Proximal Business Intelligence on the Semantic Web

Authors: David Bell and Thinh Nguyen

Department of Information Systems and Computing (DISC), Brunel University, UK
david.bell@brunel.ac.uk, thinh.nguyen@brunel.ac.uk

Abstract: Ubiquitous information systems (UBIS) extend current Information System thinking to explicitly differentiate technology between devices and software components with relation to people and process. Adapting business data and management information to support specific user actions in context is an ongoing topic of research. Approaches typically focus on providing mechanisms to improve specific information access and transcoding but not on how the information can be accessed in a mobile, dynamic and ad-hoc manner. Although web ontology has been used to facilitate the loading of data warehouses, less research has been carried out on ontology based mobile reporting. This paper explores how business data can be modeled and accessed using the web ontology language and then re-used to provide the invisibility of pervasive access; uncovering more effective architectural models for adaptive information system strategies of this type. This exploratory work is guided in part by a vision of business intelligence that is highly distributed, mobile and fluid, adapting to sensory understanding of the underlying environment in which it operates. A proof-of-concept mobile and ambient data access architecture is developed in order to further test the viability of such an approach. The paper concludes with an ontology engineering framework for systems of this type – named UBIS-ONTO.

Keywords: Pervasive informatics, Business intelligence, Semantic web.

1. Introduction

The Ubiquitous computing (Ubicomp) goal of an enhanced computer that makes use of the many devices embedded within the physical environment - effectively invisible to the user -impacts all areas of computing, including hardware components, network protocols, interaction substrates (e.g. software for screens and haptic entry), applications, privacy, and computational methods (Weiser 1993). Since this vision, the Web has provided a platform for applications to outgrow the local machine (Penyala and Shim 2009) and provide a rich source of ubiquitous content (often invisible to the mobile user). Invisibility within the physical environment is a central theme in Ubicomp. John Seely Brown at PARC calls it the periphery (Weiser 1991). In order to de-couple the information that we associate with our current applications and move it into the periphery, a means to transform the content and support the user in their current place is required.   A number of opportunities and challenges need to be met if business systems (and business data) are to be made available in such environments.

A large numbers of devices (from mobile phones to ambient screens), variation in how information should be adapted and the large number of source systems combine to make architecting such system challenging. The cost of display technology is dropping and traditional posters and billboards are being replaced with digital displays (Schmidt 2009). Accessing enterprise data warehouses or enterprise system database back-ends has had little coverage in a pervasive computing context.

In contrast, "wireless sensor networks based on mobile devices will drive a wide range of new urban scale and eventually global-scale applications" (Cambell et al., 2008 pp. 12-21). Raman and Chebrolu (2008) identify a number of issues with regard to sensor network success though; arguing the superficial level at which problems are investigated is a limiting factor. Only by widening (and connecting) research at all levels from hardware

design to information system strategy can substantial progress be made. Chen, Finin & Joshi (2003) identified weakness of traditional UBIS as: (1) limited support for knowledge sharing and context reasoning, (2) limited agreement between programs that wish to share data, (3) Software is typically modified in response to the dynamic nature of contextual information and the environments.

This paper presents an exploration into how commercial business data warehousing solutions can benefit from both sensor and semantic web technologies in order that business data can be both mobile and adaptive. The paper is structured as follows and opens with background material on commercial data warehouse approaches using SAP and introduces proximity sensors. A number of semantic web tools are then presented; followed by a description of a research plan that brings together business reporting, ontology and sensors. This is followed by more detailed description of the artifacts that resulted from the research – comprising two ontology, a Web service based architecture and performance measures – before the work is summarized.

2. Background

2.1. Business Data Meets Ubiquitous Architecture

Data warehousing allows organizations to organize and store a large amount of business data in a format that can be easily analyzed (Bose & Mahapatra, 2001); while data mining techniques use methods of artificial intelligence to facilitate the discovery of data patterns – relationships between underlying data. These components, coupled with visualization of information, user alerting events, are the basis of a new category of Business Intelligence (BI) tools. This research project utilizes the SAP Business Warehouse (BW) as a business data source and attempts to provide a novel data presentation approach based on proximity (to people, places and devices). In a contemporary business setting, business transactions result in a large amount of electronic data stored within information systems. Heterogeneous data (e.g. text, number, format, video, audio, etc.) created by various application systems needs to be extracted, transformed, loaded and consolidated into a form that can be analyzed. These functions – extraction, transformation, loading, consolidation - are applied by many warehouse technologies in order that data can be analyzed and stored using forms (Bose & Mahapatra, 2001). Constant flow of electronic transaction data can make the analytical process difficult (one reason for such external snapshots of transactional and other data). Data warehouses are designed to provide query processing of integrated data views. Large amounts of data are brought together in data warehouses (from different sources) providing multi-dimensional views of the data.
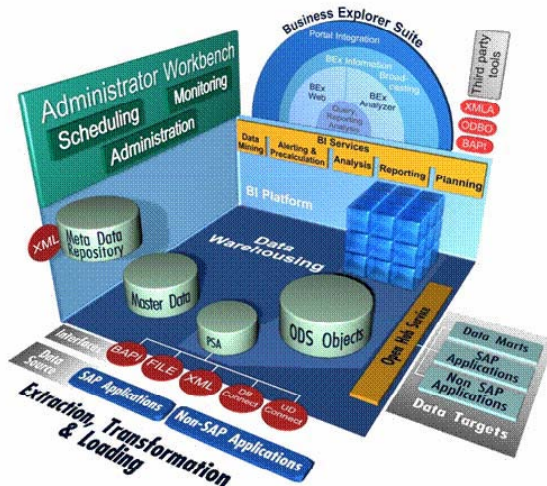

Figure 1: SAP Business Warehousing (http://www.sap.com)

Using SAP's toolset, end-users are able to create queries and aggregate related data in

order to produce appropriate business reports. The Business Explorer's interface (based on an Excel add-on) allows SAP end users to create and save such reports. The popular SAP infrastructure (typical of many data warehousing systems) provides a number of data loading and reporting options – all reliant on successful data cleansing and importation. Exploring the decoupling of both information storage and presentation is at the heart of this paper – moving away from the centralized approaches and using context to filter data for presentation (to support decision making). A need for more timely data access using pervasive business intelligence has gained some recent interest (Watson and Wixom 2007).

In order to determine appropriate devices (mobile, ambient or other) for rendering business information, proximity measurement and response is required. Importantly, proximity not only determines the rendering device in view but is able to direct inferences on what specific business data is appropriate.

2.2. Scale of Proximity – Starting with RFID Technologies

An RFID tag is built with an antenna, a small silicon chip including a radio receiver, a radio modulator, control logic, memory and power system (Simson & Beth, 2006). The silicon chip is fundamentally an IC-based tag chip consisting of an integrated circuit (IC) with memory, essentially a microprocessor. There are various kinds of tags on the market. However, tags may be divided into three basic categories:

• <u>Passive tags:</u> powered by incoming radio frequency from a reader and with a range of a few centimetres to 9 meters.
• <u>Active tags:</u> powered by on-board battery and with a range of 30. "High end onboard capabilities can integrate analogue and digital interfaces to the outside world" (Stanford, 2003, p. 11).
• <u>Semi-active tags (semi-passive tags):</u> powered by on-board battery but using incoming radio signals for power. Reliability is improved (over passive tags) but still has the passive tags short reading range (Simson & Beth, 2006).

Variation in tag functionality provides some interesting possibilities for associating data of interest with particular scales of proximity. This is also applicable to a range of location based devices – Bluetooth, Wifi, GPS and others. Simon and Bath (2006) categorize such positioning systems as either: (1) Coarse-Grained systems such as Global positioning system (GPS), Assisted GPS, Wi-Fi, Bluetooth technology, and few other types of radio frequency infrastructures including RFID tag and RFID reader, Mobile phones (e.g. Cambridge Positioning systems has accuracy of location tracking up to 20 meters and the Rosum system has the accuracy about 3 meters to 25 meters) and (2) Fine-Grain systems that use ultrasound to detect distance from a tag to a dedicated point. Ubisense use ultra wideband radio signals for real time location system tracking system that can detect a target (tag) in a 15 centimeter range (Simson & Beth, 2006).

In order to make use of proximity measures in data provisioning, the source data must first be modeled in such a way that it can then be selected for subsequent presentation on the mobile or ambient devices present. Tools of the semantic web provide one approach to this problem - ontology and ontology query languages in particular – allowing relationships between business data, device capabilities and proximity to be modeled.

2.3. Semantic Web Tools

An ontology is "an explicit specification of a conceptualization" (Gruber, 1995, pp. 1) and provide a formal description of concepts and their relationships within a domain. This result is a shared understanding and/or shared vocabulary for a domain of interest. The W3C Web Ontology (OWL) is itself based on XML using the RDF/XML specification. OWL provides more expressive power, extensibility, modifiability and interoperability (Mizoguchi, 2004). Constraints can be placed on property values, instances of classes and on classes themselves. For example, cardinality constraints, equivalence and reverse property functions. To support the semantic web in a distributed environment, OWL increases

functionality for interoperability and scalability, allowing mapping (via Web URIs) between existing ontologies, importing existing ontology for reuse and interoperability (Chen, Finin & Joshi, 2003). Two projects are relevant to this research -CoBrA and SOUPA.

The CoBrA (Context Broker Architecture) system (Chen, Finin & Joshi, 2003) is a context-broker that detects context-aware devices or contexts in intelligent spaces. The system is able retrieve context-information from devices or environments. Co-BrA-ONT is an ontology written in OWL and provides knowledge on contexts, agents and external sources. For example, a context broker in XCo detects a person's unique RFID number when entering Office Y. Policies on context privacy are included, which determine which contexts/data should be displayed at which locations. Ridhawi et al. (2008) created context policy ontology being used in the authors' own context-aware system.

Architectural bottlenecks limit each context broker to a specific geographical environment such as meeting rooms, laboratories, etc. In practice, Broker agents in CoBrA are "responsible for maintaining and aggregating a shared model for context information. The broker agent facilitates the distributed reasoning capabilities for service agents that make use of CoBrA by including a knowledge model and therefore removes the need to deal with the reasoning part for each service and application." (Ridhawi et al., 2008, p.559). SOUPA extends COBRA-ONT and reuses a number of existing ontologies: (1) FOAF for people, (2) DALM for Time, (3) OpenCyc and RCC for location, (4) COBRA-ONT for contexts, (5) MoGATU BDI for modeling the belief, desire, and intention of human users and software agents and (6) Rei Policy Ontology for access control rules Others have used ontology with data warehousing (Baumbach et al. 2006; Salguero et al. 2008), typically as a means to integrate data when uploading.

3. Research Approach

One question in ubiquitous business intelligence is how proximal and business data can be synthesized more effectively in a mobile setting – bringing together hardware devices and software/application data services. The research described in this paper initiates this process by exploring the use of semantic web middleware (the OWL language and SPARQL query engine), applications (SAP's Warehouse) and proximity knowledge. The research follows a design research approach, which is a search process to discover an effective solution to a problem (Hevner et al. 2004 p.88). The research process can be seen in Figure 2.
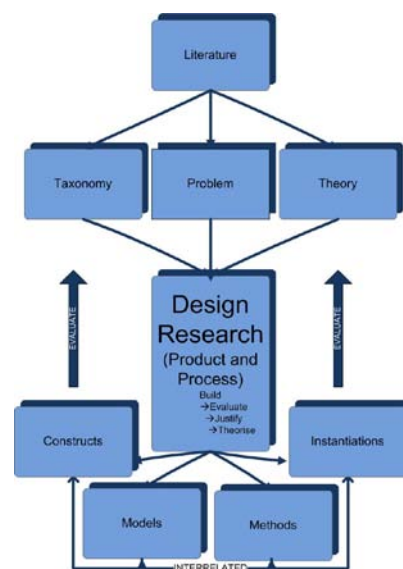

Figure 2. Design research framework (terminology from March and Smith, 1995)

The design research process presented in this paper, and depicted in the diagram above, is

methodologically based on and adapted from the approach described by Nunamaker et al. (1991) and the guidelines presented by Hevner et al. (2004). The following sections address this problem of scalable business intelligence, confirming relevance of the problem being investigated, both in the architecture, ontology and software artifacts. The resulting artifacts are now presented. The starting point for the research is a number of reports that form part of the SAP training material on sales figures and the like. The data in these reports is used to develop a number of ontology and an associated querying mechanism.

4. Research Artefacts

4.1. Proximity Based BI Architecture

Figure 3 depicts our finalized UBIS reporting environment. The system comprises a domain ontology (deployed on the web server) that contains both business and proximity vocabulary. This ontology is then used to generate a number of BiGraphs (Business Intelligence graphs) – business instance data in the form of N3 triples (http://www.w3.org/2001/sw/RDFCore/ntriples/). Triples describe specific elements of business reporting data and relate the data to proximity objects. For example, describing a piece of data that is of interest to Salesman A when within proximity of Office B or City C. A tag in reading range will be read by an RFID reader passing the RFID tag's unique identification number to a host computer (a service named ProxiBI). A simple SPARQL processor on the host computer then chooses an appropriate SPARQL query in order to retrieve business data and render this on an LCD TV within view. A number of these processors are readily available on the Web already. In order to undertake the transcoding, the host computer uses references from a domain ontology document – BusinessStaticNTT.owl published on a web server - in order to understand reporting data in N3. The retrieved data can then be displayed on ambient screen or on a mobile device depending on user preference for this locale (again modeled in the ontology). It can be seen that ontology is at the heart of the system and its creation is of paramount importance in order to support the description of: 1) physical objects (both static and mobile) in the environment, 2) business data that is important to specific users or user categories and 3) the presentation form on specific devices (mobile or static).
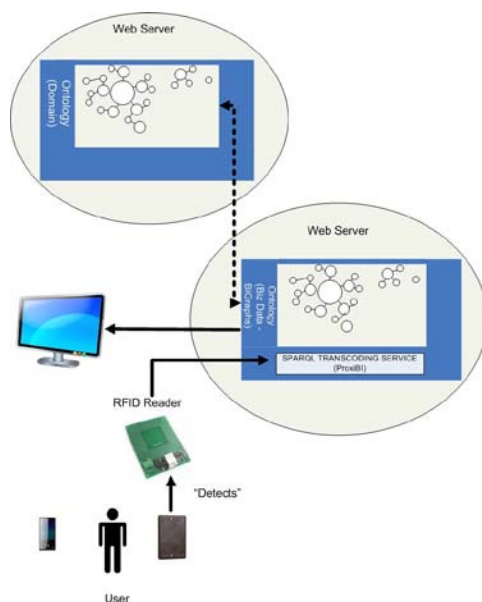

Figure 3. High Level Architecture

4.2. Ontology Topology

Enterprise Ontology (Uschold et al., 1998) and TOVE (Gruninger and Fox, 1995) are well-known methodologies for modeling enterprise structures. Although their ontology specification outputs are quite similar, the development processes are different. There are

a number of steps in the Enterprise Ontology approach defined by Uschold et al. (1998), starting with scoping boundaries for ontology. Brainstorming follows (typically from an unstructured list of words and phrases). Categorization is next – elaborating identified concepts to find relations - placing them into groups or work areas (i.e. parking space and building into a Place group). Elaborating on concepts in each group and arranging them in priority order is then possible (helping to discard or move irrelevant ones to other groups). Each group is then analyzed, defining a term for each concept in a group with concepts being renamed in order that they can be understood more easily by users who will use them as context information for the domain. For example, AtomicPlaceNotInBuilding terms are created to group ParkingSpace and Garden. Building and zone are assigned to the newly created term called CompoundPlace. Terms are then refined in the group by deleting existing ones or adding new terms if necessary - adding definition to each term. The definition is not analogous to the definition of a word in the dictionary but must describe its own meaning and relations with other terms (Uschold et al., 1998). To capture/define precise term definitions, Uschold et al. (1998) identified building blocks such as Entity – class or instance of class - , Relationship – predicate - , and State of Affair. The purpose is to identify types of terms and assign them accordingly. This process was used as an input method for this research.

The Protege-OWL editor supports working with RDF and OWL and was used at the outset of this project. The domain ontology is iteratively constructed using Protégé. In contrast, the graphical environment is somewhat cumbersome when dealing with business instance data. Turtle/Notation 3 (N3) – a quick way of describing RDF triples – is used for this part of the process. Triples in N3 are then able to reference the domain ontology. Berners-Lee (http://www.w3.org/DesignIssues/Notation3) mentions the advantages of readability when using triples – applying also to derivative works such as Turtle (http://www.w3.org/TeamSubmission/turtle/). Instead of creating instances of classes in OWL documents directly, the creation of Notation 3 (N3) files (i.e. SAP.n3) provides a more usable alterative creating statements representing these instances – e.g. about their proximal relevance. Users are able to easily create and maintain instances of classes using vocabulary described in the OWL document. Understandably, the domain ontology in OWL must be developed first. The methodology above was followed with a specific focus on linking reporting objects to proximity objects – places, distances, people etc. The result of this work is a domain ontology document, published on the web server and named Busi-nessStaticNTT.owl. The interpretation process was driven by a set of business reports – systematically analyzing report headers, headings and data. Unsurprisingly, as more reports are interpreted the domain ontology takes shape. Seven test reports were interpreted resulting in an extended version of the ontology presented in Figure 4. It can be seen that proximity vocabulary such as place and proximity are included (including subclasses with respect to an example business environment). Although the domain ontology was created, this process would likely involve the re-use of existing ontology where applicable. Creating a new ontology with both business and proximity vocabulary enables a more controlled testing environment (i.e. in a production environment, future reviews would then harmonize the ontology with golden source ontology).
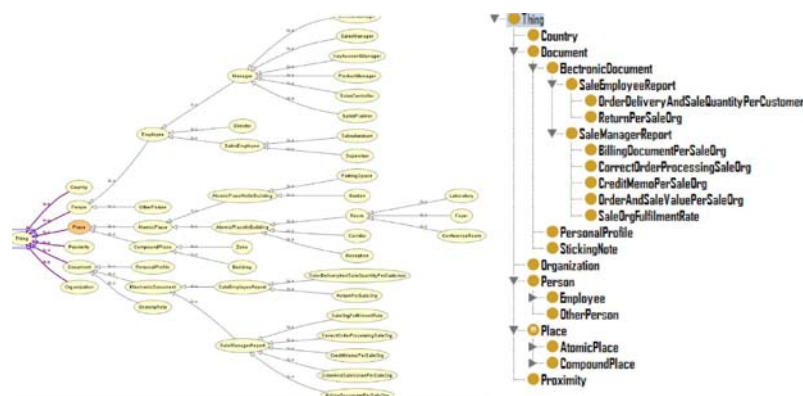
Figure 4. Domain Ontology Example (after 1st Report Interpretation Pass)

The next step in the process is the production of a BiGraph – RDF ontology in N3 that contains report instance data. The example below (Figure 5) presents a small part of the N3 store created using the OWL domain ontology syntax and terms. Creating N3 files was an easier process with a vocabulary already defined and although undertaken manually can easily be automated from source documents. The storage of business data in N3 also opens a number of opportunities for distributing data over the network with placement directed by the query service, source system or available host. While building the N3 document (and relating business data to object proximity) it became apparent that rules can be formulated and placed within the domain ontology. Future work on automating this process is envisaged. Examples of such rules were groupings of specific data elements and patterns of use related to specific locations.

```
@prefix sapd: http://......./BusinessStaticNTT.owl# .
@prefix ns:    <http://example.org/ns# .
@prefix :       http://example.org/services/ .


:Report1
        sapd:Document        "SaleOrgFulfilmentRate_FrankfurtOrg";
        sapd:Place           "St.John building" ;
        sapd:Proximity       sapd:YellowZone ;
        sapd:Interest         sapd:SalesRoles
        sapd:BusData1        1234
        sapd:BusData2        "abcd"
......
```

Figure 5. Sample N3 document created - referencing OWL domain ontology (sapd)

The N3 document contains the following:

1. @prefix : is used to declare namespaces (URIs).
2. sapd : is attribute name – prefix namespace – referring to the namespace for the domain ontology - BusinessStaticNTT.owl.
3. The "#" at the end of the namespace is a symbol to link namespace and entity in the document. If it is not included in the namespace, it should be included inside document. For example, in this case, sapd:Document should be rewritten as sapd:#Document.
4. sapd:Document refers to " http://...../BusinessStaticNTT.owl#Document".

In summary, the N3 file can be seen as simple triple stores. Most semantic query languages are compatible and support N3, and are able to process (i.e. retrieve, remove, edit, etc.) data in triple store appropriately. The ARQ query engine for Jena written in Java is used to experiment with the ontology (as the Jena SPARQL libraries are embedded within a web based data access service -Proxibi). It allows users to create standard SPARQL which facilitates query processing of RDF triples. Basically, SPARQL defines pattern matching statements for interrogation of N3 and OWL documents and retrieves appropriate data. The purpose of this proof of concept is to test the efficiency of using ontology in a simple UBIS for intelligent data provisioning.

Therefore, the motivating use cases are made up for testing only and the proposed scenarios are used to demonstrate how part of the developed system will operate rather than focusing too much on developing the whole application system.

Three scenarios were used for experimentation in this project:
•   Scenario 1: A user enters a building ("St Johns") reception and some available reporting (BiGraphs) should be displayed (i.e. reports of interest to the user).
•   Scenario 2: Two users (a Salesman and Client) are standing next to each other. The system should infer a meeting and provide appropriate documents for their discussion.

- **Scenario 3**: A salesman ("Salesman1") is attending a meeting in Frankfurt and he would like to view data related to sale fulfilment for the Frankfurt Organisation.

Queries are created for each of the above scenarios and stored under the names sap1.q, sap2.q and sap3.q respectively. The queries are subsequently tested using reporting data stored in N3 documents of various sizes (replicating existing triples for performance analysis). This testing will show whether the retrieved data is actually extracted from defined SPARQL queries and if they conform to the users' requirements and more importantly whether the performance of such a system is adequate.

The query used for Scenario 3 (sap3.q) is shown below in Figure 6:

```
PREFIX sapd: <http://........./BusinessStaticNTT.owl#>
SELECT ?Document ?calendarYearMonth ?openOrderQuantity ?incomingOrderQuantity
?fulfilmentRateQuantityOnPercentage
FROM <SAP.n3>
WHERE
{
        ?loc    sapd:Document        ?Document .
        ?loc    sapd:Organization    ?Organisation .
        ?loc    sapd:Organization    "Frankfurt" .
        ?loc    sapd:Document        "SaleOrgFulfilmentRate_FrankfurtOrg".
        ?loc    sapd:calendarYearMonth   ?calendarYearMonth .
        ?loc    sapd:openOrderQuantity           ?openOrderQuantity .
        ?loc    sapd:incomingOrderQuantity   ?incomingOrderQuantity .
        ?loc sapd:fulfilmentRateQuantityOnPercentage?fulfilmentRateQuantityOnPercentage.
}
```

Figure 6. Test SPARQL query

The query has some similarity to SQL with select/from/where terms. The interesting aspects of the query are that the files (domain ontology and InfoGraph) are on the Web – and can be placed anywhere. The resulting output of the query is able to be rendered on a number of devices (simple ambient and mobile output in the case of this research). An example of the output can be seen in Figure 7. The choice of device used is based on proximity detection (altering the query in some cases).
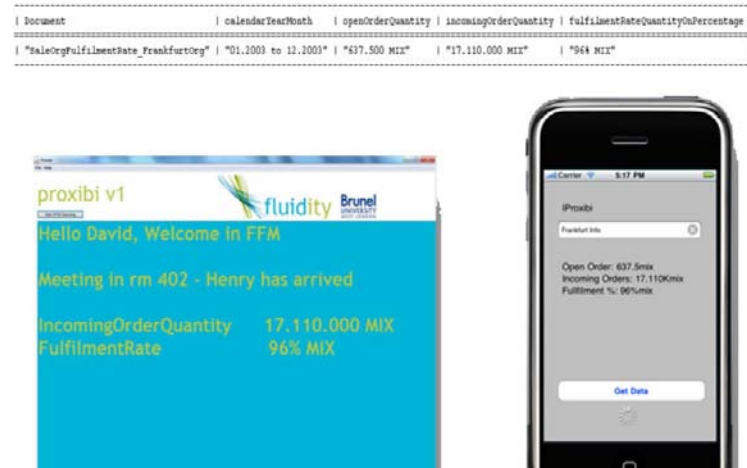

Figure 7. Test output from the SPARQL query and service (test environment, ambient and mobile)

4.3. Performance Analysis

Three scenarios (three queries) are executed (see Table 1 and Figure 8) - each query is

executed ten times for each of the N3 files listed below (mean timings being reported). The results are encouraging with adequate performance when running even with 20,000 triples (detailing 7000 BiGraphs). Obviously, the underlying network speed is a large dependency here and the figures presented are taken from a campus network, local N3 store and internet sourced domain ontology.

DataSet Name	Document	Lines	Description	Query time
Small size	7	95	N3 list of 7 document	2.278 seconds
Medium size	70	244	10 times of N3 list (document name changed)	2.409 seconds
Large size	700	1452	100 times of N3 list (document name changed)	2.786 seconds
Extra large size	7000	24004	1000 times of N3 list (document name changed)	5.246 seconds

Table 1. N3 Input file sizes

Although the performance results are encouraging, the process by which the artifacts came about is of equal or more interest. This process is presented as an UBIS ONTO framework – presenting both a list of concerns and steps to engineer the ontology. The interplay between the domain ontology, N3 Business instance data and the queries – namely where and when knowledge and rules are created, extended and stabilize – is apparent as each artifact was developed and subsequently integrated.
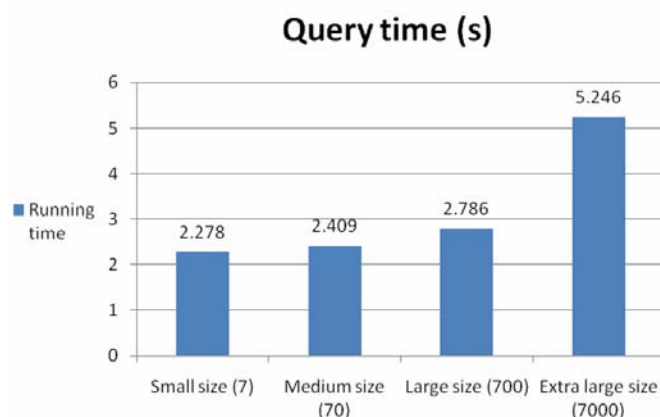

Figure 8. SPARQL Query execution time

5. UBIS ONTO Framework

Reflecting on the process that unfolded while developing the presented artifacts, a number of areas required analysis and subsequent decision making (see Figure 9). The starting point is the source documents/data sources that underpin current business reporting functionalities. It is this data, coupled with user requirements/scenarios, that is able to direct initial domain ontology production. Once initial domain ontology has been created, a number of activities come into play in order to extend the ontology with necessary detail – taking into account the objects residing in the environment and the target ontology topology (placement of domain and triple based ontology over the network). In addition, user preferences and scenarios (how the user interacts with the system) direct ontology re-organization over time. It should be noted that personal data privacy and security are not within the scope of this paper.
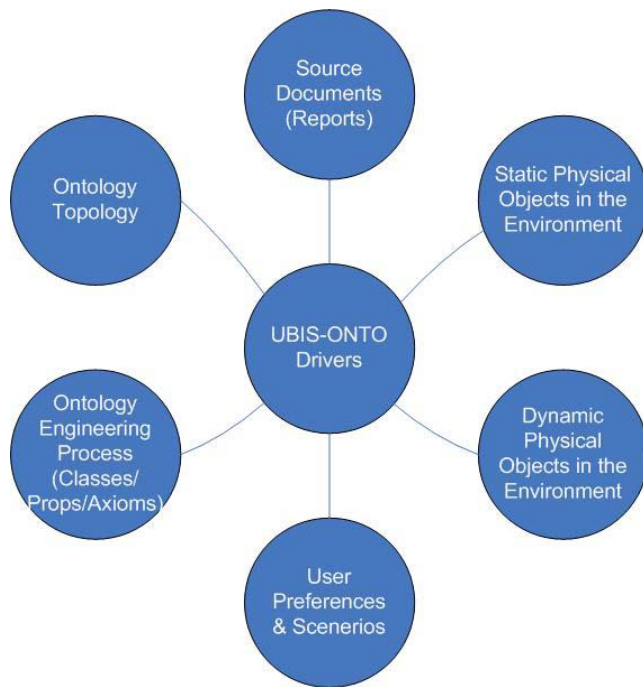
Figure 9. UBIS-ONTO Evolution

5.1. Recommendations for Engineering

Based on the resulting research artifacts and the process by which they were developed, a number of steps are presented summarizing the UBIS-ONTO process (Table 2). The process includes many of the steps identified by Uschold et al. (1998), but extends traditional methodologies with the addition of two key steps: (a) detailed interpretation and integration within the ontology of proximity objects and measures and (b) a harmonizing process that re-organizes ontology and queries based on their use. The work also addresses weaknesses identified by Chen, Finin and Joshi (2003) through the use of commodity semantic web technology and by: (1) adopting OWL and N3 as an easy to use knowledge representations platform supporting data sharing and 2) a focus on the ontology (as opposed to software) in order to better support the dynamic nature of contextual information and the environments (i.e. less if any impact on software systems). The ability to distribute business data across the network (in separate N3 stores) is also better able to support transactional flow – placing aggregate data in separate stores and more dynamic content nearer to source if hosting options are available. Ontology centered approaches also provide a level of independence from hardware and software – a more general advantage highlighted by Pendyala and Shim (2009).

	UBIS-ONTO STEPS
1	Source documents are interpreted – extracting classes and properties from structural labels (headers, headings etc.)
2	Review and organize interpretation of the domain – with output into an OWL ontology. Identification, organization, re-naming, re-categorization of classes and properties.
3	Add additional detail to classes and properties. Detail in this case is driven by projected future query requirements.
4	Analyse the physical environment and identify objects (static and dynamic) of interest.
5	Transform instance data in N3 triples (automated where possible). Transforming current reporting data into a single or network of triple files/stores.
6	Extend triple document/store with proximity definitions – with interest in particular concepts when "near" to identified objects. This will include the definition of "near" with respect the objects in question.
7	Develop SPARQL queries (and support presentation adaptors) to support identified user scenarios.
8	Harmonize ontologies and queries in use – moving indentified groupings and/or patterns into the domain ontology.

Table 2. Ontology Engineering Steps (UBIS-ONTO)

6. Conclusion

The paper presents a novel approach to mobile business data reporting that interweaves proximity sensors, business data warehousing and semantic web tools. The resulting system and methodology is relatively light weight allowing business data to be stored over the network and displayed when and where most appropriate to the user (radically different from current approaches that rely on centralized data cleansing, loading and storage). The paper focuses on how ontology tools can be used to realize a UBIS business intelligence vision – virtually integrating business data on the Web for presentation on mobile or ambient devices. The practical work includes experimentation that emphasizes the viability of the approach in terms of performance. The research itself is exploratory in nature and much is left to investigate. One key area for the future is testing the approach in a real-world business environments (moving beyond the SAP test data being utilized here), and investigating the evolution of ontological artifacts (including queries) over time.

References

Baumbach, J., Brinkrolf, K., Czaja, L.F., Rahmann, S., Tauch, A. (2006) CoryneRegNet: An ontology-based data warehouse of corynebacterial transcription factors and regulatory networks. BMC Genomics 7(24).

Bose, I., Mahapatra, R. (2001) Business Data Mining – A Machine Learning Perspective. Information & Management 39, 211–225.

Campbell, A.T., Eisenman, S.B., Lane, N.D., Miluzzo, E., Peterson, R.A., Lu, H., Zheng, X., Musolesi, M., Fodor, K., Ahn, G. (2008) The Rise of People-Centric Sensing. IEEE Internet Computing 12(4), 12–21, doi:10.1109/MIC.2008.90.

Chen, H., Finin, T., Jishi, A. (2003) Using OWL in a Pervasive Computing Broker. In: Proceedings of Workshop on Ontologies in Open Agent Systems, AAMAS 2003.

Chen, H., Perich, F., Finin, T., Joshi, A. (August 2004) SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In: Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services.

Daniel, H., Albert, P., Mike, P. (2007) RFID, A Guide to Radio Frequency Identification. Wiley, USA.

Gruber, T.R. (1995) Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human and Computer Studies 43(5/6), 907–928.

Gruninger, M., Fox, M. (1995) Methodology for the design and evaluation of ontologies. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing held in Conjunction with IJCAI 1995, Montreal, Canada.

Hevner, A., March, S., Park, J., Ram, S. (2004) Design Science in Information Systems Research. MIS Quarterly 28(1).

Mizoguchi, R. (2004) Tutorial on Ontological Engineering Part 2, Ontology Development, Tools and Languages. New Generation Computing 22, 61–96.

Nunamaker, J., Chen, M., Purdin, T. (1991) System Development in Information Systems Research. Journal of Management Information Systems 7(3), 89–106.

Pendyala, V.S., Shim, S.S.Y. (2009) The Web as the Ubiquitous Computer. Computer

42(9).

Raman, B., Chebrolu, K. (2008) Censor networks: a critique of "sensor networks" from a systems perspective. SIGCOMM Comput. Commun. Rev. 38(3), 75–78.

Ridhawi, Y., Harroud, H., Karmouch, A., Agoulmine, N. (2008) Policy Driven Context-Aware Services in Mobile Enviroments. Innovation in Information Technology, 558–562.

Salguero, A., Araque, F., Delgado, C. (2008) Ontology based framework for data integration. WSEAS Trans. Info. Sci. and App. 5(6), 953–962.

Simson, G., Beth, R. (eds.) (2006) RFID Applications, Security, and Privacy. Addison Wesley, United States.

Schmidt, A. (2010) Ubiquitous Computing: Are we there yet? Computer 43(2).

Stanford, V. (2003) Pervasive computing goes the last hundred feet with RFID systems. Pervasive Computing 2(2), 9–14.

Uschold, M., King, M., Moralee, S., Zorgios, Y. (1998) The Knowledge Engineering Review 13, 31–89.

Weiser, M. (1993) Hot topics-ubiquitous computing. Computer 26(10), 71–72.

Weiser, M. (1991) The Computer for the 21th Century. Scientific American 265(3), 94–101.