a *medRec* of type *text* where information (and/or knowledge) of the case history is held. The *testRec* contains two elements, one of type *text* and one of type *image* (e.g. X-ray results).

Schema definition	Instance example
patient :: IBentity;	create patient $p;
doctor :: IBentity;	
medRec :: OBentity;	create medRec $pMRec txt;
testRec :: OBentity;	create testRec $pT1 img;
diaRec :: OBentity;	create testRec $pT2 txt;
name: patient -> string;	name $p <= “C.S.Poorly”;
address: patient -> string;	address $p <= “1, Narrow Lane, London SE1, UK”;
sex: patient -> char;	sex $p <= ‘M’;
age: patient -> integer;	age $p <= 25;
height: patient -> float;	height $p <= 1.78;
weight: patient -> float;	weight $p <= 75.5;
allergies: patient -> [string];	allergies $p <= null;
category: patient -> string;	category $p <= “in patient”;
docname : doctor -> string;	
docid: doctor -> string;	$cons = head inv_docid [123];
gp: patient -> doctor;	
consultant: patient -> doctor;	consultant $p <= $cons100;
admission: patient -> {date, date};	admission $p<= {1/9/02, null};
temperature: patient -> [float];	temperature $p <= [37.5, 38.7, 38.2, 38.2];
caseHistory: patient -> medRec;	caseHistory $p <= $pMRec;
caseFindings: patient -> [testRec];	caseFindings$p<= [$pT1, $pT2];
diaGnosis: patient -> diaRec;	

- There is no need for distinguishing between string data types of fixed or variable length. Although relational databases support varying length columns and use them to save space, they often do it to the expense of the performance of update operations. In FDM, data of type string have a variable length that can be a few characters long, or span multiple lines such as the *address* element in the example without affecting performance.
- On the same theme, missing or incomplete information is also efficiently handled, without the need to record *null* values, unless such recording is semantically important. The patient in this example has no known allergies, therefore the query

  allergies $p;

  returns null. Similarly, there has been no diagnosis so far. However, assuming that there will be a diagnosis later on, this information is incomplete rather than unknown. Therefore, should the function *diaGnosis* be applied to our patient:

  diaGnosis $p;

  the result will be unknown and not null. This is done by not assigning an explicit value to *diaGnosis $p*, in which case the database returns *unknown* as a default.

- Unlike new relational applications that need a new set of application programs every time, functional databases provide the flexibility of code reusability. There is only one language used both as DDL and DML. Consider the following example:

```
hospital_bill: patient -> float;
hospital_bill x <=
        hospital_stay x + operation_costs x +
        prescription_costs x;
```

- The function *hospital_bill* (defined as above "extensionally", contrary to the *name*, *age,* etc. functions that were "intentionally" defined) can be re-used even if the database schema is to change, as long as there is an entity equivalent to *patient*. Further coding of the functions *hospital_stay, operation_costs* and *prescription_costs* however may be schema specific.
- Combining of similar database schemas is also possible, by means of the *equivalency* operation. The example below shows the equivalency operation for a base type entity.

```
decimal_number == float;
comment == string;
```

- In the same way, we can extend database schemas by using the equivalency operation between abstract entities:

```
person == patient;
specialist == doctor;
```

- User views are easily tailored to particular groups of users and implemented without the expensive parameterisation and customisation involved in relational databases. The following function creates a view for Dr Who, whose *docid* is 255:

```
Dr_Who's_Patient_List <=
        [ x || x <- All_patient & 255 = docid consultant x];
```

The above view is created assuming that each doctor needs access to his or her own set of patients. The query itself is formulated as a list abstraction, and it makes use of the in-built function All_patient which, in its generic form All_entity, returns all objects of a kind.

Similarly, the function hospital_bill defined earlier may be relevant to the accounts department but not to the specialist. Equally, the accounts department employees should not be able to access sensitive data such as the patients records and medical history. Customisation can be easily implemented by a login procedure. The users of the database log in with a user name and a password and they download the schema relevant to their group privileges.

All updates to the database are immediately reflected in all relevant views. Likewise, updates carried out in a view are also reflected in the database. Since the database can handle incomplete information, there are no integrity problems, unlike in the case of relational databases.

3. Interoperability and content management.

Medical information exchange over the Internet is governed by the compatibility (or lack of) among a variety of information storage media. Figure 5 pictures a typical web environment for information exchange.

In view of the diversity of architectures and implementations the feature of compatibility has been of high priority as it guarantees the interoperability between our system and other existing e-health applications. The rest of this section investigates the fundamental schema transformation framework that grants full interoperability between our system and other medical data banks over the Internet.

3.1. The role of XML in data description and validation

The eXtensible Markup Language [9] is a mark-up language like HTML and as such they are both subsets of the Standard Generalised Markup Language (SGML). Unlike HTML, it allows users to create their own mark-up tags for virtually any type of information. While HTML was designed to *display* data and focus on how data looks, XML was designed to *describe* data and focuses on what data is, separating thus the content from its presentation. Its principal features are extensibility, structure and validation. Also, it contains no formatting instructions and can be parsed easily.

Validation is handled by means of a DTD (Definition Type Document) or a XML Schema and a *validating* parser that checks whether a XML document conforms to a given DTD. By means of the DTD (or XML schema) independent groups of people can agree to use a common format for interchanging data, verify the validity of either received or own data and provide an application-independent way of sharing and exchanging information. As a consequence, XML documents have proved to be highly portable and allow for information modelling and extensive data manipulation, attributes especially relevant to medical record applications [10, 11, 12].

The fact that XML is currently used as a standard framework for data exchange over the Internet, has been recognised by software developers worldwide, who have proceeded by integrating XML into their applications in order to gain Web functionality and interoperability between heterogeneous knowledge banks.

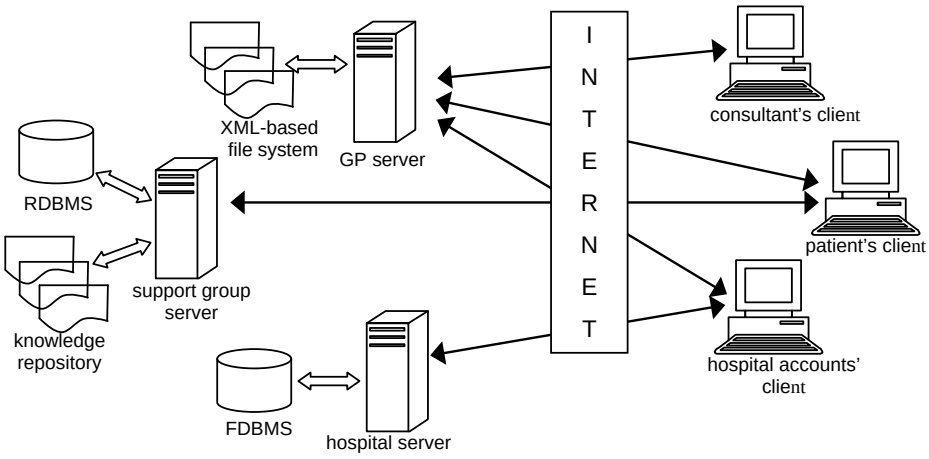


Figure 5: Information exchange on the Internet

Since most existing relational or object-relational medical systems in use are XML compatible, a two-way translator between XML and the functional model was considered necessary [13, 14].

Furthermore, the issue of validation was deemed especially relevant to medical applications [10, 15], as it allows for agreement on a common format for interchanging data among independent sources. To that effect we sought a DTD representation that has been already used successfully in data exchange between heterogeneous databases, but is flexible enough to allow for the inclusion of updates and queries. The DTD used was the GNOME project's XML queries proposal [16] (Figure 6).

3.2. Schema and query transformation

Although the XML translator grants our system interoperability with all other XML-compliant systems, we have chosen to follow the transformation examples between the relational and the functional architectures, mainly due to the market dominance of the RDBMSs.

```
<!ELEMENT QUERY (TARGET?, SOURCES?, VALUES, QUALIFICATION?)>
<!ATTLIST QUERY op(SELECT|CREATE|INSERT|UPDATE|DELETE) #REQUIRED>
<!ELEMENT TARGET (TABLE|VIEW)*>
<!ELEMENT SOURCES (TABLE|VIEW)*>
<!ELEMENT VALUES (CONST|QUERY|FIELD|FUNC)+>
<!ELEMENT QUALIFICATION
(AND|OR|NOT|EQUAL|NONEQUAL|INF|INFEQ|SUP|SUPEQ|NULL|LIKE|CONTAINS)*>
<!ELEMENT TABLE (#PCDATA)>
<!ATTLIST TABLE id ID #IMPLIED
          temp(yes|no) #IMPLIED
          alias CDATA  #IMPLIED>
<!ELEMENT VIEW (#PCDATA)>
<!ATTLIST VIEW id ID #IMPLIED
          alias CDATA #IMPLIED>
<!ELEMENT CONST (#PCDATA)>
<!ATTLIST CONST printname #IMPLIED>
<!ELEMENT FIELD (#PCDATA)>
<!ATTLIST FIELD source IDREF #REQUIRED
          name CDATA #REQUIRED
          printname #IMPLIED
          group(yes|no) #IMPLIED>
<!ELEMENT FUNC (FIELD|CONST|FUNC)*>
<!ATTLIST FUNC name CDATA #REQUIRED
          printname #IMPLIED>
<!ELEMENT AND
(AND|OR|NOT|EQUAL|NONEQUAL|INF|SUP|SUPEQ|NULL|LIKE|CONTAINS)+>
<!ELEMENT OR
(AND|OR|NOT|EQUAL|NONEQUAL|INF|SUP|SUPEQ|NULL|LIKE|CONTAINS)+>
<!ELEMENT NOT
(AND|OR|NOT|EQUAL|NONEQUAL|INF|SUP|SUPEQ|NULL|LIKE|CONTAINS)>
<!ELEMENT EQUAL ((CONST|FIELD|FUNC),(CONST|FIELD|FUNC))>
<!ELEMENT NONEQUAL ((CONST|FIELD|FUNC),(CONST|FIELD|FUNC))>
<!ELEMENT INF ((CONST|FIELD|FUNC),(CONST|FIELD|FUNC))>
<!ELEMENT INFEQ ((CONST|FIELD|FUNC),(CONST|FIELD|FUNC))>
<!ELEMENT SUP ((CONST|FIELD|FUNC),(CONST|FIELD|FUNC))>
<!ELEMENT SUPEQ ((CONST|FIELD|FUNC),(CONST|FIELD|FUNC))>
<!ELEMENT NULL (CONST|FIELD|FUNC)>
<!ELEMENT LIKE ((CONST|FIELD|FUNC),(CONST|FIELD|FUNC))>
<!ELEMENT CONTAINS ((CONST|FIELD|FUNC),(CONST|FIELD|FUNC))>
```

Figure 6: The GNOME project DTD

Based on the DTD above the XML interface makes it possible for the client to send queries in a generic form, without having to know the architecture of the underlying DBMS. The XML code example below can be translated into a *create patient table* with attributes *name, address, sex* and *age* for a relational system (the current GNOME query specification does not support the CREATE operation, so the DTD had to be extended accordingly). The operation was simplified by omitting the type definitions, and assuming the default ones. Alternatively, it can be translated into a *create entity patient,* with functions *name, address sex* and *age* for a functional system.

```
<QUERY op = “create”>
    <SOURCES><TABLE id= “t01”>patient</TABLE></SOURCES>
    <VALUES>
        <FIELD source = “t01” name= “name”/>
        <FIELD source = “t01” name= “address”/>
        <FIELD source = “t01” name= “sex”/>
        <FIELD source = “t01” name= “age”/>
    </VALUES>
</QUERY>
```

The databases can then be populated in a similar way:

```
<QUERY op = “insert”>
    <SOURCES><TABLE id= “t01”>patient</TABLE></SOURCES>
    <VALUES>
        <CONST> “C.S.Poorly”</CONST>
        <CONST> “1, Narrow Lane, London SE1 UK” </CONST>
        <CONST> “M”</CONST>
        <CONST> 25 </CONST>
  </VALUES>
</QUERY>
```

Obvious as it may be, it is worthwhile observing that the XML formatted output is the same result one would get, from either database. As a result, from the end-user’s point of view the resulting output is independent of the database itself and dependent only on an agreed specification of an interchange format.

Query transformation is done in the same manner. Consider the following SQL query:

```
select name, age
from patient;
```

It is an example of the relational operation *projection*. The same query in the functional model is formulated as the list abstraction:

```
[ {name x, age x} || x <- All_patient ];
```

Using the GNOME project XML queries proposal, we now convert the SQL query into its equivalent functional one and vice versa. Following the project’s DTD, the query description is divided into 2 distinct parts: <sources> and <values>.

```
<QUERY op = “select”>
    <SOURCES><TABLE id = “t01”>patient</TABLE></SOURCES>
    <VALUES>
        <FIELD source= “t01” name = “name”/>
        <FIELD source= “t01” name = “age”/>
    </VALUES>
</QUERY>
```

The first part (<sources>) provides the field of the *from* clause in SQL (from patient), or the *All_* part in FDL (All_patient). In similar way, the second part provides the field which corresponds to the table column in the relational, or the function in the functional model.

The second example shows the SQL version of the *restriction* relational operation:

```
select *
from patient
where name = “C.S.Poorly”;
```

In the functional model the *where* clause is appended to the list abstraction as an “&” (AND) qualifier. Note the in-built *inverse* function (inv_name).

```
[{name x, address x, age x, sex x} || x <- All_patient
    & “C.S.Poorly” = head inv_name];
```

According to our DTD, the restriction operation example in XML would be:

```
<QUERY op = “select”>
    <SOURCES><TABLE id = “t01”>patient</TABLE></SOURCES>
    <VALUES>
        <FIELD source= “t01” name = “name”/>
        <FIELD source= “t01” name = “address”/>
        <FIELD source= “t01” name = “age”/>
        <FIELD source= “t01” name = “sex”/>
    </VALUES>
    <QUALIFICATION>
        <EQUAL>
            <FIELD sorce= “t01” name = “name”/>
            <CONST>“C.S.Poorly”</CONST>
        </EQUAL>
    </QUALIFICATION>
</QUERY>
```

The above XML formulated query is divided into 3 distinct parts: <sources>, <values> and <qualification>. The first (<source>) and second (<field>) parts are used as before. The third

part (<qualification>) provides the components of the *where* clause in SQL and the "&" qualifier in the functional list abstraction.

The last query returns the GP name for every patient. The SQL version deploys the *theta-join* operation (in this specific case it is the equijoin).

```
select name, dname
from patient, doctor
where patient.pid = doctor.pid;
```

The functional version would be:

```
[{name x, docname y} ||
        x <- All_patient & y <- All_doctor          & y = gp x ];
```

The XML version of the same query would be:

```
<QUERY op = "select">
        <SOURCES>
                <TABLE id = "t01">patient</TABLE>
                <TABLE id = "t02">doctor</TABLE>
        </SOURCES>
        <VALUES>
                <FIELD source= "t01" name = "name"/>
                <FIELD source= "t02" name = "docname"/>
        </VALUES>
        <QUALIFICATION>
                <EQUAL>
                        <FIELD source= "t01" name = "pid"/>
                        <FIELD source= "t02" name = "did"/>
                </EQUAL>
        </QUALIFICATION>
</QUERY>
```

The third part in this example provides the components of the equijoin condition, or the final condition in the list abstraction.

Once again it is worth pointing out that all the above XML data representations could have been derived from either database model. We have therefore shown that both the output of the relational and that of the functional databases can be formatted to conform to the specified DTD, with no information loss.

3.4. User views and reports

Suppose that we want to create a user view of Dr Who's patients' details. The relational view would be created as:

```
create view DrWho'sList as
select name, address
from patient, doctor
where patient.pid = doctor.pid
        AND doctor.dname = "Dr Who";
```

For simplicity, but without affecting the general case (i.e. select *), only the *name* and *address* fields have been selected whereas the rest have been suppressed. The equivalent functional view is defined as:

```
DrWho'sList = [{name x, address x}
        || x <- All_patient & docname consultant x = "Dr Who" ];
```

Employing the same DTD as before, the XML code would then be:

```
<QUERY op = "create">
    <TARGET><VIEW>DrWho'sList2002</VIEW></TARGET>
    <SOURCES><TABLE id= "t01">patient</TABLE></SOURCES>
    <VALUES>
        <QUERY op = "select">
            <SOURCES>
                <TABLE id = "t01">patient</TABLE>
                <TABLE id = "t02">doctor</TABLE>
            </SOURCES>
            <VALUES>
                <FIELD source= "t01" name = "name"/>
                <FIELD source= "t01" name = "address"/>
            </VALUES>
                <QUALIFICATION><AND>
                    <EQUAL>
                        <CONST>Dr Who</CONST>
                        <FIELD source= "t02" name = "dname"/>
                    </EQUAL>
                    <EQUAL>
                        <FIELD source= "t01" name = "pid"/>
                        <FIELD source= "t02" name = "did"/>
                    </EQUAL>
                </AND></QUALIFICATION>
        </QUERY>
    </VALUES>
</QUERY>
```

Although not, strictly speaking, part of data manipulation, reports have proved to be popular with e-medicine applications, and as such they were deemed a viable component of the current project. Contrary to queries and views, reports were not supported by the existing DTD. An extension of the QUERY to include the (optional) element REPORT was needed. Figure 7 shows the basic extension of the document type definition to cover reports. The transformation of a report example between the two databases by means of XML and the extended DTD was considered trivial.

4. The implementation

A prototype system following the above design and specification has been implemented by means of an extended version of the Functional Database Language (FDL) as the back-end functional database and a schema dependent XML compatible web interface.

```
<!ELEMENT QUERY (REPORT?, TARGET?, SOURCES?, VALUES, QUALIFICATION?)>
<!ELEMENT REPORT (FORMATTING?, BREAK?, COMPUTE?)
<!ELEMENT FORMATTING (TITLE, PAGESIZE?, LINESIZE?, COLSFORMATTING?)>
<!ELEMENT TITLE (TTITLE | BTITLE)>
<!ATTLIST TTITLE (left | right) “right” >
<!ATTLIST  BTITLE (left | right) “right” >
<!ELEMENT PAGESIZE #PCDATA>
<!ELEMENT LINESIZE #PCDATA>
<!ELEMENT COLSFORMATTING (COLNAME, HEADING?, FORMAT)>
<!ELEMENT COLNAME #PCDATA>
<!ELEMENT HEADING #PCDATA>
<!ELEMENT FORMAT #PCDATA>
<!ELEMENT BREAK #PCDATA>
<!ATTLIST BREAK skip CDATA #IMPLIED>
<!ELEMENT COMPUTE(AGRFUN+, COLNAME, COLNAME)>
<!ELEMENT AGRFUN #PCDATA>
```

Figure 7: Report DTD extension

FDL [17,18] is implemented over a persistent, semantic, free software triple store [19] in which all information is held. The triple store contains ordered 3-tuples comprising fixed-length internal identifiers, and the lexical token converter maps such identifiers to external printable representations. It has a small set of primitive instructions and provides FDL with complete persistence for all type and function declarations and for the function defining equations.

The database consists of the function definitions. A function is defined by its type declaration and a set of equations specifying its value for the various possible values of its argument(s). The type and function declarations can be regarded as the database schema. These declarations may be introduced or deleted at any time, subject only to minimal constraints to ensure that the database remains well defined.

As the current version of FDL does not fully support multimedia types, the language had to be enhanced with a facility that incorporates the OBentity identifiers into the database, providing the users with the function *retrieve* which triggers the relevant application for displaying each entity's particular multimedia type. For example, suppose that the query:

```
caseFindings $p1;
```

returns the list:

```
[$sc0, $sc1, $sc2]
```

which contains two files of type image (i.e. the X-rays $sc0, $sc2) and one video stream (the file $sc1). In order to view the contents of these files the user has to type in the following queries:

```
retrieve  $sc0;
retrieve  $sc1;
retrieve  $sc2;
```

The function retrieve invokes an image viewer for cases $sc0 and $sc2 and a video streaming application for $sc1.

The implementation of the prototype translator between the functional database and XML has been done by means of a server-based Perl/cgi script running on a SPARC Enterprise 450 Solaris 8 platform which supports an Apache Web server (chosen for its open source and for allowing safe connections of medical data through SSL). The web interface itself is application dependent. Following a slightly cut-down version of the schema of the toy database introduced in section 2.4, the interface of figure 8 was constructed as an example.

The left hand screen corresponds to the *Patient Details Insertion Form* and the right hand is the *Patient Records Queries Form*.

The top of the *Details* form provides the choice of the back-end functional database and the type of update that can be either a new record or an existing one. If the same interface is required for accessing a relational system as well as the functional database, a choice of architecture may be supplied. The user supplies as many fields as are available and the rest are left blank. There is a browsing facility for the files of multimedia content. The user needs to supply the filename along with the relation between the patient and the multimedia file (pull-down menu of choices). In this particular form layout there is the possibility of inserting three multimedia files and four known allergies. Any further entries would have to be inserted via a second, *existing record update* screen. The example screen corresponds to the insertion update of section 3.2 . The Perl/cgi receives the user input and creates the XML code. As the back-end database, in this case, is a functional one, the XML to FDL module will then be activated and will in turn generate the following FDL :

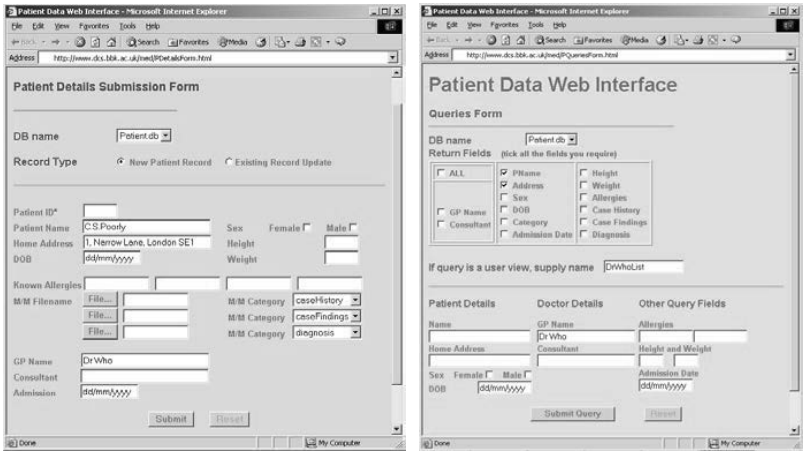


Figure 8 The Patient Details and Record Queries Forms

```
create patient $p;
name $p <= "C.S.Poorly";
address $p <= "1, Narrow Lane, London SE1";
sex $p <= 'M';
dob $p <= 01/01/1978;
```

Regarding the system's interoperability, had a RDBMS been at the back-end, the XML-SQL module would have been triggered, resulting in the generation of the equivalent SQL. The screen that follows informs the user that the update was successful or, if there has been any problem, displays the error message.

The *Query* form provides the choice of database and of returned fields. If all information is required, the user can tick the *All Fields* checkbox. When the submit button is pressed, the Perl/cgi script receives and formats the query input into XML. Although not part of the example query screen, comparison, logic and arithmetic operators can also be implemented, so can queries employing specific, extensionally defined FDL functions that are already part of the schema.

The example screen corresponds to the XML-formatted Dr Who's patient list of section 3.4. The query is consequently translated and passed on to FDL and the output is then fed back to the script, which formats it again into XML. Although the code can be altered to conform to a given DTD, at the moment the XML output follows a generic format with tags based on function names. Once on the client side, the output can be saved or fed to a local database.

Figure 9 shows the screen corresponding to the above query.

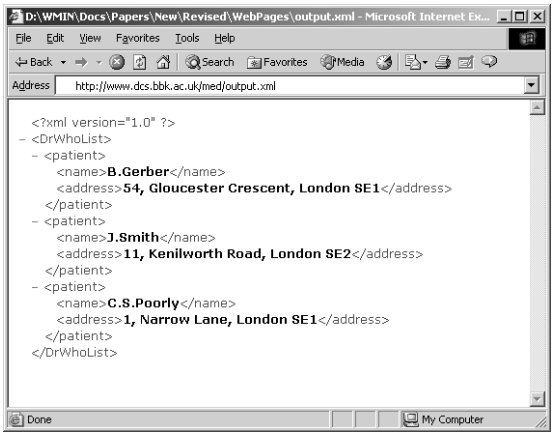


Figure 9 The query output screen

The prototype has been tested with data borrowed from other applications and the web and various interface screens have been created. In all cases the performance of the system was satisfactory, though the volume of data and transactions in the testing environment were medium to low.

5. Conclusions and further work

The research described in the previous sections contains the detailed workings, from design to implementation, of a prototype system that continues and extends a short paper presented at Mednet2002 [20]. The work carried out adds to the on-going exploration of medical knowledge management, focusing in particular on information modelling and retrieval of e-health applications over the world wide web. Based on a number of problem areas encountered, a system was sought that would successfully address and resolve the highlighted trouble issues, but would, at the same time, communicate and co-operate if necessary, with already existing software.

The proposed functional architecture was consequently chosen as it tackles the previously mentioned inadequacies of current products in information modelling, amplifies multimedia content, and ensures cost-effective customisation, code re-use and low maintenance. Information modelling is thus facilitated, permitting the collaboration of the developer, the practitioner and the end user in the modelling process.

The use of XML in connecting heterogeneous knowledge repositories and databases was extended to include the developed system, via a two-way translator interface allowing for interoperability with a variety of existing products. The developer is therefore allowed the choice, deployment, and merging of different models to fit particular circumstances.

However, dealing with real life e-health applications presents challenges not necessarily encountered when testing our prototype system with trial data. Complex cases and considerably large volumes of information may need a finer tuning on each and every level. Based on that, our future research plans are three-fold: (a) enhancement of the main system engine, (b) expansion of the XML interface, and (c) testing. In more detail:

(a) On the main system front, the following aspects are currently being explored: the addition of constraints, the possibility of supporting ad-hoc queries over the web and the broadening and further integration of accessing the knowledge sources.

The current system evokes *knowledge entities* residing in external files, which can be then viewed or possibly queried via keyword search. In spite of treating KB sources the same way as other external entities, more investigation is needed in order to provide a seamless integration and also improve and diversify the methods of access.

Besides the *retrieve* operation (see section 4), a *load* operation will be added, which will enable the local storing of information. Apart from the obvious advantages of making the information locally available, *load* could effectively enhance performance and network traffic, by replacing a long and unwieldy interaction with the server with a short interaction followed by local processing of information based on image, video and other multimedia content.

(b) The inherent coarseness of the data definition document may lead to inept information modelling and the possibility of ambiguity issues. Despite the conceptual simplicity of the GNOME project's query proposal DTD, it is arguable that one would need a more varied specification for an interchange format employed by a particular group of health organisations. Therefore, although DTDs are widely used for data format interchange specification, from the functional database point of view, XML Schemas may prove to be more appropriate, as they support primitive types. Use of XSLT and the more advanced XML features such as Xlink and XPointer need also be investigated.

(c) Finally, vigorous testing of the system with appropriate, real life data needs to be carried out, and performance tests need to take place. This further testing will determine whether the cgi framework is effective and efficient, or needs to be replaced by a more scalable alternative. Although we are at present searching for an appropriate set of data, it is certain that a variety of sources - rather that a single supply - of information will be required in order to achieve an optimum analysis and assessment of the system described.

References

1. D.S. Batory et al., Implementation concepts for an extensible data model and data language, ACM Transactions on Database Systems (TODS), September 1988, Vol 13, Issue 3, pp.232-262
2. David Shipman, The functional data model and the data language DAPLEX, ACM Transactions on Database Systems (TODS), March 1981, Vol 6, Issue 1, pp. 140-173
3. Gene Bellinger, Durval Castro, Anthony Mills, Data, Information, Knowledge and Wisdom, http://www.outsights.com/systems/dikw/dikw.htm, 2000
4. E.F. Codd, A relational model of data for large shared data banks, Commun. ACM 13, 6, June 1970, pp. 377-387
5. Won Kim. Introduction to Object-Oriented Databases**,** Computer Systems, MIT Press, Cambridge, MA, 1990
6. H. Ishikawa et al, The model, language and implementation of an object oriented multimedia knowledge base management system, ACM Transactions on Database Systems, Vol 18, No 1, March 1993, pp. 1-50
7. C.J. Date, H. Darwen, Foundation for Object Relational Databases: The Third Manifesto, Addison-Wesley , May 1998
8. Joan Peckham , Fred Maryanski, Semantic data models, ACM Computing Surveys (CSUR), September 1988 Vol 20 Issue 3, pp.153-189
9. Bray T., Paoli J., Sperberger-McQueen (W3C) Extensible Markup Language (XML) 1.0 http://www.w3.org/TR/2000/REC-xml-20001006 06/10/2000
10. Simon Hoelzer et al., Value of XML in the implementation of clinical practice guidelines - the issue of content retrieval and presentation, Medical Informatics and the Internet in Medicine, 2001, Vol 26, No 2, pp131-146, April 1, 2001
11. Carol Friedman et al, Representing Information in Patient Reports Using Natural Language Processing and the Extensible Markup Language, Journal of American Med. Inform. Assoc., Vol 6 No1 pp76-87, 1999
12. Richard N. Shiffman et al. GEM: A Proposal for a More Comprehensive Guideline Document Model Using XML, Journal of American Med. Inform. Assoc. 2000, 7(5): 488-498
13. Dotsika F., Watkins A., XML and Functional databases, Proceedings of the IASTED Intelligent Systems and Control Conference ISC2001, pp 242-246
14. Dotsika F., Watkins A., Integrating Web-Based Information Systems: WWW and the Functional Model, Proceedings of the IASTED Conference on Internet and Multimedia Systems and Applications IMSA2002, pp 74-80
15. M. Dugas, K. Kuhn, N. Kaiser, K. Überla, XML-based visualisation of design and completeness in medical databases, Journal of Medical Informatics and the Internet in Medicine, Vol 26, No 4, October 2001, pp 237 - 250
16. GNOME project, The XML Query Specification, GNOME-DB database integration, http://www.gnome-db.org/docs/white-papers/xml-queries.php
17. King P.J.H, Poulovassilis A. FDL: A Language which integrates Databases and Functional Programming Actes du Congres INFORSID 88 pp 167-181

18. Poulovassilis A.: FDL: an integration of the functional model and the functional computational model, Proc. BNCOD-6, 1988, pp 215-236
19. M.Derakhshan, A Development of the Grid File for the Storage of Binary Relations, *PhD Thesis,* Department of Computer Science, Birkbeck College, University of London, 1989
20. Dotsika F., Modelling medical operational knowledge for e-health applications, MEDNET2002, International Journal of Health Care Engineering, Technology and Health Care, Vol. 10, No 6, 2002, pp474-476.

[15]

An interoperable, graphical environment for the capturing of medical information.

Fefie Dotsika
Department of Business
Information Management and Operations
Westminster Business School
University of Westminster
E-mail: dotsikf@westminster.ac.uk

Andrew Watkins
School of Computer Science
and Information Systems
Birkbeck College
University of London
E-mail: andrew@dcs.bbk.ac.uk

Introduction

From pathology, diagnostics and treatment, to patient history and lifestyle, medicine is a true science of information. As medical information is growing, its management and utilisation becomes more challenging. While the current generation of electronic healthcare applications keeps on multiplying, doctors, patients and medical administrators are faced with the task of choosing the right application that will enable them to find and use the relevant information at the right time.

Resulting from the recent experimental deployment of functional database management systems for the storage, manipulation and retrieval of medical information [1, 2], MedISD (*Med*ical *I*nformation *S*ystem *D*esign) has been developed, a web-based, graphical, information modelling environment, which enables practitioners to model their own custom-made healthcare information systems. The development of MedISD was deemed necessary following the agreement for the trial use of the system with NHS primary healthcare data.

MedISD focuses on improving healthcare practice by enabling custom schema modelling, direct representation and flexible use of medical knowledge, and support of metadata and multimedia content. The aim of the system is thus to significantly reduce the complexity of developing medical information systems, from primary healthcare data pools to distributed e-health applications. No technical knowledge or database expertise is required apart from basic desktop environment skills. The tool captures information in the form of directed graphs and automatically generates tailor-made medical database schemas based on the functional data model. The system supports complex objects, user views and it is further integrated by providing an XML interface that allows for interoperability with other databases and medical knowledge repositories in general.

Material and methods

The architecture of MedISD is modular. After analysing the healthcare administrators requirements, the graphical environment was designed to provide the following components: (a) the *model visualisation panel* where the user can edit and manipulate the primary data objects as well as the relations among them, (b) the *information capture component*, where the edited schema is translated into the entities and binary relations of the underlying database, (c) the *data dictionaries*, (d) the (explicit) *schema manipulator* for extensionally defined functions that might be added by the more sophisticated user or administrator, and (e) the (automatic) *schema generator*.

On the information-modelling front the system had to be able to provide both global and user views and offer a variety of modelling techniques that are especially relevant to medical applications, such as the support of incomplete or missing information, partially structured data and entities of multimedia content.

Appreciating the paramount importance of interoperability in electronic healthcare, the environment was designed to be fully integrated and interoperable by means of an XML interface [3], which is linked directly to the schema generator component. Depending on the user choice, the schema generator can create either the database schema, or the equivalent XML DTD. This interface provides MedISD with the ability to communicate with other medical information repositories of alternative architectures. Apart from the obvious benefit of compatibility, it addresses the issue of validation, a service especially relevant to medical applications, as it allows for agreement on a common format for interchanging data among independent sources.

Results

MedISD has been implemented using Java 2 Platform, Standard Edition (J2SE), version 1.4.1 on both Solaris 9 and Windows 2000 Operating Systems. It has been tested with all major revisions of Java since Java 1.3, so that it works on all platforms with the relevant Java Virtual Machine. Java was chosen because it is architecture independent, provides portable user interface, and can enable loading on demand of the application front end as an applet over the web. Furthermore, it is well suited for distributed object computing with CORBA or Java/RMI.

MedISD supports the functional data model and is equipped with a two-way translator between XML and the underlying functional schema, providing facilities for the automatic generation of valid XML documents and DTD's. Figure 1 presents a snapshot of the system.

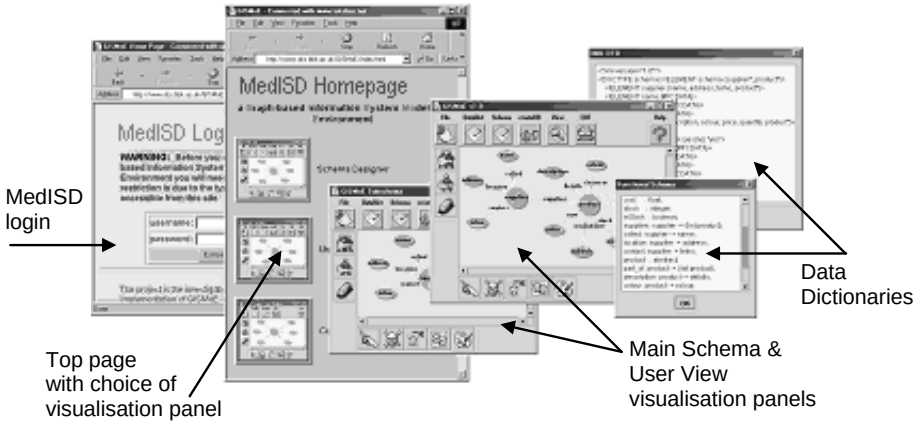


Figure 1: MedISD snapshot

The system has been tested with data borrowed from other applications and the web and its performance has been particularly satisfactory. However, dealing with real life e-health applications presents challenges not necessarily encountered when testing the system with trial data. Complex cases and considerably large volumes of information may need a finer tuning on each and every level. Although MedISD will be soon be used with NHS primary healthcare data, it is certain that a variety of

sources - rather that a single supply - of information will be needed in order to achieve an optimum analysis and assessment.

Conclusions

A great part of the success of medical information management relies upon the effective modelling and efficient maintenance of data that is relevant to the medical transaction processing and the clinical decision-making. Based on this principle, the research outlined in this paper enables e-health to follow in the steps of other disciplines whose application developers have been using graphical information modelling tools for a number of years. MedISD was designed to provide health practitioners with a tool that models information quickly and effortlessly, generates the relevant schema, creates the corresponding back-end database and allows for interoperability and communication between the application's native functional db server and other information repositories currently in use. The tool's modelling flexibility can hide technical complexity from the end-user group (typically consisting of healthcare administrators with basic IT application skills but no technical background) while enabling the more sophisticated type of practitioner to model information explicitly, via a number of advanced modelling implements.

References

[1] Dotsika F., *Modelling medical operational knowledge for e-health applications*, International Journal of Health Care Engineering, Technology and Health Care, Vol. 10, No 6, 2002, IOS Press, Amsterdam.

[2] Dotsika F., '*From data to knowledge in e-health applications: an integrated system for medical information modelling and retrieva*l' to appear in the International Journal of Medical Informatics and the Internet in Medicine.

[3] Dotsika F., Watkins A., *Integrating Web-Based Information Systems: WWW and the Functional Model*, Proceedings of the IASTED Conference on Internet and Multimedia Systems and Applications IMSA2002.

Proceedings of the 7th IASTED International Conference
Internet and Multimedia Systems and Applications
August 13-15, 2003, Honolulu, Hawaii, USA

GISMoE : A GRAPH-BASED INFORMATION SYSTEM MODELLING ENVIRONMENT

Fefie Dotsika[1], Andrew Watkins[2]
[1] University of Westminster, Department of Business Information Management and Operations,
dotsikf@wmin.ac.uk
[2] University of London, Birkbeck College, School of Computer Science and Information Systems,
andrew@dcs.bbk.ac.uk

Abstract

The aim of this paper is the investigation, design and implementation of GISMoE, a web-based automated environment for information modelling, based on the functional paradigm. The resulting system develops a user-friendly, interactive graphical interface that assists the systems analyst and designer in developing interoperable information systems solutions and facilitates data and information modelling. GISMoE supports the functional data model and generates functional database schemas, maintains up-to-date data dictionaries and creates new databases based on the designed user models. The environment is fully integrated and interoperable by means of a two-way translator between XML and the underlying functional schema, providing facilities for the automatic generation of valid XML documents and DTD's.

Keywords: Web and internet tools and applications, databases and the web, data modelling, Java technology and applications, information systems.

1 Introduction

Successful data management relies upon the effective modelling and efficient maintenance of data that is relevant to transaction processing and decision-making. While data maintenance is the task mainly undertaken by the database management system itself, data modelling is the responsibility of the systems designer. The growing demand for new information systems that cover an ever-expanding variety of application fields along with the need to maintain existing systems assure that data and information modelling score high among the list of IS expertise.

For many years application developers have been using graphical tools that automatically generate significant quantities of code. Although the efficiency, scope and completeness of the generated code varies from product to product, they increase productivity and have thus remained popular with the information systems professionals.

The research carried out introduces GISMoE, a web-based, interoperable, information system modelling environment that automatically generates functional database schemas, user views and their XML equivalent. The aims of the tool are to (a) considerably facilitate information modelling, providing an efficient and quick response to environmental changes, (b) supply a central, up-to-date data dictionary, a functional schema and the corresponding XML DTD based on the user design, (c) support the creation of functional databases and customised user views, and (d) act as two-way translator between the functional database schema and XML.

The creation of the system was deemed necessary for the advance and further continuation of two research projects that are presently using functional database management systems. One project lies on the area of e-medicine and is a collaboration between the local health authorities and the authors. It deals with the modelling, storage, retrieval and manipulation of medical data of both plain and multimedia content [1, 2]. The second project investigates the use of functional databases with crime data, the modelling of all relevant information and the retrieval, identification and analysis of possible crime clusters. Both projects need a tool that models information quickly and effortlessly, generates the relevant functional schemas and allows for interoperability and communication between the functional database servers and other information repositories currently in use. The crime cluster analysis project in particular requires frequent re-modelling of the information that comes from cid reports, victim statements etc., a task that can prove to be too cumbersome if no functional database expertise is available at top level. The developed system bypasses these concerns by allowing the user to model information by means of directed graphs and automatically generates the database schema that corresponds to the designed diagram. It further simplifies modelling by supporting complex objects, sub-schemas and user views. It is assumed that the user is aware of the basic *binary relational* schema concepts (as opposed to the n-ary relations, generally concerning the relational model) and is fairly comfortable working in a desktop environment.

The rest of the paper is organised as follows: section 2 deals with the basics of information and data modelling, and contains a brief synopsis of different methods and data models. Section 3 looks into issues of interoperability among heterogeneous systems. Section 4 introduces the new system and presents its usage and section 5 draws our conclusions and outlines future work.

2 Information modelling environments

With *relational* databases taking up the lion share in the market, data modelling techniques concentrate on the development of tabular schemas. The *relational model* stores data in (what is perceived by the user as) tables, each holding data about a particular theme. The rows represent instances and the columns represent attributes. Techniques used as an aid to relational database modelling include the non-loss decomposition method of *normalisation* [3], the *entity relationship* model [4] and the *semantic object* [5] method.

The following example shows the relational representation of a P*roduct-Supplier* database with three tables, containing information about the supplier, the products and the supply respectively:

Supplier

Sno	Name	Address	Telno
123	Ash	1,Sea Rd	1234
555	...	...	...

Product

Pno	Price	InStck	Description	Colour	Partof
231	12	Y	bolt	green	234
234	...	...	...	...	

Supply

Sno	Pno	Qty
123	234	3
123	231	...

Following the techniques mentioned above, a number of application development modelling tools is available, which provide system designers with data modelling assisted database design. Most leading database vendors provide data modelling tools: Oracle's most comprehensive and widely used products are the *Designer Environment* [6] (for relational) and *Object Database Designer* [7] (for object-relational modelling). Sybase offers the *Power Designer* [8], and other well known products include the *ERwin data modeller* [9] and the *DeZign* [10], both supporting the ER-modelling technique.

However, even with the use of professional tools, relational modelling requires considerable expertise, and remains a cumbersome undertaking. Besides, in spite of their commercial success, conventional relational database systems lack the richness of conceptual models and cannot satisfy the special requirements of non-traditional, non-business-oriented database applications [11]. CAD, hypermedia and medical applications are examples of such systems. These applications require modelling based on complex objects that can take the form of image data, structured text and component hierarchies. Especially in the case of web-based applications, the database systems involved need to be able to model, store and manipulate extensive multimedia data efficiently.

Conceptual models on the other hand, provide users with a more flexible and easy way of modelling applications, especially those of the multimedia type. Among them, the object model of data was originally developed to provide persistent storage for CAD programs, and has proved to be enduringly popular for this type of applications [12] and there's a number of UML-based modelling tools that support the OO design. However, the object database market never really took off despite the high commercial expectations. Some of the characteristics of the model proved to be drawbacks in real life applications, especially over the Internet. In certain aspects of query and transaction processing the OO approach never proved a match to the relational model. The concept of encapsulation in particular creates a processing overhead when populating or querying the database. Due to encapsulation, there is less granularity in an object database than in a relational db.

Having had neither the commercial success of the relational model nor the following of the OO paradigm, the *functional* data model (FDM) [13] emerged as one of the different flavours of the *semantic* models and remained a favourite of the "alternative", mainly academic scene of database applications. Its recent use in a variety of different projects that range from e-applications to data mining however has proved remarkably successful, and it has re-emerged as a popular choice for side- or partial storage of complex and, generally, unconventional information.

FDM offers a finer semantic granularity, which facilitates data modelling. Different type of information can be held for different data. There is no need for data types of fixed length like in the case of relational databases. On the same theme, missing or incomplete information is efficiently handled, without the need to record *null* values, unless such recording is semantically important. Unlike new relational applications that need a new set of application programs every time, functional databases provide the flexibility of code reusability. There is only

one language used both as DDL and DML. Contrary to the relational model, which is aggregation founded, FDM is based on directed graphs, and the modelling of information is done in a simple, conceptual way, with no need for expensive normalisation procedures and expert care. The concept nodes represent entities, attributes, states and events. The relation nodes show how the concepts are interconnected. *Figure 1* shows the graphical representation of the functional schema for the *Supplier-Product* database.

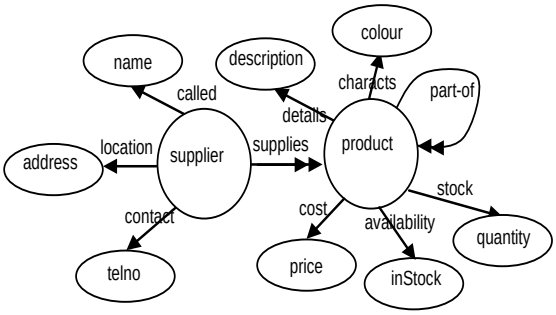


Figure 1: The Supplier-Product database

Unlike the extended automated modelling support for relational databases, and the conceptual UML-based modelling of object oriented applications, there are no designer tools for DBMSs based on the functional architecture. With the increase in the FDBMS range of use, information modelling has become vital for the users of the provided for applications. Similarly, the support of interoperability between the FDBMS and any other architecture is essential. This is the ground the developed environment is set to cover. The prototype system supports an extended version of the functional database language FDL [14,15], equipped with facilities for the storage and manipulation of multimedia types of text, image, audio and video format.

3 Issues of interoperability

At the age of information distribution, it is virtually impossible to discuss information modelling without mentioning system integration and XML. The eXtensible Markup Language [16] separates the content from its presentation. Its principal features are extensibility, structure and validation. Validation is handled by means of a DTD (Definition Type Document) or a XML Schema and a *validating* parser that checks whether a XML document conforms to a given DTD. Very appropriately to the task at hand, it supports complex structures, including deep nesting that traditional relational DBMSs cannot store.

As the internet becomes faster and more integrated, XML has come to be regarded as the new standard for data distribution and system interoperability. By means of the DTD (or XML schema) independent groups of people can agree to use a common format for interchanging data, verify the validity of either received or own data and provide an application-independent way of sharing and exchanging information. As a consequence, XML documents have proved to be highly portable and allow for information modelling and extensive data manipulation.

These characteristics make XML an ideal framework for data exchange over the internet, a fact that has been recognised by software developers worldwide, who have proceeded by integrating XML into their applications in order to gain Web functionality and interoperability between heterogeneous knowledge banks.

Recognising the importance of interoperability, GISMoE supports XML by automatically translating the functional schemas into data type definition documents and maintaining XML data dictionaries in parallel to those coded in FDL. The prototype deploys the transformation rules that have already been developed and which map the functional model to XML [17, 18]. It thus generates both XML documents describing the user-defined models and the equivalent DTDs, based on the produced database schemas (see section 4).

4 The prototype tool

The GISMoE architecture is modular. The environment consists of the following distinct modules:

1. the model visualisation and refinement,
2. the DB object capture, and
3. the schema generator.

The modules' interconnections and interfaces are pictured in *Figure 2*.

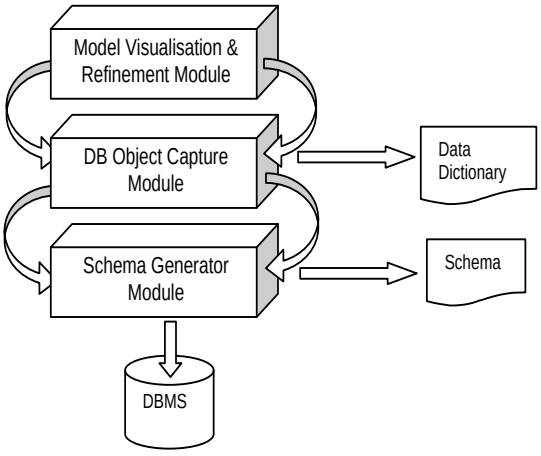


Figure 2: the GISMoE architecture

4.1. The environment layout

On GISMoE home page users are required to sign in to the system. Once access is given, the main page comes up with the choice of three options: apart from the main

model visualisation screen, GISMoE maintains another two sub-screens, depending on the modelling technique adopted, the *subschema* support screen and the *complex objects* visualisation screen.

The subschema sub-screen works in much the same way as the main schema screen and provides user views (the *createDB* in the main menu panel is substituted by *createUV*). Parts of the main schema can be dragged and dropped from the main screen, while subschema updates and further development are verified for compatibility by means of an extra option of the Schema sub-menu. If there are no clashes, the changes can be consequently accepted, in which case the main schema and any existing user views are also updated accordingly, or rejected. In Figure 3a the view depicted in the *Subschema* window (bottom) corresponds to the colour co-ordinated part on the main schema window (top).

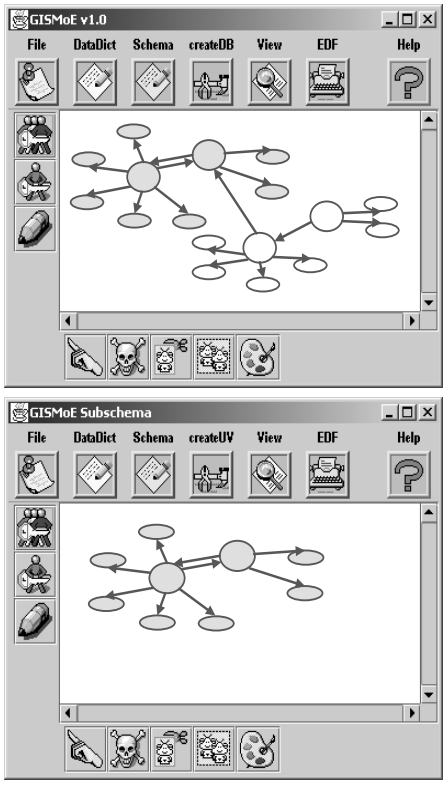


Figure 3a: user views

The complex objects' support screen provides the extra functionality of modelling entities whose behaviour and detailed attributes are to be considered at a later stage, or, whose presence is desired in a higher level of abstraction (i.e. for reasons of simplicity in extended and complicated schemas). Figure 3b demonstrates a possible use of complex objects in the same schema. The entities A and B are modelled as complex entities in the main window (top) and then expanded in the visualiser (bottom).

The db objects' *capture pane* allows the user to draw, edit and manipulate the database entities and functions.

The *Edit* buttons are responsible for the editing functions `Select, Cut, Paste` and `Delete`. The *Fonts* and *Colour* button provide further editing assistance.

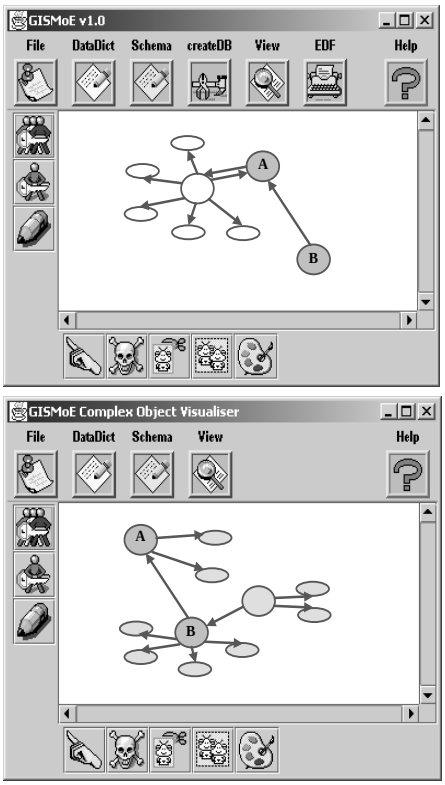


Figure 3b: complex objects

The *Menu* buttons consist of the *File* (`open, save` and `save_as` operations), *View* (`zoom, rotate, expand_complex_object`), *Data Dictionary* (`FDM entities, functions, XML elements`), *Schema* (choice of functional or XML DTD) and *Help* options. The *CreateDB* of the main window creates a database. The web version allows users with special privileges to create the database on the server. Creation of the database on the client side is only possible with the stand-alone application. *CreateUV* is the equivalent operation of the subschema window that creates customised user views. The *EDF* option provides the user with the facility of inserting *extensionally* defined functions (edf's) to the generated database. The following example introduces such a function called *discount* to the Supplier-Product database. The function introduces a 10% discount for big stocks of products (=stocks of over 100), whose parts are no longer available:

```
discount: product -> float;
discount x <=
    if AND (stock x >= 100)
```

```
(part-of x = [ ]) 0.10 0;
```

The *Mode* buttons provide the editing of the schema itself. There are 3 different modes: *Abstract Entity*, *Base Entity* and *Function*. Each mode changes the cursor into a mode-related shape (arrow-ended crosshair for *Abstract*, plain crosshair for *Base*, double arrow for *Function* mode). Once the cursor is positioned to the required place, a dialog window appears prompting the user to provide information about the entity or function. In the case of functions, the user has to select two entities before the function line materialises on the screen. The first entity selected corresponds to the *domai*n (from) entity, whereas the second selected entity corresponds to the *range* (to) entity. Recursive functions can be drawn by selecting the same entity twice. A tool snapshot is pictured in *Figure 4* below.

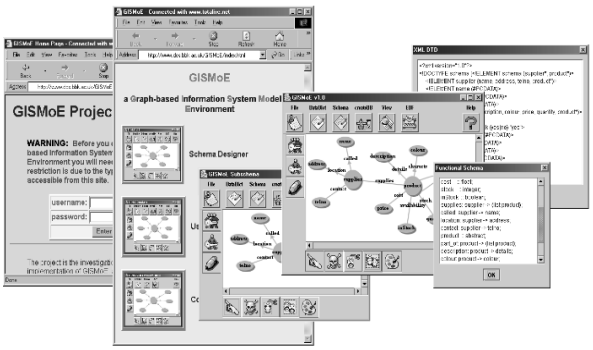


Figure 4: GISMoE snapshot

4.2. Creating database schemas.

Lets now assume that the user has designed the *Supplier-Product* database (see Figure 1) on the capture pane. Based on this schema, the data dictionaries contain the following information (the first two rows correspond to the FDM entities and functions, the third is the XML data dictionary):

Entities (type)	**Functions**	**XML tags**
supplier (abstract)	Supplies	<supplier>
name (string)	Called	<name>
address (string)	Location	<address>
telno (integer)	Contact	<telno>
product (abstract)	part_of	<product>
description (string)	details	<description>
colour (string)	characts	<colour>
price (float)	cost	<price>
quantity (integer)	stock	<quantity>
inStock (boolean)	availability	<inStock>

By clicking on the Schema button the user can choose to create the functional database schema, or the corresponding XML DTD. The functional schema is generated as follows:

```
supplier :: abstract;
name:: string;
address:: string;
telno :: integer;
product :: abstract;
details :: string;
characts :: string;
cost :: float;
stock :: integer;
inStock :: boolean;
supplies: supplier ->(list product);
called: supplier -> name;
location: supplier -> address;
contact: supplier -> telno;
part_of: product -> (list product);
description: product -> details;
colour: product -> colour;
price: product -> cost;
quantity: product -> stock;
availability: product -> inStock;
```

Based on the same schema, the equivalent DTD is also automatically generated. Entities of type abstract, string, integer or float are translated into XML elements, while boolean entities are turned into attributes. Abstract entities are given an occurrence of 0 or more and are part of the top entry which is named *schema*.

```
<?xml version="1.0"?>
    <!DOCTYPE schema [
    <!ELEMENT schema (supplier*,product*)>
    <!ELEMENT supplier
       (name, address, telno, product*)>
    <!ELEMENT name (#PCDATA)>
    <!ELEMENT address (#PCDATA)>
    <!ELEMENT telno (#PCDATA)>
    <!ELEMENT product(description, colour,
       price, quantity, product*)>
    <!ATTLIST product inStock (yes|no)
       "yes">
    <!ELEMENT description (#PCDATA)>
    <!ELEMENT colour (#PCDATA)>
    <!ELEMENT price (#PCDATA)>
    <!ELEMENT quantity (#PCDATA)>
]>
```

GISMoE has been implemented using Java 2 Platform, Standard Edition (J2SE), version 1.4.1 on both Solaris 9 and Windows 2000 Operating Systems. It has been tested with all major revisions of Java since Java 1.3, so that it works on all platforms with the relevant Java Virtual Machine. Java was chosen because it is architecture independent, provides portable user interface and can enable loading on demand of the application front end as an applet over the web. Because of security issues arising from the sensitive nature of the data, web use of the system is restricted to practitioners and researchers working on said projects. A full-blown, stand-alone version will also be available to those working off-line. For demo purposes, a cut-down version of GISMoE will be ready for downloading soon at http://www.dcs.bbk.ac.uk/~andrew/GISMoE.html

5 Conclusions and future work

The research carried out highlights the design and implementation of a web-based graphical information modelling environment, that automatically generates database schemas based on the functional data model. The system supports complex objects, user views and extensionally defined functions, and it is further integrated by providing an XML interface that allows for interoperability with other databases and knowledge and information repositories in general. The prototype implementation of GISMoE runs both as an applet and as a stand-alone application and we are currently working on the latest version that will allow for its internet demo distribution by means of a free downloading facility (see section 4.2). Once this is finalised, two trial runs of GISMoE will take place, with two different sets of data: one with medical and another one with crime data.

6 References

[1] Dotsika F., Modelling medical operational knowledge for e-health applications, International Journal of Health Care Engineering, Technology and Health Care (ISSN 0928-7329), Vol. 10, No 6, 2002, IOS Press, Amsterdam.

[2] Narayana Jayaram, Dotsika F., The impact of the three key healthcare technology standards on evidence-based healthcare practice, International Journal of Health Care Engineering, Technology and Health Care (ISSN 0928-7329), Vol. 9, No 6, 2001, IOS Press, Amsterdam, pp 501-503.

[3] Codd E.F., Further Normalisation of the Data Base Relational Model, Database Systems, Courant Computer Science Symposia Series,Vol. 6, Prentice Hall 1972.

[4] Pin-Shan Chen P. The Entity-Relationship Model – toward a unified view of data, ACM TODS 1, No 1, March 1976

[5] Hammer M.M , McLeod D.J., The Semantic Data Model: a modelling mechanism for databae applications, Proc. ACM SIGMOD International Conference on management of Data, Austin Texas, 1978

[6] S.W.Day, Modelling J2EE Applications using Oracle9/Designer and Oracle Developer, Oracle Corporation White Paper, April 2002

[7] A family of products with Oracle8, Oracle Corporation White Paper, June 1997, http://otn.oracle.com/products/oracle8/htdocs/xo8twps2.htm

[8] Sybase Power Designer 9.5, Product Overview Paper, June 2002, http://www.sybase.com/content/1019470/PD95_overview.pdf

[9] Don LeClair, ERwin and Managing eBusiness Development, Computer Associates White Paper, April 2002.

[10] Datanamic, DeZign for databases, http://www.datanamic.com/dezign

[11] D.S. Batory et al., Implementation concepts for an extensible data model and data language, ACM TODS, September 1988, Vol 13, Issue 3, pp.232-262

[12] H. Ishikawa et al, The model, language and implementation of an object oriented multimedia knowledge base management system, ACM Transactions on Database Systems, Vol 18, No 1, March 1993, pp. 1-50

[13] David Shipman, The functional data model and the data language DAPLEX, ACM Transactions on Database Systems (TODS), March 1981, Vol 6, Issue 1, pp. 140-173

[14] King P.J.H, Poulovassilis A. FDL: A Language which integrates Databases and Functional Programming Actes du Congres INFORSID 88 pp 167-181

[15] Poulovassilis A.: FDL: an integration of the functional model and the functional computational model, Proc. BNCOD-6, 1988, pp 215-236

[16] Bray T., Paoli J., Sperberger-McQueen (W3C) Extensible Markup Language (XML) 1.0 http://www.w3.org/TR/1998/REC-xml-19980210 10/02/1998

[17] Dotsika F., Watkins A., Integrating Web-Based Information Systems: WWW and the Functional Model, Proceedings of the IASTED Conference on Internet and Multimedia Systems and Applications IMSA2002 (ISSN 1482-7905), pp 74-79.

[18] Dotsika F., Watkins A., XML and Functional databases, Proceedings of the IASTED Intelligent Systems and Control Conference ISC2001, pp 242-246.

Towards the new generation of web knowledge

Fefie Dotsika and Keith Patrick
Business, Information, Organisation and Process Management Research Centre, Westminster Business School, University of Westminster, London, UK

Abstract

Purpose – As the web evolves its purpose and nature of its use are changing. The purpose of the paper is to investigate whether the web can provide for the competing stakeholders, who are similarly evolving and who increasingly see it as a significant part of their business.

Design/methodology/approach – The paper adopts an exploratory and reviewing approach to the emerging trends and patterns emanating from the web's changing use and explores the underpinning technologies and tools that facilitate this use and access. It examines the future and potential of web-based knowledge management (KM) and reviews the emerging web trends, tools, and enabling technologies that will provide the infrastructure of the next generation web.

Findings – The research carried out provides an independent framework for the capturing, accessing and distributing of web knowledge. This framework retains the semantic mark-up, a feature that we deem indispensable for the future of KM, employing web ontologies to structure organisational knowledge and semantic text processing for the extraction of knowledge from web sites.

Practical implications – As a result it was possible to identify the implications of integrating the two aspects of web-based KM, namely the business-organisational-users' perspective and that of the enabling web technologies.

Originality/value – The proposed framework accommodates the collaborative tools and services offered by Web 2.0, acknowledging the fact that knowledge-based systems are shared, dynamic, evolving resources, whose underlying knowledge model requires careful management due to its constant changing.

Keywords Knowledge management, Modelling, Knowledge sharing, Worldwide web, Semantics

Paper type Viewpoint

1. Introduction

Technology has been heralded as the answer to our information requirements, a charge that has been extended to meet our knowledge requirements as well. Intranets have been cited as examples of such a solution and success, so have web-enabled databases and portals. But to what extent does this address the needs of the user in the creation and particularly the ability to search for and share this information and knowledge? And how effectively does this facilitate the creation of new knowledge? It is our proposition that users and organisations need to beware of the balancing act of successful web-based search and sharing.

The web was originally designed as a text and image repository for human use. Its unprecedented expansion however has triggered a significant increase in the expectations for web-based information retrieval, knowledge sharing and collaborative working. Search engine indices have become too large, with every search producing an enormous amount of results. Search engines are often limited by poor indexing, ranking of pages according to inappropriate metrics, the absence of keywords on relevant pages and inaccessibility to distributed information repositories of different

VINE: The journal of information and knowledge management systems
Vol. 36 No. 4, 2006
pp. 406-422

0305-5728
DOI 10.1108/03055720610716665

formats, such as databases. At the end of every query the searchers are inundated with a great amount of links that they need to go through in order to gather the knowledge they require. Companies often try to second-guess the "magic" words used by searchers, or employ search engine optimisers. Organisations often end up paying for content that could be found for free on the web.

Looking into web knowledge search and sharing, users can search for knowledge using a number of means: following a path of hypertext links, using search engines, web-directories or intelligent agent software. Based on the actor of the search two distinct approaches can be identified: the end-user practice and the automated approach. The first one (also termed "cognitive" approach) is considered the traditional method and relies on the user going through websites in order to gather the required knowledge. The second method is the technical equivalent of the same process and relies on intelligent agents ("bots") for the gathering of knowledge. Each method has its advantages and disadvantages and, depending on the task at hand, each is associated with particular quality and suitability issues and/or specific limitations.

A further reflection is required when we observe the ubiquitous and pervasive nature of technology throughout our lives, which impacts on both our working practices and attitude toward that technology, in terms of how it is used and it continues to evolve. There is a shift from a specialised, centralised and controlled application and implementation of technological solutions to one that sees distributed, and multiple solutions that are local and enterprise-wide. An additional shift is from a smaller number of centralised specialists to increasingly involved and sophisticated end-users. This confronts the issues of technological design and development (what should be made to fit what or whom and should the user fit the system/software or the reverse) with intrinsic implications for developers, users and organisations alike. It is not untypical to organise around business processes with a tendency to embed them within rigid bureaucracies with the inherent procedures and rules, technology systems, and structures such as ERP and SAP systems. So if for some reason, the process needs to be changed, it becomes very difficult to make any adjustments because so much structure has been wrapped around it. Allee (1997) saw the need where strategies are human-centred and not technology centred and for a culture that addresses and supports knowledge creation, sharing and learning.

These technological shifts can be seen to be further reflected in the changing nature of the economy from manufacturing to knowledge and information-based economics, which focus more toward productivity, new products and services, new modes of delivery/supply, time-based competition, and shorter product life cycles. This economy is global in terms of both the market-place and the internet/web-mediated market-space which is engendering a workforce characterised by three significant types of worker:

(1) data workers who process and disseminate organisation's paperwork;

(2) information workers who primarily create and process information; and

(3) knowledge workers who design products or services, or create new knowledge for the organisation (Laudon and Laudon, 2005).

This growth in knowledge work and knowledge workers requires not only the ability to find and access information and knowledge, but also ability to share this synchronously and asynchronously in terms of both time and location. Newell *et al.*

(2002) saw the knowledge worker in a more evolved form than Laudon and Laudon (2005), characterised by higher levels of education, specialist skills and ability to apply these skills to identify and solve problems. According to them these workers effectively "own the primary means of production", and have the knowledge, skills and ability to apply them.

During this period, with a significant causal effect from the introduction and spread of technology, organisations can also be seen to have evolved by changing their structure. There is evidence of flattening, reduction in the number of level of management and reporting within the structure and decentralisation. Satellite structures are often used that include geographic relocation of parts of either the organisation or particular activities or tasks, including their outsourcing or off-shoring. These changes sought to derive flexibility, location independence, low (lower) transaction and co-ordination costs, empowerment, and create the need for collaborative work.

Zack (1999) suggested:

> To remain competitive, organisations must efficiently and effectively create, locate, capture, & share their organisation's knowledge & expertise.

This, results in a series of questions for any organisation: what do we know about our customers, services, products, markets and environment? How well do we know what we know? With the additional proviso "is this known by the right people?", queries regarding information quality and information integrity, and with the final question "How well do we act on what we know?". This series of activities and actions is commonly identified as knowledge management (KM), and aptly described by Elliott (2004) as: "the coordination and management of human understanding and knowledge within an organisation". Hence the need to be able to search and find the information and knowledge required at that time and to be able to share and reuse concurrently and at subsequent occasions.

The rest of the paper is organised as follows. Section 2 examines the different methods adopted when searching for knowledge on the web. In section 3 we explore the emerging trends and technologies that influence the future of the web, identify the ones most pertinent in knowledge search and share and assess their overall impact in web-based KM. Section 4 identifies the problems of each approach, determines possible solutions and sets the foundation of the proposed framework. Finally in section 5 we sum up our conclusions and outline future work.

2. Searching for knowledge on the web

The web, as it stands, holds information using natural language, multimedia content and hypertext marking. Such information can be combined from various sites via a search engine and can then be processed either by humans (cognitive approach) or by intelligent agents (automated approach). The cognitive method places the burden of knowledge discovery on the end-user who is required to go searching through a number of web pages. Equivalency of terminologies is not an issue – since humans can make associational mappings on the fly – neither is deduction. However, humans cannot process this information when it comes in overwhelming quantities. This is where the intelligent agents take over, though, for this to happen, a number of standards have to be met and appropriate technologies need to be adopted.

2.1. The cognitive approach

The web was designed primarily for human interpretation and use. The end-user searches across pages either through hyperlinks (free-style web surfing and/or use of hyperlink indices), subject directories, or search engine results. In order to be successful, this method requires strong and sound indexation that enhances navigation. It further relies on the searchers' mental model that is their personal experience and domain knowledge. Mental models influence the actual search and determine how the searchers will interpret the information gathered. Both pre-existing and found knowledge are mapped into a contextual cognitive structure, a "schema". Schemata facilitate the organisation of knowledge and incomplete information around a basic framework and affect future search behaviour and further evaluation of knowledge (Greve and Taylor, 2000).

However, finding information on the web is not necessarily the direct result of searching. We identify three different factors that influence the users' search behaviour and the overall task success in the cognitive approach: search strategy; choice of keywords (associated with the user, not with the dynamics of the search engine, which are addressed in section 3); and usability and navigation issues.

(1) Depending on the focus of the search, three strategies can be identified, that are directly linked to the task of the search (Navarro-Prieto *et al.*, 1999):

- Top-down strategy, where the searchers start with a general area and proceed by narrowing down their search by following the links provided. Favoured when the topic of the search is contained within a general site consisting of a well-organised list of subtopics.
- Bottom-up strategy, where the choice of keyword(s) is specific. This method is chosen for precise, fact-finding searches.
- Mixed strategy, where both the above methods are used within the same search, either in parallel (multiple searches) or alternating strategies for better results. This strategy is typically used by the more experienced searchers.

(2) Depending on the choice of keywords, the searcher can opt for a plain keyword search, a Boolean (a search using OR, NOT and AND operators), or an exact phrase search. Search engine findings provide additional help by including the results from fuzzy searches (matches which are returned even when words are partially spelled or misspelled) and precision indicators (how close the result link is to the original query in percentage of relevancy).

(3) Apart from the search strategy and keyword choice, some searches never return the information sought, simply because users often get lost in hyperspace. Quantitative assessment of the success rate of web navigation has identified a number of factors that influence the results. Successful searches are correlated with shallow hierarchical navigation (high compactness), while failure is related to a linear style of navigation (high stratum) (McEneaney, 2001). A number of algorithms (longest repeated sequence, sequence alignment, etc.) have been used to assess the similarity between optimal and user navigation paths (Pitkow and Pirolli, 1999; Wang and Zaïane, 2002) proving that the higher the similarity to the optimal path, the better the chances of finding information successfully.

The main advantages of the cognitive approach include low costs (end-users instead of the increased expenditure of specialised, often tailor-made software and equally costly maintenance) and increased suitability when dealing with "open" domains and community-based applications. The limitations are typically two-fold. On the one hand there is often a (potentially) overwhelming volume of results returned by the search engine. Going through them harvesting the knowledge sought is not always easy especially as time constraints are often involved. On the other hand, there are the search-engine bound problems: poor indexing, ranking of pages according to a range of not always appropriate metrics, the absence of keywords on relevant pages and inaccessibility to distributed information repositories of different formats.

2.2. The automated approach

The technical approach ranges from the use of general search engines to the specialised search of intelligent agents (bots). Search engines employ robot programs (known as spiders) that roam the internet in search of information, rank the results according to relevance and list them for the user. In a similar way meta-search engines transmit user queries to multiple individual search engines and subject directories at once and then compile and consolidate the results into a uniform format and listing. The top search engines employ intelligent agent software, which navigates the internet searching for information.

However, conventional web mark-up provides syntax but lacks semantics, a fact that severely limits the task of intelligent agents. The new generation of web standards add semantics and deduction capabilities to traditional mark-up. The semantic web is about sharing knowledge between communities, individuals and machines. It expands the web by supporting semantic mark-up, transforming it into a distributed knowledge base which provides the ideal environment for intelligent agents performing various automated tasks (McIlraith *et al.*, 2001). The linking of information is done by re-usable, task-specific, high-level generic procedures, featuring user-specific customising constraints over a framework of standards and an ontological approach that determines a shared common concept of a domain. This "new" web, still relies on old technologies such as HTML and XML for looks and content structure, but it is further enabled with new languages and standards such as Resource Description Framework (RDF) (Brickley, 1999) and a variety of ontology languages (Fensel, 2001) such as the World Wide Web Consortium's standard, OWL (Web Ontology Language) (W3C OWL, 2004). These languages are used to create vocabularies that add semantics, inference tools and formal specifications of contents and relationships. As a result, web content becomes process-able (and, thus, ultimately "understandable") by intelligent agents, that is autonomous, interactive and adaptable software programs that search, gather and filter information.

The more the information, the greater the degree of complexity involved (and required). Benjamins *et al.* (2004) plot the dimensions of (web) information overload, (intelligent agent) task delegation and relevant complexity as depicted in Figure 1.

Overload of information corresponds to higher intelligence requirements (y). Equally, where intelligent agents are concerned, greater task delegation corresponds to greater autonomy (x), where autonomy represents the agents' primal characteristic of being able to operate on their own, without that is, human interference. The complexity

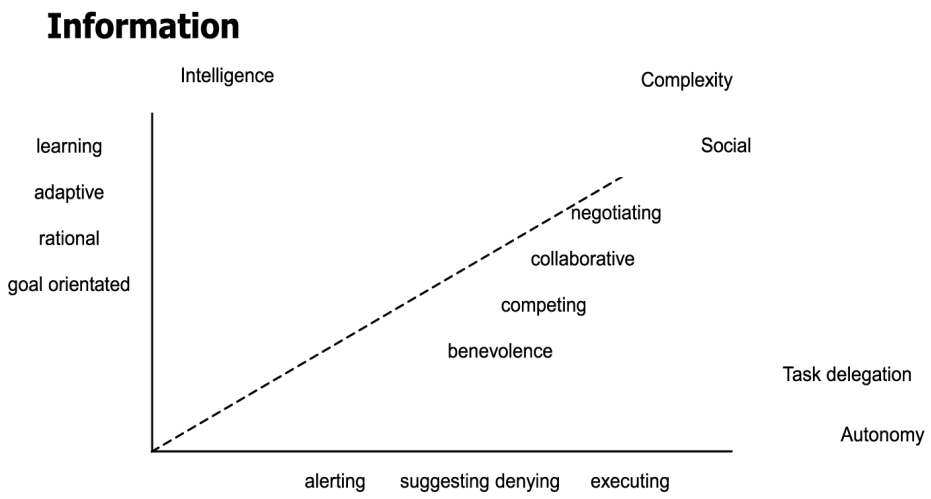


Figure 1.
Complexity of agents

of an intelligent agent can then be defined as a function f: $x \rightarrow y$. The rate of complexity multiplies with the increase of information and task delegation.

The automated approach fares best with "closed" consensual domains of knowledge and when highly precise information needs to be retrieved automatically, especially when semantic mark-up, ontologies and intelligent agents are deployed. The limitations of this approach include storage and scalability problems but also requirements for specialist end-users (Dotsika and Patrick, 2005a). However, the main drawbacks of the method arise from ontology quality issues, as we will see in the next section.

3. The future of the web

While the web is an essential repository of information, the simple use of search engines often fails to capture and interpret the users' real information needs. It is said that "a quality result is not a long list of links but the correct list". The semantic gap between the users' perception of the search domain and the results provided by the search may be the outcome of the sheer volume of answers returned, low quality, or plain irrelevance. Despite the fact that a part of corporate KM usually relies on web-based collaborative computing technologies by means of intranets, KM suites, corporate portals, etc., the quality of information retrieval, reuse and sharing is rather disappointing. Organisations and individuals are looking into the emerging trends and technologies for a possible solution. As a consequence there has been much speculation about the future of the web and its use as an efficient KM platform.

The idea of enhancing KM by enabling it to tap into the semantic web is to make a huge amount of electronically information more accessible by using ontologies to make searches more intelligent. The principle is simple: keyword searches are based on matching word patterns, whereas intelligent searches are based on answering questions. The semantic web supporters declare that the future lies in formal semantics, standardisation and intelligent agents. The semantic web key technology for managing knowledge is ontologies.

The Web 2.0 (O'Reilly, 2005) enthusiasts on the other hand proclaim that the future should be all about collaboration, sharing and end-users. According to this scenario,

the future lies in the tools supporting these activities, which are collectively known as social software.

Our framework proposal seeks to reconcile the two trends, since, although the sets of followers of the two camps seem disjoint in the first instance, they clearly have overlapping goals. It then furthers the notion of the web knowledge platform to include the "invisible" web. This "hidden" part of the web (referred to as the "invisible", "dark" or "deep" web) contains a huge amount of information that is not accessible by search engines.

3.1. Semantic web and ontologies

The application of ontologies as the conceptualisation of a given domain is well documented within the context of enterprise models (Fox and Gruninger, 1998). With the arrival of the semantic web there is a growing demand for facilitating ontologies' re-use and deployment, coupled with an increasing concern about the quality and validity of the information provider. Re-use (and/or extension) of existing ontologies is possible, and knowledge engineers are called upon to determine their suitability and decide on the best possible choice. One way to develop new ontologies is to identify and adapt existing ones from a neighbouring field. This method can increase consistency while keeping costs low. But, regardless of the technique employed, the quality of the ontology is of the utmost importance. We identify the following quality issues: ontology modelling features, express-ability/re-usability and application environment issues:

- In accordance with the principles of conceptual modelling, ontological quality comes in three flavours: syntactic, semantic and pragmatic (Lindland *et al.*, 1994). Syntactic quality reflects the syntactic correctness of the model. Semantic quality addresses the question "Does the model cover the domain of interest?" Finally, the pragmatic dimension indicates whether the model is comprehensible by the user.
- In modelling ontologies, express-ability is a synonym to complexity. Complexity hinders re-usability, one of the most important characteristics of ontologies. A high-quality ontology is specific in modelling the domain's attributes, but should not be more specific than necessary.
- Ontologies should be able to integrate with a variety of applications and interfaces. They should therefore be language independent (not tied to a particular natural or programming language) an aspect that may affect the ontology's express-ability.

The use of semantic mark-up and ontologies have led to the deployment of an increasing number of intelligent agent information retrieval systems. They often employ a combination of agent types (brokers, mediators and wrappers), search technologies (natural language understanding, filtering and domain modelling, conceptual search techniques) and architectures (simple or multi-agent, local or distributed). These systems tend to be task-specific and, consequently, the quality of the search results depends on the particular assignment.

Nevertheless, information retrieval is not intelligent agents' only suitable application. Agent software provides a specialised form of "push" technology, a dynamic form of electronic publishing that automates the transfer of information to

end-users. Push technologies are an increasingly popular type of sharing content as well as applications. The agents undertake the time-consuming task of monitoring web information resources and are controlled by end-users who can specify the type of information they want to receive.

There is a number of existing RDF tools, developers APIs and ontology editors that can be combined to provide semantic web-enabled KM platforms. The best-known open-source ones are Protégé-2000 (Noy *et al.*, 2001) and Sesame (2004), while OntoEdit (2002) and Jena 2 toolkit (HP Laboratories Research, 2002) are commercial suites. Other products include OILEd (Bechhofer *et al.*, 2001), Ont-O-Mat (Handschuh *et al.*, 2001) and the more recent Swoop (Kalyanpur *et al.*, 2005). They invariably offer ontology browsing and editing and may provide querying facilities (Sesame, Jena 2, etc.) and/or plug-ins (Sesame, Swoop, etc.)

A pick-and-mix combination of tools like the above has lead to complete ontology assisted KM platforms. KAON (Bozsak *et al.*, 2002) and On-to-Knowledge (Davies *et al.*, 2002) are the most comprehensive among them. KAON is an open-source ontology management platform targeted for business applications. KAON's front-end consists of the user-level applications and its core addresses the developer needs and comprises two APIs and a number of libraries. On-to-Knowledge comprises an ontology-based environment that provides tools for the support of KM, a bottom layer of machine-processable metadata and a core repository that uses semantics to describe meanings for annotated data

3.2. Web 2.0 and social software

While the technical/automated approaches have been viewed as the solution to meeting information requirements they do not represent a complete solution, as they do not follow the patterns of cognitive practice of individuals. Reflecting the question "Where is the fit?" there are two possible and opposing views: "technology to the user" and "the user to the technology" (Dotsika and Patrick, 2005b). Historical evidence however shows a pattern which is not always in line with theory or discussion. According to it, developer approaches typically take the former view, while practice echoes the latter. The result of the inherent compromise impacts on the proposed efficiency gains of any solution and the current and future goodwill toward subsequent technology solutions. This is not necessarily an aversion to technology or technology solutions, but dissatisfaction with how the solution fails to meet or fit the requirements and behaviour of the proposed and potential user. This can be seen in the provision of information (and subsequent information overload) of a 24/7 technologically connected world, whose need to be able to "pull" the information required is far greater than the overwhelming nature of the "push" of the continuous stream of information broadcasting to customers and employees. The problem of the latter is the notion that it throws information by the bucket, when a glassful was all that was needed, with these buckets rarely being other than tangential to the actual need.

Two observations can be made in relation to this scenario regarding technology, users, and organisations. Organisations can be seen to have two fundamental structural components in their make-up. At the core there is the typically formal structure with its levels and responsibilities and reporting hierarchies. Then there is the informal organisation centred on personal connections, common interest or goals, "... an invisible force influencing resource allocation ...", and "... an antidote to

inflexible bureaucracy …" (Gabriel *et al.*, 2000). The second observation regards technology and how individuals can and will use it, seen in how they use technology to interact and cluster with other individuals, through mobile telephony, instant messaging, mailing list and groups, etc. It should perhaps be noted that typically this grows organically, and this aspect has significance in examining how to balance the technical and end-users aspects of search and sharing. This collection of technologies, being branded as social software, supports the desire of individuals to be pulled into groups to achieve goals (Boyd, 2003). Figure 2 depicts the potential components of social software (adapted from Bryant, 2003, 2005a).

Although this tag can be applied to many types of software, there are several key elements, such as a means for conversational activity that is both synchronous and asynchronous, and feedback in the form of contributions and comments from others, with evidence of the personal relationships of the participants, who together form the social network.

Social software effectively is a convergence of the thinking of the domains of social networks, human-computer interaction (HCI) and web services. In relation to the question of the technology-to-user fit, social software adapts to its environment, as opposed to the environment being required to adapt to the software. Successful software can be seen to be intuitive so that it enables the user to adapt and continue to use it. An additional feature in relation to the organisation is the duality of its informality and typically bottom-up development. The more interesting aspect and relevant to the examination of the balance between end-user and bots is how the adoption of social software in organisations is also seeing a different approach, drawing on the ethos and nature of social software itself, with vendors and proponents (like Headshift) seeking to shift from IT-centric solutions and implementations to building on the information and knowledge store within the organisation (Table I (source Bryant, 2005b)).

A characteristic of this approach is the centring on the users without over-burdening them from above. The key population of taxonomy or ontology is from the bottom, although within a top-down framing or seeding. There is additional support for the lateral bridging of elements across groups, rather than the traditional/typical top-down constraining, enabling collaboration with the users instead of shaping them to the technology. In general, this technique seeks to join

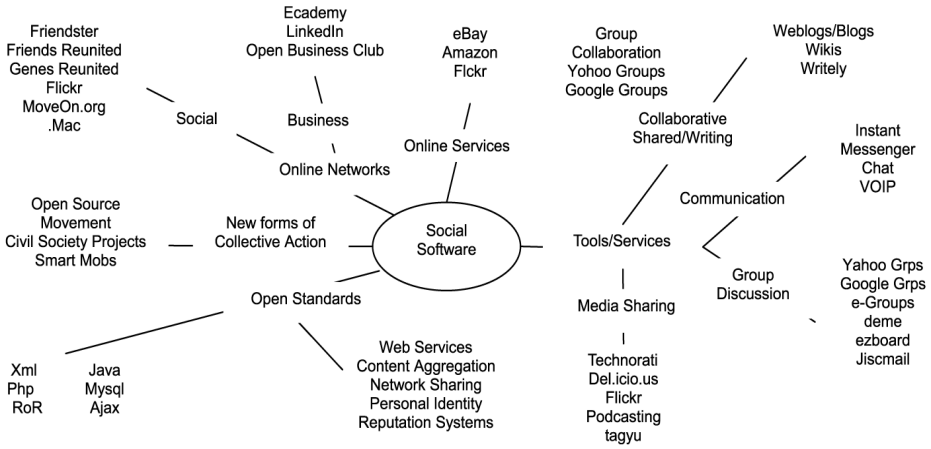


Figure 2.
Social software

across the differing and diverse individuals and workgroups within an organisation, but also to allow for the re-factoring of stored information and knowledge around the current and changing needs, creating flexibility and scope for innovation. Core to this approach is the encouragement and stimulation of the social networks and interaction, especially the conversational aspects. These elements seek to expand user attitudes, from single-loop learning and rigid focusing upon direct problem solving, to the adoption of double-loop learning.

Web 2.0 is a reference to perceptions of what the next generation web will look like and can be seen in aspects of the social software, services like flickr (the online photo sharing community site) or technorati (the blog internet search engine), places/spaces for sharing, an environment providing users with web based applications and collaborative environments and resources that are accessible from any computer and location, regardless of operating systems or software installed on that machine. It reflects a coming of age of aspirations underpinning the thin-client and network-computer approaches proposed in the 1990s. In essence, Web 2.0 is a development from the wellsprings that fed the social software movement, but increasingly involving larger technology and web-focused organisations like Yahoo and Google. Yahoo purchased flickr, while Google followed with the acquisition of writely, the web word-processor environment that enables the sharing of documents and collaboration in real-time, with the ability to limit access and edit documents from anywhere (Ukn Google blog, 2006). Google has recently launched a web-based collaborative spreadsheet application (Ukn BBC, 2006) and an online sharable calendar, with further linked support through really simple syndication (RSS), enabling links to content deemed relevant to and for the collaborating users.

3.3. The invisible web (IW)

In 2001, BrightPlanet, a search technology company, speculated that IW possibly contained 550 billion documents, perhaps 500 times the content of the conventional web, when Google – which claims to index the most comprehensive collection of documents on the internet – had identified 1.2 billion documents and was actually capable of searching a mere 600 million of those (Bergman, 2001).

The IW comprises content that search engines either cannot or will not index. Most of the IW is made up of the contents of specialised databases that can be queried via the web. The results are then delivered in dynamically generated web pages, whose storage is expensive and are therefore discarded as soon as the user reads them. Technical barriers related to the design and functionality of spiders mean that search engines cannot find or create these pages. Spiders navigate the web by following

Traditional solutions	Social software
Top-down command and control	Bottom-up, devolved
One-to-many, impersonal	Many-to-many, personal
Formal, bloated, inflexible	Informal, lightweight, flexible
Corporate voice	Human voice
Large, slow, expensive	Small, iterative, cheap
Owned by the vendors or it	Owned by you and your people

Table I.
Traditional solutions vs social software

hyperlinks (a page with no links becomes "invisible") but can neither type nor "think". Hence, specialised databases that are searchable over the web are inaccessible if they have no static pages with links containing information, so are web sites that require login. The rest of the IW consists of the so-called excluded pages. They are certain types of pages that the search engines exclude by policy. They either contain special formats that hinder indexing (e.g. contents in Flash, Shockwave, images only, etc.), or script-based pages (e.g. sites with URLs that contain the "?" sign).

Although there are not general tools for searching the IW, there are an increasing number of links and subject directories to invisible web databases, such as The Invisible Web Directory (IWD, 2005). Integration of the traditional and the IW is, of course, problematic. Directed query technology and pre-assembled storehouses provide some (far from seamless) support. The former is cumbersome and places the burden on the user, who has to download the appropriate software and issue effective queries. The latter supports selected content and query customisation which disadvantages general requests and needs.

Quality issues are similar to those encountered in the conventional web: matters of availability, quality of information and duplication. Duplication is particularly difficult to assess, though the guidelines are similar to those of the "traditional" web sites. Sites whose content is unique include topical and scientific databases, library holdings, satellite imaging data and internal site indices. Duplicated sites (and information) include product listings, software, press releases, mirrored sites and search engine results. Nonetheless, assessing IW's overall information quality can be tricky, as there is no standardisation of retrieval methods and no availability of proper statistics of depth and volume. Similarly, the sharing of the information retrieved from the IW is not straight-forward: resulting pages are dynamic and lack of relevant organisational strategy in their storing for sharing purposes can well lead to storage inefficiency. Moreover, neither the semantic web nor the Web 2.0 tools and methodologies can be applied here.

4. Towards a new framework

The frameworks visited lack a number of tools/facilities that we deem essential for supporting KM. Our proposal criticises both approaches by pinpointing their respective advantages (features we need to retain) and disadvantages (issues we need to resolve). Our framework therefore differs in the following points:

(1) Knowledge modelling tools of existing or proposed systems are usually editor and/or form based. As such they are largely counter-intuitive and require expertise not always present where end-users are concerned. The alternative to editor-based schema design is conceptual modelling: the process of constructing a model of the information at hand that is independent of the implementation details, application programs and software/hardware considerations. As a concept it applies to the modelling of information and knowledge and plays a central role in the creation of any information repository, from web content to KM systems. Conceptual modelling tools fitted with a graphical user interface have proved to be more appropriate than editor-based environments (Dotsika and Watkins, 2004). They facilitate knowledge capture by hiding complexity, are user friendly and can be cost-effective since they automatically generate code.

(2) Whatever the future of the web, there always will be information repositories residing outside the boundaries of the new technologies. Therefore, an integrated approach should try to maintain interoperability with such sources for as long as needed (Dotsika, 2003). Current systems provide some access to existing sources, such as KAON's access to relational data sources via OntoMat-REVERSE (Bozsak *et al.*, 2002), however a full integration with legacy systems would require a more flexible approach that transcends schema architectures.

(3) The idea of enhancing KM by enabling it to tap into the semantic web is to make a huge amount of electronically information more accessible by using ontologies to make searches more intelligent. The adoption of a common ontology language has been considered a must for the support of semantic interoperability, resulting in the Web Consortium's OWL recommendation (W3C OWL, 2004). Ontology language standardisation however is inversely proportional to ontology content design. The quality criteria particularly relevant to semantic web ontologies are accuracy (inaccurate ontologies would produce wrong results), transparency (opaqueness would affect reusability) and reason-ability (otherwise inference would be disabled) (Svatek, 2004). There are a number of methods offering ontology content quality support, such as meta-properties, pre-fabricated patterns support, collected hints, etc. (Svatek, 2004). While most methods fare well with accuracy control, their performance in controlling transparency and reason-ability varies significantly depending on the application area.

(4) However, this typically top-down approach runs the risk of failing in capturing the detail required. This detail usually resides at the bottom, where the key people often find themselves constrained by technology, rigid software support and bad system design. Inability to engage and involve the end-user results in systems that do not get employed efficiently and can potentially lead to system failure. The solution is to combine the flexible top-down framing/bottom-up populating of social software with the formal semantics of the semantic web.

(5) When it comes to semantic mark-up, storage, scalability and retrieval are problematic areas. Storing semantic web data has led to the debate over the implementation architecture (relational vs. graph-based) while scalability and constant increase of storage requirements have given birth to further storage concerns. The storage debate is well timed as it coincides with the launching of the new file system implementations brought out by the major operating system vendors (Sun Microsystems with ZFS as part of their OS Solaris 10 and Microsoft with WinFS as part of Longhorn). On the retrieval front, query languages at present do not always have the flexibility required (eg. query across multiple graphs and sub-graphs).

(6) The SW framework has been described as overestimating the value of deductive logic, while underestimating the difficulty of a shared worldview (Shirky, 2003). Even if the automation of web information retrieval by means of intelligent agents is successful, web contents will always be used and processed by humans as well as agents, with or without the involvement of some partial automated tasks. In this "traditional" use of the web, indexation takes

precedence over formal semantic mark-up, as navigation is more pertinent than inference. Although this approach lacks the advantages of computational deduction it may nevertheless prove enduring due to its low-cost, easily maintenance and no requirements for specialist end-users. Therefore new systems should take this into consideration and look into integrating the cognitive approach with the automated one.

Figure 3 sums up the proposed framework.

5. Conclusions and future work

Our exploration identified several non- exclusive trends that represent views on how the next generation of web could evolve and how the latency of web knowledge can be unlocked. However, there is the inherent problem that each trend may overwhelm the previous one and not allow its full exploration. Indeed computer history is littered with ideas left behind which remain unfulfilled and never fully explored: a problem associated with technology is the penchant for riding the front of the wave, the cutting edge.

It is possible to observe several patterns in how these trends and ideas are driven; from within the existing web-developer environment and from the collaboration and swarming of IT-literate web users seeking to build or help build a shared vision of a web that is customisable and delivers what users want and not what developers think they want. At the same time technology companies seek to create and/or exploit the commercial benefits of the next wave. This can be discerned in the interests of the significant players within the web environment, such as Yahoo, E-bay, Google, Microsoft, etc. that seek to harvest the commercial benefit of the web. This behaviour is shown through their own development, acquisitions and manoeuvring in the marketplace. The significant patterns lie in the collaborative views of the social software movement, which are now solidifying in the Web 2.0 framework and being consolidated into web applications and services. Another significant trend is that of assisting the management of the exponential growth of the web, in relation to the data,

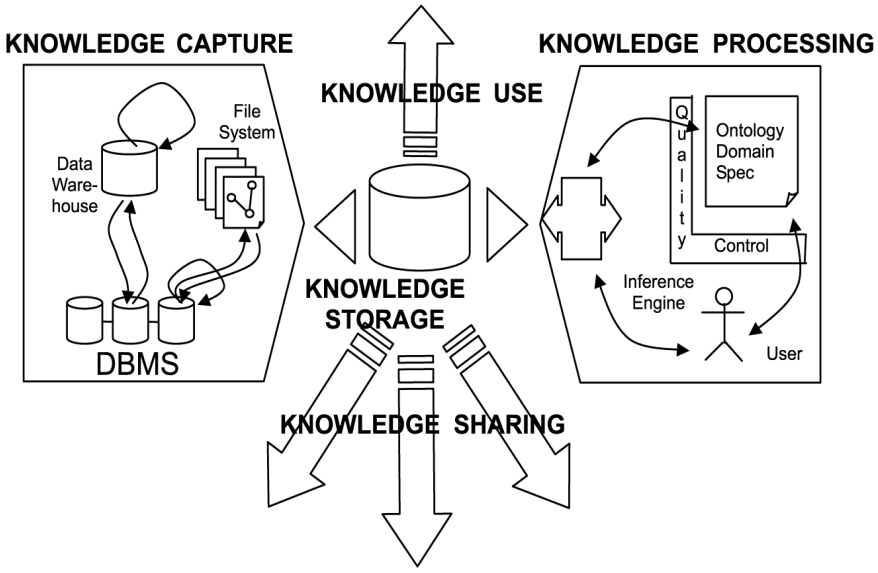


Figure 3.
Web-based KM

information and latent knowledge, which is the base of the semantic web infrastructure with its established potential in information retrieval and knowledge discovery. To this extent we present a framework that could reenergise the development of the potential that lies within the semantic web and support the creation of a web of knowledge that is no longer a latent hop.

Based on the above we investigated the main requirements for the support of KM in the next generation of web, looked into existing developments and solutions and provided an independent framework for the capturing, accessing and distributing of web knowledge. This framework retains the semantic mark-up, a feature that we deem indispensable for the future of KM, employing web ontologies to structure organisational knowledge and semantic text processing for the extraction of knowledge from websites. Furthermore, our proposal accommodates the collaborative tools and services offered by Web 2.0, acknowledging the fact that knowledge-based systems are shared, dynamic, evolving resources, whose underlying knowledge model requires careful management due to its constant changing.

However, web search and sharing is only part of the problem. An increasing problem lies in user expectation, as more systems are clothed in web-based front-ends that mask the underlying disparate nature of the information repositories, legacy systems and databases that are at the back-end. This suggests to users of all levels functionality that is neither realistic nor practicable, with consequences for systems developers, administrators and managers. It further indicates the need for proactive management of users and has an impact on how their expectations are encouraged and supported.

While our research was based upon web-based knowledge, the next step should include non-web-based sources of information, such as office documents, e-mail messages and news feeds. A recent Butler Group Review (Thornton, 2005) reports that anywhere up to 80 per cent of a knowledge worker's time is spent hunting for information and 80 per cent of corporate information is held on users' desktop PCs. Search strategy and practice should include desktop search, thus integrating web servers, file servers, DBMSs and e-mail storage. There are currently a number of desktop search environments that do just that, with Google, Copernic, Yahoo! and MSN Toolbar Suite leading the market.

References

Allee, V. (1997), *The Knowledge Evolution: Expanding Organisational Intelligence*, Butterworth-Heinemann, Boston, MA.

Bechhofer, S., Horrocks, I., Goble, C. and Stevens, R. (2001), "OilEd: a reason-able ontology editor for the semantic web", *Proceedings of KI2001, Joint German/Austrian Conference on Artificial Intelligence, 19-21 September 2001, Vienna, Austria*.

Benjamins, V.R., Contreras, J., Corcho, O. and Gomez-Perez, A. (2004), "Six challenges for the semantic web", *SIGSEMIS Bulletin*, Vol. 1 No. 1, April, p. 2004.

Bergman, M.K. (2001), "The deep web, surfacing hidden value", *The Journal of Electronic Publishing*, Vol. 7 No. 1, August.

Bozsak, E. *et al.* (2002), "KAON – towards a large scale semantic web, e-commerce and web technologies", *Proceedings of the Third International Conference, EC-web 2002, Aix-en-Provence, France, September 2-6, 2002*, 2455 of Lecture Notes in Computer Science, Springer, Berlin, pp. 304-13.

Boyd, S. (2003), "Are you ready for social software?", available at: http://darwinmag.com/read/050103/social.html

Bryant, L. (2003), "Smarter, simpler, social", available at: http://headshift.com/moments/archive/sss2.html

Bryant, L. (2005a), "Introduction to social software for the networked social enterprise", available at: http://headshift.com (accessed June 2006).

Bryant, L. (2005b), "Making knowledge work", available at: http://headshift.com (accessed June 2006).

Brickley, D. (1999), "Semantic web history: nodes and arcs 1989-1999 the WWW proposal and RDF", available at: www.w3c.org/1999/11/11-WWWProposal/ (accessed November 2005).

Davies, J., Fensel, D. and van Harmelen, F. (Eds) (2002), *On-to-Knowledge: Semantic Web Enabled Knowledge Management*, J. Wiley & Sons, New York, NY.

Dotsika, F. (2003), "From data to knowledge in e-health applications: an integrated system for medical information modelling and retrieval", *International Journal of Medical Informatics and the Internet in Medicine*, Vol. 28 No. 4, December, pp. 231-51.

Dotsika, F. and Patrick, K. (2005a), "Knowledge capture, sharing and maintenance in the semantic web age: a framework proposal", *Proceedings of the 4th International ISOneWorld Conference, Las Vegas, April.*

Dotsika, F. and Patrick, K. (2005b), "From end-users to bots: the balancing act of web-based knowledge search and sharing", *Proceedings 2nd International Conference on Intellectual Capital, Knowledge Management and Organisational Learning, Dubai, November*, pp. 143-50.

Dotsika, F. and Watkins, A. (2004), "Can conceptual modelling save the day: a unified approach for modelling information systems, ontologies and knowledge bases", *Proceedings of the 15th IRMA International Conference, May*.

Elliott, G. (2004), *Global Business Information Technology: An Integrated Systems Approach*, FT/Prentice-Hall, London.

Fensel, D. (2001), *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, Springer-Verlag, Berlin.

Fox, M.S. and Gruninger, M. (1998), "Enterprise modelling", *AI Magazine*, Fall, pp. 109-21.

Gabriel, Y., Fineman, S. and Sims, D. (2000), *Organizing and Organisations*, 2nd ed., Sage Publications, London.

Greve, H. and Taylor, A. (2000), "Innovations as catalysts for organizational change: shifts in organizational cognition and search", *Administrative Science Quarterly*, Vol. 45 No. 1, pp. 54-80.

Handschuh, S., Staab, S. and Mädche, A. (2001), "CREAM – creating relational metadata with a component based, ontology driven annotation framework", *Proceedings of First International Conference on Knowledge Capture ACM K-CAP 2001, October 2001, Vancouver*.

HP Laboratories Research (2002), "Jena 2 – a semantic web framework", available at: www.hpl.hp.com/semweb/jena.htm (accessed July 2002).

IWD (2005), "The invisible web directory", available at: www.invisible-web.net/ (accessed October).

Kalyanpur, A., Parsia, B., Sirin, E., Cuenca-Grau, B. and Hendler, J. (2005), "Swoop – a web ontology editing browser", *Journal of Web Semantics*, Vol. 4 No. 1.

Laudon, K., C, . and Laudon, J.P. (2005), *Management Information Systems Managing the Digital Firm*, 8th ed. (international ed.), Pearson/Prentice-Hall, Englewood Cliffs, NJ.

Lindland, O.I., Sindre, G. and Sølvberg, A. (1994), "Understanding quality in conceptual modelling", *IEEE Software*, Vol. 11 No. 2, pp. 42-9.

McEneaney, J.E. (2001), "Graphic and numerical methods to assess navigation in hypertext", *International Journal of Human-Computer Studies*, Vol. 55 No. 5, pp. 761-86.

McIlraith, S.A., Son, T.C. and Zeng, H. (2001), "Mobilizing the semantic web with DAML-enabled web services", *Proceedings of the 2nd International Workshop on the Semantic Web, Hong Kong*.

Navarro-Prieto, R., Scaife, M. and Rogers, Y. (1999), "Cognitive strategies in web searching", *Proceedings of 5th Conference on Human Factors and the Web*, http://zing.ncsl.nist.gov/hfweb/proceedings/proceedings.en.html

Newell, S., Robertson, M., Scarbrough, H. and Swan, J. (2002), *Managing Knowledge Work*, Palgrave, London.

Noy, N.F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R.W. and Musen, M.A. (2001), "Creating semantic web contents with Protege-2000", *IEEE Intelligent Systems*, Vol. 16 No. 2, pp. 60-71.

OntoEdit (2002), "Knowledge modelling with OntoEdit, OntoPrise, semantics for the web", available at: www.ontoprise.de/products/ontoedit_en (accessed: July 2002).

O'Reilly, T. (2005), "What is Web 2.0?", available at: www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

Pitkow, J.E. and Pirolli, P. (1999), "Mining longest repeated subsequence's to predict world wide web surfing", *Proceedings Second USENIX Symposium on Internet Technologies and Systems, 11-14 October*.

Sesame (2004), "'openRDF.org', Home of Sesame", available at: www.openrdf.org (accessed July 2004).

Shirky, C. (2003), "The semantic web, syllogism, and worldview, economics and culture, media and community", November, available at: www.shirky.com

Svatek, V. (2004), "Design patterns for semantic web ontologies: motivation and discussion", paper presented at the 7th Conference on Business Information Systems, Poznaň, 21-23 April.

Thornton, R. (2005), "On the road to business integration", *Butler Group Review Journal*, February, available at: www.butlergroup.com/review/default.asp (accessed September 2005).

Ukn BBC (2006), "Google launches web spreadsheet", available at: http://news.bbc.co.uk/1/hi/business/5051610.stm (accessed June 2006).

Ukn Google blog (2006), "Googler insights into product and technology news and our culture", available at: http://googleblog.blogspot.com/2006/03/writely-so.html (accessed June 2006).

W3C OWL (2004), "OWL web ontology language overview", available at: www.w3.org/TR/owl-features/ (accessed April 2005).

Wang, W. and Zaïane, O.R. (2002), "Clustering web sessions by sequence alignment", *Proceedings of DEXA Workshops*, IEEE Computer Society, Los Alamitos, CA, pp. 394-8.

Zack, M.H. (Ed.) (1999), *Knowledge and Strategy: Resources for the Knowledge-based Economy*, Butterworth-Heinemann, Oxford.

Further reading

Berners-Lee, T., Hendler, J. and Lassila, O. (2001), "The semantic web", *Scientific American*, May, pp. 35-43.

Phillips, N. and Patrick, K. (2003), "Personality type and the natural development of knowledge evolution", in Coakes, E., Willis, D. and Clarke, S. (Eds), *Knowledge Management in the Sociotechnical World: The Graffiti Continues*, Springer, Berlin.

About the authors

Fefie Dotsika is a Senior Lecturer in the Business School of the University of Westminster. Her background, expertise and research interests include system interoperability, web information modelling and retrieval and web-based knowledge management. She is currently involved in research in the areas of: semantic web technologies and common interchange formats; web information search and share; emerging web technologies, their role, importance and applications; and web standards and their implications in the future of e-business. Fefie Dotsika is the corresponding author and can be contacted at: F.E.Dotsika@westminster.ac.uk

Keith Patrick is a Senor Lecturer in the Westminster Business School at the University of Westminster, London, UK. He takes a managerial, organisational and user perspective on technology and its deployment. His background and research interests lie in the domains of Business Information and Knowledge Management. His current research interests include: trust and risk in online transactions; web- based technologies and knowledge management; and social capital (trust, relationships, networks).

An IT Perspective on Supporting Communities of Practice

Fefie Dotsika
University of Westminster, UK

INTRODUCTION AND BACKGROUND

An increasing number of organisations have come to recognise the fact that encouraging and maintaining communities of professionals with common interests, aims and objectives can reduce costs and increase profits. From enhancing customer responsiveness to increasing innovation and preventing reinvention, Communities of Practice (CoPs) are seen as an important vehicle to the improvement of organisational performance.

Even as the role of CoPs has been gaining momentum, the IT community has become aware of the evolving opportunities and is consequently involved in attempting to provide the relevant software tools. This article investigates the requirements for the efficient IT support of CoPs, explores the advantages and pitfalls of supporting 'computerised' versions of these communities, reviews a number of existing software tools and looks into emerging technologies considering their role and appropriateness.

CoPs AND IT

CoPs are often viewed as a catalyst to the success of a particular organisation's KM system. Their mission is the capturing and sharing of knowledge among practitioners: a task that has traditionally relied upon communicating organisational knowledge via personal interaction and sharing of experiences, problems and best practices.

One might question whether the deployment of IT in supporting CoPs is justifiable, and whether it would offer a clear return on investment. Those who are for IT support argue that providing easy access to critical market intelligence through, say, a portal, is always good for business. Those who are against tend to overemphasise the problems that electronic systems have created over the years for managers and users alike.

But in spite of such problems, bad press and disaster cases that come under the umbrella of system failure scenarios, it is an undeniable fact that an ever-increasing amount of vital business information spends its whole life-cycle in digital format. This fact alone challenges the nature of old-fashioned communication/collaboration

between the members of a group and adds to the need of consolidating the way information is handled.

While many communities are supported by websites providing knowledge sharing by means of online libraries, knowledge centres, specialist databases, information repositories and white pages, only few of them get the full necessary support. Terms like *online* and *virtual CoPs* are becoming commonplace, reflecting thus the increasing tendency to form expanded and even globalised versions of the traditional groups of people who come together to share their knowledge. Despite the spatial difference between traditional groups and their online counterparts, the actual requirements remain the same. Figure 1 summarises these requirements, and depicts the main flow of basic activities within a CoP.

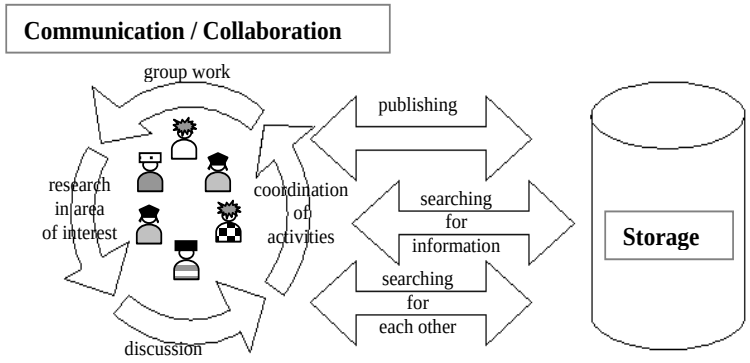


Figure 1. Flow of activities

Following the framework proposed by Ngwenyama and Lyytinen [1] the four types of social actions can be seen at work here: instrumental, communicative, discursive and strategic. According to this division we can now look at the four action clusters from an IT viewpoint, identifying what type of software/groupware would be appropriate for carrying out their respective tasks.

1. Instrumental actions. This category is supported by the so-called *research tools*. These are tools that provide the person executing the instrumental action with the relevant resources, i.e. the relevant knowledge. Databases, data warehouses, data marts, electronic document management systems (EDMs), knowledge bases and knowledge servers all play the role of *knowledge repositories* under this category. The research tools that extract knowledge from these repositories come in all shapes and guises, from database query languages and search engine facilities to data mining and intelligent agents.

2. Communicative actions. Traditionally the earliest and possibly most efficiently supported category. Use of e-mail, list servers, internet, corporate intranets and even remote login facilities, file transfer and electronic messaging are examples of communication tools.

3. Discursive actions. Apart from the possible overlap with the previous category – such as the use of e-mail and listserv facilities – there are dedicated groupware packages that assist the setting up, customisation and

configuration of on-line discussion groups. Chat rooms and e-conferencing are also popular applications. In general, collaboration services come under two categories: *synchronous* and *asynchronous*. Instant messaging facilities, e-conferencing and all sorts of audio and/or video streaming belong to the former category, whereas discussion forums, calendar postings and e-mail belong to the latter.

4. Strategic actions form the last category, the only one with no evident IT support. Although closely related to instrumental actions to the extent that they both strive to achieve rational objectives, the two categories differ in their view of the opponent: the person executing the instrumental action treats the adversary as an organisational resource and not as a person capable of intelligent counteraction [2] (which is the case in the strategic action). This "quirkiness" alone makes things hard as it predefines a requirement difficult to resolve with conventional IT tools. The solution is likely to come from the Artificial Intelligence community, with the use of intelligent agents. These are adaptive computer programs capable of reasoning and learning, and are collectively known as *bots*. There are many types of agents, each performing specific, specialised tasks (search bots, chatter bots, shopping bots etc). Their potential to support strategic actions derives from the fact that they are sociable - they can interact and communicate with humans and other bots.

Apart from the above, there is a number of collaborative computing technologies used in the support of knowledge management that can also be put into use with CoPs. These tools can usually service the above action categories to varying degrees, with the exception of the strategic actions.

- *Knowledge management suites* provide solutions for creating centralized repositories for storing and sharing knowledge, allow for communication between the members of the group and support groupwork. They thus integrate the storage, communications and collaboration services into a single environment.

- *Portals*. Also known as *super-sites* or *enterprise knowledge portals*, they are an electronic doorway providing a comprehensive array of resources and services. Portals typically contain newsletters, e-mail services, search engines, online shopping, chat rooms, discussion boards and personalised links to other sites. While portals attract a large number of visitors offering a wide range of contents, vortals (*vertical portals*, also known as *online communities*) are narrower in focus and address a specific industry, theme, or particular interest, a feature that has made them more appropriate for the support of CoPs.

- *Collaboration tools*, often referred to as *groupware*. A difficult to define class due to the diversity in the functions offered. Most packages comprise an information repository that can be accessed by team members who can collaborate working on common documents and can hold electronic discussions. Some groupware packages integrate calendars, group schedulers and e-mail. Others offer e-conferencing facilities or other real-time meeting support.

EXISTING SOFTWARE PLATFORMS

We divide the software platforms into two distinct categories: software that offers IT support aimed especially at Communities of Practice and software designed to assist Knowledge Management in general, but also meets the requirements for the support of CoPs. Generally speaking, both KM and CoP support requirements are similar, though different emphasis is given to certain components. For instance, KMware demands broader content management techniques leading to more rigorous system interoperability requirements, whereas CoP support relies heavily on the communication layer. With a large and constantly increasing number of available platforms in each category, the list of products presented below is only representative of the range of services available but is by no means exhaustive.

COPS DEDICATED IT SUPPORT

1. iCohere [3] provides web collaboration software tools for online communities, project teams and distributed organizations. Specific applications include extranets, workgroup and virtual team collaboration and online learning. Their technology and supporting processes enable engaging member communication, networking, knowledge sharing, collaboration and learning. The groupware is available either as a hosted application on the iCohere servers or for use in the customers own servers as a site licence. It supports a back-end MS SQL Server and web-based dynamic DBMS access. Whether hosted or licensed, the software claims advanced security considerations. It supports https option, configurable password formats and login timeouts. iCohere partners include universities, education-focused professional societies, corporate business, government agencies, healthcare associations and non-profit organisations.

2. Tomoye's CoP platform Tomoye Simplify 4.0 [4] offers a similar set of resources. In efforts to meet the increasing customer demand for integrated services, Tomoye has recently become a Microsoft Certified Partner (March 2004). On its web site, the company demonstrates the different functionalities of the Simplify platform through two case studies: (a) oneFish, a Cop at the UN in Rome and (b) Global Knowledge Partnership at the World Bank.

Tomoye built oneFish to enable 15,000 fisheries researchers from around the world to pool their knowledge, identity experts and collaborate in online conversations. oneFish features 10,000+ records, cross referenced across 1700 topics. It allows for easy navigation, provides threaded discussion forums, e-mail lists and digests, FAQs, content ratings and a search engine over an XML database that includes multimedia content.

The second case study is the Global Knowledge Partnership at the World Bank, an organisation that comprises 65+ partner organisations, dedicated to the sharing of knowledge and best practices for sustainable development. Knowledge is modelled as knowledge objects, and each object (including people) can have its own discussions and FAQ. Users can further subscribe to a subject of interest and receive regular e-mail updates, digests and links to new related objects. The online environment provides login facilities and membership privileges, customisation, navigation via bookmarks, search for knowledge and experts, discussion forums and instant messaging.

3. Not all software houses providing CoP support software are big. KnowNet is a small company that was founded in 2000 to research and develop new architectures, ideas and internet software for collaborative knowledge development and learning [5]. The company supports virtual online communities through integrated portals, collaborative content management interactive XML document repositories, structured discussion groupware, collaborative resource sharing and metadata management. Its customers include the European Commission (Leonardo da Vinci Vocational Training, CEDEFOP and STRATA programmes) and the British Library.

GENERAL KM SUPPORT

1. Open Text [6] is one of the biggest players in groupware services, especially since it acquired fellow enterprise content management software firm IXOS in 2003. Both Open Text - better known for its collaboration and document management software – and IXOS – known for archiving and content management – had made several acquisitions in the months prior to the take over, with the result to end up with a surplus of software packages that needed sorting and integrating. The company offers Livelink, a KM software environment that manages corporate knowledge assets. Marketed as a "scalable and modular platform for the acquisition, creation, aggregation, management and delivery of content", the Livelink interface brings together various collaborative applications supported on the Open Text platform and can be successfully used for CoP support. By leveraging best practices and lessons learned across different communities, Livelink connects and organises knowledge entities into knowledge-sharing networks and delivers an integrated system for collaborative work to globally distributed teams.

2. Another major KM software player is Hummingbird [7] a global provider of enterprise software solutions. Their integrated platform Hummingbird Enterprise 2004 offers a comprehensive number of capabilities: content, document and record management, e-mail management, enterprise workflow, collaboration platform, wireless mobility, query and reporting facilities, and data integration. Their portal framework integrates all components of Hummingbird Enterprise 2004 to deliver personalised content, applications and collaboration capabilities within dynamic views or virtual workspaces, based on the role of the user in the business process. Their customers cover a vast cross-section of industries: aerospace and defence, government, chemical, oil and gas, energy and utilities, automotive, telecommunications, financial services, life sciences and healthcare, education, manufacturing, retail etc. Although the software is mainly marketed as enterprise content management, wherever available, the platform has enough attributes that make it an efficient CoP support tool.

3. iLevel Software [8] provides solutions that enable teams to collaboratively manage the entire life-cycle of business content using a unified, tightly integrated platform and repository. The iLevel environment offers extensive XML content management, web-based document management, web content management and intranet/extranet access to business information, but also a number of services that improve knowledge exchange and retrieval, such as enterprise search, categorisation facilities, alerts and collaborative capabilities.

THE SEMANTIC WEB AND THE USE OF ONTOLOGIES

The unprecedented expansion of the World Wide Web has triggered a significant increase in the expectations for web-based information retrieval, knowledge sharing and collaborative working, all of which work well within a tight frame of reference but become problematic when this frame expands. With the appearance of the Semantic Web [9], the rapidly developing form of web content that is readable by computers, web-based knowledge representation relies on languages that express information in a machine process-able form.

The "conventional" Web relies on encoding schemes based on technologies such as HTML and XML (eXtensible Markup Language) [10]. However, information that adheres to this encoding lacks explicit semantics. To this extend, the Semantic Web deploys two further enabling technologies: RDF (Resource Description Framework) [11] and ontologies [12]. If we think of HTML as a mark-up language for displaying data and XML as another for describing it, then RDF provides the semantic mark-up and ontology languages supply a shared common understanding of a domain.

More specifically, RDF models knowledge as directed graphs, represented as triples. The semantic structure of these triples is the assertion that *subjects* are associated with *objects* by means of *predicates*, hence the subject-predicate-object relationship. Each of these terms can be represented by a URI (Universal Resource Identifier).

With the semantic mark-up in place, ontologies provide the formal specification of a knowledge domain, often along with an inference engine. A particular knowledge domain consists of classes, their instances and the relationships between them. This domain specification can then be communicated between heterogeneous application systems, enhancing knowledge sharing and retrieval [13]. Consequently ontologies are particularly useful for (a) sharing a common understanding of a domain among the members of the community, (b) analysing and/or reusing domain knowledge and (c) making explicit any domain assumptions.

The deployment of semantic mark-up together with ontologies revolutionises web information retrieval and sharing, a fact that is of particular interest to CoPs [14, 15], some of which are already working towards common encoding standards. Among them, the linguistic community is developing GOLD (General Ontology for Linguistic Description) [16].

Nevertheless, the Semantic Web is not the only use of ontologies related to CoPs. Another use focuses on systems used to identify CoPs within an organisation, a process presently done by means of structured interviews. ONTOCOPI (Ontology-based Community of Practice Identifier) [17] is such a system, capable of identifying CoPs by examining the connectivity of instances in a knowledge base with regard to their type, weight and density.

CONCLUSION

There is a number of software platforms that are designed to assist Communities of Practice. Some of them provide dedicated support, whereas others are general KM environments able to offer CoPs the required IT

facilities. But while online communities benefit from technology and face-to-face member interaction can be substituted by virtual contact to various degrees, knowledge manipulation still poses a significant and often decisive obstacle to the flow of knowledge inside these communities. The emergence of the Semantic Web seems to tackle a number of these problems, though the process of migration is currently rather cumbersome and requires specialist knowledge of the technologies involved. However, software for the computerised adding of semantics to web information is being developed, while the design and development of tools for the automated capturing, sharing and retrieval of knowledge are under way.

REFERENCES

[1] Ngwenyama, O. K., & Lyytinen, K. (1997). Groupware environments as action constitutive resources: A social action framework for analyzing groupware technologies. Computer Supported Cooperative Work: *The Journal of Collaborative Computing*, *6*(1), pp. 71-93.

[2] Ngwenyama, O. K., & Lee A. S. (1997). Communication richness in electronic mail: critical social theory and the contextuality of meaning. *MIS Quarterly*, Volume 21, Number 2, pp. 145-167

[3] iCohere: Creating collaborative communities http://www.icohere.com (June 2004)

[4] Tomoye Simplify 4.0, http://www.tomoye.com/ (June 2004)

[5] KnowNet, http://www.theknownet.com/ (June 2004)

[6] Open Text and IXOS http://www.opentext.com (June 2004)

[7] Hummingbird Enterprise 2004: Enterprise Content Management Platform, http://www.hummingbird.com/products/enterprise/index.html (June 2004)

[8] iLevel Software http://www.iLevelSoftware.com/ (June 2004)

[9] Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web, Scientific American, May 2001

[10] Decker S. et al. The Semantic Web: the roles of XML and RDF, IEEE Internet Computing, Sep-Oct 2000, pp 63 –75

[11] Dan Brickley, Semantic Web History: Nodes and Arcs 1989-1999 The WWW Proposal and RDF, http://www.w3c.org/1999/11/11-WWWProposal/ 1999

[12] Fensel D., Silver Bullet for Knowledge Management and Electronic Commerce, Springer-Verlag, Berlin, 2001

[13] J. Davies, A. Duke, and A. Stonkus: OntoShare: Using Ontologies for Knowledge Sharing, Proceedings of the WWW2002 Semantic Web workshop, 11th International WWW Conference , Hawaii, USA, 2002.

[14] Domingue J. et al, (2001) Supporting ontology-driven document enrichment within communities of practice. In *Proceedings 1st International Conference on Knowledge Capture (K-Cap 2001)*, Victoria, BC, Canada.

[15] Motta E., Buckingham-Shum, S. and Domingue, J. (2000). Ontology-Driven Document Enrichment: Principles, Tools and Applications. International Journal of Human-Computer Studies, 52, 1071-1109

[16] Farrar S., Langendoen T., A Linguistic Ontlogy for the Semantic Web, Glot International Vol. 7, No. 3, March 2003 pp 97 –100

[17] Alani Harith, O'Hara Kieron, Shadbolt Nigel (2002) ONTOCOPI: Methods and Tools for Identifying Communities of Practice . In *Proceedings Intelligent Information Processing 2002*, Montreal - Canada.

Terms and Definitions

Portal: an electronic doorway providing a comprehensive array of resources and services. Portals typically contain newsletters, e-mail services, search engines, online shopping, chat rooms, discussion boards and personalised links to other sites.

Vortal: a vertical portal. A vertical industry, or market, or specific group portal on the Internet.

Knowledge management suites: Software packages that provide solutions for creating centralized repositories for storing and sharing knowledge, support content management, allow for communication between the members of the group and assist group-work.

Semantic web: A collaboration of the World Wide Web Consortium (W3C) and others to provide a standard for defining data structures on the Web. www.w3.org/2001/sw

XML (eXtensible Markup Language): A subset of SGML (Standard Generalised Markup Language), designed to describe data. It incorporates features of extensibility, structure and validation and is currently playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.

RDF (Resource Description Framework): A recommendation from the W3C for creating meta-data structures that define data on the Web. It is designed to provide a method for classification of data on Web sites in order to improve searching and navigation

Ontology: (originally) a branch of metaphysics the study of the essence of beings, or first principles. In IT it is the working model of entities and interactions in some particular domain of knowledge or practices. In AI an ontology is the specification of a conceptualisation.

Knowledge sharing: developing from within

Keith Patrick and Fefie Dotsika
Westminster Business School, University of Westminster, London, UK

Abstract

Purpose – If collaboration and knowledge sharing lie at the core of providing added-value to either services or products can we improve this process? The purpose of this paper is to suggest that it can be improved and this lies in how we develop the systems that support collaboration and knowledge sharing. This can be achieved within the development process, focusing on the knowledge sharers and developing from within.

Design/methodology/approach – The underpinning has been the examination of traditional system development methodologies, the emergence of social computing and its underlying approaches and ethos. The approach draws upon knowledge management concepts, overlaid onto the purpose and motivation for knowledge sharing.

Findings – The paper continues the premise that better systems are derived from fully engaging with the systems users. Although existing methodologies have this at their heart, the systems produced still fall short. The argument presents how developing systems from within can improve the likelihood of success through the adoption of social computing practices. It shows that the involvement of those expected to collaborate or share through the proposed system in the development process, enhances the collaborative relationships and increases the probability of sharing through engagement and empowerment.

Originality/value – This paper frames how a known problem in systems development and the greater sensitivity of knowledge management systems may be overcome. It highlights how the collaborative and inclusive nature of social computing practice can serve to bridge the sociotechnical divide through the reduction of barriers and providing alternative bridges.

Keywords Knowledge sharing, Knowledge management, Worldwide web

1. Introduction

We are constantly reminded that it is the employees who are the most important asset in any organisation (CIPD, 2001). Similarly, that collaboration and sharing represent the key to value-added (Lord Sainsbury of Turville, 2006; Porter and Ketels, 2003) which are described as the only means for developed countries to be able to compete in the growing global knowledge economy. This has resulted in a great expenditure of time and money in developing information systems, no longer just standalone information systems or databases, but integrated ones, or at least having a means to interact with each other, typically through the use of Internet technology, web-based interfaces, intranets, and portals. These systems increasingly become larger in size and complex in their nature and reach across geographical dispersed organisations and collaborators. Despite systems development methodologies increasing the role of end-user's, the specification and requirements for a system are set from above, in a top down manner, by managers often remote from the day-to-day tasks and activities. However, the history of Information Systems development has demonstrated that the likelihood of the solution meeting the original requirements or matching expectations is low. Haag *et al.* (2004) offer the low figure of only 20 per cent of systems meeting user requirements or the functionality sought. Furthermore, these systems provide greater

The Learning Organization
Vol. 14 No. 5, 2007
pp. 395-406

0969-6474
DOI 10.1108/09696470710762628

amounts of information, a "glut" to some extent, that overwhelms users and inhibits its use and therein its value. This hampers the productivity of knowledge workers and their ability to generate the knowledge required to provide necessary "value-added" for an organisation to remain competitive. It also reflects the need to shift to an information and knowledge sharing environment where "pull" (i.e. users active search for information), and not "push" (i.e. broadcasting of information) is predominant and where the users are being able, to an extent, to shape what is received and how it is received.

This growth in knowledge work and knowledge workers requires not only the ability to find and access information and knowledge, but also the ability to share this synchronously and asynchronously in terms of both time and location. Newell *et al.* (2002) see the knowledge worker as characterised by higher levels of education, specialist skills and ability to apply these skills to identify and solve problems. These knowledge workers effectively "own the primary means of production", and have the knowledge, skills and ability to apply them.

What we are proposing is an approach that encourages knowledge sharing through the development of systems from within. The social interaction inherently involves a sharing of both the goal and a favourable outcome, which is centred on problem solving. It focuses on the processes and the people involved, solving the problem within, rather than a solution imposed from outside or above. It seeks to involve, engage and empower, creating an environment amongst those who need to share and hold the knowledge. We suggest that this can be achieved through designing closer to the needs of the ultimate users and the organisation needs and requirements and through tapping the local knowledge of "what works" and "what does not work". Similarly, involvement of the users increases the likelihood of the system being used for the benefit of the organisation, while the involvement itself generates engagement and empowerment, so that ownership should follow. This approach adopts the blending of the social and the technical that is inherent in the emerging developments of Social Software, the coalescing of Web 2.0, and the Semantic Web.

The rest of the paper is organised as follows. In section 2 we address the social and technical systems contexts. Section 3 deals with the social software movement and its role in shaping and defining the concept of Web 2.0. In section 4 we investigate the problems in delivering the practical context following the methods outlined. Section 5 takes a step forward towards exploring less known and rarely used knowledge repositories along with the problems facing knowledge sharing in such environments. Finally we draw our conclusions in Section 6.

2. The social and technical systems context

Our proposed approach to system development from within aspires to the harnessing of the characteristics of social systems to overcome the differences of the technical system through the development process. Herrmann *et al.* (2004) identify these characteristics as "...communication and cooperation between individuals, especially the emergence of meaning systems, self-referential development of structures and learning process..." and contrasts them to the technical systems with their "...artefacts, control, anticipation, learning in respect to purpose, and determination from without the system...". This essentially recognises the complexity in the relationship existing between the social and technical, whilst placing the emphasis

upon the social in order to shape the technical outcomes. It also highlights the need to bridge what Ackerman (2000) identified as the social-technical gap "... what we know we must support socially and what we can support technically." Whitworth and de Moor (2003) also identified significance in this gap and a need to meet social requirements (see Figure 1).

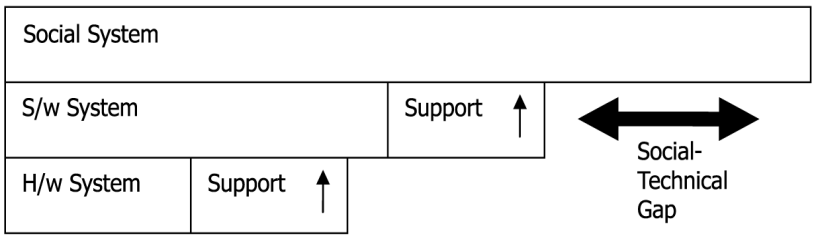
They further proposed the requirement for legitimacy in creating a trusted system or a system user's trust to overcome this gap. The strength in our proposed approach lies in the collaboration and negotiation amongst the users, be they sharers or creators, and more significantly the level of re-negotiation during development. This indicates how involvement, and subsequent engagement and ownership are essential as the need and resultant solution may over time be required to adapt or to evolve in the application or use.

3. Socially driven emergent technologies

The burgeoning growth of the Internet, accelerated through the spread of high bandwidth broadband, has now witnessed the emergence and outcomes of tools facilitating socially based interaction and participation. This virtual environment is characterised through self-organisation around a common interest or causes, typically, non-hierarchical and meritocractic, requiring only interest, time, application, and contribution for membership. This can be particularly seen in a number of socially-driven technology based developments: the Open Source Movement, the GNU General Public License (GNU GPL) and the Creative Commons approach to copyright. Each emphasises increased levels of sharing, sociability and contribution, with a diverse and non-hierarchical end-user involvement, adopting a more bottom-up oriented approach over the typically rigid corporate locking down of the top-down approach. The Social Software movement can be seen to build directly upon this ethos, with Web 2.0 emerging as a similar off-shoot, although providing additional questions, such as, how does the social approach cope when meeting the reality of commercial world? We are additionally witnessing the emergence of a generation of the future workforce currently in their teens, immersed in their mySpace and YouTube personas, to whom these environments are second nature.

3.1 Social software

Social Software effectively, is a convergence of the thinking of the domains of Social Networks, Human-Computer Interaction (HCI) and web services. In relation to the question of the technology-to-user fit, Social Software adapts to its environment, as opposed to the environment being required to adapt to the software. Successful



Source: Whitworth and de Moor (2003)

Figure 1.
The social-technical gap

software can be seen to be intuitive so that it enables the user to adapt and continue to use it. An additional feature in relation to the organisation is the duality of its informality and typically bottom-up development. The more interesting aspect and relevant to this examination is how the adoption of Social Software in organisations is also seeing a different approach, with vendors and proponents seeking to shift from IT-centric solutions and implementations to building on the information and knowledge stored within the organisation. The development approach adopted by Social Software takes a bottom-up, devolved approach, which is personal but many-to-many, informal, lightweight, flexible, presenting a human voice and taking small iterative steps. This approach is not costly and its ownership lies with the creators/users. This is opposed to the traditional approaches whose top-down command and control nature are typically impersonal, one-to-many, formal, bloated, inflexible, reflecting a corporate voice, and whose development is slow and expensive with a large product that is owned by the vendors or IT department.

This approach is characterised by its user focus and the limitation of the extent of burdening from above. It provides only top-down framing or seeding, as opposed to the rigidity of the formal constraints of a traditional managerial lead development and the locking down of the business/information requirements. However, this approach requires additional support that enables lateral bridging of elements across groups, rather than the typically traditional top-down constraining. This method seeks to link across the organisation addressing the differing and diverse individuals and workgroups present. It also allows the re-factoring of stored information and knowledge around the current and changing needs, creating flexibility and scope for innovation. Central to this approach are the encouragement and stimulation of the social networks, and the interaction within an organisation, particularly the conversational aspects. This practice can potentially assist in expanding user attitudes, from the single-loop learning and rigid focusing upon direct problem solving to the adoption of double-loop learning.

3.2 Web 2.0

Social Software is deemed one of the main components of Web 2.0, a new web concept allowing for the creation of web sites that improve the sharing of knowledge and services and are of a more collaborative, interactive and dynamic nature than plain pages. The origins of the concept of Web 2.0 can be seen in the lack of clarity in the available definitions. This also reflects its emergent nature, its evolution as it shifts toward the mainstream and the changes in nature and intent of those involved. For Tim O'Reilly (2005) Web 2.0 arrived through a conference and subsequent brainstorming session exploring the question "what is the role of the web post the dot-com boom and bust?". This discussion also derived a set of principles and practices set around the view that Web 2.0 has a central core and no real or hard boundary, representing a web platform rather than a PC platform. According to this definition the core competencies of Web 2.0 companies are:

- services, not packaged software, with cost-effective scalability;
- control over unique, hard-to-recreate data sources that get richer as more people use them;
- trusting users as co-developers;

- harnessing collective intelligence;
- leveraging the long tail through customer self-service;
- software above the level of a single device; and
- lightweight user interfaces, development models, and business models.

Wikipedia describes Web 2.0 as the "second generation of network-centric services available on the internet that let people collaborate and share information online in a new way – such as social networking sites, wikis, communication tools, and folksonomies." (Wikkipedia, 2006). Boutin (2006) outlines it as: "the web as a platform for interacting with content", reflecting a set of online tools that enable the "aggregation" and "remixing" of content, through interfaces that combine content from different sources in a manner not possible through a single domain. Davis (2005) takes this to suggest "Web 2.0 is an attitude not a technology".

The emergent nature of what is Web 2.0 is further apparent in the discussions of the 2nd Web 2.0 Conference (web.2.con.com, 2006) agenda – seeking to "clarify what it is" – or the O'Reilly view as owner of the Web 2.0 trademark, with a further lead topic of "Defining Web 3.0: What's Next?".

The key element we propose lies in this combination of PC and user independence and is in essence an architecture that will enable individuals and groups to use common tools in order to create and share information and knowledge. There are a varied range of examples appearing; "Writely" a word-processing application that is accessed via the web with the potential for multiple writers to be creators and/or editors of the same document. "Flickr", a photo sharing website, that has been used by utility engineers to share and distribute digital images of faults and breakages and thereby help monitor faults and share fixes and work-arounds. The key is that specialised technical skills are not necessarily needed: the availability of a range of tools and services enables the creation and sharing of new applications or instances, which are determined by the users and their needs. It could be ventured that this is a throwback to applications-based systems development approach, with many of the disadvantages such as lack of compatibility removed. Further examples of these type of combinations, colloquially termed "mash-ups", can be either websites or Web 2.0 applications that utilise content from multiple sources, generally third parties. These could be the incorporation of news feeds (e.g. BBC RSS feeds of headlines), links to relevant books and related topics (e.g. Amazon), and linking maps with geographic location data through addresses and postcodes (e.g. Google Earth). Such techniques provide a means to leveraging the increasing publicly available information from business and government sources. Once again, the emerging MySpace/YouTube generation appears both adept in this environment and willing to use it, interact and expect more!

4. Issues in delivering the practical context

As software and systems design methodologies have evolved the significant change has been the increased use of business analysts, and ultimately end-users in the development process (Patrick and Dotsika, 2006). However, when developing knowledge management systems, or managing people-centric problems, or trying to solve them with technology alone is insufficient, there is a need to both address and adopt social focused approaches (Patrick and Dotsika, 2006). To this extent the employment of social software collaborative tools is emerging. IBM is taking a

bottom-up approach by using Dogear, a social book-marking system, to categorize web content and other material using user-suggested tags. Microsoft is using Quests, an internal communications system which includes a wiki system and will be building a wiki into its SharePoint Server 2007 Web portal.

The sceptics however doubt that Web 2.0 is really going to make a difference in sharing knowledge. Some go further to suggest that it is nothing more than a marketing ploy. And how does one distinguish between Web 1.0 and Web 2.0 in practice? Is it simple adherence or the partial application of some form of Web 2.0 like behaviour? As shown earlier there are a few conflicting definitions around, besides Tim O'Reilly's own. In any case, one thing is for certain; technology should work for people, not the other way around. And Web 2.0 seems to be doing just that, mixing the services from different providers and users in a user-controlled way, except when its design falls short of delivering the right results. Our research has identified four problem areas of knowledge sharing when developing from within: knowledge modelling, standardisation, security and maintenance.

4.1. Knowledge modelling

Current applications supporting knowledge sharing and interoperability between incompatible knowledge repositories rely on annotating data and maintaining a syntactic consistency. This process adds structure and semantics to an otherwise unstructured or semi-structured mass of text-based information which, when in great quantity, becomes almost impossible to retrieve.

Web 2.0 classification schemes include tagging, taxonomies and folksonomies. Tags are labels (keywords) that categorise content. Taxonomies are hierarchical tree structures of classifications where every node maintains an is-a-relationship with the parent node. Folksonomies are open-ended, collaboratively generated taxonomies. Their novelty and popularity (in true, bottom-up, Social Software fashion) stem from the fact that their creators are also their users.

Tags and folksonomies however are rather informal classification systems, their main drawback being that they allow ambiguity in the classification process. This ambiguity can take the form of multiple meanings for the same word or synonyms for the same meaning. Taxonomies on the other hand, although formal, are rather restrictive and do not allow for flexibility in modelling complex information and knowledge.

Ontologies, a formal classification scheme based on explicit specification of the conceptualisation of a given domain (Gruber, 1993), are also used in Web 2.0, albeit in a non-standardised way. The role of formal semantics is to remove ambiguities in the interpretation of complex expressions providing thus a single, unified view of data and information across platforms and applications. That is to say that, in order to get information silos communicating with one-another, one needs to create compatible abstract models that can incorporate the schemas of all given silos regardless of their particular local syntax. In addition to the issue of interoperability, formal semantics make the content compliant to machine processing and therefore provide the foundation for system integration and knowledge sharing by means of intelligent agent systems.

This has been the founding stone of the Semantic Web (Berners-Lee *et al.*, 2001), a rapidly developing form of web content that is readable by computers, where

web-based knowledge representation relies on languages that express information in a machine process-able form. Although the concept of the Semantic Web precedes that of Web 2.0, they now have their own dedicated following, consisting of users and developers. Despite the often antagonistic relations of the two groups, there is a substantial overlap of aims and objectives. The Semantic Web platform provides formal semantics by means of the Resource Description Framework (RDF) (Brickley, 1999) and the Web Ontology Language (OWL) (W3C OWL, 2004). One of the main anti-Semantic Web arguments has always been the (flawed) assumption that the Semantic Web is nothing but the attempt of academics to create a single ontology or schema that can link together all sources of knowledge. While the "single ontology" claim is rather naive and unrealistic, the Semantic Web relies indeed on formal standardised semantics.

Whilst the Semantic Web-compliant ontologies are a solution to the problem of interoperability, they do not fare well on two occasions. The first is when we take into account the temporal attribute of knowledge. The constant evolution of communities and their inherent knowledge often demands the re-shaping, if not complete re-modelling, of the ontologies used. The second is their nature: semantic Web ontologies are traditionally top-down rather than bottom-up constructs.

The solution to the above problems seems to be somewhere in between the current methodologies. The general consensus is that open dynamic environments do not benefit from traditional semantic reconciliation techniques that depend upon shared vocabularies and global ontologies (Aberer *et al.*, 2004). The adoption of emergent semantics as a possible solution is based on the adoption of new heuristics which are founded on a domain's emerging properties and locally agreed semantics (Aberer *et al.*, 2003, Cudré-Mauroux and Aberer, 2004). This methodology merges successfully, formal semantics and bottom up design.

4.2. Standardisation

Not everybody agrees on the importance of standardisation of information and access to services. Social Software as well as Web 2.0 enthusiasts see often little point in it. During the AAAI 2006 conference in Boston, Google Director of Search Peter Norvig (in)famously argued with Tim Berners-Lee that leading commercial providers see no need to standardise. However without standardisation there can be no real integration of services, no interoperability and no cross-platform knowledge retrieval. Successful knowledge sharing relies on a common meaning, syntax, definition and delivery mechanism, so that, standardising on information interchange increases the ability to share data throughout organisations (Home Office and PITO, 2006).

A possible solution appears when bottom-up development (key in knowledge sharing) is paired with formal modelling (key in knowledge retrieval). A recent development is SPRQL, a query language and data access protocol that incorporates the flexibility of the RDF data model, which has the potential to become a key component and could provide a common query language for all in Web 2.0 applications (Dodds, 2006).

4.3. Security

Security is another consideration and Web 2.0 is not immune to security breaches. Take for instance AJAX (Asynchronous JavaScript and XML), one of the key enablers

of the new generation of more interactive Web sites. One of the first Web applications to showcase this technique was Google Maps. While information in old-fashioned Web sites is passed through forms, AJAX allows for many more interactions with the browser and may run JavaScript on the client PC. It is thus open to a number of risks such as cross-site scripting, code correctness issues, object model violations, insecure randomness and poor error handling (Twynham, 2006).

However, to an extent we have been here before when organisations were seeking the benefits of the Internet without the inherent disadvantages. This was largely achieved through the deployment of Internet technologies within the organisations network boundaries and firewalls, giving rise to the intranet, and subsequently extranets and privileged access to customers and collaborators.

4.4. Maintenance and scalability

Who looks after the application? We have spoken of involvement and ownership by the user but there is the additional aspect of systems administration. The dot-com come e-commerce explosion of the late 1990s has left many organisations with a plethora of local e-commerce applications, which soak network bandwidth, equipment and maintenance time for minimal business return. This is seen in active programmes in many larger organisations seeking to locate and decommission these local solutions that prove to be an overhead rather than an asset. This indicates that local solutions still need to be framed in a wider organisational context and a strategy of collaborative activities and sharing that extends beyond individuals, workgroups or departments.

Closely related to maintenance, scalability issues can be both technical (network effects in the case of particularly popular applications) and financial (economic effects when the revenue does not scale with the application usage). While open source software can potentially minimise the financial burden of scaling/changing applications and platforms it has the drawback of lack of technical support, which dictates the need for in-house technical expertise.

5. Knowledge sharing beyond the conventional Web

It is often called deep, dark, invisible or hidden and is defined as the part of the Web that cannot (or will not) be indexed by search engines. It mostly comprises the contents of specialized databases that can be queried via the Web. The query results are delivered in dynamically generated web pages, whose storage is expensive and are therefore discarded as soon as the user reads them. Technical barriers related to the design and functionality of web crawlers mean that search engines cannot find or create these pages. Crawlers navigate the Web by following hyperlinks (a page with no links becomes "invisible") but can neither type nor "think". Hence, specialised databases that are searchable over the Web are inaccessible if they have no static pages with links containing information, so are Web sites that require login. The rest of the deep Web consists of the so-called excluded pages. They are certain types of pages that the search engines exclude by policy. They either contain special formats that hinder indexing (e.g. contents in Flash, Shockwave, images only etc.), or script-based pages (e.g. sites with URLs that contain the "?" sign).

In order to differentiate between the deep Web and the conventional one, the latter is also called surface Web. Writing for the BrightPlanet, a search technology company, Michael Bergman speculates that public information on the deep Web is currently 400

to 550 times larger than the conventional Web and that "a full ninety-five per cent of the deep Web is publicly accessible information - not subject to fees or subscriptions" (Bergman, 2001).

Looking at similar facts and metrics it becomes apparent that, if the surface Web plays a vital role in knowledge sharing, then, so does the deep Web. Yet, our tried and tested methods for knowledge retrieval and sharing can hardly be applied to so idiosyncratic a knowledge repository: there are no general tools for searching the deep Web although there are a number of subject directories to invisible Web databases, such as The Deep Web Directory (BrightPlanet, 2006).

Matters become even more complicated if we consider integrating the surface and deep Webs. Directed query technology and pre-assembled storehouses provide some support. However, directed query languages are cumbersome and require user expertise along with identification and downloading of the correct tools. On the other hand, pre-assembled storehouses support selected content and query customisation which disadvantages general requests and needs.

6. Conclusions and future work

If future competitive advantage lies in providing value-added to products and services, then knowledge creation becomes essential to providing this value-added. At the core of knowledge creation lies knowledge sharing and therein the need for collaboration. All of which, takes place in a global 24/7 environment where organisations themselves and their partners are geographically dispersed. This situation is mitigated and bridged through the adoption of information technology, which unfortunately has a history of falling short in its delivery.

Through our examination we have indicated that potential solutions already exist which can be harnessed accordingly, especially when utilising the underlying drivers as apparent in the social phenomena, something not unlike the necessities that underpin knowledge sharing and collaboration. More specifically our research acknowledges, investigates and addresses:

(1) the need for development with business analysts and end-user involvement;

(2) the importance and contribution of Social Software in bottom-up modelling and end-user empowerment;

(3) the impact of Web 2.0 technologies in bringing (1) and (2) together, bridging thus the socio-technical gap; and

(4) the shortcomings of this approach, as we move onto a more automated era of Internet applications and repositories beyond the conventional Web.

Essentially "developing from within" centres upon the location of the key knowledge and the understanding of the requirements, for these are both critical to the notion of knowledge creation and just-in-time knowledge and fundamental in providing the necessary value-added.

Located at the centre of the process, the knowledge worker is involved, engaged, empowered and owns the solution. Through usage by knowledge workers, this solution is neither a capital overhead nor one of the bottlenecks that litter the history of information systems development and knowledge management systems roll-outs.

For if we are better able to exploit the benefits of the information and knowledge that lies in the "surface web" we may then be able to consider the greater challenge of exploiting the latent value that lies "deep or invisible web".

Our conclusion is that "developing from within" provides an effective solution to the problem of knowledge sharing by means of the combination of the social and technical systems. This solution is facilitated by the social phenomena that underpin emerging technology developments, as apparent in Social Software, Web 2.0 and Semantic Web, but also hindered by a number of potential weaknesses, which we tried to foresee and identify. In future research we intend to further address these problems, detect their dynamics in relation to knowledge sharing and retrieval and propose possible solutions.

References

Aberer, K., Cudré-Mauroux, P. and Hauswirth, M. (2003), "Start making sense: the chatty Web approach for global semantic agreements", *Journal of Web Semantics*, Vol. 1 No. 1, pp. 89-114.

Aberer, K., Cudré-Mauroux, P., Ouksel, A.M., Catarci, T., Hacid, M.S., Illarramendi, A., Kashyap, V., Mecella, M., Mena, E., Neuhold, E.J., Troyer, O.D., Risse, T., Scannapieco, M., Saltor, F., deSantis, L., Spaccapietra, S., Staab, S. and Studer, R. (2004), "Emergent semantics principles and issues", Database Systems for Advanced Applications 9th International Conference, DASFAA 2004, *LNCS*, Vol. 2973.

Ackerman, M.S. (2000), "The intellectual challenge of CSCW: the gap between social requirements and technical feasibility", *Human Computer Interaction*, Vol. 15, pp. 179-203.

Bergman, M.K. (2001), "The deep Web, surfacing hidden value", *The Journal of Electronic Publishing*, University of Michigan, Michigan, IL.

Berners-Lee, T., Hendler, J. and Lassila, O. (2001), "The semantic Web", *Scientific American*, May.

Boutin, P. (2006), "Web 2.0: the new Internet 'boom' doesn't live up to its name", available at: www.slate.com/toolbar.aspx?action = print&id = 2138951 (accessed August 2006).

Brickley, D. (1999), *Semantic Web History: Nodes and Arcs 1989-1999 The WWW Proposal and RDF*, available at: www.w3c.org/1999/11/11-WWWProposal/ (accessed November 2005).

BrightPlanet (2006), *Deep Web Directory*, available at: www.completeplanet.com (accessed August 2006).

CIPD (2001), *Raising UK Productivity: Why People Management Matters*, CIPD, March.

Cudré-Mauroux, P. and Aberer, K. (2004), "A necessary condition for semantic interoperability in the large", CoopIS/DOA/ODBASE, (2), pp. 859-872.

Davis, I. (2005), "Talis, Web 2.0 and all that", *Internet Alchemy*, available at: http://iandavis.com/blog/2005/07/talis-web-20-and-all-that (accessed August 2006).

Dodds, L. (2006), "SPARQLing services", *Proceedings of the XTech 2006 Conference, Amsterdam*.

Gruber, T.R. (1993), "Towards principles for the design of ontologies used for knowledge sharing", in Guarino, N. and Poli, R. (Eds), *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Kluwer Academic Publishers, Deventer.

Haag, S., Cummings, M. and McCubbery, D.J. (2004), *Management Information Systems: for the Information Age*, 4th ed., McGraw Hill Irwin, New York, NY.

Herrmann, T., Hoffman, M., Kunau, G. and Loser, K-U. (2004), "A modelling methods for the development of groupware applications as socio-technical systems", *Behaviour & Technology*, Vol. 23 No. 2, pp. 119-35.

Home Office, PITO (Police Information Technology Organisation) (2006), *Implementing ISS4PS*, Vol. 2, available at: http://iss4ps.police.uk/

Lord Sainsbury of Turville (2006), *Launch of the Materials Knowledge Transfer Network*, available at: www.dti.gov.uk/ministers/speeches/sainsbury190106.html (accessed 19 January 2006).

Newell, S., Robertson, M., Scarbrough, H. and Swan, J. (2002), *Managing Knowledge Work*, Palgrave, Basingstoke.

O'Reilly, T. (2005), "What is Web 2.0: design patterns and business models for the next generation of software", available at: www.oreilly.com/lpt/a/6228/ (accessed August 2006).

Patrick, K. and Dotsika, F. (2006), "Reconsidering the development process when building knowledge management systems", *Proceedings of KMAC 2006, The Third Knowledge Management Aston Conference, Aston Business School, Aston University, Birmingham*, 17-18 July 2006.

Porter, M.E. and Ketels, C.H.M. (2003), "UK competitiveness moving to the next stage", DTI Economics Paper, No 3, May, DTI.

Twynham, S. (2006), "Ajax security", available at: www.it-observer.com/articles/1062/ajax_security/ (accessed August 2006).

W3C OWL (2004), *OWL Web Ontology Language Overview*, available at: www.w3.org/TR/owl-features/ (acessed April 2005).

web.2.con.com (2006), 2nd Web Conference, available at: www.web2con.com/ (accessed January 2007).

Whitworth, B. and de Moore, A. (2003), "Legitimate by design: towards trusted socio-technical systems", *Behaviour & Information Technology*, Vol. 22 No. 1, pp. 31-51.

Wikkipedia (2006), "Web 2.0", available at: http://en.wikipedia.org/wiki/Web_2.0

Further reading

Appelbaum, S.H. (1997), "Socio-technical systems theory: an intervention strategy for organisational development", *Management Decision*, Vol. 35 No. 6, pp. 452-63.

Barnett, A. (2005), "Web 1.0 and Web 2.0, Alex Barnett Blog", available at: http://blogs.msdn.com/alexbarn/archive/2005/08/13/451282.aspx (accessed August 2005).

Bloomfield, B.P. and Danieli, A. (1995), "The role of management consultants in the development of information technology: the indissoluble nature of socio-political and technical skills", *Journal of Management Studies*, Vol. 32 No. 1, pp. 23-46.

Bradley, P. (2006a), "Web 2.0 – a new generation of services, Part 1", *Library + Information Update*, p. May.

Bradley, P. (2006b), "Web 2.0 – a new generation of services, Part 2", *Library + Information Update*, June.

Bryant, L. (2003), *Smarter, Simpler, Social*, available at: http://headshift.com/moments/archive/sss2.html

Bryant, L. (2005a), *Making Knowledge Work*, available at: http://headshift.com (accessed August 2006).

Bryant, L. (2005b), *Introduction to Social Software for the Networked Social Enterprise*, available at: http://headshift.com (accessed August 2006).

Dotsika, F. and Patrick, K. (2005a), "Knowledge capture, sharing and maintenance in the semantic web age: a framework proposal", *Proceedings of the 4th Annual ISOneWorld Conference and Convention, March 30-April 1, 2005*.

Dotsika, F. and Patrick, K. (2005b), "From end-users to bots: the balancing act of Web-based knowledge search and sharing", *ICICKM 2005, Proceedings of the 2nd International Conference on Intellectual Capital, Knowledge Management and Organisational Learning, November*.

Eason, K. and Harker, S. (1996), "Representing socio-technical systems options in the development of new forms of work organisation", *European Journal of Work and Organisational Psychology*, Vol. 5 No. 3, pp. 399-420.

Elizen, B., Enserink, B. and Smit, W.A. (1996), "Socio-technical networks: how a technology studies approach may help to solve problems related to technical change", *Social Studies of Science*, Vol. 26, pp. 95-141.

Governer, J. (2005), "The differences from Web 1.0 to Web 2.0", *MonkChips*, available at: www.redmonk.com/jgovernor/archives/000884.html (accessed August 2005).

Kelly, K. (2006), "We are the Web", *Wired Magazine*, available at: http://wired.com/wired/archive/13.08/tech_pr.html (accessed August 2006).

LaMonica, M. (2006), "Google deal highlight Web 2.0 boom", *CNET News. Com*, available at: www.cnet.com.au/software/internet/0,39029524,40061041,00.htm (accessed 14 March 2006).

Lin, H-F. and Lee, G-G. (2006), "Effects of socio-technical factors on organisational intention to encourage knowledge sharing", *Management Decision*, Vol. 44 No. 1, pp. 74-88.

MacManus, R. and Porter, J. (2005), "Web 2.0 for designers", *Digital Web Magazine*, available at: www.digital-web.com/articles/web_2_for_designers/ (accessed 4 May 2005).

Margulies, N. and Colflesh, L. (1982), "A socio-technical approach to planning and implementing new technology", *Training and Development Journal*, December, pp. 16-29.

Ukn BBC (2006), "Google launches web spreadsheet", http://news.bbc.co.uk/1/hi/business/5051610.stm (accessed June 2006).

Ukn Google blog (2006), "Googler insights into product and technology news and our culture", available at: http://googleblog.blogspot.com/2006/03/writely-so.html (accessed June 2006).

Corresponding author
Keith Patrick can be contacted at: K.Patrick01@westminster.ac.uk

International Journal of Information Management 29 (2009) 407–415

Contents lists available at ScienceDirect

International Journal of Information Management

journal homepage: www.elsevier.com/locate/ijinfomgt

Uniting formal and informal descriptive power: Reconciling ontologies with folksonomies

Fefie Dotsika*

University of Westminster, Business Info Management, 35 Marylebone Road, London NW1 5LS, United Kingdom

ARTICLE INFO

Keywords:
Ontologies
Folksonomies
Web classification schemes
Information modelling
Web information management

ABSTRACT

Ontologies and folksonomies are currently the most prominent web content classification schemes. While their roles are similar, their engineering is different. In an attempt to combine and harness their distinct powers, web and information scientists are attempting to integrate them, merging the flexibility, collaboration and information aggregation of folksonomies with the standardisation, automated validation and interoperability of ontologies. This paper explores the basics of web information classification engineering, identifies the strengths and weaknesses of the existing methodologies, assesses their effectiveness and investigates a number of key quality issues. It then investigates the existing methods for integrating ontologies and folksonomies and examines the integration requirements. It finally proposes a common framework for reconciliation of the two classification approaches and quality assurance.

1. Introduction

Information management problems in organisations tend to be frequent, consistent and well defined. Their common parameter is a range of applications that communicate poorly, if at all, between them, due to a variety of data formats, points of data entry and overlapping or ineffectually defined views. Contrary to applications involving sets of well-defined and well-contained data, organisational knowledge is gradually accumulated, over a period of time—often without a strategy. As a result content management systems often lack the ability to compare and contrast data and often fail to retrieve the right information at the right time, even when all of the data required is stored in the same format and/or location.

Web information management and retrieval inherits the problems of information management, slightly intensified by the common (albeit unjustified) expectation that search engines and web content management systems are all the tools organisations need in order to locate and retrieve the information sought in a timely manner. While search and web content retrieval are the heart of web-based knowledge management, the web is a system in constant flux and its scale is well beyond that of traditional information retrieval. The inability to find the right information is mainly caused by the absence of a centralised mechanism for cataloguing data. Incompatible metadata lead to systems that fail to retrieve, compare and contrast information in a timely manner and system interoperability becomes unattainable. As a consequence, information retrieval and sharing are greatly compromised.

The interpretation, processing and retrieval of information according to a given knowledge representation schema are at the heart of information and knowledge management. Information classification of high quality helps to capture relationships and links between different pieces of knowledge and leads to well-designed representation schemas, which, in turn, facilitate the findability, retrieval and processing of the information. High quality classification is crucial in safeguarding system interoperability as it ensures the correct and successful mapping between the attributes, behaviour and functionality of systems' components. Apart from the obvious downside of information loss, lack of interoperability can result in a number of problems, such as semantic inconsistency, logical ambiguity and information redundancy. In order to avoid such problems, the semantics of the domain of discourse need to be agreed upon, formalised and represented.

Information modelling focuses on the representation of the semantics (Halpin, 1995) and is an essential part of the design process, whether formal or informal. It relies mainly on classification systems, as defined by the ISO Terminology Standards (ISO 704 and ISO 1087-1). These standards can record the basic classification features of any part of an information system, from typical classification (e.g. keyword list, taxonomy) to data modelling and behaviour (e.g. state diagram, organisation chart, computer program, narrative description). Classification schemes are subject-based systems. At their most basic they are *controlled vocabularies* and represent lists of concepts with no relations between them. Folksonomies, taxonomies and ontologies are currently the most prominent web content classification schemes. Their roles are sim-

* Tel.: +44 20 79115000.
E-mail address: F.E.Dotsika@westminster.ac.uk.

0268-4012/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijinfomgt.2009.02.002

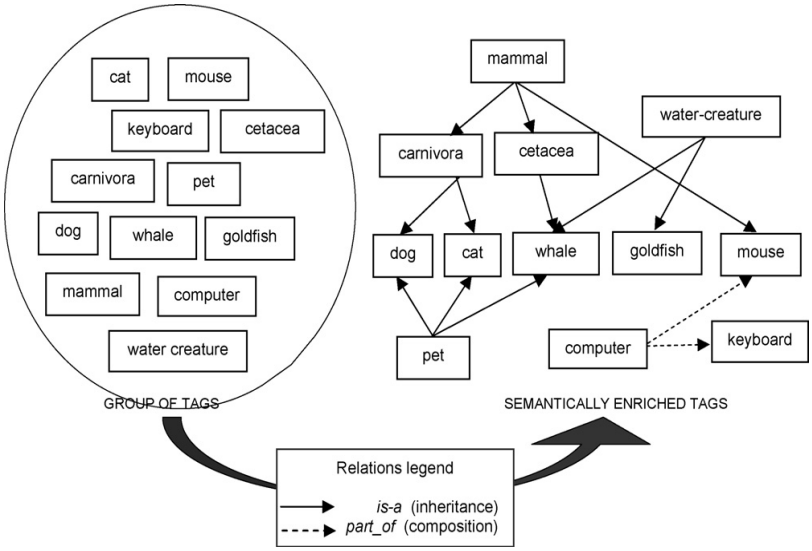


Fig. 1. Semantic enrichment of folksonomies.

ilar: they are all used for finding information on the web. Their engineering nevertheless is different. In particular:

Folksonomies are collaborative, user-generated metadata. They offer an informal way of online information categorisation, search and sharing. Folksonomies are a faceted classification scheme and are created bottom-up, in an analytical synthetic way, where the subject area is first divided into individual concepts which can be composed to construct complex subjects via appropriate sets of rules. Emergence of new concepts can be accommodated. They are characterised by their collaborative bottom-up design and are the prominent classification tagging scheme for Web 2.0.

Taxonomies are hierarchical classification schemes: metadata organised in hierarchical tree structures. They are used mainly in content management and they model entities and *is–a* relationships. Taxonomies are typically engineered top-down, in a hierarchical enumerative approach where the subject area is divided into increasingly narrower and more detailed categories, systematically enumerated. If the semantics they represent are extended with associational and equivalent relationships they are known as *thesauri*. While still widely used to provide a conceptual framework for analysis and information retrieval, taxonomies have been superceded by ontologies. Within the context of web classification schemes, and therefore within this paper, taxonomies can be thought of as a specialised case of ontologies (often refered to as *lightweight ontologies*).

Ontologies are sets of shared, explicit and formal concepts used to organise and classify content. They are structured metadata representing sets of concepts and their relationships within a domain. They model entities, logical constraints and relationships in the form of directed graphs. Like taxonomies, they are top-down schemes where knowledge is modelled in classes, properties and relationships and where the taxonomic *is–a* associations are extended to include additional types that allow a more refined semantic modelling. They enable the use of automated reasoning tools that can provide conceptual search and retrieval, decision support and knowledge management services. The ontology language OWL is the top part of the Semantic Web framework.

In an attempt to combine and harness the distinct powers of the two classification approaches, web and information scientists are attempting to integrate them, merging the bottom-up flexibility, collaboration and information aggregation of folksonomies with the top-down standardisation, automated validation and interoperability of ontologies. Folksonomies become thus semantically enriched (Fig. 1). Depending on the process, the resulting metadata are known as *folksontologies* (Van Damme, Hepp, & Siorpaes, 2007), *semantically enriched folksonomies* (Angeletou, Sabou, Specia, & Motta, 2007), *flexonomies* (Kapetanios & Schaal, 2007), or, simply, *integrated folksonomies* (Specia & Motta, 2007). However, neither the process of mapping folksonomy tags to domain ontologies nor that of mining folksonomies from ontologies are straightforward operations. Folksonomies come with inherent ambiguity and a flat organisation. Ontologies are unambiguous and semantically rich but are rigid and lack user consensus on a domain view.

The resulting methods are mainly experimental, lack serious automation, and/or fail to address certain quality issues. Without agreed guidelines and even practitioners' consensus, there is no methodology for the integration of the two modelling methods. This is the gap that this paper aims to bridge. It examines the existing approaches, identifies their strengths and weaknesses, assesses their effectiveness and investigates a number of key quality issues. It determines the integration requirements and finally proposes a common framework for the reconciliation of the two classification approaches, which facilitates quality assurance.

The rest of the paper is organised as follows: in Section 2 we examine the various types of web information modelling systems. In Section 3 we explore the existing methods for integrating ontologies and folksonomies and examine the processes involved. Section 4 gives a review of the most prevalent existing methodologies that assess quality issues. Section 5 investigates the integration requirements, identifies the strengths and weaknesses of the existing methodologies and proposes a common framework for reconciling the two classification approaches. In Section 6 we draw our conclusions and outline future work.

2. Modelling information for the web

While the evolution of the next generation of web is already underway, the speculation about its model's origin is widening. The two most prominent trends are the *Semantic Web* (Berners-Lee, Hendler, & Lassila, 2001) and the *Web 2.0* (O'Reilly, 2005). Following the dual trend, web information modelling relies mostly on folksonomies (best suited for the sharing and co-operative model

of Web 2.0) and ontologies (best supported by the Semantic Web infrastructure).

2.1. Folksonomies: modelling information bottom-up

The power of user-generated tagging arises from its bottom-up consensus that associates keywords with content. Users decide on their own tags and tagging schemes without the use of restricted vocabularies, pre-defined categories or domain expertise. Rafferty and Hidderley (2007) point out that *democratic indexing* is an alternative approach to concept-based retrieval. It differs from expert-led models by focusing on user interpretation and from other image retrieval systems such as *Flickr*, by being user-indexed rather than author-indexed. The system creates a number of classification templates capturing a wide range of image information, such as biographical, structural, content-related (overall and object-based) and interpretative (overall mood and object-based mood).

Against all odds and the belief that collaborative tagging is useless and chaotic, it has proved to be effective for organising personal (Mathes, 2004) and corporate (Patrick & Dotsika, 2007) information, blog searching (Ohkura, Yoji, & Hiroshi, 2006), facilitating innovation (Udell, 2004; Hayman, 2007) and enabling the discovery of marginalised information such as in the area of the so-called *long tail* (Anderson, 2006). Despite the differences in the tagging process, user interactions and the complexity of individual categorisation, users of folksonomies share universal behaviour following simple activity patterns (Cattuto, 2006).

Folksonomy enthusiasts however fail to give serious evidence that bottom-up tagging can deal with issues of interoperability across distributed knowledge repositories, automated search and quality of information retrieval. The problems inherent in uncontrolled vocabularies, namely ambiguity, inconsistent granularity, duplications and synonyms, lead to compromised content retrieval (e.g. searching for objects tagged as 'pet' will not necessarily retrieve those tagged as 'cat' or 'hamster') and quality problems which are difficult to resolve.

Folksonomies can be grouped according to the system supporting the tagging process (e.g. plain labels for webpage content, numbers for tagging related to ratings or geographical information etc.). They can be either

(a) narrow, when one or few people provide tags mostly for their later personal retrieval (e.g. tagging in Flickr; Rafferty & Hidderley, 2007) or
(b) broad, when many people publicly tag the same items for their own use, each one with their own vocabulary (e.g. tagging in *del.icio.us*; Vander Wal, 2005).

There is no one correct way for developing folksonomies. Their creators can only rely on best practice.

2.2. Ontologies: modelling information top-down

The power of ontologies lies in their expressiveness and overall effectiveness in modelling information accurately and consistently so that they enable automatic reasoning, concept-based searches and knowledge discovery by means of intelligent agents (Hendler, 2001). They reduce ambiguity, enable validation and standardisation and facilitate sharing (W3C, 2004). By enabling querying and reasoning support at runtime, quality is enhanced and costs cut through increases in consistency (by reducing maintenance overhead) and reuse potential (W3C, 2006). Compared to folksonomies, while ontologies can do everything that folksonomies can, the opposite is not true.

The Semantic Web vision however has been slow in delivering its promise of interoperability. While the first article on the Semantic Web appeared in 2001 in the Scientific American (Berners-Lee et al., 2001), it still remains a work in progress. Criticisms include the elusiveness of the "global ontology" and the expertise needed for engineering formal specifications, a process that takes web content publishing away from the end-user. There is also the difficulty of ontologies to deal with uncertain knowledge (Ding, Peng, & Pan, 2004) and the underlying technologies' and inference tools' expensive needs in terms of memory and processing time (Preuveneers & Berbers, 2006).

Web ontologies are used to categorise a range of resources, from web sites to products sold online. The different types of ontologies are therefore dictated by the application area and the modelling process.

In terms of *granularity* (level of abstraction) ontologies can be identified as low (domain), mid-level and upper ontologies. In terms of *language formality* ontologies can be informal (natural language), semi-formal (defined in either restricted and structure natural language or semi-formally defined language) and formal (defined by means of formal semantics).

Depending on the *domain* they define, ontologies can be grouped in four categories (IBM, 2004):

(a) role-based ontologies (terminology and concepts relevant to a particular user, person or application)
(b) process ontologies (terms, relationships, constraints, input and output relevant to a particular process or group of processes)
(c) domain ontologies (terminology and concepts relevant to a particular topic)
(d) interface (structure and content restrictions relevant to an interface).

There is no one correct way for developing ontologies. Their modelling is an iterative process and the result a solution among other viable answers. However there are a few methodologies for their design. Guarino (1998) proposed four ontology design principles (domain clarity, application of the identity criterion, identification of a basic taxonomic structure and explicit identification for roles) which we will examine in further detail in the following section. Despite the possible alternatives, the clues for best practice are provided by the requirements and future scalability of the application and/or domain.

3. Existing methods in reconciling the classification approaches

While the Semantic Web creators see the power of Web 2.0 sophisticated data interfaces, collaborative content generation, search and sharing, most Web 2.0 enthusiasts know that their collaboratively engineered content fails to deliver a platform for automated search, intelligent agents and inter-application integration. As a result there is an increasing number of methods that aim to bring together the bottom-up approach of folksonomies with the traditional top-down design of ontologies. These methods seek to reconcile the differences of the two classification schemes while preserving their advantages.

3.1. FolksOntologies (Van Damme et al., 2007)

By far the most comprehensive approach, this method integrates multiple resources and techniques for deriving ontologies from folksonomies following a number of steps.

In the first instance folksonomies and their associated data are analysed so that relevant relations may be determined. Emergent semantics of sub-communities of interest are discovered by analysing the tags used. These communities can be explicit or

implicit. In explicit communities the users have similar interests and/or expertise that they have made explicit by joining a specific social network, user group or other domain of interest. Implicit communities are identified by the sharing of the same tags and/or objects. Associated data of folksonomies can also be derived from folksonomy-driven web sites (systems). Similarly, the linking of systems can be either explicit (through social networks their users are members of) or implicit (through shared sub-communities of interest).

The output of previous process is then complemented with information from online lexical resources. Such resources can be online dictionaries but also Google and Wikipedia. While dictionaries provide reliable definitions of words, Wikipedia contains definitions of words not yet well established to be found in dictionaries. Google's dictionary functions provide suggestions, if for instance the user has misspelled a word, or if a similar keyword results in more hits than the user word of choice.

The third step involves a secondary level of resources derived from open source ontologies and Semantic Web resources such as the search engine Swoogle and Wordnet. Swoogle indexes metadata and computes relationships between them, while Wordnet provides information on synonyms, homonyms and meronyms which facilitates communication between different ontologies.

The final step is ontology mapping techniques where conceptual elements can be matched based on the labels, ontology structure or both. This method can be used to identify relationships between tags and between tags and elements in existing ontologies. Mapping and matching techniques can be based on formal classification theory, semantic matching etc.

3.2. Making explicit the semantics behind the folksonomy tag space (Specia & Motta, 2007)

The method aims to integrate folksonomies with the Semantic Web by employing occurrence analysis and clustering techniques to identify ontologies corresponding to (meaningful) groups of tags, while ontology querying by means of Semantic Web search engines can provide relationships between tags. While preliminary results have been encouraging, more work needs to be carried out towards the improvement of the clustering technique and the automation of the system before the method can be adequately evaluated.

3.3. Semantically enriched folksonomies (Angeletou et al., 2007)

This approach extends the research carried out by Specia and Motta (2007) by employing the ontology matching algorithm presented in (Sabou, d'Aquin, & Motta, 2006) to automate the *harvesting* process that is the dynamic selection, combination and exploitation of relevant knowledge derived from online ontologies. It therefore enriches folksonomies semantically by means of harvesting Semantic Web resources and enables the automated discovery of semantic relations between the tags of various clusters of related tags.

The method was tested through two main experiments. The first one ran the Sabou et al. algorithm on the clusters generated by Specia and Motta. The results were rather poor. In the second experiment the clusters of tags were selected directly from Flickr's cluster generator. The same algorithm was applied and the results were better. However the algorithm used enables the discovery of only *is–a* or *no* relations between tags, leaving out the more generic types of relations.

3.4. Flexonomies (Kapetanios & Schaal, 2007)

This method presents flexonomies, a theoretical approach to organising and sharing tags. The proposed model is based on a high-dimensional space and an algebra for accessing and manipulating a flexonomy. The model and algebra act as a "mathematical and ontological foundation for organising and sharing contextualised and personalised semantic tagging and annotation in digital libraries". Further work is currently under way on the specification of a query language and on the implementation of a flexonomy prototype that will provide a collaborative environment for semantic tagging of shared bibliographic entries.

3.5. From folksonomies to networks of terms to ontologies (Lux & Dosinger, 2007)

The emergence of semantics from folksonomies here is a three-step process during which a lightweight ontology is derived from an original folksonomy. In the first instance the tags are statistically analysed and a *tag cloud* (i.e. a set of related tags depicted in different font sizes and colours according to their weight/cardinality) is produced.

During the next step, a weighted, directed *network of tags* is created by means of computing their co-occurrence (i.e. their similarity, as determined by their assignment to the same resources). In order to reduce a number of quality drawbacks associated with tag networks, merging and filtering are applied. Dictionary-based filtering eliminates tags that do not appear in the given dictionary, thesaurus-based merging combines synonyms, algorithm-based cleaning based on the *string edit distance* (Navarro, 2001) eliminates typos and *stemming* (Baeza-Yates & Ribeiro-Neto, 1999) merges alternative spellings (e.g. plural and singular forms, such as *toy* and *toys*). The tag network is thus turned into a *term network*.

The creation of an ontology from the term network is the third step. The method highlights the difficulties involved, especially deriving concepts from terms and identifying the relations between concepts. A number of strategies are mentioned, such as the use of previously generated knowledge from a similar domain or community or the semi-automated classification and filtering of terms for concept identification.

3.6. Capturing latent emotional semantics in social tagging systems (Baldoni et al., 2008)

This approach adds a semantic layer to the social tagging of *Arsmeteo*, a web portal for sharing works of art containing a folksonomy of over 10,000 tags. These tags are related to *OntoEmotions*, an OWL ontology chosen for fitting the application purposes. The method is based on measuring the relationship strength between the Arsmeteo tags and the OntoEmotions terms by means of correlation coefficients. The correlation between tags and terms is calculated based on the occurrences of the corresponding words, counted in Google.

Other methods attempt to bridge the gap by employing the use of *faceted ontologies* (Schmitz, 2006) or *enrich folksonomy tags with hierarchical relations* (Heymann & Garcia-Molina, 2006). A number of techniques identify groups of related tags but do not identify the particular semantics of these relations (Mika, 2005; Begelman, Keller, & Smadja, 2006; Wu, Zhang, & Yu, 2006).

4. Quality issues

Quality issues are at the core of web information systems and as such need to keep up with the dynamic expansion of the information set. The need for quality control in web information classification systems is particularly prevalent and forms one of the main requirements for the support of web knowledge integration and management. This type of quality control has been identified as a vital part of the framework used for the capturing, accessing

and distributing of web knowledge in the next generation of web (Dotsika & Patrick, 2006).

4.1. Folksonomies and quality

While professionally created metadata are often considered of high quality, they are costly in terms of time and effort required to produce them. This has resulted in a considerable increase in the adoption and popularity of folksonomies whose tagging mechanism, even if semantically inferior to ontologies, allows an intuitive browsing of the information collection.

With no controlled vocabulary involved, folksonomies also appeal because of their lack of political, social or cultural bias and the opportunity they present to represent the *long tail* that is minority tastes and interests. They are central to developing systems from within, a technique which improves the likelihood of success through the adoption social computing practices (Patrick & Dotsika, 2007).

The Proof of Concept studies at The Metropolitan Museum of Art, carried out research that compared tags assigned by trained and untrained cataloguers to existing museum documentation (Trant, 2006). The aim of the project was to explore the potential for improving access to museum collections by means of social tagging. The results were particularly encouraging and illustrated that the less formal, participatory and distributed nature of folksonomies can enhance museum documentation with content that reflects the interests and perspectives of the museum communities.

Despite their popularity however, folksonomies are particularly affected with quality problems, due to their nature: tags are not replacement for formal systems. They can be (and usually are) ambiguous and inexact. Lack of a controlled vocabulary means masses of tags describing the same things. The consequence of this is poor searching. There are currently no actual methodologies and no rigorous design methods addressing these problems. Guidelines and best practice are all the users are armed with most of the time.

Quality troubles with user-created keywords fall into one of the following categories (Golder & Huberman, 2005):

(a) Polysemes and homonyms: words with many meanings which cause ambiguity. Polysemes share etymologies (e.g. the *bank* and to *bank*), whereas homonyms do not (e.g. river *bank*).
(b) Synonyms: multiple words with the same or closely related meaning that cause inconsistency) and
(c) Discrepancies in granularity: when there is variation of basic level tags (*bank*, *banks* and *banking*).

Finding ways to minimise – if not eliminate – these problems can enhance the quality of the folksonomy. However there are no actual methodologies addressing this. Commonly used tags are often encouraged instead of infrequently or single-use tags. The reason behind this is that repeated tags often share a social shared meaning, alongside the personal meaning (Mejias, 2004).

Another quality issue is that of the so-called 'sloppy' tags. Their critics want them extinct, their supporters claim that they can be very helpful in particular searches. Tidying up too neatly can cost the flexibility that made folksonomies so popular in the first place (Shirky, 2005) and can compromise or totally eliminate the presence of the *long tail*.

4.2. Ontologies and quality

Despite their role as the backbone of automation in the Semantic Web, ontologies are not free of quality trouble. For instance, it is difficult to derive ontologies out of large domains with no formal categories, unstable (or simply dynamic) entities and naive users of no authority. Interestingly enough, the web is such a domain.

Colomb and Weber (1998) focus on the issue of *ontological completeness* before addressing that of quality. They define it in terms of the relationship between an organisation's semiotic system and the information system that describes it. However ontological completeness is not well defined, as, invariably, it depends on the views of different stakeholders. Colomb and Weber propose the use of the better-defined *ontological adequacy* instead, a property of the semiotic system used to specify the information system. Ontological adequacy is evaluated with respect to the generalised ontology developed by Bunge (1977, 1979) and later summarised by Weber (1997). Having ensured ontological adequacy, *ontological quality* is defined as the degree of visibility of the organisation's semiotic system in the information system's semiotic system. This visibility is in fact determined by how much the systems' semiotics match those of the organisation.

According to Rector, Wroe, Rogers, & Roberts (2001) quality assurance criteria are, ideally, established during the phase of the original design and modified iteratively. They are based on a combination of the following four factors:

(a) intermediate representations adapted to domain expert authors' user views,
(b) domain expert authors' guidelines,
(c) underlying ontology schemas and transformation rules to the intermediate representation and
(d) natural language generation lexicons and grammars for result displaying.

Kashyap (2003) introduces the *structural* and *atomic* ontological qualities. Structural quality consists of the notions of semantic richness, internal consistency and completeness of domain coverage. Atomic quality encompasses the quality of concepts and relationships, the quality of axioms and constraints and the notion of ontological commitments. This approach shows that, in all aspects of information retrieval and integration, trust is linked to quality.

Looking into the subject from a different perspective, quality problems have also been blamed on the (mis)use of the expression. Bringing the term ontology closer to the logical theory and semantics deprived it from all direct relation to reality (Smith & Welty, 2001). As a consequence of losing their grounding, ontologies cease to perform well in linking different conceptual models with overlapping semantics.

Formalising ideas further, Guarino (1998) links the overloading of the *is–a* relation to semantic problems related to the use of linguistic ontologies (he uses *is–a* to mean the main taxonomic relation, pointing out that it is not the same as the *InstanceOf* relation which links a node to the class it belongs to and is not a partial order). He demonstrates how overloading compromises quality and proposes four design principles that solve the overloading problem as follows:

(a) Domain clarity based upon the nature of the entities modelled. They can be (i) particulars (that is individuals of any world), (ii) universals (conceptual properties and relations) or (iii) linguistic entities (such as nouns, verbs etc.). While particulars and universals justify two separate ontologies, lexical items should be kept out of the domain.
(b) Identity issues are associated with the *identity criterion*[1] (IC). ICs are, in practice, difficult to express for classes corresponding to natural language. Instead of the optimum *sufficient* conditions for identity, we can determine *necessary* conditions that will allow to: (i) identify an entity as an instance of a given class

[1] Identity Criterion is formally defined as: for a property P an IC is a binary relation Ip (P *carries an* IC for its instances) such as: $Px \wedge Py \wedge Ipxy \rightarrow x = y$.

C, (ii) re-identify an instance of C across time (persistence) and (iii) count the instances of C.

(c) Follow a basic taxonomic structure (a *basic backbone*) of categories and types. Under the assumption that each type has a different set of ICs, the types form a tree of mutually disjoint classes. Categories can also form a shallow tree of mutually disjoint classes.

(d) Explicit identification of roles. The advantages of this principle are (i) tags can easily be hidden to isolate the *basic backbone* and (ii) deduction is possible involving mutual disjointness while avoiding explicit declarations.

Another approach that arises from ontological analysis in Philosophy, discusses identity, unity, essence and dependence of formal ontologies (Welty & Guarino, 2001). This methodology aims to clarify modelling assumptions and, as a result, claims to facilitate conceptual modelling, analyse taxonomic links and help identify the backbone taxonomy.

4.3. Existing standards: RDF and Topic Maps

One can debate that the next generation of web sets the foundation for the next generation of information architecture. New standards (or, rather, enforcement of existing ones) will address the issues raised and will improve quality of information. However, there is a standards' debate concerning both strategic and quality issues. The two possible standards are RDF (Resource Description Framework), a model developed by the W3C for representing information about resources in the World Wide Web and Topic Maps, a model for knowledge integration developed by the ISO.

Topic Maps (Pepper, 1999) originate in the early 1990s from work on managing documentation indices. They support high-level indexing of sets of information resources in order to enhance information find-ability (Maicher & Park, 2005). Topic Maps were adopted as an ISO work item in 1996 and the Topic Map standard ISO 13250 was published in early 2000.

RDF originates in MCF (Meta Content Framework) (Guha & Bray, 1997) which was made into an XML application and published as a W3C Recommendation in 1999 (Brickley, 1999). It is part of the *Semantic Web* framework and provides structured metadata about resources and a foundation for logical inference.

The similarities between Topic Maps and RDF are noteworthy, though the two communities became aware of one another in 1999. WC3 has carried out a survey of interoperability proposals for integrating the two standards (W3C, 2005). A closely related debate concerns the use of *Subjects* vs. that of *Resources*, as in the case of URIs (Universal Resource Identifiers) and PSIs (Published Subject Indicators), a resource which describes (part of) a vocabulary and provides URIs for terms in that vocabulary.

Mere adoption of standards however is not enough to safeguard web information quality. Information modelling is of paramount importance here and, if done ineffectively, no amount of standardisation can overcome the problems created. Quality issues in web information modelling are directly related to the choice of the classification scheme.

There are an increasing number of methodologies that deal with assessing the quality of ontologies. However, it is a different story when it comes to folksonomies. The problems inherent in an uncontrolled vocabulary, such as ambiguity, duplications and synonyms, lead to quality problems which are difficult to resolve. Due to their different nature (ontologies are rigorous and standardise-able classification schemes with a top-down design, whereas folksonomies are often ad hoc and bottom-up), the quality issues associated with them differ so that the two schemes cannot possibly adhere to the same quality assurance methodologies.

5. Integration requirements

But what exactly are the requirements in integrating the two approaches? And where does one start? Choosing tags from within the boundaries of controlled vocabularies is the first recommended step (Macgregor & McCulloch, 2006). However there is evidence suggesting that end-users should participate in the development of controlled vocabularies (Abbott, 2004; Mai, 2004). Folksonomies evolve and grow faster than ontologies: high occurrence of new popular tags can benefit the Semantic Web by enriching existing ontologies. As we move from folksonomies to structure semantic networks and ontologies there is the need for expertise not available in the broad spectrum of end-users, hence the need for automated processes (Hess, Maass, & Dierick, 2008). Bringing everything together we can now evaluate the existing methodologies, assess the quality requirements and finally propose a common framework for the integration of the two approaches to information modelling.

5.1. The methodologies revisited

The methods described in Section 3 have shown the way forward but they are not without problems and shortcomings. The folksontologies method (Van Damme et al., 2007) provides no actual case studies, experiments with sets of data or metrics of success or failure. There is no clear indication about which steps and sub-procedures are carried out manually and which ones are automated, especially the steps that involve online lexical resources, ontologies and Semantic Web resources. This approach also introduces certain issues of trust in social networks, which, although not a problem of the particular method as such, highlights possible obstacles in fully exploiting online resources. For instance, in order to establish mappings between own tags and those of one's peers one needs to import those tags belonging to friends and connections in the first place, an action that implies trust.

From the two methods that aim to enrich folksonomies by identifying the relationships between tags, the first one (Specia & Motta, 2007) lacks automation and needs further improvement of the clustering technique. The second one (Angeletou et al., 2007) takes the effort further and proves that it is possible to automate the process of semantically enriching folksonomy tags. However there are still some drawbacks, such as the enrichment algorithm used (it does not identify generalised relations and is based on strict string matching) and some difficulties in dealing with certain inherent characteristics of folksonomies and ontologies (handling of attributes, novel terminology, complex tags, lack of ontologies etc.). All this will be considered and addressed in future work.

The flexonomies model (Kapetanios & Schaal, 2007) constitutes an interesting premise, although it is purely theoretical at this stage and therefore lacks the impact of case studies and experimentation or simply worked examples. Despite its mathematical rigor, its practicality is not immediately clear. Unless the tag creation process is largely undertaken in an automated environment not requiring specialised expertise (i.e. in goes the multiset of personalised and contextualised tags and out comes the relevant flexonomy) the method potentially creates the exact same problems as the ones it tries to resolve.

The Lux and Dosinger (2007) approach that derives lightweight ontologies from tag and term networks is mainly experimental and incomplete. The term network generation is based on simple methods that can be applied to any folksonomy. However these methods are efficient only when the folksonomy is small, that is when the tags and resources involved are limited. Also, the part that deals with the generation of the actual ontology gives the results of some experiments on concept identification and clustering but no

Table 1
Existing methods.

Feature	Method					
	Van Damme et al.	Specia & Motta	Angeletou et al.	Kapetanios & Schaal	Lux & Dosinger	Baldoni et al.
Analysis to determine relations	√	Partial	Partial (*is–a*)	N/A	√	√
Tag cleaning/quality control	Partial	×	×	N/A	√	×
Ontology mapping techniques	√	√	√	√	√	√
Handling of attributes/complex tags	×	×	×	N/A	×	×
Use of multiple resources	√	×	Partial	√	√	×
Automation	Partial	×	Partial	√	Partial	Partial
Worked examples/case studies	×	×	×	×	×	√
Evaluation/metrics/results	×	×	Some	×	Some	Some

Table 2
Quality criteria framework.

Criteria	Ontologies	Folksonomies
Quality assurance criteria should be established during the original design	√	√
The relationship between an organization's semiotic system to the information system that describes it should be proportional.	√	By default
Overloading of the *is–a* relation should be avoided	√	×
Balance between logical theory and reality	√	×
Use of a basic taxonomic structure	√	Could benefit
Avoidance of ambiguity and inconsistency	By default	√
Discrepancies in granularity should be avoided	By default	√
Issues of trust	√	√

indication of work on edge classification. It overall highlights the problems rather than propose an actual method or procedure.

Baldoni et al. (2008) finally is a case study rather than complete methodology. It uses a pre-defined ontology to map the folksonomic tags generated in the Arsmeteo portal.

Table 1 is a comparative matrix that summarises our findings.

5.2. Quality assurance revisited

Web classification schemes' quality requirements are dependent upon the actual application of the information involved. Tagging information for personal use requires almost minimal quality assurance, whereas information quality in web-based knowledge management is of paramount importance. Information quality is as central to system interoperability, as system interoperability is to knowledge management. Despite its importance, quality assurance does not seem to be an integral part of the integration approaches examined.

Bringing together the existing quality assurance methodologies, design processes and best practice guidelines we have examined so far, we can now recommend a list of criteria that can be used as a quality assurance framework for web classification schemes. Some of the criteria are applicable to ontologies only, some to folksonomies and some to both. Entries marked "by default" assume the traditional design of the particular classification.

Table 2 summarises the quality criteria.

5.3. Towards a common framework

Putting together all the information and methods gathered so far we can now identify a number of requirements for integrating ontologies and folksonomies.

(a) Quality issues. Although not explicitly addressed by the methods reviewed, quality assurance is central to the selection, analysis, enrichment and mapping of tags. This requirement applies to both folksonomies and ontologies.

Folksonomy cleansing is probably the first integration requirement. There are a number of problems associated with the choice of tags used. Folksonomies are particularly affected with quality issues and can be (and usually are) ambiguous and inexact. Lack of a controlled vocabulary means masses of tags describing the same things. There are currently no actual

Table 3
Integration requirements.

Requirement	Nature/specifics	Domain/focus	Potential problems
Quality issues	Concatenated tags (bank account) Variations (bank, banks, banking) Polysemes (the bank and to bank) Homonyms (bank and river bank) Synomyms (bank, deposit, pay-in)	Folksonomy	Lack of automated tools Ambiguity and inconsistency issues Intended vs. accidental granularity
	Completeness Use of a basic taxonomic structure Overloading of the is-a relation Domain experts' guidelines	Ontology	Non-proportionality between an organization's semiotic system and the information system that describes it
Semantic enrichment	Analysis and clustering Relation identification Attributes and properties	Folksonomy	Broad vs. narrow folks Lack of sophisticated automated tools
Mapping completeness	Novel terminology Instances Multilingual tags	Folksonomy	Inability to map through
	Specialised expert knowledge	Ontology	
Trust and ethics	Harvesting information across systems implicitly or explicitly connected	Social networks	Information used against the wishes of stakeholders.

methodologies and no rigorous design methods addressing these problems. Guidelines and best practice are all the users are armed with most of the time.

Ontology quality assurance is also of paramount importance. Finding online ontologies to enrich and ultimately match user-defined tags when the ontology in question is of poor quality is clearly unwise. Several methods have been proposed related to the completeness, domain expert authors' guidelines, underlying ontology schemas and transformation rules, natural language generation lexicons and grammars, levels of structural and atomic ontological qualities etc. (Colomb & Weber, 1998; Rector et al., 2001; Kashyap, 2003).

(b) Semantic enrichment. The tags are analysed, clustered and semantically enriched by means of harvesting online resources. The existing methods deal with this process with various success rates. However they all gloss over the additional information contained in tags that semantically corresponds to attributes and/or properties.

(c) Mapping completeness. The requirement that tags can be mapped to some part of an existing ontology. Currently there are a number of issues that may prohibit this, ranging from an altogether absence of a relevant ontology to special tags that cannot be adequately mapped (see Table 1). The inverse situation (i.e. an ontology, usually specialised, with no corresponding folksonomy) is also included, though not properly addressed at this stage.

(d) Trust and ethics. These originate in the explicit and implicit links between systems using tags for information modelling. Explicit links arise from social network members' profiles and/or multiple such memberships. Implicit links are created through shared sub-communities of interest and/or common objects. The problem here is that harvesting information from several systems (e.g. importing someone's personal tags and creating mappings between them and own tags) implies a level of trust and possibly raises a certain level of ethical questions.

Table 3 presents the integration requirements.

6. Conclusions and future work

The research carried out in this paper addresses the engineering, reconciliation and integration of web information classification schemes. Its main contribution is that it attempts to bring together a number of recent and current projects which address the same issues but have been operating in silos. In particular:

- It identifies folksonomies and ontologies as the main web information classification schemes and briefly outlines the facts of their engineering and their spheres of influence.
- It investigates existing methods for the reconciliation of the two approaches, evaluates their effectiveness and determines possible shortcomings.
- It ascertains the need for quality assurance throughout the reconciliation process and determines that the adoption of a standard, such as Topic Maps or RDF, enhances web information quality and improves findability.
- It establishes that, due to the essential difference in design between the two classification schemes (top-down for ontologies, bottom-up for folksonomies), there can be no common framework for quality assurance.
- It determines both the need for and the absence of a standard methodology for web information modelling to complement the design and development of ontologies and folksonomies.
- Finally it proposes a list of requirements and a common framework for the integration of the two methods and explores obvious and not-so-obvious obstacles.

Due to the nature and currency of the research a number of online resources often considered less authoritative had to be consulted.

The next generation web, be it Semantic Web-based, Web 2.0-based or a hybrid mix, will intensify the efforts towards system interoperability and therefore successful information modelling and retrieval. The adoption of the dual approach of 'bottom-up population, top-down standardisation' necessitates a flexible yet rigorously regulated interface between the two parts. Although this paper proposes a sound foundation for reconciliation and delivers a comprehensive specification of requirements, it is by no means an all-inclusive solution. Future research will further investigate the integration requirements and will address certain problems and practicality issues raised in the previous section. In particular it will explore the *organisational* aspect of this dichotomy, that is, how organisations manage (or intend to manage) the reconciliation of the two classification approaches on a practical level.

References

Abbott, R. (2004). Subjectivity as a concern for information science: A Popperian perspective. *Journal of Information Science*, *30*(1), 95–106.

Anderson, Chris. (2006). *The long tail: Why the future of business is selling less of more*. New York: Hyperion. ISBN 1-4013-0237-8.

Angeletou, S., Sabou, M., Specia, L., & Motta, E. (2007). Bridging the gap between folksonomies and the Semantic Web: An experience report. In *Workshop: Bridging the gap between Semantic Web and Web 2.0, European Semantic Web conference*.

Baeza-Yates, R. A., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. Addison-Wesley Longman Publishing Co., Inc.

Baldoni, M., Baroglio, C., Horváth, A., Patti, V., Portis, F., Avilia, M., & Grillo, P. (2008). Folksonomies meet ontologies in ARSMETEO: From social descriptions of artifacts to emotional concepts. In *Proc. of Formal Ontologies Meet Industry (FOMI 2008)* Torino, Italy, June.

Begelman, G., Keller, P., & Smadja, F. (2006). Automated tag clustering: Improving search and exploration in the tag space. In *Proc. of the collaborative web tagging workshop at WWW'06*.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). *The Semantic Web*. Scientific American. May.

Brickley, D. (1999). [Online] *Semantic Web history: nodes and arcs 1989–1999 the WWW proposal and RDF*. Available from www.w3c.org/1999/11/11-WWWProposal/.

Bunge, M. (1977). *Treatise on basic philosophy: Volume 3: Ontology I: The furniture of the World Reidel*. Holland: Dordrecht.

Bunge, M. (1979). *Treatise on basic philosophy: Volume 4: Ontology II: A world of Systems Reidel*. Holland: Dordrecht.

Cattuto, C. (2006). Semiotic dynamics in online social communities. *The European Physical Journal C-Particles and Fields*, *46*(August (Suppl. 2))

Colomb, R. M., & Weber, R. (1998). In N. Guarino (Ed.), *Proceedings of the international conference on formal ontology in information systems (FOIS'98) Trento, Italy, 6–8 June 1998. Formal ontology in information systems*. Amsterdam: IOS-Press. pp. 207–217.

Ding, Z., Peng, Y., & Pan, R. (2004). A Bayesian approach to uncertainty modeling in OWL ontology. In *Proceedings of the international conference on advances in intelligent systems – theory and applications, IEEE, Luxemburg, November 2004*.

Dotsika, F., & Patrick, K. (2006). Towards the new generation of web knowledge search and share. *VINE: Journal of Information and Knowledge Management Systems*, *36*(4), 406–422.

Halpin, T. A. (1995). *Conceptual schema and relational database design* (2nd ed.). Prentice-Hall.

Hayman, S. (2007). Folksonomies and tagging, new developments in social bookmarking. In *Ark group conference: Developing and improving classification schemes* Sydney, June 2007.

Hendler, J. (2001). Agents and the Semantic Web. *IEEE Intelligent Systems*, *16*(2), 30–37.

Hess, A., Maass, C., & Dierick, F. (2008). Web 2.0 to Semantic Web: A semi-automated approach. In *ESWC 2008 workshop on collective semantics: Collective intelligence and the Semantic Web (CISWeb 2008), Tenerife, Spain, 2008*. Available from http://www.andreas-hess.info/publications/hess-cisweb08.pdf.

Heymann, P., & Garcia-Molina, H. (2006). Collaborative creation of communal hierarchical taxonomies in social tagging systems. Stanford InfoLab Technical Report 2006-10.

Golder, S., & Huberman, B. A. (2005). [Online] The structure of collaborative tagging systems. *HP Labs Technical Report*. Available from http://www.hpl.hp.com/research/idl/papers/tags/.

Guarino, N. (1998). Some ontological principles for designing upper level lexical resources. In *Proceedings of the first international conference on lexical resources and evaluation* Granada, Spain,

Guha, R. V., & Bray, T. (1997). Meta content framework using XML. W3C Technical Note, June 1997.
IBM. (2004). [Online] *Ontology-based web services for business integration*. <http://www.alphaworks.ibm.com/tech/owsbi> Accessed 04.04.07.
ISO 704:2000. (2003). *Terminology work—principles and methods*. International Organisation for Standardization, ISO/TC 37 International Standards.
ISO 1087-1:2000. (2003). *Terminology work – Vocabulary – Part 1: Theory and application*. International Organisation for Standardization, ISO/TC 37 International Standards.
Kapetanios, E., & Schaal, M. (2007). An algebra and conceptual model for semantic tagging of collaborative digital libraries. In *The second workshop on foundations of digital libraries in conjunction with 11th European conference on research and advanced technologies on digital libraries* Budapest, Hungary, September,
Kashyap, V. (2003). Trust and quality for information integration: The data-metadata-ontology continuum. In *Workshop on data quality* Dagstuhl, Germany, September.
Lux, M., & Dosinger, G. (2007). From folksonomies to ontologies: employing wisdom of the crowds to serve learning purposes. *International Journal of Learning Technology*, *3*(4/5), 515–528.
Macgregor, G., & McCulloch, E. (2006). Collaborative tagging as a knowledge organisation and resource discovery tool. *Library Review*, *55*(5), 291–300.
Mai, J. E. (2004). Classification in context: relativity, reality, and representation. *Knowledge Organization*, *31*(1), 39–48.
Maicher, L., & Park, J. (Eds.). (2005). *Lecture notes in computer science Charting the Topic Maps research and applications landscape*. Springer.
Mathes, A. (2004). [Online] Folksonomies—Cooperative classification and communication through shared metadata, *EServer TC Library*. <http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html> Accessed 05.05.08.
Mejias, U., 2004. [Online] *Bookmark, classify and share: A mini-ethnography of social practices in a distributed classification community*. Available from http://ideant.typepad.com/ideant/2004/12/a_delicious_stu.html.
Mika, P. (2005). Ontologies are us: A unified model of social networks and semantics. In *Proc. of ISWC'05*
Navarro, G. (2001). A guided tour to approximate string matching. *ACM Computing Surveys*, *33*(1), 31–88.
Ohkura, T., Yoji, K., & Hiroshi, N. (2006). Browsing system for weblog articles based on automated folksonomy. In *WWW 2006, the third annual workshop on the weblogging ecosystem* Edinburg, May 2006.
O'Reilly, T., 2005. [Online] *What is Web 2.0?* <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html> Accessed 28.03.07.
Patrick, K., & Dotsika, F. (2007). Knowledge sharing: developing from within the learning organization. *The International Journal of Knowledge and Organizational Learning Management*, *14*(July (3))
Pepper, S. (1999). [Online] Euler, revolution, and Topic Maps. Steve Pepper, Ontopia. *XML Europe 1999*. Available from http://www.ontopia.net/topicmaps/materials/euler.pdf.
Preuveneers, D., & Berbers, Y. (2006). [Online] *Prime numbers considered useful: ontology encoding for efficient subsumption testing*. Technical Report CW464, Department of Computer Science, Katholieke Universiteit Leuven. <http://www.cs.kuleuven.ac.be/~davy/publications/cw464.pdf> Accessed 06.05.08.
Rector, A. L., Wroe, C., Rogers, J., & Roberts, A. (2001). Untangling taxonomies and relationships: personal and practical problems in loosely coupled development of large ontologies. *K-CAP, 2001*, 139–146.
Rafferty, P., & Hidderley, R. (2007). Flickr and Democratic Indexing: dialogic approaches to indexing. *Aslib Proceedings: New Information Perspectives*, *59*(4/5), 397–410.
Sabou, M., d'Aquin, M., & Motta, E. (2006). Using the Semantic Web as background knowledge for ontology mapping. In *Proc. of the international workshop on ontology matching (OM-2006)*
Schmitz, P. (2006). Inducing ontology from flickr tags. In *Proceedings of the Collaborative Web tagging workshop at the 15th WWW conference (WWW2006)* Edinburgh, Scotland, 2006.
Shirky, C. (2005). [Online] Folksonomy, or how I learned to stop worrying and love the mess. In *O'Reilly emerging technology conference, San Diego*. Available from http://craphound.com/etech2005-folksonomy.txt.
Smith, B., & Welty, C. (2001). Ontology: Towards a new synthesis. In C. Welty, & B. Smith (Eds.), *Formal ontology in information systems, Ongunquit*. Maine: ACM Press.
Specia, L., & Motta, E. (2007). Integrating folksonomies with the Semantic Web. In *The proceedings of 4th European Semantic Web conference* Innsbruck, Austria,
Trant, J. (2006). Exploring the potential for social tagging and folksonomy in art museums: Proof of concept. *New Review in Hypermedia and Multimedia*, *12*(1), 83–105.
Udell, J. (2004). [Online] *Collaborative knowledge gardening, InfoWorld*. <http://www.infoworld.com/article/04/08/20/34OPstrategic_1.html> Accessed 05.05.08.
Van Damme, C., Hepp, M., & Siorpaes, K. (2007). FolksOntology: An integrated approach for turning folksonomies into ontologies. In *Proceedings of the ESWC 2007 workshop on bridging the gap between Semantic Web and Web 2.0* Innsbruck, Austria,
Vander Wal, T. (2005). [Online] *Explaining and showing broad and narrow folksonomies*. <http://www.personalinfocloud.com/2005/02> Accessed 28.03.07.
W3C. (2004). [Online] *OWL Web ontology language use cases and requirements, recommendation*. <http://www.w3.org/TR/webont-req/> Accessed 06.05.08.
W3C. (2005). [Online] *RDFTM: Survey of interoperability proposals*. Available from http://tesi.fabio.web.cs.unibo.it/RDFTM/DraftSurvey.
W3C. (2006). [Online] Ontology driven architectures and potential uses of the semantic web in systems and software engineering. *Working Draft*. Available from http://www.w3.org/2001/sw/BestPractices/SE/ODA/060211/.
Weber, R. (1997). *Ontological foundations of information systems*. Coopers & Lybrand.
Welty, C., & Guarino, N. (2001). Support for ontological analysis of taxonomic relationships. *Journal of Data and Knowledge Engineering*, *39*(1), 51–74.
Wu, X., Zhang, L., & Yu, Y. (2006). Exploring social annotations for the Semantic Web. In *Proc. of WWW'06*

Fefie Dotsika is a senior lecturer in the Business School of the University of Westminster. Her background, expertise and research interests are in the area of system interoperability, information modelling, Semantic Web and Web 2.0 technologies.

International Journal of Information Management 30 (2010) 335–342

Contents lists available at ScienceDirect

International Journal of Information Management

journal homepage: www.elsevier.com/locate/ijinfomgt

Semantic APIs: Scaling up towards the Semantic Web

Fefie Dotsika*

University of Westminster, Business Info Management, 35 Marylebone Road, London NW1 5LS, United Kingdom

ARTICLE INFO

Article history:

Keywords:
Ontologies
Folksonomies
Web classification schemes
Information modelling
Web information management

ABSTRACT

Web information retrieval and knowledge discovery are undergoing changes. The size of the Web and the heterogeneity of web pages generate new challenges in meeting user needs. This paper investigates the different methods deployed that add semantics to web content: semantic tagging and semantic APIs. The research carried out investigates existing systems in each category, outlining their primary features and functionality. It then proposes a framework for the evaluation of semantic tagging based on the main requirements for information discovery and recommends a number of comparative assessments, ranging from basic product information and requirements' analysis to the evaluation of the APIs information modelling functionality.

1. Introduction

There is wide speculation about the next generation of Web architecture. Information retrieval and knowledge discovery score high among the usual requirements and the leading trends put proprietary arms around emergent Web technologies, services and long-awaited applications, claiming that the next Web architecture is their offspring.

Tim Berners-Lee introduced the Semantic Web in 2001 (Berners-Lee, Hendler, & Lassila, 2001) as a form of web content where knowledge representation relies on languages expressing information in a machine process-able form, by means of a framework based on RDF (Resource Description Framework) and ontologies. The information modelling is predominantly top-down and it is done formally, without the participation of end-users.

Web 2.0 (O'Reilly, 2005) is more of a term rather than architecture, encompassing a number of Web technologies and applications. It sees the Web as a platform and focuses on end-user involvement, co-operation and information sharing. As far as the user-content is concerned, the information modelling is informal and carried out bottom-up by means of user-generated tag systems called folksonomies.

A number of hybrid web architectures are also present, such as *Web 3.0* and *3D Web*. Web 3.0 (Berners-Lee, 2006) is defined rather poorly but the consensus presents it as a mixture of Semantic Web content and Web 2.0 co-operative applications. 3D Web is an even vaguer term, involving the evolution of the Web into the three-dimensional space. Another definition attributes the three Ds to Decentralisation, Disaggregation and Democratisation of the Web (Loux, 2008).

Whatever the architecture, the nature of the Web remains the same: it is an ever-expanding system of hypertext documents containing text and multimedia that can be accessed over the Internet. First and foremost it is an information repository expected to yield information whenever searched. However, while search engine technology is maturing, it is still relatively young compared to, say, database technology. Furthermore, research has shown that there is a difference between the way users search the Web and the way they search traditional information retrieval systems and online public access catalogues (Jansen, 2001). Enhancements to current practices come from a variety of sources. Systems that engage cultural and local meaning are shown to expand and disseminate research that includes multiple communities and cultures (Boast, Bravo, & Srinivasan, 2007). Other methods include improved Web information modelling, adaptive methods for personalisation of search, advances in natural language processing technologies and information relevance measuring metrics. Apart from the metrics, a number of these methods are present in a new family of Web application programmers' interfaces commonly known as Semantic APIs, which are the topic of this paper.

The rest of the paper is organised as follows: in Section 2 we give a comprehensive overview of semantic tagging. Section 3 introduces the semantic APIs, examines their functionality and assesses their role. In Section 4 we analyse our findings and in Section 5 we present our conclusions and highlight future work.

2. Semantic tagging for the web

Web content is readable by humans, but, unless it is semantically annotated, it is not machine readable, in the sense that it cannot be

* Tel.: +44 20 79115000.
E-mail address: F.E.Dotsika@westminster.ac.uk.

0268-4012/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijinfomgt.2009.12.003

automatically interpreted in any reasonable manner. The following are different ways that allow the semantic annotation of web content.

2.1. HyperText Markup Language

The simplest form of semantic tagging is done using HTML. Although, as a markup language, HTML is meant to display information on browsers and enable navigation between hypertext documents, it also allows the embedding of multimedia objects and scripting languages (such as JavaScript). Strictly speaking, HTML does not deal with semantics. However, there is a number of elements and attributes that can be used to describe semantics such as:

(a) <meta> specifies associative key-value pairs usually by means of the attributes *name* (key) and *content* (value). This element is widely used in Search Engine Optimisation (SEO).
(b) is used in wrapping-up specific names of the attribute *class* and the attributes *rev* and *rel* which accept link-type values, specifying the links defined by <a> or <link>.
(c) is used in the same way as .

2.2. Microformats

Microformat is a method that uses existing HTML (and XHTML) tags to semantically annotate web data (Allsop, 2007). It provides a consistent manner for the identification of common data. From the above elements, while <meta> is widely used in Search Engine Optimisation (SEO), consistent semantic annotation with and is carried out with microformats.

Application of microformats is currently centred in the annotation of information on contact details (hCard, proving information about telephone number, postal and email address), events (hCalendar with date, location, etc.) and reviews of products, etc. (hreview with item being reviewd, date, hCard of reviewer, etc.). Their simplicity makes their adoption popular (LinkedIn, Flickr, Technorati), but there is no standardisation and their semantic power is rather limiting. Lack of hierarchies and inability to define complex relationships are the main culprits of their narrow design capability. On the other hand, lack of standardisation prevents wide interoperability.

2.3. Resource Description Framework (RDF)

RDF is a WC3 specification (Beckett, 2004). It is a formal, general-purpose language for modelling Web-based information. Modelling is done by means of subject-predicate-object expressions, known as *triples*. RDF subjects can be Uniform Resource Identifiers (URIs) or blank nodes (anonymous resources). Predicates are URIs and represent relationships. Objects can be URIs, blank nodes or literals.

The modelling flexibility of RDF is considerable. It provides the facility to represents containers of resources and supports reification. Containers can be unordered, ordered or lists of alternatives. Reification enables making statements about statements.

RDF is part of the Semantic Web infrastructure. It is the only standardised semantic annotation method, an attribute that enables interoperability. However, its complexity has prevented widespread adoption. Its formality requires expertise and makes it difficult to master, an attribute that limits its popularity. It has been argued that the complexity refers to RDF/XML, as RDF is in fact a simple concept.

RDF is difficult to publish. Web content annotated with RDF requires XHTML for its textual presentation but also a parallel RDF/XML part to publish the semantic information.

2.4. Notation 3 (N3)

Notation 3 (Berners-Lee, 1998) is a simpler, more readable, non-XML-based version of RDF. The language is extended to include variables and nested graphs, enabling thus greater expressiveness. N3 has subsets, one of which is RDF itself. Modelling of information is done in triples, just like with RDF.

2.5. RDFa (Resource Description Framework – in – attributes)

RDFa is a specification of the W3C (Adida & Birkbeck, 2008) and represents an easier than RDF way to provide metadata. It continues the tradition of microformats. RDFa allows XHTML documents to be marked-up with machine-readable indicators. Like in the case of microformats, RDFa makes use of standard XHTML attributes such as *rel* and RDFa-introduced ones such as *property*. Namespaces are also used to import vocabularies, such as the Dublin Core and FOAF taxonomies. The traditional RDF triples are then created in an easier, lighter format. Any existing RDF schema can be used by RDFa.

2.6. Ontologies and OWL

Ontology is a set of shared, explicit and formal concepts used to organise and classify content. It models entities, logical constraints and relationships in the form of directed graphs. Web ontologies are used to categorise a range of resources, from web sites to products sold online. The Web Ontology Language OWL (Smith, Welty, & McGuinness, 2004) is a family of languages endorsed by the World Wide Web Consortium. OWL ontologies are built using XML/RDF syntax and can model:

(a) entities as classes, which are subclasses of the OWL class *Thing*,
(b) properties which are binary relations modelling the characteristics and attributes of a class,
(c) instances of a class and
(d) operations on classes. Such as union, intersection, etc.

There are three variants of OWL, OWL Lite, OWL DL and OWL Full. The first two variants have semantics that are based on description logic and its family of formal knowledge representation languages. OWL Lite has low expressiveness and was designed to provide modelling for classification hierarchies such as thesauri and taxonomies and simple constraints. OWL DL has more expressiveness, preserves computational completeness and enables logical reasoning. OWL Full has different semantics and was designed to be RDF Schema – compatible.

2.7. Topic Maps

Topic Maps is a model for knowledge integration developed by the ISO. It started as a project on managing documentation indices in the early 90s. They support high-level indexing of sets of information resources in order to enhance information find-ability (Maicher & Park, 2005). In 1996 Topic Maps were adopted as an ISO work item and the Topic Map standard ISO 13250 was published in 2000.

There are distinct similarities between Topic Maps and RDF. WC3 maintains a working document of interoperability proposals for integrating the two standards (W3C, 2005). According to their analysis, semantic mappings seem to fit the interoperability requirements better than object mappings.

2.8. Folksonomies

Folksonomies are collaborative, user-generated metadata. They are a faceted classification scheme and are created bottom-up, in

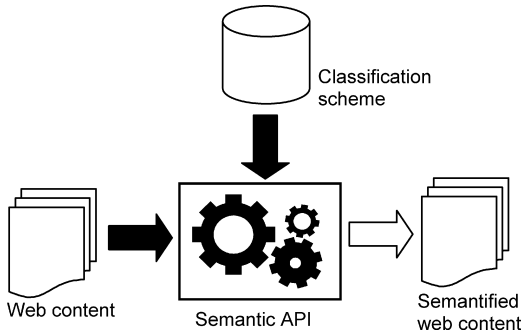


Fig. 1. Semantic APIs.

an analytical synthetic way, where the subject area is first divided into individual concepts which can be composed to construct complex subjects via appropriate sets of rules. They are characterised by their collaborative bottom-up design and are the prominent classification tagging scheme for Web 2.0 (Smith, 2008).

However, contrary to recent claims, folksonomies cannot be considered semantic tagging. The user-generated metadata carry implied only semantics. As a consequence, unless the tag semantics are explicitly stated they cannot be automatically processed. There is a growing area of research that aims to retrieve folksonomy semantics and engineer corresponding ontologies (Dotsika, 2009). These methods seek to reconcile the differences of the two classification schemes while preserving their advantages.

3. Semantic APIs

Without semantic markup, information discovery can be enabled by an Applications Programmers Interface (API). APIs that take unstructured text (including web pages) as input and return the content's contextual framework are termed *semantic* APIs. The increasing popularity of web semantics has resulted in a rise of semantic APIs. While the SW-fuelled web services require formal markup and an RDF/OWL support, there are APIs that offer web content classification and discovery outside of the SW framework. Fig. 1 depicts the function of Semantic APIs.

All Semantic APIs generate XML code, and most of them RDF. The Topic Maps group has created the Topic Maps API (TMAPI 2.0), a project originally developed as an interface for Topic Map processors (Heuer & Schmidt, 2008). It is a user-friendly enabler of Topic Map applications development, currently in alpha status. Another relevant project has developed a semi-automatic methodology for generating Topic Maps (Kásler, Venczel, & Varga, 2006). However, currently, there are no Semantic APIs supporting Topic Maps.

Here we will limit our list to the most popular semantic API projects.

3.1. Dapper

The Dapper (Data Mapper) API (Dapper, 2005) enables developers to extract semantics from web content in the form of an XML document that can then be used to build mashups, RSS feeds and other applications. The Dapper Semantify Web Service allows the user to define the content of interest, reads the website and creates a feed (a *Dapp*) of the specific content. In order to work with semantic search engines, the Dapps have to be created with field names from a number of supported RDFa namespaces (Dublin Core, FOAF, Creative Commons, MediaRSS and GeoRSS). There is the plan to incorporate the namespaces and determine the appropriate one automatically. At the moment the Semantify service works with PHP pages (a PHP script is required for every Dapper on the page). Dapper is currently a free web service.

Dapper's core semantic engine was created in 2005 and is based on genetic algorithms and machine learning techniques. It was aimed for the gathering, manipulation and dissemination of web content in new formats without the need for programming.

Their flagship application is the Dapper advertiser Dapper Ads. The motive is to create relevant, targeted, dynamic web advertising. The targeting is based on consumer behaviour and there is focusing on areas with emphasis on real-time pricing, such as travel and financial services. The system enables the extraction of live offers from advertising websites (Dapper's Live Offer Platform), the targeting of potential customers by analysing user profiles (Dapper's Intent Platform) and the dynamic discovery of the best match (Dapper's Ad Monkey).

3.2. OpenCalais

OpenCalais (Calais, 2008) is an automatic generator of semantic metadata in RDF format from web content. It is based on natural language processing (NLP) that was originally developed by ClearForrest, a company now owned by Reuters. It works on text only (no other media files are supported) and offers support only for English. Calais operates as a web service and supports SOAP and REST APIs.

The API reads in unstructured documents (plain text, HTML, XML), recognises a number of different entities and annotates them semantically in RDF. Existing entities include person, company, place and event. Performance is in the scale of under a second, even for large documents and the creators claim that the four entities will be expanded to include more. Apart from the list of entities, the service returns number of occurrences and relevancy scores measuring the semantic importance of the various entities. The latest version, Calais 4.0, can assign content to ten different categories: health, politics, sports, technology, law, business & finance, entertainment & culture, travel, weather and environment.

Calais supports a list of related tools that aid systems developers:

- Calais Collection integrates Calais into the Drupal platform, an open source content management system.
- Gnosis is a browser extension for FireFox or Internet Explorer. While browsing a website, the plugin can identify a number of entities automatically, such as people, locations, etc. and enables searches based on the type of entity identified.
- Tagaroo automatically generates tags and image location for WordPress blogs.
- Calais Marmoset invokes the OpenCalais web service and generates and embeds metadata to be used with semantic search engines, such as Yahoo!'s SearchMonkey.

Just like Dapper, Calais is free of charge (*Open*Calais). However, for a service where the amount of daily submissions is expected to exceed 40,000, there is CalaisProfessional, a paid equivalent to OpenCalais. CalaisProfessional offers a higher class service, a service level agreement and five times higher submission rate capability.

3.3. SemanticHacker

SemanticHacker (SemanticHacker, 2008) is an API that takes text as input and classifies the document content into categories, creating what TextWise calls its "Semantic Signature". The classification is done by identifying and returning a number of entities from a given classification scheme, the Open Directory Project. Their weight is then measured and a relevance score returned. The Semantic Signature is presented by means of a list of vectors which

are produced by TextWise's Trainable Semantic Vectors (TSV) technology. The system employs NLP and text mining techniques.

Apart from the Semantic Signature call, which analyses the content provided, the API comprises a number of other relevant services:

- The concept service identifies the key concepts of the input and orders them by weight.
- The category service extracts the main topic categories and orders them by weight.
- The matching service provides a similarity search that matches the text's semantic signature to a number of context indexes such as Wikipedia, YouTube videos, etc., and includes a number of links that can be relevant for further reading. The listed items are ordered based on their match score.
- The index call is available only after licensing a Custom Content index for a fee. The API then allows users to perform similarity searches against their own custom content.
- The filter call discards useless text from an indicated content. There are a number of filter algorithms available depending on the type of input.

For easy access to the API, TextWise have released the *WordPress* plugin and a widget, which allow bloggers and content publishers respectively to use the SemanticHacker's similarity search and display the results.

SemanticHacker works under a licence agreement. Users are sent a token upon registration which enables access to the API and allows for a limited number of queries. Additional queries, along with custom dictionary development can be purchased after contacting TextWise.

3.4. Semantic Cloud API

The service identifies and extracts semantics from a web page or a document, creates a semantic cloud of concepts and generates a list. Alternatively it can take a set of URLs as input and return a multi-document summary about the main concepts present and/or an essay on a specific topic (Semantic Cloud API, 2009). The web service provides the user with two methods, *ExtractConcepts* and *CreateSummary*. The Semantic Cloud operates as a paid web service and supports SOAP and REST APIs.

Semantic Cloud bases pricing on the requested bandwidth. There are two packages of different capacity, the *Small Virtual Search Server*, which covers approximately 40K *ExtractConcepts* requests per day and the *Large Virtual Search Server* capable of servicing twice as many, or 15K *CreateSummary* requests per day, or a combination of the two.

3.5. Zemanta API

The Zemanta API (Zemanta, 2009) takes in unstructured text and returns tags, categories, links, photos, and related articles. The service acts as a single-point entry to various, pre-indexed, content databases. Zemanta analyses the postings, discovers relevant content and adds it to the page or document. The system is powered by NLP and semantic algorithms. It categorises content by comparing it to their pre-indexed database. The categorisation process is constantly enhanced by end-user input and machine learning methods.

There have been some issues with Zemanta's performance as it has been shown to slow down users' browsers (especially Internet Explorer versions 6 and 7). This is due to Zemanta's high use of JavaScript and can be fixed by switching to a different browser (best current choice: Firefox 3).

The Zemanta API is free of charge for up to 10,000 calls per day. After that there are two packages, depending on user needs. The first one covers up to 50,000 calls per day for $1200 a month and the second one services twice as many calls for $2000 a month.

3.6. Ontos API

The Ontos API Semantic Web Service (Ontos, 2009) is currently in beta version. It was launched in July 2009 as an API that provides the means to personalise the NLP platform that returns named entities and semantic relations when fed with non-semantically annotated text. Users can define their own semantic content via external dictionaries and can tune concepts from core ontologies. Ontos supports visual representations in the form of cognitive maps, dynamic reports and summaries from document collections. The retrieved information is stored in their Expert Knowledge Base, which is a scalable RDF store.

The Ontos architecture is modular.

- The Ontos Annotation Server consists of the Expert KB, the resource crawler, the OntosMiner which performs the information extraction and the language processing, and the Scalable Early Object Identification module. OntosMiner is currently available in English, German, French and Russian.
- The Ontos Navigation Server consists of the semantic annotation module and the semantic ranking and relevance service.
- The Ontos Inference Server carries out the API's semantic digesting and summarization services.

LightOntos for Workgroups is a package that manages documents and their annotations. It offers further possibilities for investigation and analysis, such as the visualisation of annotations. The diagrams can be edited and manipulated to form news feeds, generate relevant tags and link the content of individual documents.

4. Making a decision: analysis and method comparison

The various methodologies and systems that annotate web content with semantic metadata have all the same objective: to enhance information retrieval and knowledge discovery. In order to assess the assorted methods we first compare the different approaches to semantic tagging and then the semantic APIs.

4.1. Semantic tagging methods

With the purpose of assessing the different schemes and facilitate decision making, we need to look into the basic requirements for information retrieval. The methods are then compared against these requirements in order to determine how they fare and identify the extent of their influence.

Researching information retrieval in a pre-Web age, Cleverdon, Mills, and Keen (1966) identified *coverage*, *time lag*, *recall*, *precision*, *presentation* and *user effort* as the six main criteria to be used when evaluating an information retrieval system. From these criteria, *recall* (perceived as a measure of completeness) and *precision* (measuring fidelity) have been the most popular metrics in the various evaluation methods and algorithms (Salton, 1992).

Traditional information retrieval is the original base of Web information services (Agosti & Melucci, 2001; Pokorny, 2004). We identify a number of issues relevant to the intended evaluation where Web information retrieval necessitates a shift of focus:

- Web-based information is interlinked. While the requirement of *coverage* remains, the focus is on standardisation and system

Table 1
Semantic tagging.

Method	Feature					
	Standardisation/system interoperability	Semantic modelling power/granularity	Presentation		Cost	
			Browser support	XHTML attributes	Simplicity	Implementation
HTML	×	Low	×	√	High	£
Microformats	×	Low	√	√	High	£
RDF	√	High	×	×	Low	£££
N3	√	High	×	×	Low	££
RDFa	√	High	×	√	Medium	££
Ontologies/OWL	√	High	×	×	Low	£££
Topic Maps	×	High	×	×	Medium	££
Folksonomies	×	Medium	Depends	×	High	£

interoperability. The interconnectivity of Web information highlights the significance of relevance: a page with information that bears little or even no relationship to the information required may have a number of links to highly relevant resources. It then becomes itself partially relevant (Samalpasis, Tait, & Bloor, 1998). Standardisation and system interoperability contribute to information interconnectivity and further enhance coverage.

- Precision of the information retrieved has two contributing factors: the modelling of the information in the first place and the efficiency, standards and power of the search engine. The latter is of no interest to this paper. The former is a consequence of the modelling method used and relates to the method's semantic power and modelling granularity, and is influenced by a number of quality criteria (Dotsika, 2009).
- The ever-expanding user population is, in its majority, naïve end-users, compared to the information workers of the pre-Web era. While recall and precision remain critical, the user-centric character of the Web makes presentation, simplicity of solution, expertise requirements and issues of code integration and maintenance equally essential (Dotsika & Patrick, 2006). In the case of semantic tagging, presentation focuses on browser compatibility and the method's support of *XHTML* attributes.

Taking the above points into account we modify the original list accordingly and derive the following principal categories (not in order of importance):

(a) Coverage: system interoperability & standardisation.
(b) Precision: issues of information modelling (completeness, granularity) and quality.
(c) Presentation: issues of usability, navigation. Can be divided into:
  i. browser support,
  ii. use of XHTML attributes for semantic tagging.
(d) Cost: pricing the solution and user-effort. Can be divided into:
  i. simplicity of solution, expertise requirements, issues of code integration and maintenance
  ii. issues of custom-made vs. off-the-self, open-source vs. bespoke.
(e) Performance: speed and promptness of retrieval.
(f) Recall: relevance and timeliness of retrieved information.

Arasu, Cho, Garcia-Molina, Paepcke, and Raghavan,

The last two categories, performance and recall are not examined here, as they are not pertinent: they both relate to the search engine rather than the method employed and therefore were not deemed relevant to the current paper. Performance is also dependent upon the level of Internet traffic at the time of the query. Recall is further influenced by the volatility of Web-based information. Sites and individual pages appear and disappear constantly, while Web content changes on a regular (or rather irregular) basis. According to Arasu, Cho, Garcia-Molina, Paepcke, and Raghavan (2001) 40 percent of commercial pages change daily (compared to 23 percent of general Web pages) and have a half-life of 10 days. Pages can be static (in formats such as HTML, PDF, Postscript, etc.) or dynamic (generated by scripts such as PHP, JSP, etc.).

Browser support is currently offered for microformats from Firefox (since version 1.5–2), Internet Explorer (as an add-on) and Flock 1.0. The granularity for microformats is recorded as "low", which may seem unfair. However, the comparison is done with the bigger picture in mind. Therefore the modelling power of microformats is compromised, due to the fact that it is restricted to a few only entities and attributes.

Table 2
Semantic APIs: product information.

API	Feature						
	Developer	Tools & plugins	Web service	WS protocol	User support forums & blogs	Cost	Performance
Dapper	Dapper Inc.	Semantify	√	REST	√	Free	High
OpenCalais	Reuters	Tagaroo, Gnosis, Marmoset, SemanticProxy, Drupal modules	√	SOAP REST	√	Free – ££ (CalaisProfessional)	High
SemanticHacker	TextWise	SemanticSignature, Categorisation, ConceptTagging WordPress	√	REST	√	Free – ££	Medium–high
Semantic Cloud	Semantic Engines LLC	×	√	SOAP REST	Limited to email	££	High
Zemanta	Zemanta Ltd	×	√	REST	√	Free – £££	Medium–high
Ontos	Ontos AG	Ontos Miner, Navigation Server, Inference Server	√	REST	√	Free demo versions available (currently in beta)	High

Table 3
Requirements-based decision making.

Requirement	API					
	Dapper	Calais	SemanticHacker	Semantic Cloud	Zemanta	Ontos
Identify key concepts and categories	√	√	√	√	√	√
Relevance scores measure semantic importance	×	√	√ Semantic signatures	√ Also essay in specific topic	×	√
Create new format, mashups	√	×	×	√	×	×
Enhance content presentation	√	×	√	×	√	√
Add to content	×	×	√ Hyperlinks to rel. content	×	√ Hyperlinks, multimedia	√ Hyperlinks
Multidoc summary from URLs	×	×	×	√ Based on concepts	×	Summaries & reports
Enhance document findability	×	Metadata for semantic SE	×	×	×	×

Table 4
Information modelling.

API	Feature							
	Input: classification scheme		Output: semantic tagging method					
	Custom taxonomy	Standard taxonomy	Microformats	RDF	N3	RDFa	OWL	Topic Maps
Dapper	√	√	×	×	×	√	×	×
OpenCalais	×	√	√	√	×	×	×	×
SemanticHacker	√	√	×	√	×	×	×	×
Semantic Cloud	√	√	×	√	×	×	×	×
Zemanta	√	√	×	√	×	×	×	×
Ontos	√	√	×	√	√	×	×	×

Table 5
Input sources and output formats.

API	Feature	
	Classification scheme used	Output formats
Dapper	User thesaurus or standard taxonomy	XML JSON CSV RSS
Dapper's Semantify (beta)	Dublin Core FOAF Creative Commons, MediaRSS GeoRSS	RDFa
OpenCalais	RDFS schema [a number of set entities, events and facts]	XML RDF Microformats JSON CSV
SemanticHacker's Semantic Signatures	Open Directory Project (ODP)	XML RDF JSON
SemanticHacker's Categorisation	Open Directory Project (ODP)	XML RDF JSON tag cloud formats
SemanticHacker's Concept Tagging	Open Directory Project (ODP)	XML RDF JSON tag cloud formats
Semantic Cloud	SC's database of content and/or user thesaurus	XML
Zemanta	Pre-indexed database of content	XML RDF JSON
Ontos	Fixed ontology that can be enhanced with additional concepts/instances	XML, RDF N3 JSON

Based on the above, we distinguish six features for the comparison of semantic tagging methodologies.

Table 1 presents our findings.

4.2. Semantic APIs

As we saw earlier, when semantic tagging is not an option, knowledge discovery and content categorisation can be carried out by means of the semantic APIs which take Web content as input and annotate it with semantic metadata. However, there are inconsistencies in the way that different products annotate Web content and this makes their comparison troublesome. To overcome this, we divide our comparison into two sections. In the first instance we look into basic product information (Table 2) and requirement-based decision planning (Table 3). The second part concentrates on information modelling and the comparison is based on the APIs' common ground of enhancing Web information retrieval and discovery (Tables 4 and 5).

The semantic APIs general characteristics consist of developer, availability of extra tools, Web service information, online user support, cost and performance. All APIs are offered as Web Services. Most of them support a number of extra tools and/or plugins.

Performance here relates to the semantic API (i.e. how fast is the tagging done), as opposed to the performance related to the information retrieval. Performance has been difficult to establish and compare due to lack of consistent information. For instance, according to Dapper pages, their API takes less than 5 ms to read a profile (based on a large scale database, the World Ocean DB 2001), whereas Calais takes less than a second to process a sizeable article. On the other hand the performance of Zemanta depends on the browser and there is next to no information about the Semantic Cloud (then again it is used by Amazon, so one can deduce that it is more than adequate). Another matter that makes performance questions difficult is the differences in the actual functionality of the APIs, which makes comparisons on equal terms problematic. Queries to the relevant companies have not yielded detail and performance measurements are relative and approximate.

The results can be seen in Table 2.

All the available systems have facilities for the identification of key concepts and categories, each one of them provides a set of other services that can help to make a choice depending on the user requirements. For instance, whilst all of them can identify key concepts and categories, only one enhances document find-ability (Calais) and only two can create a multi-document summary. The following table lists the available services/requirements opposite the systems that support them.

The next step is to consider the classification input and output of the semantic APIs. Apart from Web content the APIs require a classification scheme and a schema which they base their semantic annotation on. Most systems accept custom and standard taxonomies (types and particulars can be seen in Table 5). The outputs examined here are the same as those compared in Table 1.

Table 4 lists our findings.

If a custom taxonomy is not (or cannot be) used as input, the semantic APIs use a number of standard classification schemes such as pre-indexed content, or off-the-shelf taxonomies and ontologies such as:

(a) Dublin Core: a syntax-independent, expandable classification scheme created and maintained by a cross-disciplinary group of professionals.
(b) FOAF (Friend Of A Friend): an ontology written in RDF and OWL describing people, links among them and activities.
(c) MediaRSS and GeoRSS: RSS extensions for multimedia files and encoding locations, respectively.
(d) Open Directory Project (ODP): a multilingual ontology owned by Netscape for listing Web sites.

Web information systems have championed the development of new approaches in modelling and describing knowledge and therefore the diversity of formats used (Boast et al., 2007). As a consequence, apart from the standard semantic metadata mentioned above, some APIs (or API tools and/or plugins) provide additional output formats that cater for a variety of other applications. Such formats are XML, Comma Separated Values (CSV), Web feed formats used to publish frequently updated content (RSS), text-based, simple data structures in JavaScript Object Notation (JSON) and tag cloud formats, which are a visual representation of weighted folksonomies.

Based on the above, the following table lists the APIs and API tools along with the sources of their (non-user-defined) input and output formats.

5. Conclusions and future work

The research carried out in this paper addresses web information discovery by means of semantic markup. It identifies semantic annotation and APIs as the key methods for enhancing Web information retrieval and investigates existing systems in each category, assessing and comparing their primary features and functionality.

Following the basic requirements for information retrieval and knowledge discovery, we determined the crucial issues that influence Web-based information and proposed a framework of categories for the evaluation of semantic tagging. We reviewed the market leaders of the increasingly popular semantic APIs and devised a number of comparative assessments, ranging from basic product information and requirements' analysis to the evaluation of the APIs information modelling functionality.

There is enough evidence to suggest that the next generation of the Web will be a hybrid mix of Semantic Web infrastructure and Web 2.0 functionality. Whatever its architecture, the efforts towards automated information retrieval and knowledge discovery will be intensified. Future research will further investigate the current trends and will further address certain problems and practicality issues raised in Section 4. In particular it will explore the *organisational* aspects of information retrieval, that is, how organisations manage (or intend to manage) the semantic annotation of their web content on a practical level.

References

Adida, B., & Birkbeck, M. (2008). RDFa Primer, Bridging the Human and Data Webs (online). http://www.w3.org/TR/xhtml-rdfa-primer/ (accessed 4th April 2009).
Agosti, M., & Melucci, M. (2001). Information retrieval on the Web. In M. Agosti, F. Crestani, & G. Pasi (Eds.), *Lectures on Information Retrieval: Third European Summer School (ESSIR 2000)* (pp. 242–285). Springer.
Allsop, J. (2007). Microformats: Empowering Your Markup for Web 2.0. Friends of Ed, NY.
Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., & Raghavan, S. (2001 August). Searching the Web. *ACM Transactions on Internet Technology*, *1*(1), 2–43.
Beckett, D. (Ed.). (2004). RDF/XML Syntax Specification (online). http://www.w3.org/TR/rdf-syntax-grammar/ (accessed 24th April 2009).
Berners-Lee, T. (1998). Notation 3 Specification (online). http://www.w3.org/DesignIssues/Notation3.html (accessed 4th April 2009).
Berners-Lee, T., Hendler, J., & Lassila, O. (2001 May). *The Semantic Web*. Scientific American.
Berners-Lee, T. (2006). A more revolutionary Web (online). http://www.iht.com/articles/2006/05/23/business/web.php (accessed 28th March 2009).
Boast, R., Bravo, M., & Srinivasan, R. (2007). Return to Babel: Emergent diversity, digital resources, and local knowledge. *The Information Society*, *23*(5), 395–403.
Calais. (2008) [online]. http://www.opencalais.com/ (accessed 24th April 2009).

Cleverdon, C. W., Mills, J., & Keen, E. M. (1966). *An inquiry in testing of information retrieval systems (2 vols.)*. Cranfileld, U.K.: Aslib Cranfield Research Project, College of Aeronautics.
Dapper. (2005) [online]. http://www.dapper.net/ (accessed 30th April 2009).
Dotsika, F., & Patrick, K. (2006). Towards the New Generation of Web Knowledge Search and Share. *VINE: The Journal of Information and Knowledge Management Systems*, *36*(4), 406–422.
Dotsika, F. (2009). Uniting formal and informal descriptive power: Reconciling ontologies with folksonomies. *The International Journal of Information Management*, *29*(October (5)), 407–415. ISSN 0268-4012
Heuer, L., & Schmidt, J. (2008). TMAPI 2.0, subject-centric computing. In *Fourth International Conference on Topic Maps Research and Applications, TMRA 2008* Leipzig, Germany.
Jansen, B. J. (2001). A review of Web searching studies and a framework for future research. *Journal of the American Society for Information Science and Technology*, *52*(February (3)), 235–246.
Loux, K. (2008). Bloggers and startups, challenge the big companies and embrace open standards (online). http://thenextweb.com/2008/04/03/keynote-khris-loux-bloggers-and-startups-challenge-the-big-companies-and-embrace-open-standards/ (accessed 28th March 2009).
Kásler, L., Venczel, Z., & Varga, L. Z. (2006). Framework for Semi Automatically Generating Topic Maps. In *Proceedings of the 3rd international workshop on text-based information retrieval* Riva del Grada,
Maicher, L., & Park, J. (Eds.). (2005). *Charting the Topic Maps Research and Applications Landscape, Lecture Notes in Computer Science*. Springer.
Ontos. (2009) [online]. http://www.ontos.com/o_eng/index.php?cs=2-1 (accessed 25th August 2009).
O'Reilly, T. (2005). What is Web 2.0? (online). http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html (accessed 28th March 2009).
Pokorny, J. (2004). Web searching and information retrieval. *Computing in Science and Engineering*, *06*(July–August (4)), 43–48.
Salton, G. (1992). The state of retrieval system evaluation. *Information Processing & Management*, *28*(4), 441–449.
Samalpasis, Tait, J., & Bloor, C. (1998). Evaluation of information seeking performance in hypermedia digital libraries. *Interacting with Computers*, *10*, 269–284.
Semantic Cloud API. (2009) [online]. http://www.semanticengines.com/api.aspx (accessed 30th April).
SemanticHacker. (2008) [online]. http://www.semantichacker.com/ (accessed 24th April 2009).
Smith, G. (2008). *Tagging: People-Powered Metadata for the Social Web*. Berkeley, CA: New Riders.
Smith, M. K., Welty, C., & McGuinness, D. L. (2004). OWL Web Ontology Language (online). http://www.w3.org/TR/owl-guide/ (accesses 4th August 2009).
W3C. 2005. RDFTM: Survey of Interoperability Proposals (online). http://tesi.fabio.web.cs.unibo.it/RDFTM/DraftSurvey (accessed 28th March).
Zemanta. (2009) [online]. http://www.zemanta.com/api/ (accessed 30th April 2009).

Fefie Dotsika is a senior lecturer in the Business School of the University of Westminster. Her background, expertise and research interests are in the area of system interoperability, information modelling, Semantic Web and Web 2.0 technologies.

The next generation of the web: an organisational perspective

Fefie Dotsika

Westminster Business School

Westminster Business School, University of Westminster

Working Paper Series in Business and Management
Working Paper 12-1
March 2012

WESTMINSTER
BUSINESS SCHOOL

WORKING PAPER SERIES
IN BUSINESS AND MANAGEMENT

WORKING PAPER 12-1
March 2012

The next generation of the web: an organisational perspective

Fefie Dotsika

Westminster Business School

Corresponding author:

Fefie Dotsika
Westminster Business School
University of Westminster
35 Marylebone Road
London NW1 5LS, UK
F.E. Dotsika@westminster.ac.uk

ISBN ONLINE: 978-1-908440-07-5
http://www.westminster.ac.uk/about-us/schools/business/research/working-paper-series

UNIVERSITY OF
WESTMINSTER

The next generation of the Web: an organisational perspective.

Fefie Dotsika
Senior Lecturer
University of Westminster
F.E.Dotsika@westminster.ac.uk

Abstract:

The web has revolutionised information sharing, management, interoperability and knowledge discovery. The union of the two prominent web frameworks, Web 2.0 and the Semantic Web is often referred to as Web 3.0. This paper explores the basics behind the two paradigms, assesses their influence over organisational change and considers their effectiveness in supporting innovative solutions. It then outlines the challenges of combining the two web paradigms to form Web 3.0 and critically evaluates the impact that Web 3.0 will have on the social organisation. The research carried out follows action research principles and adopts an investigative and reviewing approach to the emerging trends and patterns that develop from the web's changing use, examining the underpinning enabling technologies that facilitate access, innovation and organisational change.

Keywords: *Knowledge management, web technologies, web information modelling, Web 3.0.*

1. Introduction

Web 2.0 is a user-centric web environment where information modelling is based on non-standardised user-generated folksonomies and innovation originates in social interaction. The Semantic Web is a machine-centric framework of web standards, semantic-driven, built top-down with formal classification schemes and highly searchable content. Information modelling is supported by a standardised, precise framework of XML, RDF and ontologies. Innovation is built on find-ability.

Both paradigms are based on the interlinking of information, way beyond the hypertext linkage that Web 1.0 introduced and web users took for granted. They both create information networks which are highly dynamic, interactive, adaptive and searchable. The Web 2.0 network is firmly based on the social aspect of its technologies. The Semantic Web network is the standardised principle of linked resources by means of Uniform Resource Identifiers (URIs), so that knowledge representation is web-embedded, with URIs assigned to terms and relationships. What would it be like joining the two?

Merging the power of the two network models, namely the social aspect with the standardised and interoperable information framework, leads to the new generation of web applications referred to as Web 3.0. Disregarding attempts to refer to the Semantic Web as Web 3.0 (Lassila & Hendler 200, Hendler 2008), in this paper we will use the term Web 3.0 to refer to the union of Web 2.0 and Semantic Web. While a fully functioning Web 3.0 is probably years away, there has been endless speculation about its impact.

In the early 80s Robert Metcalfe claimed that the value of a (telecommunications) network was proportional to the square of the number of users, despite the fact that its cost grew linearly with the number of connections (Gilder 1993). Metcalfe's heuristic has been cited, debated and replaced by alternatives many times since (Reed 2003; Brisco et. al 2006; Hendler & Golbeck 2008). The phenomenon is referred to as the *network effect* and, despite the lack of definitive algorithm consensus and hard mathematical proof, it is still part of web network analysis and provides an indication of the impact the merging of the two paradigms will have.

Semantic technologies coupled with social networking can instigate innovative influence with wide organisational implications that can benefit a considerable range of industries. The scalable and sustainable business models of social computing and the collective intelligence of organisational social media can be resourcefully paired with internal research and knowledge from interoperable information repositories, accounting systems, back-end databases etc. Web 3.0 can free human resources so that they can be used to better serve business development, support innovation and increase productivity.

Examples of Web 3.0 applications have appeared in various areas, such as medicine and bioinformatics (Giustini, 2007, Mesco 2007) the travel industry (Gruber 2007), publishing (Shaw 2010) and, of course education (Ohler 2008).

Since Web 3.0 is a combination of Web 2.0 and the Semantic Web, supporting and enhancing the applications with considerable organisational impact, we start by re-visiting the two well-known architectures and build on our findings. The rest of the paper is organised as follows: in section 2 we give a comprehensive overview of Web 2.0 and its use and role in today's organisation and Enterprise 2.0, from basic technologies and tools to innovation potential. Section 3 sums up the Semantic Web architecture and examines information modelling issues, challenges and its impact within the social organisation. In section 4 we look into integration

and we investigate tools for automation, quality issues and obstacles. Section 5 focuses on Web 3.0's organisational impact and section 6 presents our conclusions.

2. The user aspect: Web 2.0

Web 2.0 (O'Reilly 2005) was coined in 2005 by Tim O'Reilly and is a selection of technologies and applications rather than an architecture. Web 2.0 focuses on social interaction, end-user involvement and information sharing. The content is user-generated and the information modelling is informal, carried out bottom-up by means of user-generated tag systems. Data and information are seen as the driving forces. Paired with the relevant business practices, Web 2.0 gave birth to Enterprise 2.0, a term that describes the set of Web 2.0 technologies enabling access to collective intelligence within organisations. These core technologies enable innovation through websites/sources of collective content with functionality that gets enriched as more people use them. There are different ways to partition Web 2.0 technologies in order to examine their functionality, organisational impact and effectiveness in supporting innovation. The scope of the paper suggests that we follow the life-cycle of Web 2.0 content, from creation, distribution and re-use to its role as a vehicle of social interaction and then through to retrieval and deployment.

Compared to the traditional static web pages, Web 2.0 content can be dynamically generated by means of blogs, wikis, Ajax applications and RSS feeds. Organisational blogs are particularly widespread in both the private and public sectors (Kim et al 2008) and have a considerable effect on employee engagement, communication and collaboration. Integrated tools that combine data from more than one sources called *mashups* are used as situational applications that solve immediate business problems (Jhingran, 2006). Rigid content management systems are successfully aided or even replaced by collaborative wikis (Melhrose et al 2009). Information sharing and syndication are enabled by aggregators and RSS feeds, a widely adopted family of formats used to publish frequently updated content that improves organisational communication by streamlining smart information within employees' communities of practice, on their desktops, mobile devices or through their email clients.

The heart of Web 2.0 is social. The word "social" is used to form numerous compound terms such as social- computing, media, software and networks. Social computing has transformed digital economics with business models that are scalable, have low barriers for entry and are sustainable in the long term. Harnessing the power of social computing has created the need for organisational strategies that reflect the shift in online culture (Shuen 2008, Li & Bernoff 2008). The social organisation can be enclosed within the firewall when social interaction is limited to organisational networking and in-house communities of practice, or can tap into the rest of the web and maximise its use of collective intelligence. In the case of organisations with digital presence, user interactions in social networks, paired with effective communication govern the revenue models. Increasing the member base becomes crucial when the revenue model is advertising, willingness to pay is the prominent driver for a subscription model and trust is of paramount importance for revenue based on transactions (Enders et al 2008).

Web 2.0 information modelling is done by means of user-generated tags known as folksonomies (Smith 2008). Folksonomies are collaborative metadata, created bottom-up in an analytical synthetic way. They are successful in organising corporate (Patrick & Dotsika 2007) information and enable innovation (Hayman 2007). Information find-ability and organisational visibility are further improved by search engine optimization (SEO). SEO replaced the trend of acquiring Internet domain names relevant to the nature of the business carried out and ended the lucrative domain name speculation of the 90s.

Web 2.0 technologies gave marketing a great boost. Apart from Enterprise 2.0 and SEO-based marketing, there are a number of other methods that have evolved in parallel. With trend forecasting, marketing specialists look into web searches and keyword databases for sophisticated and accurate market predictions (Rangaswamy et al 2009). With web analytics, the analysis of a set of metrics provides information about website traffic and can be used in business research. In social media marketing, social networks are exploited to increase brand awareness, promote customer interaction, facilitate monitoring and achieve marketing objectives.

Web 2.0 deploys web services which are applications requested and executed remotely and which interface with one another providing a standard means of interoperating between different software applications. Web services share business logic, data and processes and promote interoperability and re-use. Web services' composition creates business processes and complex workflows and is regulated by standards such as *orchestration* and *choreography* (Busi et al, 2006). Adoption of web services is on the increase due to the fact that organisations associate competitive advantage with a process of ongoing adaptation through flexible business processes and web services are proven to be a key determinant on business process flexibility (Deependra & Jay 2005). Large organisations are not the only ones to benefit. Use of web services by small and medium enterprises (SMEs) can improve agility and deliver strategic benefits such as higher profit margins and better competitive positioning (Ray & Ray, 2006).

The table below expands the customary comparative analysis between Web 1.0 and Web 2.0, to include assisting technologies and ensuing organisational applications. The third column (Web 2.0) consists of the additional features that are thought of as Web 2.0, but also assumes the contents of the second column, that is the attributes, technologies and methods associated with Web 1.0. The final column of organisational innovation examples contains a small sample of relevant applications and is suggestive rather than exhaustive.

	Attribute	Web 1.0	Web 2.0 (Web 1.0+)	Web 2.0 assisting technologies	Organisational innovation & Enterprise 2.0
Content generation, distribution and re-use.	**Content nature**	Static	Dynamically generated pages	Blog publishing tools, AJAX	Organisational blogs, dynamic websites.
			Mash-ups	Mash-able APIs such as GoogleMaps, eBay, Amazon, Yahoo Traffic	Business, enterprise and advertising mashups.
	Content management	Content Management Systems	Wikis	Wiki technologies such as Mediawiki	Organisational wikis, Wikipedia
	Content manipulation and sharing	Screen/Web scraping, hyperlinks	Data and media sharing, syndication	Content aggregators, XML-based feeds	Adoption/use of RSS feeds
Social interaction	**Social interaction/ communication**	Websites & their visitors. Newsgroups & bulletin boards.	Social computing, social media	Social networking, virtual communities, online auctions, reputation systems, prediction markets.	Specialist groups (LinkedIn, Facebook etc.) Organisational social networks
Retrieval and deployment	**Information modelling**	HTML/XHTML	User tagging	Folksonomies	Organisational tag clouds, use of flickr, Del.icio.us, technorati, etc.
	Find-ability	Domain name speculation	Enhanced search algorithms	Search engine optimisation (SEO)	Search Engine Marketing (SEM), user profiles, targeted, social and viral marketing
	Marketing	Advertisements	Social media-based marketing	SEO, cookies, RSS, Web analytics.	
	Deployment	Websites	Web as a platform	Web Services	Amazon Web Services, Google Apps, etc.

Table 1. Web 2.0 technologies and tools

There are problems with Web 2.0, just like there are problems with everything that has participation and collaboration at its core (Ebner et al 2007, Vickery & Wunsch-Vincent 2007). Quality of information is at the centre of the disadvantages cited about Web 2.0 (Antiqueira et al 2007). Information modelling with folksonomies presents a number of further quality issues (Dotsika 2009). Other organisation-centred problems include technology dependence, security concerns, information overload and difficulties in finding relevant context. Ethical and legal issues such as privacy, anonymity, reputation, intellectual property rights, copyright violations, monetary function and trust are other often-quoted concerns. On the web services front, adoption is affected by low performance, basic forms of service invocation and service discovery issues (Wang et al 2004). While business adoption increases, organisations are reluctant to establish service registries, repositories and service level objectives.

3. The technology aspect: Semantic Web.

Tim Berners-Lee introduced the *Semantic Web* (SW) in 2001 (Berners-Lee 2001) as a form of web content where knowledge representation is standardised and relies on languages expressing information in a machine process-able form, by means of a framework based on RDF (Resource Description Framework) and ontologies. The information modelling is predominantly top-down and it is done formally, without the participation of end-users.

The organisational impact of the Semantic Web is based on system interoperability and adaptive, personalised information access. Interoperability addresses heterogeneity issues present in data and business processes and it ensures information integration across systems, a process too costly for any organisation. Interchange, distribution and creative reuse are a Semantic Web inherited standard, while scalability is dependent upon increasingly powerful implementations (Ankolekar et al, 2007). Adaptive technologies facilitate the tailoring of information access according to given user profiles. Intelligent information integration and agents such as information brokers, filters, personalised search agents and Knowledge Management Systems (KMS) are examples of innovative applications. Public sector adoption of web-based integrated KMS has overcome earlier challenges and the designated systems have proven their ability to support knowledge work and deliver strategic change (Butler et al, 2008).

The SW framework consists of XHTML, XML, the Resource Description Framework (RDF), a range of data interchange formats and notations and the Web Ontology Language (OWL).

On the semantic annotation front, XHTML supports *microformats*, a method that uses existing XHTML (or HTML) tags to semantically annotate web data (Allsop 2007). Their application is currently centred in the annotation of certain information such as contact details (hCard) and events (hCalendar) etc. The simplicity of microformats has made their adoption popular.

The Resource Description Framework (Beckett 2004) is an XML-based, standardised semantic annotation method, and, as such, interoperable. RDF modelling is done by means of subject-predicate-object expressions, known as *triples.* The RDF Schema (RDFS) adds basic ontology description power to plain RDF and many of its components are included in OWL. Together with RDF they form Semantic Web's RDF layer which adds semantics to web content and enhances machine process-ability. The model is scalable and searches are improved as the information can be processed in relation to the modelled relationships between data and/or resources. SPARQL is an RDF query language, part of the Semantic Web framework (WC3 2008A).

A number of "easier" interchange formats have been also used instead of RDF/XML. No major applications of these formats are currently adopted widely by organisations but they are briefly reviewed in the interest of completeness. RDFa is a specification of the W3C (Adida & Birkbeck 2008) and represents a simpler alternative to RDF that allows XHTML documents to be marked-up and allows the import of area specific vocabularies. Notation 3 (Berners-Lee, 1998) is a simpler, not XML-based, more readable version of RDF. Turtle (Terse RDF Triple Language) is a serialisation format for RDF graphs and a subset of N3. N-Triples (W3C 2001) is a line-based, plain text format for RDF graphs and a subset of Turtle.

The top part of the SW framework are ontologies, sets of shared, explicit and formal concepts used to organise and classify content. From an organisational point of view, ontologies are used to model enterprise information and processes accurately and consistently, enabling automatic reasoning, concept-based searches, process composition and knowledge discovery by means of intelligent agents (Hendler 2001). The Web Ontology Language OWL (Smith et al 2004) is a family of languages built using XML/RDF syntax and part of the Semantic Web framework.

Table 2 summarises the role, functionality and applications of Semantic Web technologies. Unlike Table 1, the focus here is the technologies supporting the content, rather than the content itself. The shaded part signifies technologies also used by Web 2.0.

	Assisting Technology	Role	Functionality	Support	Application	Innovation/ applications with organisational implications
Display	**HTML**	Data displaying mark-up	Web 1.0 content	Web browser	Web pages	
	XHTML	Data displaying mark-up XML-compatible	Web 2.0 content	Microformats	hCard, hCalendar, hNews etc.	XHTML Friends Network (XFN)
Syntax and Semantics	**XML**	Data describing mark-up	Modelling of data, data structures	Ajax, Data Object Model (DOM), Java applications, Web APIs	Dynamically updated web pages, web services	Web services supporting technologies and languages such as UDDI, WSDL, BPEL, etc.
	RDF, RDFa	XML-based semantics	Modelling information about resources	Data interchange, machine processing, increased findability	Semantic search, system interoperability, information remix and reuse, semantic mashups, etc.	RSS, AlchemyAPI, Wikipedia[3], etc.
	N3, Turtle, NTriples	Non-XML based semantics				Search-engine adoption, (Search-monkey), use in SEO, Wapedia, etc.
Rules and inference	**RDFS**	RDF + basic vocabulary	Modelling class hierarchies & properties	Intelligent agents, personalisation, adaptive information access	Natural language search engines, dynamic and adaptive contextual information builders, information brokers and filters, personalised search agents, intelligent adaptive social media, etc.	FOAF, SIMILE, OpenPSI, etc.
	OWL	Reasoning power	Modelling of rules and constraints			

Table 2. Functionality and application of Semantic Web technologies.

The problems with the Semantic Web are mostly of a technical nature and come as a consequence of the complexity that is associated with its technologies.

RDF is difficult to publish. Web content annotated with RDF requires XHTML for its textual presentation but also a parallel RDF/XML part to publish the semantic information. Any

development of RDF/RDFS or OWL requires specialised expertise and this has prevented widespread adoption. Its formality makes it difficult to master and limits its popularity.

Scalability is another concern. Once we take the Semantic Web applications outside the relatively few semantically annotated sites, it becomes apparent that the size of the web and the sheer amount of data it contains present a challenge. The creation of common ontologies and the mass transition to semantic annotation are more than a few years off.

When large ontologies are created, their quality can be an issue. The main problem is semantic uncertainty, which can be divided into ambiguity, randomness, inconsistency, incompleteness and vagueness (W3C 2008B). Handling semantic uncertainty plays an important role in ontology languages for the Semantic Web.

All this makes organisational adoption expensive and cumbersome. While large companies and high budget projects embrace the Semantic Web readily in order to take better advantage of intellectual assets, enhance productivity and increase competitiveness, smaller companies with web presence have remained reluctant to do the same.

4. Web 2.0 and Semantic Web integration.

The advantages of merging the Web 2.0 technologies with the Semantic Web infrastructure are obvious. But what exactly are the practicalities involved? And how can organisations achieve such transition? There are three different approaches for reconciling Web 2.0 and the Semantic Web.

The first is the obvious, “straightforward” method: start from scratch and create web resources which follow the standards of the Semantic Web platform before end-users are allowed to add their (probably somewhat restricted) bottom-up markup and collaborative tagging. Organisational or off-the-shelf ontologies might be used and interoperability will be ensured. However, this is a scenario that aligns almost exactly with the creation of Semantic Web pages and applications, and therefore it is not addressed at this stage. Instead, we will concentrate on the two other existing methods of integration: the transformation of folksonomies into ontologies and the use of semantic APIs.

4.1. Transforming folksonomies into ontologies

This approach makes use of the richness of Web 2.0 by retaining the flexibility, collaboration and information aggregation of existing folksonomies and transforming them into ontologies. There are a number of methods that follow this route. The most popular/known are:

- The creation of *FolksOntologies* (Van Damme et al. 2007) is a method that derives ontologies from folksonomies analysing the latter and their associated data to determine relations, complements the output with online lexical resources and employs ontology mapping techniques where conceptual elements can be matched based on the labels, ontology structure or both.

- Another method makes explicit the semantics behind the folksonomy tag space (Specia & Motta 2007) and integrates folksonomies with the Semantic Web by employing occurrence analysis and clustering techniques.

- Deriving semantics from folksonomies can be done by statistically analysing the tags and creating a *tag cloud* (i.e. a set of related tags depicted in different font sizes and colours according to their weight/cardinality) (Lux & Dosinger 2007). By means of computing the tags

co-occurrence, the cloud is transformed into a weighted, directed *network of tags* which in turn is used to create an ontology.

- Another approach is to capture latent emotional semantics in social tagging systems (Baldoni et al. 2008) by means of adding a semantic layer to the social tagging of *Arsmeteo*, a web portal for sharing works of art containing a folksonomy of over 10,000 tags. These tags are related to *OntoEmotions*, an OWL ontology chosen for fitting the application purposes.

All the above methods share common problems with quality assurance, mapping efficiency and ethical issues.

Quality issues are present in both folksonomies and ontologies. In folksonomies the problems are ambiguity (polysemes and homonyms), inexactness (synonyms), granularity discrepancies (Golder & Huberman 2005) and, of course, misspellings and inaccuracies. Ontologies suffer from issues of completeness, transformation rules, domain expertise, structural and atomic qualities (Colomb & Weber 1998; Rector et al. 2001; Kashyap 2003).

When it comes to information mapping, the existing methods are inefficient in mapping certain additional information contained in tags that semantically corresponds to attributes and/or properties. A further problem is the possible absence of a relevant ontology so that special tags cannot be adequately mapped.

In the area of ethics, transforming folksonomies to ontologies requires to harvest information from several systems, a process that implies a level of trust and raises a certain level of ethical questions.

In order to alleviate these problems and regulate the process, an integration framework has been proposed (Dotsika 2009). The framework identifies the existing shortcomings, groups them according to the integration requirements and suggests four steps that can be followed to regulate the transformation: (a) quality assurance, (b) semantic enrichment, (c) mapping completeness and (d) issues of trust and ethics.

From an organisational point of view, the main advantages of the above methods of integration are the preservation of the organically grown tag systems and the safeguarding of the bottom-up design, collective intelligence assets and end-user involvement. However, the alignment of folksonomies is not an inexpensive operation, especially since the methods presented are only partially automated and therefore need to be tailored to specific organisational requirements.

4.2. Semantic APIs

This method adds semantics to existing web content automatically, by means of specialist *semantic* Applications Programmers Interfaces (APIs) which take unstructured text input and return the content's contextual framework. There are a number of semantic APIs available, offering a variety of options and flexibility. The best known are:

- The Dapper (Data Mapper) API (Dapper, 2005) enables developers to extract semantics from web content in the form of an XML document that can then be used to build mashups, RSS feeds and other applications. The Dapper Semantify web service allows the user to define the content of interest, reads the website and creates a feed of the specific content.

- OpenCalais (Calais, 2008) is an automatic generator of semantic metadata in RDF format from web content, based on natural language processing (NLP). It works on text only and operates as a web service. The API reads in unstructured documents, recognises a number of different entities and annotates them semantically.

- SemanticHacker (SemanticHacker, 2008) is an API that takes text as input and classifies the document content into categories. The classification is done by identifying and returning a number of entities from a given classification scheme (the Open Directory Project). Their weight is then measured and a relevance score returned. The system employs NLP and text mining techniques.

- The Semantic Cloud service (Semantic Cloud API, 2009) identifies and extracts semantics from a web page or a document, creates a semantic cloud of concepts and generates a list. As an alternative it can take a set of URLs as input and return a multi-document summary about the main concepts present and/or an essay on a specific topic.

- The Zemanta API (Zemanta 2009) takes in unstructured text and returns tags, categories, links, photos, and related articles. The service acts as a single-point entry to various, pre-indexed, content databases. Zemanta analyses the postings, discovers relevant content and adds it to the page or document. The system uses NLP and semantic algorithms and categorises content by comparing it to their pre-indexed database.

- The Ontos API Semantic web service (Ontos 2009) provides the means to personalise the NLP platform that returns named entities and semantic relations when fed with non-semantically annotated text. Users can define their own semantic content via external dictionaries and can tune concepts from core ontologies. Ontos supports visual representations in the form of cognitive maps, dynamic reports and summaries from document collections.

There have been studies to evaluate and compare the various systems in order to inform and steer organisational adoption (Dotsika 2010; DiCiuccio 2010). Due to the disparity of the products and the inconsistencies in the way the semantic APIs annotate web content, the evaluation is generally troublesome. The comparisons take into account performance and other basic product information, requirement-based decision planning and information modelling capabilities, and, in terms of classification schemes adopted, input sources and output formats.

All products identify key concepts and categories but depending on the original input, disambiguation issues and low entity-return seem to affect most APIs. The majority provide extra tools and plugins to customise results. Apart from the APIs own taxonomies, they allow custom taxonomies to be used as input and support a variety of output formats. Overall however, while the performance is not a problem, content annotation is fairly dependent upon the original content.

From an organisational point of view, the semantic APIs are the cheapest method of integration available. Since the design is top-down, the preservation of user-generated tags is problematic. The quality of the end product is also an issue, though most APIs allow for custom taxonomies which can theoretically improve the quality of the semantic tagging and, depending on the application, more than one semantic APIs can be used. Nevertheless, a lack of case studies of official adoption means that a fuller evaluation is not yet possible. The table below summarises our results and compares the different methods of integration.

	Folks→ Ontos	SemAPIs	Ab initio
Design	bottom-up	top-down	top-down
End user involvement	✓✓	✓	×
Folksonomy ⇔ ontology mapping	✓	×	possible
Information loss avoidance	limited	limited	✓
Flexibility – customisation	limited	possible	according to spec
Attributes/ complex tags	×	×	✓
Automation	partial (some methods)	✓✓	✓
Cost	~~£~~£	~~£~~	~~£~~£~~£~~
Evaluation/metrics/results	×	some	some

Table 3. Integration methods

5. The organisational implications of Web 3.0.

One of the main practical implications of Web 3.0 is the quality of information attained, as it has a direct impact on organisational success and profitability. Gathering the above facts we adopt the four-category quality model (Wang et al 1997; Zhu & Wang 2010) to create the comparative analysis table for web information quality. The focus is on organisational information rather than individual data. The first group of dimensions (*accuracy, objectivity* and *reliability*) are inherent qualities and therefore their values are, strictly speaking, unknown. However, the Semantic Web provides a logical, if weak, guarantee of quality control, due to the high cost of its application. Web 2.0 reputation systems can be deployed to enable reputation quality. The next group addresses contextual quality and is dependent on the nature of the task to be performed. While Web 2.0 technologies offer the potential to enhance all related dimensions (*relevancy, value-added, timeliness, completeness* and *volume*), it is the Semantic Web and Web 3.0 that provide the means for actual improvement. The categories of accessibility and representational quality focus on the employed infrastructure and are compared accordingly. Attributes such as *ease of understanding* and *concise representation* for instance score the same, although the underlying enabling technologies are different and therefore cannot be thought of as interchangeable. Security assessment is “naive” and does not involve particular web service security, data storage and information leakage issues. Table 4 bellow shows our findings.

Category	Dimension	Web 1.0	Web 2.0	SW	Web 3.0 [inherits from Web 2.0 & SW]
Intrinsic Data Quality	Accuracy	?	weak control	possible	improved
	Objectivity	?	weak control	possible	improved
	Believability	?	weak control	possible	improved
	Reputation	?	control mechanisms available	possible	control mechanisms available
Contextual Data Quality	Relevancy	?	improved	✓✓✓	✓✓✓
	Value-Added	×	improved	✓✓✓	✓✓✓
	Timeliness	?	improved	✓✓	✓✓✓
	Completeness	×	improved	✓✓	✓✓✓
	Amount of Data	?	improved	✓✓	✓✓✓
Accessibility	Accessibility	✓	✓✓	✓✓✓	✓✓✓
	Access Security	✓	✓✓	✓✓	✓✓
Representation	Interpretation	✓	✓✓	✓✓✓	✓✓✓
	Ease of Understanding	✓	✓✓✓	✓✓✓	✓✓✓
	Concise Representation	✓	✓✓✓	✓✓✓	✓✓✓
	Consistent Representation	✓	✓✓	✓✓✓	✓✓✓

Table 4. Web information quality

The next step is to sum up the information gathered about other aspects of significant impact from an organisational point of view and create a second table for reference and comparison. For consistency we maintain the facets we identified in section 2, focused on content generation, distribution, retrieval and deployment. Content generation is the category that stands out in terms of enhanced performance. The result is not a surprise as Web 3.0's main strengths are personalisation, custom and on-demand content. Distribution does not fare any different to previous web frameworks and there is no evidence that content search would improve that of the Semantic Web. Advanced automation enables networking to be content- as well as consumer-directed. The scalability and tractability attributed to Web 2.0 are not that

clear in Semantic Web environments and they have been deliberately left undefined. Table 5 presents the results of the analysis.

	Organisational aspect	Web 1.0	Web 2.0	SW	Web 3.0
Content generation, distribution and reuse	**Seamless, on-demand content**	×	✓✓	✓✓	✓✓✓
	Info analysis: Personalisation - tailoring	×	✓	✓✓	✓✓✓
	Info synthesis: Custom mashups	×	✓	✓	✓✓✓
	Interchange, distribution, creative reuse	×	✓✓	✓✓	✓✓
	Ownership	individual	shared	either	either
Social aspect	**Networking**	×	content-directed	content-directed	content- or consumer-directed
Content retrieval and deployment	**Search**	✓	✓✓	✓✓✓	✓✓✓
	Scalability - tractability	×	✓	?	?
	Web services – cloud computing	×	✓	✓✓	✓✓
	Media-centric capabilities	×	×	limited	limited

Table 5. Web 3.0 benefits for the enterprise

Finally we look into the need of organisations to respond to technological as well as socio-economic trends as a means to improve competitiveness and promote sustainability. The assembled facts were analysed in order to derive information on how Web 3.0 supports organisational change and sustainable development.

In order to assess the impact of web technologies on organisational change we follow the four categories classification of changes (Flood, 1996; Cao et al, 1999):

(a) changes in organisational processes (business processes, process-driven workflow);
(b) changes in organisational functions (structural change, possibly affecting decision systems and policies, resource allocation mechanisms, organising human resources);
(c) changes in values (cultural change in stakeholders' behaviour and values);
(d) changes in power (power distribution within the organisation, factors that influence power dynamics, scalability issues).

Web adoption brought on change that is often explained by means of the *e-adoption ladder* model (Martin and Matlay, 2001; Jones *et al*., 2003). The model depicts web-based organisational change as a linear process. The arrival of Web 2.0 and the consequent adoption of social software and web services revolutionised business processes and employees' behaviour and lead to radical changes that are yet to be sufficiently measured and analysed. The actual extend of this change is neither matched by that of the Semantic Web, nor is predicted for Web 3.0.

All types of organisational change are, depending on the type of company, present in and influenced by Web 2.0, Semantic Web and Web 3.0 environments. Change in organisational processes is predominant in organisations adopting web services and cloud computing. It is process-focused and largely dependent on workflow optimisation. As such, it is prevalent in the implementation and composition of web services, especially when composition standards are present (web service orchestration and choreography), and therefore best supported by the Web 3.0 infrastructure. Automated service discovery, a field that has proven problematic, is another area that stands to benefit (Klusch et al, 2006; Henze et al, 2006). Organisational functions are equally influenced by resource allocation and decision support mechanisms (Bonatti et al, 2006). Value changes identified were predominantly cultural and changes in power were linked to scalability issues. There is no evidence to suggest that Web 3.0 facilitates scalability more than its predecessors.

The web's promotion of sustainable development can be thought of as three-fold: it applies the forefront of technological development in information and communication technologies to business processes, aides the emergence of new markets and creates a new generation of smart and creative stakeholders. Sustainable development is assessed following the three aspects that analyse the conceptual developments underpinning sustainability (Faber et al., 2005): artefact (entity-construct), goal orientation (absolute-relative) and behavioural interaction (static-dynamic). Looking into the part of the organisation (assets, functions and processes) that is realised by means of the web infrastructure, the kind of artefact is identified as a construct in all categories. Goal orientation relates to the point of reference that is used to determine the artefact's sustainability. There is no absolute approach to web-based sustainability, therefore goal orientation is recorded as relative. The final aspect of behavioural interaction explores the dynamics of the artefact and the environment. In all occasions this is dynamic as both the artefact and the environment change. As a result, while organisational web adoption follows the construct-oriented approach and a relative, dynamic perspective that enhance sustainability (Faber et al., 2005), there is no evidence that Web 3.0 will make organisations more or less sustainable than the other web frameworks. Table 6 summarises our findings.

	Organisational aspect	Web 1.0	Web 2.0	SW	Web 3.0
Change	**Processes**	limited	✓✓	✓	✓✓
	Functions (structural change)	×	✓	✓✓	✓✓
	Values (cultural change)	✓	✓✓	✓	✓
	Power (scalability mainly)	×	✓✓	✓	✓✓
Sustainability	**Artefact**	×	construct	construct	construct
	Goal orientation	×	relative	relative	relative
	Behavioural interaction	×	dynamic	dynamic	dynamic

Table 6. Organisational change and sustainability

6. Conclusions and discussion.

This paper addresses the impact and implications of Web 3.0 from an organisational perspective. Having defined Web 3.0 as the integration of Web 2.0 and the Semantic Web, the research carried out investigated the parent web frameworks as a first step and recorded their

distinctive capabilities in order to set the base for comparative analysis and impact assessment for the new generation of web technologies.

Automated means of migration to Web 3.0 and organisational adoption were explored. The methods for transforming folksonomies into ontologies were deemed disappointing in terms of automation and derived information quality. However these methods safeguard bottom-up design and entail the highest end-user involvement. The alternative methods of semantic APIs provide a fully automated solution and are the cheapest. Their top-down design however limits end-user involvement. While waiting for the semantic APIs to evolve, deriving Web 3.0 web resources from scratch is presumed to be the best method. Nonetheless, the skills' level required and overall cost, make mass-adoption of this method a theoretical rather than practical approach.

Information quality was evaluated by means of a comparative analysis table based on information quality aspects. Apart from intrinsic data quality, where effects were mostly speculated at, the Web 3.0 framework yields the best results. Contextual data quality, accessibility and representation fared better than, or as well as, the best other category.

Web 3.0 contributed equally positively in all aspects addressing organisational content generation, distribution, retrieval and reuse. The content-directed networking of previous web generations is maintained and supplemented with the consumer-directed choice. Deployment of web services and cloud computing remain the major promoters of scalability and sustainability, despite their unassuming presence in the matrix.

There is enough evidence to suggest that the next web generation will be a hybrid mix of Web 2.0 technologies reinforced with semantic markup. Whether this markup will be the formal, robust variety of the Semantic Web or an automated, user-friendly approach, easier to implement and therefore better suited for organisational adoption, is yet to be seen. An obvious stepping stone towards this direction is the use of semantic APIs. Their continuing evolution requires further investigation and their detailed assessment and evaluation is part of our future research. Another aspect is the investigation of how the different web generations influence organisational change and sustainability. This one is also part of future research.

References

Adida B., Birkbeck M. (2008) [online], RDFa Primer, Bridging the Human and Data Webs, http://www.w3.org/TR/xhtml-rdfa-primer/ accessed 24 July 2010

Allsop J. (2007), Microformats: Empowering Your Markup for Web 2.0, Friends of Ed, NY

Anderson, Chris (2006), The Long Tail: Why the Future of Business Is Selling Less of More. New York: Hyperion. ISBN 1-4013-0237-8

A. Ankolekar, M. Krötzsch, T. Tran and D. Vrandecic, (2007), The Two Cultures, Mashing up Web 2.0 and the Semantic Web, *Proceedings of the 16th International Conference on World Wide Web* Banff, Alberta, Canada (May 2007), pp. 825–834.

Antiqueira, L.; Graças, M.; Nunesm V.; Oliveira O. N.; Da F. Costa, L. (2007), "Strong correlations between text quality and complex networks features" *Physica. A*, 373, 2007. pp. 811-820

Baldoni, M., Baroglio, C., Horváth, A., Patti, V., Portis, F., Avilia, M., and Grillo, P., Folksonomies meet ontologies in ARSMETEO: from social descriptions of artifacts to emotional concepts, In *Proc*. of Formal Ontologies Meet Industry (FOMI 2008*)*, Torino, Italy, June 2008

Beckett D. (ed) (2004) [online] RDF/XML Syntax Specification, http://www.w3.org/TR/rdf-syntax-grammar/ , accessed 24 July 2010

Berners-Lee T., (1998) Notation 3 Specification, (online) http://www.w3.org/DesignIssues/Notation3.html , accessed 24 July 2010

Berners-Lee T., Hendler J., Lassila O. (2001), The Semantic Web, Scientific American, May 2001

Briscoe, B., Andrew Odlyzko, and Benjamin Tilly. 2006. Metcalfe's Law is Wrong., IEEE Spectrum, July 2006

N. Busi, R. Gorrieri, C. Guidi, R. Lucchi, and G. Zavattaro. Choreography and orchestration conformance for system design. In *COORDINATION*, volume 4038 of *LNCS*, pages 63.81, 2006.

Butler, T., Feller, J., Pope, A., Emerson, B., Murphy, C., (2008), Designing a core IT artefact for Knowledge Management Systems using participatory action research in a government and a non-government organisation The Journal of Strategic Information Systems , Volume 17 Issue 4, December 2008

Calais, (2008) [online], http://www.opencalais.com/ accessed 24 July 2010

Cao,G., Clarke, S., Lehaney, B (2000), Towards systemic management of diversity in organisational change", Strategic Management, Vol. 8 No. 4, pp. 205-16.

Colomb R.M. and Weber R. 1998, Proceedings of the *International Conference on Formal Ontology in Information Systems* (FOIS'98) Trento, Italy, 6-8 June, 1998. In N. Guarino (ed.) *Formal Ontology in Information Systems* IOS-Press (Amsterdam) pp. 207-217

Dapper, (2005), [online] http://www.dapper.net/ accessed 24 July 2010

Deependra Moitra , Jai Ganesh, Web services and flexible business processes: towards the adaptive enterprise, Information and Management, v.42 n.7, p.921-933, October 2005

Dotsika, F., (2009) *Uniting formal and informal descriptive power: Reconciling ontologies with folksonomies*, International Journal of Information Management, Volume 29, Issue 5, October 2009, pp 407-415,

Dotsika, F., (2010) *Semantic APIs: scaling up towards the Semantic Web*, International Journal of Information Management, Volume 30, Issue 4, August 2010, pp 335-342

Ebner, M., Holzinger, A. & Maurer, H. (2007) Web 2.0 Technology: future interfaces for technology enhanced learning? *Lecture Notes in Computer Science*, 4556, 559-568.

Enders, A., Hungenberg, H., Denker, H., Mauch, S., (2008), The Long Tail of Social Networking: Revenue Models of Social Networking Sites, European Management Journal, Volume 26 (3), June 2008, p. 199-211. 8 15.579

Garrett, J. (2005) [online], Ajax: A New Approach to Web Applications. Adaptive Path website, http://www.adaptivepath.com/publications/essays/archives/000385.php accessed 24 July 2010 , accessed 24 July 2010.

Gilder, G., (1993), Metcalf's Law and Legacy, Forbes ASAP

Giustini, D., (2007), Web 3.0 and Medicine, British Medical Journal, 335(7633), December 2007, pp 1273-1274

Golder, S., and Huberman, B.A., 2005, [Online], The Structure of Collaborative Tagging Systems. HP Labs technical report, http://www.hpl.hp.com/research/idl/papers/tags/

Gruber, T., (2006) Where the Social Web meets the Semantic Web, 5th International Semantic Web Conference, November 7, 2006

Hayman, Sarah (2007), Folksonomies and Tagging, New Developments in Social Bookmarking, Ark Group Conference: Developing and Improving Classification Schemes, Sydney June 2007

Hendler, J. (2001) Agents and the semantic web. IEEE Intelligent Systems, 16(2). pp. 30-37

Hendler, J., (2008), " Web 3.0: Chicken Farms on the Semantic Web" *Computer*, 41 (1), 2008, pp.106-108

Hendler, J., Golbeck, J., (2008) Metcalfe's law, Web 2.0, and the Semantic Web, Web Semantics: Science, Services and Agents on the World Wide Web, v.6 n.1, p.14-20, February, 2008

Jhingran, A., (2006), Enterprise information mashups: integrating information, simply, Proceedings of the 32nd international conference on Very Large Data Bases, Seoul, Korea 2006

Johnson, D., (2005) [online], AJAX: Dawn of a new developer: The latest tools and technologies for AJAX developers. JavaWorld.com, http://www.javaworld.com/javaworld/jw-10-2005/jw-1017-ajax.html , accessed 24 July 2010.

Jones, C., R. Hecker and P. Holland (2003), 'Small Firm Internet Adoption: Opportunities Foregone, a Journey Not Begun', *Journal of Small Business and Enterprise Development* 10, 3, 287–297.

Kashyap V. 2003,*Trust and quality for Information Integration: The Data-Metadata-Ontology Continuum*, Workshop on Data Quality, Dagstuhl, Germany, September 2003

Kim, J., Lee, SH., Shin, M. S., (2008), Current usage of organisational blogs in the public sector, International Journal of Information Technology and Management 2008 - Vol. 7, No.2 pp. 201 - 216

Lassila O., Hendler, J., (2007), Embracing Web 3.0. Internet Computing, IEEE, 11(3):90-93, 2007.

Li, Charlene; Bernoff, Josh (2008). Groundswell: Winning in a World Transformed by Social Technologies. Boston: Harvard Business Press

Lux, M., Dosinger, G. (2007), From folksonomies to ontologies: employing wisdom of the crowds to serve learning purposes International, International Journal of Learning Technology, Volume 3, No 4/5, pp. 515-528

Meloche, J. A., Hasan, H. M., Willis, D., Pfaff, C. & Qi, Y. (2009). Co-creating Corporate Knowledge with a Wiki. International Journal of Knowledge Management, 5 (2), 33-50.

Mesko B. (2007), [online], Web 3.0 and medicine. ScienceRoll blog, http://scienceroll.com/2007/04/06/web-30-and-medicine/, accessed 28 July 2010

Ohler, J., (2008), The Semantic Web in Education, *Educause Quarterly,* vol. 31, no. 4, 2008

Ontos, (2009) [online], http://www.ontos.com/o_eng/index.php?cs=2-1 accessed 2 August 2010

O'Reilly, T. (2005), [online], *What is Web 2.0?*, http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html accessed 14th July 2010

Patrick, K., & Dotsika, F., 2007, Knowledge Sharing: Developing from Within The Learning Organization: The International Journal of Knowledge and Organizational Learning Management, Vol 14, Iss 3, July 2007.

Rangaswamy, A., CL Giles, Seres, S., (2009) A strategic perspective on search engines: Thought candies for practitioners and researchers, Journal of Interactive Marketing 23 (2009) 49–60

Ray, A., Ray, J., (2006) Strategic benefits to SMEs from third party web services: An action research analysis, The Journal of Strategic Information Systems, Volume 15 Issue 4, December, 2006.

Rector A.L., Wroe C., Rogers J., Roberts A. 2001 Untangling taxonomies and relationships: personal and practical problems in loosely coupled development of large ontologies. K-CAP 2001: 139-146

Semantic Cloud API, (2009) [online], http://www.semanticengines.com/api.aspx accessed 2 August 2010

SemanticHacker, (2008) [online], http://www.semantichacker.com/ accessed 2 August 2010

Shaw, T., (2010) [online], Can Web 3.0 save the publishing industry?, http://www.econtentmag.com/Articles/ArticleReader.aspx?ArticleID=67796 , accessed 2 August 2010

Shuen, A., (2008) Web 2.0: A Strategy Guide Business thinking and strategies behind successful Web 2.0 implementations, O'Reilly Media, Inc

Smith, G. (2008). Tagging: People-Powered Metadata for the Social Web. Berkeley, CA: New Riders.

Smith, M.K., Welty, C., McGuinness, D.L. (2004) [online], OWL Web Ontology Language, http://www.w3.org/TR/owl-guide/ accesses 24 July 2010

Specia, L. and Motta, E., 2007, Integrating Folksonomies with the Semantic Web, in the Proccedings of 4th European Semantic Web Conference, Innsbruck, Austria.

Udell, Jon (2004), [Online], Collaborative knowledge gardening, InfoWorld. http://www.infoworld.com/article/04/08/20/34OPstrategic_1.html , accessed 28 July 2010

Van Damme, C., Hepp, M., Siorpaes, K. (2007). FolksOntology: An Integrated Approach for Turning Folksonomies into Ontologies. Proceedings of the ESWC 2007 Workshop "Bridging the Gap between Semantic Web and Web 2.0", Innsbruck, Austria, 2007.

Vickery, G., Wunsch-Vincent, S. (2007). *Participative web and user-created content: Web 2.0, wikis and social networking*, Paris: Organization for Economic Co-operation and Development

Wang, H., Huang, J. Z., Qu, Y., & Xie, J. (2004), Web services: Problems and Future Directions. *Journal of Web Semantics, 1*, 309-320

W3C 2001, [online], W3C RDF Core WG Internal Working Draft , http://www.w3.org/2001/sw/RDFCore/ntriples/, accessed 24 July 2010

W3C 2008A, [online] , SPRQL Query Language for RDF, W3C recommendation, http://www.w3.org/TR/rdf-sparql-query/, accessed 24 July 2010

W3C 2008B, [online] , Uncertainty reasoning for the WWW, W3C Incubator Group report, http://www.w3.org/2005/Incubator/urw3/XGR-urw3-20080331/, accessed 24 July 2010

Zemanta, (2009), [online] http://www.zemanta.com/api/ accessed 24 July 2010.

Zhu, H., Wang, RY, (2010) Information Quality Framework for Verifiable Intelligence Products, Data Engineering, International Series in Operations Research & Management Science, Springer, Volume 132, 315-333 2010.

APPENDIX 2
GISMoE CODE

```
/*
        GISMoE_Applet.java
*/
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
import javax.swing.*;

public class GISMoE_Applet extends JApplet implements ActionListener {
        int c = -2;
        public static int BASE_ENTITY = 1;
        public static int ABSTRACT_ENTITY = 2;
        public static int POINTER = 3;
        public static int FUNCTION = 4;
        public static int DESTROY = 7;
        public static int WINDOW1 = 0;

        SchemaScreen drawPad, drawPad1;
        JButton abstractEntity, baseEntity;
        public static String textFont = "Arial Narrow";
        public static int textStyle = Font.BOLD;
        public static int textSize = 12;
        private boolean isApp_ = false; // Are we a Applet or NOT?
        public void init() {
                Container content = getContentPane();
                content.setLayout(new BorderLayout());
                myinit(content);
        }
        public void myinit(Container content) {
                Font defaultFont = new Font(textFont, textStyle, textSize);
                // MENUS on TOP
                // create the File menu
                JMenu fileMenu = new JMenu("File");
                fileMenu.setIcon(new ImageIcon(getmyImage("Bfile.gif")));
                fileMenu.setVerticalTextPosition(AbstractButton.TOP);
                fileMenu.setHorizontalTextPosition(AbstractButton.CENTER);
                fileMenu.setFont(defaultFont);
                fileMenu.setForeground(Color.black);
                fileMenu.setToolTipText("file");
                JMenuItem fLoad = new JMenuItem("Load");
                fLoad.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                                System.out.println("calling: Load");
                                PopupWindow p = new PopupWindow(new Frame(), drawPad, "load");
                        }
                });
                fileMenu.add(fLoad);
                JMenuItem fSave = new JMenuItem("Save");
                fSave.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                                System.out.println("calling: Save");
                                PopupWindow p = new PopupWindow(new Frame(), drawPad, "save");
                        }
                });
                fileMenu.add(fSave);
                JMenuItem fClear = new JMenuItem("Clear");
                fClear.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                                System.out.println("calling: Clear");
                                PopupWindow p = new PopupWindow(new Frame(), drawPad,
                                                "Not Implemented");
                        }
                });
                fileMenu.add(fClear);
                JMenuItem fSaveAs = new JMenuItem("Save As");
                fSaveAs.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                                System.out.println("calling: Save As");
                                PopupWindow p = new PopupWindow(new Frame(), drawPad,
                                                "Not Implemented");
                        }
                });
                fileMenu.add(fSaveAs);
                // only do in an application. Applets do not have a QUIT
                if (isApp_) {
                        JMenuItem quitItem = new JMenuItem("Quit");
                        quitItem.setMnemonic(KeyEvent.VK_Q);
                        quitItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_Q,
                                        Event.CTRL_MASK));
                        quitItem.addActionListener(new ActionListener() {
                                public void actionPerformed(ActionEvent e) {
                                        System.exit(0);
                                }
                        });
                        fileMenu.add(quitItem);
                }
                // create the Data Dictionary menu
                JMenu dictionaryMenu = new JMenu("DataDict");
                dictionaryMenu.setIcon(new ImageIcon(getmyImage("Bdictionary.gif")));
                dictionaryMenu.setVerticalTextPosition(AbstractButton.TOP);
                dictionaryMenu.setHorizontalTextPosition(AbstractButton.CENTER);
                dictionaryMenu.setFont(defaultFont);
                dictionaryMenu.setForeground(Color.black);
                dictionaryMenu.setToolTipText("data dictionary");
                JMenuItem entitydictionaryItem = new JMenuItem("XML elements");
                entitydictionaryItem.setEnabled(true);
                entitydictionaryItem.addActionListener(new ActionListener() {
```

```
        public void actionPerformed(ActionEvent e) {
                System.out.println("calling: XML elements");
                PopupWindow p = new PopupWindow(new Frame(), drawPad,
                                "XML elements");
        }
});
dictionaryMenu.add(entitydictionaryItem);
JMenu hybrid2 = new JMenu("Functional Data Dictionary");
JMenuItem fddEntities = new JMenuItem("Entities");
fddEntities.setEnabled(true);
fddEntities.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
                System.out.println("calling: Functional Data Dictionary:Entities");
                PopupWindow p = new PopupWindow(new Frame(), drawPad,
                                "Functional Data Dictionary:Entities");
        }
});
hybrid2.add(fddEntities);
JMenuItem fddFunctions = new JMenuItem("Functions");
fddFunctions.setEnabled(true);
fddFunctions.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
                System.out.println("calling: Functional Data Dictionary:Functions");
                PopupWindow p = new PopupWindow(new Frame(), drawPad,
                                "Functional Data Dictionary:Functions");
        }
});
hybrid2.add(fddFunctions);
dictionaryMenu.add(hybrid2);
// create the Schema menu
JMenu schemaMenu = new JMenu("Schema");
schemaMenu.setIcon(new ImageIcon(getmyImage("Bschema.gif")));
schemaMenu.setVerticalTextPosition(AbstractButton.TOP);
schemaMenu.setHorizontalTextPosition(AbstractButton.CENTER);
schemaMenu.setFont(defaultFont);
schemaMenu.setForeground(Color.black);
schemaMenu.setToolTipText("schema");
// viewMenu.setMnemonic('S');
JMenuItem xmlSchema = new JMenuItem("XML DTD");
xmlSchema.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
                System.out.println("calling: XML DTD");
                PopupWindow p = new PopupWindow(new Frame(), drawPad, "XML DTD");
        }
});
schemaMenu.add(xmlSchema);
JMenuItem fSchema = new JMenuItem("Functional Schema");
fSchema.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
                System.out.println("calling: Functional Schema");
                PopupWindow p = new PopupWindow(new Frame(), drawPad, "schema");
        }
});
schemaMenu.add(fSchema);
// create the Schema menu
JMenu dataMenu = new JMenu("createDB");
dataMenu.setIcon(new ImageIcon(getmyImage("BcreateDB.gif")));
dataMenu.setVerticalTextPosition(AbstractButton.TOP);
dataMenu.setHorizontalTextPosition(AbstractButton.CENTER);
dataMenu.setFont(defaultFont);
dataMenu.setForeground(Color.black);
dataMenu.setToolTipText("schema");
JMenuItem createDB = new JMenuItem("createDB");
createDB.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
                System.out.println("calling: createDB");
                PopupWindow p = new PopupWindow(getAppletContext(),
                                new Frame(), drawPad, "CreateDB");
        }
});
dataMenu.add(createDB);
JMenu edfMenu = new JMenu("EDF");
edfMenu.setIcon(new ImageIcon(getmyImage("edf.gif")));
edfMenu.setVerticalTextPosition(AbstractButton.TOP);
edfMenu.setHorizontalTextPosition(AbstractButton.CENTER);
edfMenu.setFont(defaultFont);
edfMenu.setForeground(Color.black);
edfMenu.setToolTipText("EDF");
JMenuItem edf = new JMenuItem("EDF");
edf.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
                System.out.println("calling: HELP");
                PopupWindow p = new PopupWindow(new Frame(), drawPad,
                                "Not Implemented");
        }
});
edfMenu.add(edf);
JMenu helpMenu = new JMenu("Help");
helpMenu.setIcon(new ImageIcon(getmyImage("Help.gif")));
helpMenu.setVerticalTextPosition(AbstractButton.TOP);
helpMenu.setHorizontalTextPosition(AbstractButton.CENTER);
// helpMenu.setIconTextGap(0);
helpMenu.setFont(defaultFont);
helpMenu.setForeground(Color.black);
helpMenu.setToolTipText("help");
// helpMenu.setMnemonic('H');
JMenuItem aboutHelp = new JMenuItem("About");
aboutHelp.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
                System.out.println("calling: HELP");
                PopupWindow p = new PopupWindow(new Frame(), drawPad,
```

```
                                        "Not Implemented");
                }
        });
        helpMenu.add(aboutHelp);
        // create a menu bar and use it in this JFrame
        JMenuBar menuBar = new JMenuBar();
        menuBar.add(fileMenu);
        menuBar.add(dictionaryMenu);
        menuBar.add(schemaMenu);
        menuBar.add(dataMenu);
        // menuBar.add(viewMenu);
        menuBar.add(edfMenu);
        menuBar.add(Box.createHorizontalGlue()); // Right Hand Side
        menuBar.add(helpMenu);
        content.add(menuBar);
        setJMenuBar(menuBar);
        // </MenuBar>
        // TOOLBAR on LEFT
        // create right hand toolbar
        Icon abstractEntityIcon = new ImageIcon(getmyImage("keythree.gif"));
        abstractEntity = new JButton(abstractEntityIcon);
        abstractEntity.setVerticalTextPosition(AbstractButton.TOP);
        abstractEntity.setHorizontalTextPosition(AbstractButton.CENTER);
        abstractEntity.setToolTipText("abstract entity");
        abstractEntity.setActionCommand("AbstractEntity");
        abstractEntity.addActionListener(this);
        Icon baseEntityIcon = new ImageIcon(getmyImage("keyone.gif"));
        baseEntity = new JButton(baseEntityIcon);
        baseEntity.setToolTipText("base entity");
        baseEntity.setActionCommand("BaseEntity");
        baseEntity.addActionListener(this);
        Icon functionCBIcon = new ImageIcon(getmyImage("pencil.gif"));
        JButton functionCB = new JButton(functionCBIcon);
        functionCB.setActionCommand("Function");
        functionCB.setToolTipText("function");
        functionCB.addActionListener(this);
        // TOOLBAR at BOTTOM
        JButton pointCB = new JButton(new ImageIcon(getmyImage("hand2.gif")));
        pointCB.setToolTipText("select");
        pointCB.setActionCommand("Pointer");
        pointCB.addActionListener(this);
        // Icon destroyCBIcon = new ImageIcon("crossbon.gif");
        JButton destroyCB = new JButton(new ImageIcon(
                        getmyImage("crossbon.gif")));
        destroyCB.setToolTipText("delete");
        destroyCB.setActionCommand("Destroy");
        // destroyCB.addActionListener(this);
        destroyCB.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                        System.out.println("calling: HELP");
                        PopupWindow p = new PopupWindow(new Frame(), drawPad,
                                        "Not Implemented");
                }
        });
        JButton cutCB = new JButton(new ImageIcon(getmyImage("cut.gif")));
        cutCB.setToolTipText("cut");
        cutCB.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                        System.out.println("calling: HELP");
                        PopupWindow p = new PopupWindow(new Frame(), drawPad,
                                        "Not Implemented");
                }
        });
        JButton pasteCB = new JButton(new ImageIcon(getmyImage("paste.gif")));
        pasteCB.setToolTipText("paste");
        pasteCB.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                        System.out.println("calling: HELP");
                        PopupWindow p = new PopupWindow(new Frame(), drawPad,
                                        "Not Implemented");
                }
        });
        JButton colourCB = new JButton(new ImageIcon(getmyImage("paint.gif")));
        colourCB.setToolTipText("colours");
        colourCB.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                        System.out.println("calling: HELP");
                        PopupWindow p = new PopupWindow(new Frame(), drawPad,
                                        "Not Implemented");
                }
        });
        // JButton fontCB = new JButton(new ImageIcon(
        // getmyImage("typewriter.gif") ));
        // fontCB.setToolTipText("fonts");
        JToolBar toolBar = new JToolBar();
        int VERTICAL = 1;
        toolBar.setFloatable(false);
        toolBar.setOrientation(VERTICAL);
        toolBar.add(abstractEntity);
        toolBar.add(baseEntity);
        toolBar.add(functionCB);
        toolBar.add(Box.createRigidArea(new Dimension(6, 0)));
        JToolBar toolBarBottom = new JToolBar();
        toolBarBottom.setFloatable(false);
        toolBarBottom.setOrientation(0);
        toolBarBottom.add(Box.createRigidArea(new Dimension(49, 0)));
        toolBarBottom.add(pointCB);
        toolBarBottom.add(destroyCB);
        toolBarBottom.add(cutCB);
        toolBarBottom.add(pasteCB);
        toolBarBottom.add(colourCB);
```

```
                content.add(toolBar, BorderLayout.WEST);
                content.add(toolBarBottom, BorderLayout.SOUTH);
                // <GraphicPane>
                // Create the graphic pane
                drawPad = new SchemaScreen();
                JScrollPane viewingPanel = new JScrollPane(drawPad);
                viewingPanel
                        .setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
                viewingPanel
                        .setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
                content.add(viewingPanel);
                drawPad.setCursor(new Cursor(Cursor.DEFAULT_CURSOR)); // pg541: Just
                // Java
                content.add(viewingPanel);
                // </GraphicPane>
        }
        // Which buttons have been pressed.....
        public void actionPerformed(ActionEvent e) {
                System.out.println("GISMoE_Applet: Button being pressed: "
                                + e.getActionCommand());
                if (e.getActionCommand() == "Pointer") { //
                        System.out.println("GISMoE_Applet: confirmed Pointer=");
                        drawPad.setButtonStatus(POINTER);
                        drawPad.setCursor(new Cursor(Cursor.HAND_CURSOR));

                } else if (e.getActionCommand() == "Destroy") { //
                        System.out.println("GISMoE_Applet: confirmed Destroy=");
                        drawPad.setButtonStatus(DESTROY);
                        drawPad.setCursor(new Cursor(Cursor.HAND_CURSOR));

                } else if (e.getActionCommand() == "AbstractEntity") { // abstractEntity
                        System.out.println("GISMoE_Applet: confirmed abstract entity");
                        drawPad.setButtonStatus(ABSTRACT_ENTITY);
                        drawPad.setCursor(new Cursor(Cursor.MOVE_CURSOR));

                } else if (e.getActionCommand() == "BaseEntity") { // baseEntity
                        System.out.println("GISMoE_Applet: BaseEntity");
                        drawPad.setButtonStatus(BASE_ENTITY);
                        drawPad.setCursor(new Cursor(Cursor.CROSSHAIR_CURSOR));

                } else if (e.getActionCommand() == "Function") { // baseEntity
                        System.out.println("GISMoE_Applet: FUNCTION");
                        drawPad.setButtonStatus(FUNCTION);
                        drawPad.setCursor(new Cursor(Cursor.W_RESIZE_CURSOR));
                }
        }
        // Normal getImage can not be used for Applets
        private Image getmyImage(String filename) {
                if (isApp_) { // Application
                        return Toolkit.getDefaultToolkit().getImage(filename);

                } else { // Applet
                        return getImage(getDocumentBase(), filename);
                }

        }
        public static void main(String[] args) {
                JFrame f = new JFrame("GISMoE_Applet v1.0");
                GISMoE_Applet gismoe = new GISMoE_Applet();
                gismoe.isApp_ = true;
                gismoe.init();
                gismoe.start();
                f.addWindowListener(new WindowAdapter() {
                        public void windowClosing(WindowEvent we) {
                                System.exit(0);
                        }
                });
                f.setSize(400, 320); // NOT SURE WHY?
                f.getContentPane().add("Center", gismoe);
                f.getContentPane().setLocation(500, 200);
                f.getContentPane().validate();
                f.setVisible(true);

        }
}
```

```
/*
        Nodes.java
*/
import java.awt.*;
import java.awt.event.*;
import java.awt.font.*;
import java.awt.geom.*;
import javax.swing.*;

public abstract class Nodes {
        /*
         * int no_of_fns; Functions (number of edges in conected in node) vector
         * FunctionList index // points to where funtion can be found
         */
        Color entityColour; // Colour of node
        PropertiesWindow popup;
        int type = 0; // Node type 1=base:oval, 2=abstract:circle
        public static int BASE = 1;
        public static int ABSTRACT = 2;
        public boolean isNodeSelected = false; // indicates if node selected
        public boolean nodecreated = true;
        String label; // Node Label
        String dataType; // base type int, string, float, etc
        int FontSize = 11;
        int no_of_fns = 0; // Number of functions node is connect too
        static int MAX_FUNCTIONS = 256;
        int listofFunctions[] = new int[MAX_FUNCTIONS];
        int x, y, width, height, radius; // Position and size
        int oldx, oldy; // Previous x, y before any operations
        int x1, y1, x2, y2, offset;
        final static BasicStroke stroke = new BasicStroke(2.0f);
        final static BasicStroke wideStroke = new BasicStroke(8.0f);
        public int getRed() {
                return entityColour.getRed();
        }
        public int getGreen() {
                return entityColour.getGreen();
        }
        public int getBlue() {
                return entityColour.getBlue();
        }
        public void setColour(Color c) {
                entityColour = c;
        }
        public void setLabel(String s) {
                System.out.println("label is: " + s + ":");
                label = s;
        }
        public void setFunction(int i) {
                System.out.println("Function index is : " + i + ":");
                listofFunctions[no_of_fns] = i;
                no_of_fns++;
        }
        public int getfunctionsTotal() {
                return no_of_fns;
        }
        public int getfunction(int f) {
                return listofFunctions[f];
        }
        public void alterFunctionPosition() {
                int newx = getX();
                int newy = getY();
        }
        public void setType(String s) {
                System.out.println("Type is: " + s);
                dataType = s;
        }
        public String getType() {
                return dataType;
        }
        public void setCoor(int x1, int y1) {
                x = x1;
                y = y1;
        }
        public String getLabel() {
                return label;
        }
        public int getOffset() {
                return offset;
        }
        public int getHeight() {
                return height;
        }
        public int getWidth() {
                return width;
        }
        public int getRadius() {
                return radius;
        }
        public int getX() {
                return x;
        }
        public int getY() {
                return y;
        }
        public int getNodeType() {
                return type;
        }
        public String getNodeTypeName() {
                if (type == BASE)
                        return "base";
```

```
                return "abstract";
        }
        public boolean wasNodeCreated() {
                return nodecreated;
        }
        public boolean foundNode(int cx, int cy) {
                System.out.println("Nodes  cx=" + cx + " cy=" + cy);
                return false;
        }
        public void paint(Graphics2D g) {
                if (type == BASE) {
                        // AbstactEntity node;
                        AbstactEntity node = (AbstactEntity) this;
                        node.paint(g);
                } else if (type == ABSTRACT) {
                        BaseEntity node = (BaseEntity) this;
                        node.paint(g);
                }
        }
        //
        // Node selected so highlight it
        //
        public void highlight(boolean high, Graphics2D g) {
                if (high) { // Create Highlight
                        isNodeSelected = true;
                } else { // Remove Highlight
                        isNodeSelected = false;
                }
        }
        public void moveit(int cx, int cy, Graphics2D g) {
        }
        public void unpaint(Graphics2D g) {
        }
} // End of default Class Nodes

//
// Start of AbstactEntity Class
//
class AbstactEntity extends Nodes {
        AbstactEntity(String l, int posx, int posy, int r, int w, int h, int o,
                        int red, int green, int blue) {
                label = l;
                x = posx;
                y = posy;
                radius = r;
                width = w;
                height = h;
                offset = o;
                type = ABSTRACT;
                entityColour = new Color(red, green, blue);
        }
        AbstactEntity(int posx, int posy) {
                popup = new PropertiesWindow(new Frame(), this);
                if (popup.getOKorCANCEL()) {
                        x = posx;
                        y = posy;
                        radius = 40; // was 50
                        width = 20;
                        height = 20;
                        offset = 3; // was 4
                        type = ABSTRACT;
                } else
                        nodecreated = false;
        }
        //
        // AbstactEntity
        //
        public boolean foundNode(int cx, int cy) {
                System.out.println("AbstactEntity: Current pointer is cx=" + cx
                                + " cy=" + cy);
                if ((cx >= (x - width) && cx <= (x + width))
                                && (cy >= (y - height) && cy <= (y + height))) {
                        return true;
                }
                ;
                return false;
        }
        //
        // Move AbstactEntity
        //
        public void moveit(int cx, int cy, Graphics2D g) {

                g.setStroke(wideStroke);
                g.setColor(Color.white);
                g.drawOval(oldx - width + offset, oldy - height + offset, radius,
                                radius);
                // g.drawOval(oldx-(width/2), oldy-(height/2), width, height);
                oldx = cx;
                oldy = cy;
                alterFunctionPosition();
                g.setStroke(stroke);
                g.setColor(Color.green);
                g.drawOval(oldx - width + offset, oldy - height + offset, radius,
                                radius);
                // g.drawOval(oldx-(width/2), oldy-(height/2), width, height);
        }
        //
        // AbstactEntity
        //
        public void paint(Graphics2D g) {
                float i;
```

```
                // shadow
                System.out.println("+++in Abstract paint");
                g.setColor(Color.gray);
                g.fillOval(x - width + offset, y - height + offset, radius, radius);
                g.setColor(entityColour);
                g.fillOval(x - width, y - height, radius, radius);

                g.setColor(Color.black);
                Font font = new Font("Times New Roman", Font.BOLD, FontSize);
                g.setFont(font);
                FontRenderContext frc = g.getFontRenderContext();
                LineMetrics metrics = font.getLineMetrics(label, frc);
                float messageWidth = (float) font.getStringBounds(label, frc)
                                .getWidth();
                float ascent = metrics.getAscent();
                float descent = metrics.getDescent();
                float cx = (width + messageWidth) / 2;
                float cy = (ascent + descent) / 4;
                g.drawString(label, x + (width / 2) - cx, y + cy);
        }
} // End of AbstactEntity
//
// Start of BaseEntity Class
//
class BaseEntity extends Nodes {

        BaseEntity(String l, int posx, int posy, int r, int w, int h, int o,
                        int red, int green, int blue, String t) {
                label = l;
                x = posx;
                y = posy;
                radius = r;
                width = w;
                height = h;
                offset = o;
                type = BASE;
                entityColour = new Color(red, green, blue);
                dataType = t;
        }
        BaseEntity(int posx, int posy) {
                label = "string";
                popup = new PropertiesWindow(new Frame(), this);

                if (popup.getOKorCANCEL()) {
                        x = posx;
                        y = posy;
                        radius = 40; // was 50
                        width = radius;
                        height = (radius / 2);
                        offset = 3; // was 4
                        type = BASE;
                } else
                        nodecreated = false;
        }
        //
        // BaseEntity
        //
        public boolean foundNode(int cx, int cy) {
                int x1 = x - (width / 2); // 25
                int y1 = y - (height / 2); // 12
                System.out.println("BaseEntity Current pointer is cx=" + cx + " cy="
                                + cy);
                if ((cx >= x1 && cx <= (x1 + width))
                                && (cy >= y1 && cy <= (y1 + height))) {
                        return true;
                }
                return false;
        }
        //
        // Move BaseEntity
        //
        public void moveit(int cx, int cy, Graphics2D g) {
                g.setStroke(wideStroke);
                g.setColor(Color.white);
                g.drawOval(oldx - (width / 2), oldy - (height / 2), width, height);
                oldx = cx;
                oldy = cy;
                alterFunctionPosition();
                g.setStroke(stroke);
                g.setColor(Color.green);
                g.drawOval(oldx - (width / 2), oldy - (height / 2), width, height);
        }
        //
        // BaseEntity
        //
        public void unpaint(Graphics2D g) {
                float i;

                g.setStroke(wideStroke);
                g.setColor(Color.white);
                // g.fillOval(x-(width/2)+offset, y-(height/2)+offset, width, height);
                g.fillOval(x - (width / 2), y - (height / 2), 2 * width, height * 2);
        }
        //
        // BaseEntity selected so high-light it
        //
        public void highlight(boolean high, Graphics2D g) {
                if (high) { // Create Highlight
                        isNodeSelected = true;
                        // g.setColor(Color.green);
                        // g.drawOval(x-(width/2), y-(height/2), width, height);
```

```
                        paint(g);
                } else { // Remove Highlight
                        isNodeSelected = false;
                        paint(g);
                }
                // this.repaint();
        }
        //
        // BaseEntity
        //
        public void paint(Graphics2D g) {
                float i;
                System.out.println("+++in Base paint");
                g.setColor(Color.gray);
                g.fillOval(x - (width / 2) + offset, y - (height / 2) + offset, width,
                                height);
                if (isNodeSelected) {
                        g.setColor(Color.black);
                } else {
                        g.setColor(entityColour);
                }
                g.fillOval(x - (width / 2), y - (height / 2), width, height);
                int x1 = x - (width / 2);
                int y1 = y - (height / 2);
                if (isNodeSelected) {
                        g.setColor(Color.green);
                } else {
                        g.setColor(Color.black);
                }
                Font font = new Font("Times New Roman", Font.BOLD, FontSize);
                g.setFont(font);
                FontRenderContext frc = g.getFontRenderContext();
                LineMetrics metrics = font.getLineMetrics(label, frc);
                float messageWidth = (float) font.getStringBounds(label, frc)
                                .getWidth();
                float ascent = metrics.getAscent();
                float descent = metrics.getDescent();
                float cx = (width + messageWidth) / 2;
                float cy = (ascent + descent) / 4;
                g.drawString(label, x + (width / 2) - cx, y + cy);
        }
} // End of BaseEntity
```

```
/*
        SchemaScreen.java
*/
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.util.*;
import java.awt.geom.*;

class SchemaScreen extends JComponent implements MouseListener,
                MouseMotionListener {
        Image image;
        Graphics2D graphics2D;
        int currentX, currentY;
        public int buttonStatus;
        Vector<Nodes> nodelist = new Vector<Nodes>(50);
        Vector<Functions> functionlist = new Vector<Functions>(50);
        public static int NOSTATUS = 0;
        public static int BASE_ENTITY = 1;
        public static int ABSTRACT_ENTITY = 2;
        public static int POINTER = 3;
        public static int FUNCTION = 4;
        public static int F_FIRST_NODE = 5;
        public static int F_SECOND_NODE = 6;
        public static int DESTROY = 7;
        public static int SPLINE = 8;
        public static int SPLINE_2_POINT = 9;
        public static int SPLINE_3_POINT = 10;
        public static int MOVESPLINE = 11;
        Point2D.Double startPT, endPT, controlPT;
        public static int BASE = 1;
        public static int ABSTRACT = 2;
        private boolean MOVING = false;
        Nodes first_node, second_node;
        Functions splinefn;
        public SchemaScreen() {
                setDoubleBuffered(false);
                addMouseListener(this);
                addMouseMotionListener(this);
        }
        //
        // MouseMotionEvent (MouseMotionListener)
        //
        public void mouseMoved(MouseEvent event) {
        }

        public void mouseDragged(MouseEvent event) {
                System.out.println("+In mouseDragged");
                // current_x=event.getX();
                // current_y=event.getY();

                if (buttonStatus == POINTER && first_node != null) {
                        System.out.println("+Moving.....");
                        currentX = event.getX(); // returns the X, Y position of mouse
                        currentY = event.getY();
                        reDrawAll();
                        first_node.moveit(currentX, currentY, graphics2D);
                        repaint();
                        MOVING = true;

                } else if (buttonStatus == MOVESPLINE) {
                        System.out.println("+Moving....Spline.");
                        currentX = event.getX(); // returns the X, Y position of mouse
                        currentY = event.getY();
                        controlPT = new Point2D.Double((double) currentX, (double) currentY);
                        splinefn.SetSplineC(controlPT);
                        clear();
                        reDrawAll();
                        repaint();
                        MOVING = true;
                }
        }
        //
        // MouseEvent for buttons (MouseListener)
        //
        public void mouseClicked(MouseEvent event) {
        }
        public void mouseEntered(MouseEvent event) {
        }
        public void mouseExited(MouseEvent event) {
        }
        public void mouseReleased(MouseEvent event) {
                System.out.println("+In mouseReleased");
                Functions f;
                if (buttonStatus == POINTER && first_node != null && MOVING) {
                        currentX = event.getX(); // returns the X, Y position of mouse
                        currentY = event.getY();
                        MOVING = false;
                        first_node.highlight((boolean) false, graphics2D);
                        first_node.setCoor(currentX, currentY);
                        for (int i = 0; i < first_node.getfunctionsTotal(); i++) {
                                System.out.println("NO OF FUNCTIONS ==== " + i);
                                f = (Functions) functionlist.elementAt(first_node
                                                .getfunction(i));
                                f.setCoor(currentX, currentY, first_node);
                        }
                        clear();
                        reDrawAll();
                        repaint();
                }
        }
```

```
//
// Check for button press when in draw area
//
public void mousePressed(MouseEvent event) {

        currentX = event.getX(); // returns the X, Y position of mouse
        currentY = event.getY();
        System.out.println("Button is " + buttonStatus);

        if (buttonStatus == ABSTRACT_ENTITY) {
                AbstactEntity node1 = new AbstactEntity(currentX, currentY);
                if (node1.wasNodeCreated()) {
                        node1.paint(graphics2D);
                        nodelist.addElement(node1);
                        System.out.println("Total nodes=" + nodelist.size());
                }
        } else if (buttonStatus == BASE_ENTITY) {
                BaseEntity node2 = new BaseEntity(currentX, currentY);
                if (node2.wasNodeCreated()) {
                        node2.paint(graphics2D);
                        nodelist.addElement(node2);
                        System.out.println("Total nodes=" + nodelist.size());
                }
        } else if (buttonStatus == POINTER) {
                // Search nodes for hit!!!
                System.out.println("buttonStatus = POINTER");
                if (first_node != null)
                        first_node.highlight((boolean) false, graphics2D);
                first_node = searchNodes(currentX, currentY);
                if (first_node != null) {
                        first_node.highlight((boolean) true, graphics2D);
                        System.out.println("Node Found");
                }
                //
                // If creating a Function locate FIRST node
                //
        } else if (buttonStatus == FUNCTION) {
                // Create function (line)
                System.out.println("buttonStatus = FUNCTION");
                first_node = searchNodes(currentX, currentY);
                if (first_node != null) {
                        System.out.println("Node Found");
                        buttonStatus = F_SECOND_NODE;
                }
                //
                // If creating a SPLINE 1st point
                //
        } else if (buttonStatus == SPLINE) {
                // Create function (line)
                System.out.println("buttonStatus = SPLINE ");
                // first_node = searchNodes(currentX, currentY);
                // if ( first_node != null ) {
                startPT = new Point2D.Double((double) currentX, (double) currentY);
                // startPT.setLocation( (double) 0.0, (double) 0.0);
                buttonStatus = SPLINE_2_POINT;
                System.out.println("FIRST_SPLINE_POINT");
                //
                // in DESTROY OBJECT
                //
        } else if (buttonStatus == DESTROY) {
                //
                System.out.println("buttonStatus = DESTROY");
                first_node = searchNodes(currentX, currentY);
                if (first_node != null) {
                        System.out.println("Node Found to delete");
                        first_node.unpaint(graphics2D);
                        nodelist.remove(first_node);
                        reDrawAll();
                }
                //
                // in Function mode and already selected FIRST node
                //
        } else if (buttonStatus == F_SECOND_NODE) {
                System.out.println("buttonStatus = F_SECOND_NODE");
                second_node = searchNodes(currentX, currentY);
                if (second_node != null) {
                        System.out.println("Node Found");
                        buttonStatus = FUNCTION;
                        int x1 = first_node.getX();
                        int y1 = first_node.getY();
                        int x2 = second_node.getX();
                        int y2 = second_node.getY();
                        Functions function = new Functions(x1, y1, x2, y2, first_node,
                                        second_node);

                        functionlist.addElement(function);
                        int findex = functionlist.size() - 1;
                        first_node.setFunction(findex);
                        second_node.setFunction(findex);
                        // function.paint(graphics2D);
                        reDrawAll();
                        first_node = null;
                        second_node = null;
                }
                //
                // If creating a SPLINE
                //
        } else if (buttonStatus == SPLINE_2_POINT) {
                // Create function (line)
                System.out.println("buttonStatus = 2SPLINE");
                // first_node = searchNodes(currentX, currentY);
```

```
                        // if ( first_node != null ) {
                        controlPT = new Point2D.Double((double) currentX, (double) currentY);
                        buttonStatus = SPLINE_3_POINT;
                        System.out.println("2_SPLINE_POINT");
                        //
                        // If creating a SPLINE
                        //
                } else if (buttonStatus == SPLINE_3_POINT) {
                        // Create function (line)
                        System.out.println("buttonStatus = 3SPLINE");
                        // first_node = searchNodes(currentX, currentY);
                        // if ( first_node != null ) {
                        System.out.println("3_SPLINE_POINT");
                        endPT = new Point2D.Double((double) currentX, (double) currentY);
                        Functions function = new Functions(startPT, endPT, controlPT);
                        splinefn = function;
                        functionlist.addElement(function);
                        int findex = functionlist.size() - 1;
                        buttonStatus = NOSTATUS;
                        reDrawAll();
                        // End of IF for action
                }
                // node center
                // graphics2D.setColor(Color.blue);
                // graphics2D.fillOval(currentX - 2,currentY - 2,2,2);
                // System.out.println("x="+currentX+","+currentY);
                repaint();

        }
        public void paintComponent(Graphics g) {
                if (image == null) {
                        image = createImage(600, 600);
                        graphics2D = (Graphics2D) image.getGraphics();
                        graphics2D.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
                                        RenderingHints.VALUE_ANTIALIAS_ON);
                        clear();
                }
                g.drawImage(image, 0, 0, null);
        }
        public void clear() {
                graphics2D.setPaint(Color.white);
                graphics2D.fillRect(0, 0, 600, 600);
                graphics2D.setPaint(Color.black);
                repaint();
        }
        public void setButtonStatus(int b) {
                buttonStatus = b;
        }
        //
        // Redraw all objects on the screen:
        // functions first and then nodes
        //
        public void reDrawAll() {
                Nodes n1;
                Functions f;
                System.out.println("in reDrawAll");

                for (int i = 0; i < functionlist.size(); i++) {
                        f = (Functions) functionlist.elementAt(i);
                        f.paint(graphics2D);
                        // System.out.println("Nodes="+i);
                }

                for (int i = 0; i < nodelist.size(); i++) {
                        n1 = (Nodes) nodelist.elementAt(i);
                        n1.paint(graphics2D);
                        // System.out.println("Function="+i);
                }
        }
        //
        // Given x,y find out if this is within an object
        // Nodes, BUT NOT Functions YET
        //
        public Nodes searchNodes(int x, int y) {
                // nodelist.addElement(node2);
                Nodes n, status = null;
                System.out.println("in searchNodes with " + nodelist.size() + " nodes");
                int i = 0;
                while (i < nodelist.size() && status == null) {
                        n = (Nodes) nodelist.elementAt(i);
                        int nx = n.getX();
                        int ny = n.getY();
                        System.out.println("trying Node=" + i + "; coors are x= " + nx
                                        + "," + ny);
                        if (n.foundNode(x, y)) {
                                status = n;
                                System.out.println("Node=" + i + " : HIT");
                        }
                        i++;
                }
                return status;
        }
        public String displayNodesLabels() {
                Nodes n, status = null, first_node, second_node;
                Functions f;
                String SchemaData, label;
                System.out.println("in displayNodesLabels with " + nodelist.size()+ " nodes");
                SchemaData = "";
                int i = 0;
                while (i < nodelist.size() && status == null) {
                        n = (Nodes) nodelist.elementAt(i);
```

```
                        label = n.getLabel();
                        if (n.getNodeTypeName() == "abstract") {
                                label = label + " :: " + "abstract;\n";
                        } else if (n.getNodeTypeName() == "base") {
                                label = label + " :: " + n.getType() + ";\n";
                        }
                        // System.out.println("trying Node="+i+", label=" + label);
                        SchemaData = SchemaData + label;
                        i++;
                }
                for (i = 0; i < functionlist.size(); i++) {
                        f = (Functions) functionlist.elementAt(i);
                        label = f.getLabel();
                        first_node = f.get1stNode();
                        second_node = f.get2ndNode();
                        label = label + ": " + first_node.getLabel() + " -> "
                                        + second_node.getLabel() + ";\n";
                        System.out.println("trying Function=" + i + ", label=" + label);
                        SchemaData = SchemaData + label;
                }
                return SchemaData;
        }
        public String displayDTD() {
                Nodes n, status = null, first_node, second_node;
                Functions f;
                String SchemaData, label = "";
                System.out.println("in displayDTD with " + nodelist.size() + " nodes");
                SchemaData = "<?xml version=\"1.0\"?>\n   <!DOCTYPE schema [\n   <!ELEMENT schema (";
                for (int i = 0; i < nodelist.size(); i++) {
                        n = (Nodes) nodelist.elementAt(i);
                        if (n.getNodeTypeName() == "abstract") {
                                if (label.length() == 0) {
                                        label = n.getLabel() + "*";
                                } else {
                                        label = label + ", " + n.getLabel() + "*";
                                }
                        }
                }
                SchemaData = SchemaData + label + ")>\n";
                System.out.println(":" + SchemaData);

                for (int i = 0; i < nodelist.size(); i++) {
                        n = (Nodes) nodelist.elementAt(i);
                        // System.out.println(n.getNodeTypeName() +
                        // ", label="+n.getLabel());
                        if (n.getNodeTypeName() == "abstract") {
                                label = "   <!ELEMENT " + n.getLabel() + " (";
                                // Node is connected to other nodes
                                for (int j = 0; j < n.getfunctionsTotal(); j++) {
                                        // Get Function
                                        f = (Functions) functionlist.elementAt(n.getfunction(j));
                                        // Get name of node which is connected
                                        second_node = f.get2ndNode();
                                        // Don't print yourself, so go onto next node
                                        if (second_node != n) {
                                                if (j > 0)
                                                        label = label + ", ";
                                                label = label + second_node.getLabel();
                                                if (second_node.getNodeTypeName() == "abstract")
                                                        label = label + "*";
                                        }
                                }
                                label = label + ")>\n";
                                SchemaData = SchemaData + label;
                        } else {
                                label = "   <!ELEMENT " + n.getLabel() + " (#PCDATA)>\n";
                                SchemaData = SchemaData + label;
                        }
                }
                SchemaData = SchemaData + ")>\n";
                // for (int i = 0; i < functionlist.size(); i++) {
                // f = (Functions) functionlist.elementAt(i);
                // label = f.getLabel();
                // first_node = f.get1stNode();
                // second_node = f.get2ndNode();
                // label = label +
                // ": "+first_node.getLabel()+" -> "+second_node.getLabel()+";\n";
                // System.out.println("trying Function="+i+", label=" + label);
                // SchemaData = SchemaData + label;
                // }
                return SchemaData;
        }
        public String saveData() {
                Nodes n, status = null;
                String data, label;
                System.out.println("in saveData with " + nodelist.size() + " nodes");
                data = "";
                int i = 0;
                while (i < nodelist.size() && status == null) {
                        n = (Nodes) nodelist.elementAt(i);
                        // System.out.print(n.getNodeTypeName() + ",");
                        data = data + n.getNodeTypeName();

                        System.out.print(n.getLabel() + ",");
                        data = data + "," + n.getLabel();
                        // System.out.print(n.getX() + ",");
                        data = data + "," + n.getX();
                        // System.out.print(n.getY() + ",");
                        data = data + "," + n.getY();
                        System.out.print(n.getRadius() + ",");
                        data = data + "," + n.getRadius();
```

```
                        System.out.print(n.getWidth() + ",");
                        data = data + "," + n.getWidth();
                        System.out.print(n.getHeight() + ",");
                        data = data + "," + n.getHeight();
                        // System.out.print(n.getOffset() + ",");
                        data = data + "," + n.getOffset();

                        // System.out.print(n.getRed() + ",");
                        data = data + "," + n.getRed();
                        // System.out.print(n.getGreen() + ",");
                        data = data + "," + n.getGreen();
                        // System.out.print(n.getBlue() + ",");
                        data = data + "," + n.getBlue();

                        if (n.getNodeTypeName() == "abstract") {
                                // System.out.print("blank,"); // int, string , date, bool
                                data = data + ",blank";
                        } else if (n.getNodeTypeName() == "base") {
                                // System.out.print(n.getType() + ","); // int, string , date,
                                // bool
                                data = data + "," + n.getType();
                        }
                        // System.out.println();
                        data = data + "\n";
                        i++;
                } // end Abstract and Base data
                System.out.println("\nin saveData with " + functionlist.size()
                                + " functions");
                for (i = 0; i < functionlist.size(); i++) {
                        Functions f = (Functions) functionlist.elementAt(i);
                        data = data + "function";
                        data = data + "," + f.getLabel();
                        System.out.println("label(" + i + "=" + f.getLabel());
                        data = data + "," + f.getX1();
                        // System.out.println(data);
                        data = data + "," + f.getY1();
                        // System.out.println(data);
                        data = data + "," + f.getX2();
                        // System.out.println(data);
                        data = data + "," + f.getY2();
                        // System.out.println(data);

                        data = data + "," + f.getRed();
                        // System.out.println(data);
                        data = data + "," + f.getGreen();
                        // System.out.println(data);
                        data = data + "," + f.getBlue();
                        data = data + "\n";
                } // end functionlist
                System.out.println(data);
                return data;
        }
        public void loadData(String SchemaData) {
                Nodes n, status = null;
                String label;
                int element = 1, i = 0;
                String SchemaDataNew = SchemaData.replace("\n", ",");
                System.out.println("\nin loadData\n" + SchemaDataNew);
                String data[] = SchemaDataNew.split(",");
                System.out.println("  SchemaData.length=" + data.length);
                while (i < data.length) {
                        System.out.print("load:" + data[i] + ":i=" + i + "\n");
                        String l = data[i + 1];
                        System.out.println("label=" + l);
                        if (data[i].contains("abstract") || data[i].contains("base")) {
                                int posx = Integer.parseInt(data[i + 2]);
                                int posy = Integer.parseInt(data[i + 3]);
                                int r = Integer.parseInt(data[i + 4]);
                                int w = Integer.parseInt(data[i + 5]);
                                int h = Integer.parseInt(data[i + 6]);
                                int o = Integer.parseInt(data[i + 7]);
                                int red = Integer.parseInt(data[i + 8]);
                                int green = Integer.parseInt(data[i + 9]);
                                int blue = Integer.parseInt(data[i + 10]);
                                String t = data[i + 11];
                                if (data[i].contains("abstract")) {
                                        System.out.println("   in abstract");
                                        AbstactEntity node1 = new AbstactEntity(l, posx, posy, r,
                                                        w, h, o, red, green, blue);
                                        node1.paint(graphics2D);
                                        nodelist.addElement(node1);
                                        System.out.println("Total nodes=" + nodelist.size());
                                        i = i + 12;
                                } else {
                                        System.out.println("   in base");
                                        BaseEntity node2 = new BaseEntity(l, posx, posy, r, w, h,
                                                        o, red, green, blue, t);
                                        node2.paint(graphics2D);
                                        nodelist.addElement(node2);
                                        System.out.println("Total nodes=" + nodelist.size());
                                        System.out.println("Total nodes=end");
                                        i = i + 12;
                                }
                        } else {
                                System.out.println("   in function");
                                int x1 = Integer.parseInt(data[i + 2]);
                                System.out.println("x1=" + x1);
                                int y1 = Integer.parseInt(data[i + 3]);
                                System.out.println("y1=" + y1);
                                int x2 = Integer.parseInt(data[i + 4]);
                                System.out.println("x2=" + x2);
```

```
                                        int y2 = Integer.parseInt(data[i + 5]);
                                        System.out.println("y2=" + y2);
                                        int red = Integer.parseInt(data[i + 6]);
                                        System.out.println("red=" + red);
                                        int green = Integer.parseInt(data[i + 7]);
                                        System.out.println("green=" + green);
                                        int blue = Integer.parseInt(data[i + 8]);
                                        System.out.println("blue=" + blue);
                                        Nodes f_node, s_node;
                                        f_node = searchNodes(x1, y1);
                                        if (f_node != null) {
                                                s_node = searchNodes(x2, y2);
                                                if (s_node != null) {
                                                        System.out.println("   in function");
                                                        Functions function = new Functions(l, x1, y1, x2, y2,
                                                                        f_node, s_node);
                                                        function.setColour(red, green, blue);
                                                        functionlist.addElement(function);
                                                        int findex = functionlist.size() - 1;
                                                        f_node.setFunction(findex);
                                                        s_node.setFunction(findex);
                                                        // reDrawAll();
                                                        function.paint(graphics2D);
                                                        System.out.println("   in function");
                                                }
                                                i = i + 9;
                                        }
                                }
                        }
                        repaint();
                        reDrawAll();
                        System.out.println("   end");
                }
                public String displayDataDict(String displayWhat) {
                        Nodes n, status = null, first_node, second_node;
                        Functions f;
                        String SchemaData, label;
                        System.out.println("in displayDataDict with " + nodelist.size()
                                        + " nodes doing:" + displayWhat + ":");
                        SchemaData = "";
                        if (displayWhat == "XML elements") {
                                SchemaData = "XML tags\n=====\n";
                                for (int i = 0; i < nodelist.size(); i++) {
                                        n = (Nodes) nodelist.elementAt(i);
                                        label = n.getLabel();
                                        label = "<" + label + ">\n";
                                        System.out.print("trying Node=" + i + ", label=" + label);
                                        SchemaData = SchemaData + label;
                                }
                        } else if (displayWhat == "Functional Data Dictionary:Entities") {
                                SchemaData = "Entries (type)\n========\n";
                                for (int i = 0; i < nodelist.size(); i++) {
                                        n = (Nodes) nodelist.elementAt(i);
                                        label = n.getLabel();
                                        if (n.getNodeTypeName() == "abstract") {
                                                label = label + " (abstract)\n";
                                        } else if (n.getNodeTypeName() == "base") {
                                                label = label + " (" + n.getType() + ")\n";
                                        }
                                        System.out.print("trying Node=" + i + ", label=" + label);
                                        SchemaData = SchemaData + label;
                                }
                        } else if (displayWhat == "Functional Data Dictionary:Functions") {
                                SchemaData = "Functions\n=====\n";
                                for (int i = 0; i < nodelist.size(); i++) {
                                        n = (Nodes) nodelist.elementAt(i);
                                        int j = 0;
                                        label = "";
                                        while (j < functionlist.size()) {
                                                f = (Functions) functionlist.elementAt(j);
                                                if (n.getNodeTypeName() == "abstract") {
                                                        second_node = f.get2ndNode();
                                                        first_node = f.get1stNode();
                                                        if (second_node.getNodeTypeName() == "abstract"
                                                                        && (first_node.getNodeTypeName() ==
"abstract" && n == first_node)) {
                                                                label = f.getLabel();
                                                                j = functionlist.size();
                                                        }
                                                } else {
                                                        second_node = f.get2ndNode();
                                                        if (second_node == n) {
                                                                label = f.getLabel();
                                                                j = functionlist.size();
                                                        }
                                                }
                                                System.out.print("node = " + n.getLabel() + ", Function="
                                                                + f.getLabel() + ", 2nd node="
                                                                + second_node.getLabel() + "\n");
                                                j++;
                                        }if (label == "") {
                                                label = " ";
                                        }
                                        label = label + "\n";
                                        SchemaData = SchemaData + label;
                                }
                        }
                        return SchemaData;
                }

        }
```

```
/*
        PropertiesWindow.java
*/
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.applet.*;

public class PropertiesWindow extends Dialog {
        boolean buttonStatus;
        public PropertiesWindow(Frame parent, BaseEntity node) {
                super(parent);
                JRadioButton intRadio = new JRadioButton("integer");
                JRadioButton stringRadio = new JRadioButton("string", true);
                JRadioButton boolRadio = new JRadioButton("boolean");
                JRadioButton floatRadio = new JRadioButton("float");
                JRadioButton dateRadio = new JRadioButton("date");
                ButtonGroup types = new ButtonGroup();
                types.add(intRadio);
                types.add(stringRadio);
                types.add(boolRadio);
                types.add(floatRadio);
                types.add(dateRadio);
                JRadioButton plum = new JRadioButton("plum", true);
                JRadioButton turquoise = new JRadioButton("turquoise");
                JRadioButton gold = new JRadioButton("gold");
                ButtonGroup colour = new ButtonGroup();
                colour.add(plum);
                colour.add(turquoise);
                colour.add(gold);
                JTextField label = new JTextField();
                String message1 = "Type: ";
                String message2 = "Colour: ";
                String message3 = "Label: ";
                int result = JOptionPane.showOptionDialog(parent, new Object[] {
                                message1, intRadio, stringRadio, floatRadio, boolRadio,
                                dateRadio, message2, plum, turquoise, gold, message3, label },
                                "Specify Base Entity Properiies", JOptionPane.OK_CANCEL_OPTION,
                                JOptionPane.QUESTION_MESSAGE, null, null, null);
                if (result == JOptionPane.OK_OPTION) {
                        buttonStatus = true;
                        System.out.println("Data is: " + label.getText().length());
                        if (intRadio.isSelected()) {
                                node.setType("integer");
                                node.setLabel("integer");
                        } else if (stringRadio.isSelected()) {
                                node.setType("string");
                                node.setLabel("string");
                        } else if (boolRadio.isSelected()) {
                                node.setType("boolean");
                                node.setLabel("boolean");
                        } else if (floatRadio.isSelected()) {
                                node.setType("float");
                                node.setLabel("float");
                        } else if (dateRadio.isSelected()) {
                                node.setType("date");
                                node.setLabel("date");
                        }
                        if (label.getText().length() > 0)
                                node.setLabel(label.getText());

                        if (plum.isSelected()) {
                                System.out.println("plum");
                                node.setColour(new Color(220, 162, 220));
                        } else if (turquoise.isSelected()) {
                                System.out.println("turquoise");
                                node.setColour(new Color(68, 226, 212));
                        } else if (gold.isSelected()) {
                                System.out.println("gold");
                                node.setColour(new Color(252, 214, 4));
                        }
                } else {
                        buttonStatus = false;
                }

        }

        public PropertiesWindow(Frame parent, AbstactEntity node) {
                super(parent);
                JRadioButton plum = new JRadioButton("plum", true);
                JRadioButton turquoise = new JRadioButton("turquoise");
                JRadioButton gold = new JRadioButton("gold");
                ButtonGroup colour = new ButtonGroup();
                colour.add(plum);
                colour.add(turquoise);
                colour.add(gold);
                JTextField label = new JTextField();
                String message1 = "Enter colour: ";
                String message2 = "Label: ";
                String message3 = "Please enter your username and password.";
                int result = JOptionPane.showOptionDialog(parent, new Object[] {
                                message1, plum, turquoise, gold, message2, label },
                                "Specify Abstract Entity Properiies",
                                JOptionPane.OK_CANCEL_OPTION, JOptionPane.QUESTION_MESSAGE,
                                null, null, null);
                if (result == JOptionPane.OK_OPTION) {
                        buttonStatus = true;
                        System.out.println("Data is: " + label.getText());
                        if (plum.isSelected()) {
                                System.out.println("plum");
                                node.setColour(new Color(220, 162, 220));
```

```
                        } else if (turquoise.isSelected()) {
                                System.out.println("turquoise");
                                node.setColour(new Color(68, 226, 212));
                        } else if (gold.isSelected()) {
                                System.out.println("gold");
                                node.setColour(new Color(252, 214, 4));
                        }
                        node.setLabel(label.getText());
                } else {
                        buttonStatus = false;
                }
        }
        public PropertiesWindow(Frame parent, Functions node) {
                super(parent);
                JRadioButton plum = new JRadioButton("plum", true);
                JRadioButton turquoise = new JRadioButton("turquoise");
                JRadioButton gold = new JRadioButton("gold");
                ButtonGroup colour = new ButtonGroup();
                colour.add(plum);
                colour.add(turquoise);
                colour.add(gold);
                JTextField label = new JTextField();
                String message1 = "Enter colour: ";
                String message2 = "Label: ";
                int result = JOptionPane.showOptionDialog(parent, new Object[] {
                                message1, plum, turquoise, gold, message2, label },
                                "Specify Function Properiies", JOptionPane.OK_CANCEL_OPTION,
                                JOptionPane.QUESTION_MESSAGE, null, null, null);
                if (result == JOptionPane.OK_OPTION) {
                        buttonStatus = true;
                        System.out.println("Data is: " + label.getText());
                        if (plum.isSelected()) {
                                System.out.println("plum");
                                node.setColour(new Color(220, 162, 220));
                        } else if (turquoise.isSelected()) {
                                System.out.println("turquoise");
                                node.setColour(new Color(68, 226, 212));
                        } else if (gold.isSelected()) {
                                System.out.println("gold");
                                node.setColour(new Color(252, 214, 4));
                        }
                        node.setLabel(label.getText());
                } else {
                        buttonStatus = false;
                }
        }

        public boolean getOKorCANCEL() {
                return buttonStatus;
        }

}
```

```
/*
        Functions.java
*/
import java.awt.*;
import java.awt.event.*;
import java.awt.font.*;
import javax.swing.*;
import java.awt.geom.*;

public class Functions {
        int x1, x2, y1, y2;
        String label;
        Color functionColour;
        public boolean nodecreated = true;
        public int spline = 0; // if spline = 1;
        Point2D.Double startPT, endPT, controlPT;
        int FontSize = 10;
        Nodes first, // function leaves from
                        second; // function points to

        /*
         * name label node reference_start node reference_finish type (one, many)
         *
         * Boolean highlighted // set to True if function is currently selected x,y,
         * x1, x2 // Line Colour
         */
        // Function
        public Functions(Point2D.Double s, Point2D.Double e, Point2D.Double c) {
                PropertiesWindow popup = new PropertiesWindow(new Frame(), this);
                if (popup.getOKorCANCEL()) {
                        spline = 1;
                        startPT = s;
                        endPT = e;
                        controlPT = c;
                        // System.out.println("x1="+xx1+"; y1="+yy1+"; x2="+xx2+"; y2="+yy2);
                } else {
                        nodecreated = false;
                }
        }
        // Function
        public Functions(int xx1, int yy1, int xx2, int yy2) {
                PropertiesWindow popup = new PropertiesWindow(new Frame(), this);

                if (popup.getOKorCANCEL()) {
                        System.out.println("x1=" + xx1 + "; y1=" + yy1 + "; x2=" + xx2
                                        + "; y2=" + yy2);
                        x1 = xx1;
                        y1 = yy1;
                        x2 = xx2;
                        y2 = yy2;
                } else {
                        nodecreated = false;
                }
        }
        // Function
        public Functions(int xx1, int yy1, int xx2, int yy2, Nodes f, Nodes s) {
                PropertiesWindow popup = new PropertiesWindow(new Frame(), this);

                if (popup.getOKorCANCEL()) {
                        System.out.println("x1=" + xx1 + "; y1=" + yy1 + "; x2=" + xx2
                                        + "; y2=" + yy2);
                        x1 = xx1;
                        y1 = yy1;
                        x2 = xx2;
                        y2 = yy2;
                        first = f;
                        second = s;
                } else {
                        nodecreated = false;
                }
        }
        public Functions(String l, int xx1, int yy1, int xx2, int yy2, Nodes f,
                        Nodes s) {
                label = l;
                x1 = xx1;
                y1 = yy1;
                x2 = xx2;
                y2 = yy2;
                first = f;
                second = s;
        }
        public void paint(Graphics2D g) {
                float i;
                double theta = Math.toRadians(30); // arrowhead sharpness
                int size = 10; // arrowhead length
                double angle;
                QuadCurve2D.Double quad = new QuadCurve2D.Double();
                if (spline == 0) {
                        int cx = Math.round((x1 + x2) / 2);
                        int cy = Math.round((y1 + y2) / 2);
                        g.setColor(functionColour);
                        g.drawLine(x1, y1, x2, y2);
                        // calculate points for arrowhead
                        angle = Math.atan2(y2 - y1, x2 - x1) + Math.PI;
                        int x3 = (int) (cx + Math.cos(angle - theta) * size);
                        int y3 = (int) (cy + Math.sin(angle - theta) * size);
                        int x4 = (int) (cx + Math.cos(angle + theta) * size);
                        int y4 = (int) (cy + Math.sin(angle + theta) * size);
                        int x_vals[] = { cx, x3, x4 };
                        int y_vals[] = { cy, y3, y4 };
                        // draw arrowhead
```

```
                        g.fillPolygon(x_vals, y_vals, x_vals.length);
                        g.setColor(Color.black);
                        Font font = new Font("Times New Roman", Font.PLAIN, FontSize);
                        // g.drawString(label, cx, cy );
                        FontRenderContext frc = g.getFontRenderContext();
                        LineMetrics metrics = font.getLineMetrics(label, frc);
                        float messageWidth = (float) font.getStringBounds(label, frc)
                                        .getWidth();
                        float ascent = metrics.getAscent();
                        float descent = metrics.getDescent();
                        float c1x = (messageWidth) / 2;
                        float c1y = (ascent + descent) / 2;
                        g.drawString(label, cx - c1x, cy);
                } else {
                        System.out.println("piant SPLINE");
                        quad.setCurve(startPT, controlPT, endPT);
                        g.draw(quad);
                }
        }
        public Nodes get1stNode() {
                return first;
        }
        public Nodes get2ndNode() {
                return second;
        }
        public int getRed() {
                return functionColour.getRed();
        }
        public int getGreen() {
                return functionColour.getGreen();
        }
        public int getBlue() {
                return functionColour.getBlue();
        }
        public void setColour(int red, int green, int blue) {
                functionColour = new Color(red, green, blue);
        }
        public void setColour(Color c) {
                functionColour = c;
        }
        public void setCoor(int xx, int yy, Nodes f) {
                if (f == first) {
                        System.out.println("First....");
                        x1 = xx;
                        y1 = yy;
                }
                if (f == second) {
                        System.out.println("Second...");
                        x2 = xx;
                        y2 = yy;
                }
                if (f != second && f != first)
                        System.out.println("FAIL...");
        }
        public void setLabel(String s) {
                System.out.println("label is: " + s + ":");
                label = s;
        }
        public String getLabel() {
                return label;
        }
        public int getX1() {
                return x1;
        }
        public int getX2() {
                return x2;
        }
        public int getY1() {
                return y1;
        }
        public int getY2() {
                return y2;
        }
        public boolean wasNodeCreated() {
                return nodecreated;
        }
        // Function
        public void SetSplineC(Point2D.Double c) {
                controlPT = c;
        }

}
```

```
/*
        Database.java
*/
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.applet.*;
import java.io.*;
import java.net.*;

class Database {
    Socket sock;
    BufferedReader dis;
    PrintWriter dat;
    String input, serverHost;
    static final boolean DEBUG = false;         // DEBUG
    public boolean openDB() throws IOException {
         if (DEBUG) { System.out.println("Database: openDB( )"); }
        return (openDB("193.61.29.1", 4446));
    } // openDB
    boolean openDB( String hostIP, int port ) throws IOException {
         if (DEBUG) { System.out.println("Database: openDB( String hostIP, int port )"); }
        try {
                 // Open our connection to port 4446
                 sock = new Socket( hostIP, port);

                 // Get I/O streams from the socket
                 dis = new BufferedReader( new InputStreamReader(sock.getInputStream()) );
                 dat = new PrintWriter( sock.getOutputStream() );
         return true;
         } catch(Exception e) {
         return false;
         }
    } // openDB
    public void workDB( SchemaScreen drawPad ) throws IOException {
         if (DEBUG) { System.out.println("Database: WorkDB()");  }
         // Need error checking here, but a problem at this time
         // Let the server do the hard work
         dat.println( drawPad.saveData() );
         dat.flush();
    }
    public void closeDB() throws IOException {
         if (DEBUG) { System.out.println("Database: CloseDB()");  }
         sock.close();
    } // closeDB
}
```

```
/*
        PopupWindow.java
*/
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.applet.*;
import java.io.*;

public class PopupWindow extends Dialog {
        boolean buttonStatus;
        public PopupWindow(AppletContext f, Frame parent, SchemaScreen drawPad, String action) {
                super(parent);
                System.out.println("in: PopupWindowCreate DB");
                f.showStatus("Openning FDL database...");
                try {
                    Database DB = new Database();
                    if (DB.openDB()) {
                        // Do Database work
                        DB.workDB( drawPad );
                        DB.closeDB();
                        System.out.println("in: Open OK");
                        JOptionPane.showOptionDialog(parent,
                                        "FDL Database is open for work",
"FDL",JOptionPane.DEFAULT_OPTION,
                                        JOptionPane.WARNING_MESSAGE,null, null, null);
                    } else {
                        // Security problem with applets
                        System.out.println("in: OpenFail");
                        JOptionPane.showOptionDialog(parent,
                                        "FDL Database did not open ...", "FDL",
                                        JOptionPane.DEFAULT_OPTION, JOptionPane.WARNING_MESSAGE,
                                        null, null, null);
                    }
                } catch (IOException e) { ; }
                buttonStatus = true;
        }
        public PopupWindow(Frame parent, SchemaScreen drawPad, String action) {
                super(parent);
                System.out.println("in: PopupWindow " + action);
                if (action == "schema") {
                        System.out.println("in: PopupWindowSchema ");
                        JTextArea label = new JTextArea(drawPad.displayNodesLabels(), 20,40);
                        JOptionPane.showOptionDialog(parent, new Object[] { label },
                                        "Functional Schema", JOptionPane.DEFAULT_OPTION,
                                        JOptionPane.WARNING_MESSAGE, null, null, null);
                        buttonStatus = true;
                } else if (action == "Not Implemented") {
                        System.out.println("in: PopupWindowNot Implemented");
                        JTextArea label = new JTextArea("Not Implemented in Applet...", 2,10);
                        JOptionPane
                                        .showMessageDialog(parent, "Not Implemented in Applet...",
                                                        "Not Implemented in Applet...",
                                                        JOptionPane.WARNING_MESSAGE);
                        buttonStatus = true;
                } else if (action == "XML DTD") {
                        System.out.println("in: PopupWindowXML_DTD");
                        JTextArea label = new JTextArea(drawPad.displayDTD(), 20, 40);
                        JOptionPane.showOptionDialog(parent, new Object[] { label },
                                        "XML DTD", JOptionPane.DEFAULT_OPTION,
                                        JOptionPane.WARNING_MESSAGE, null, null, null);
                        buttonStatus = true;
                } else if (action == "load") {
                        System.out.println("in: PopupWindowLoad ");
                        JTextArea label = new JTextArea(schemaData, 20, 40);
                        int result = JOptionPane.showOptionDialog(parent,
                                        new Object[] { label }, "Load: Use Copy(Control-C) & Paste
(Control-V)",
                                        JOptionPane.OK_CANCEL_OPTION, JOptionPane.QUESTION_MESSAGE,
                                        null, null, null);
                        if (result == JOptionPane.OK_OPTION) {
                                buttonStatus = true;
                                System.out.println("Data is: " + label.getText());
                                drawPad.loadData(label.getText());
                        } else {
                                buttonStatus = false;
                        }
                } else if (action == "save") {
                        System.out.println("in: PopupWindowSave ");
                        JTextArea label = new JTextArea(drawPad.saveData(), 20, 40);
                        int result = JOptionPane.showOptionDialog(parent,
                                        new Object[] { label }, "Save: Use Copy(Control-C) & Paste
(Control-V)",
                                        JOptionPane.OK_CANCEL_OPTION, JOptionPane.QUESTION_MESSAGE,
                                        null, null, null);
                        buttonStatus = true;
                } else if (action == "XML elements"
                                || action == "Functional Data Dictionary:Entities"
                                || action == "Functional Data Dictionary:Functions") {
                        System.out.println("in: PopupWindow:" + action);
                        JTextArea label = new JTextArea(drawPad.displayDataDict(action),20, 40);
                        JOptionPane.showOptionDialog(parent, new Object[] { label },
                                        action, JOptionPane.DEFAULT_OPTION,
                                        JOptionPane.WARNING_MESSAGE, null, null, null);
                        buttonStatus = true;
                }
        }
        public boolean getOKorCANCEL() {
                return buttonStatus;
        }
}
```