

MANAGING CONTEXTUAL INFORMATION IN SEMANTICALLY-DRIVEN TEMPORAL INFORMATION SYSTEMS

Sakirulai O. Isiaq, Dr Taha Osman, Dr Evtim Peytchev
Department of Computing and Technology,
School of Science and Technology, Nottingham Trent University,
Nottingham, United Kingdom.
sakirulai.isiaq@ntu.ac.uk, taha.osman@ntu.ac.uk, evtim.petchev@ntu.ac.uk

Abstract— Context-aware (CA) systems have demonstrated the provision of a robust solution for personalized information delivery in the current content-rich and dynamic information age we live in. They allow software agents to autonomously interact with users by modeling the user's environment (e.g. profile, location, relevant public information etc.) as dynamically-evolving and interoperable contexts. There is a flurry of research activities in a wide spectrum at context-aware research areas such as managing the user's profile, context acquisition from external environments, context storage, context representation and interpretation, context service delivery and matching of context attributes to users 'queries etc. We propose SDCAS, a Semantic-Driven Context Aware System that facilitates public services recommendation to users at temporal location. This paper focuses on information management and service recommendation using semantic technologies, taking into account the challenges of relationship complexity in temporal and contextual information.

Keywords- Context-aware Systems; Semantic Web; Ontology Engineering; Modelling Methodologies; Mobile Infrastructures, Semantic repository, Knowledge Management;

I. INTRODUCTION

Context-aware (CA) systems have demonstrated to be robust for the delivery of content-rich and dynamic information to nomadic users with respect to their ever-changing location and status. It permits software agents to autonomously interact with users by modeling the environment as a dynamically-evolving and interoperable context.

Contextual information is classified as internal and external in [1] and could be obtained via various means, varying from sensors, browsing profiles, network information etc.

A. Motivating Scenario

This study was motivated by the involvement of the author in a project, aimed to give a reliable services recommendation to nomadic users based on their context; these contexts could either be static contexts (e.g. geographical location, virtual location attributes like car-

park, buildings etc.) or dynamic contexts (like events, meeting, etc.).

The challenges of the project relied on investigating two main areas that include the compilation and management of contextual information and wireless communications within the framework. The framework should fulfill requirements such as (1) recommending adequate and relevant services to users based on previous and current context (2) giving adequate and response to user's timely query (query may be generated based on user's context) (3) constant update and exchange of contextual information at real time without a significant effect to the knowledgebase and information retrieval amongst others.

We propose SDCAS, a Semantic-Driven Context Aware System for the managing contextual information to recommend adequate and relevant services to users by making use of contextual information such as user profile, device, status, location etc.

II. BACKGROUND/OVERVIEW

This section gives an overview of context-aware systems surveys and the different approaches for modeling.

A. Definition

Context cannot be given a specific definition due to the different ways it is being perceived by several people. Many researchers and authors have given various definitions. Baldauf et.al in [2] highlighted common definitions as well as their authors.

For the purpose of specificity and the nature of the project, this paper goes with the widely adopted definition of context by authors in [3], they defined "Context as the information required for the characterization of an entity's (Person, Place, Object) situation having in consideration the interaction between users and the application".

B. Architecture

The target domain and context requirements play a vital role on how a context-aware system can be designed. Chen et.al [4] gave an overview on three types of context-aware

architecture. We made use of the context server type of architecture as this allows remote data accessibility due to the client/server feature, also for its comprehensiveness in context management and robustness for context information delivery.

C. Modelling

Modeling CA systems can vary from the key-value modeling where service discovery is through the use of a key-pairing matching algorithm. This is the simplest approach to context modeling but it might not be suitable for complex systems as noted in [3].

Sheng and Benatallah in [5] describe an approach that modeled a CA system using Unified Modeling Language (UML) which is a graphical approach of modeling. It was also argued in [6] that due to the powerful tools of relational databases (RDBMS) to manage large data efficiently, the synthesizing ideas could be used to build graphical models; therefore, context-aware query can be generated to query the model. However, constructing the user-define methods for the model can be time-consuming and not very efficient as discussed in [3].

The Hydrogen project in [7] makes use of the object-oriented method of modeling by utilizing the benefits of object-oriented programming features like encapsulation, reusability, inheritance etc., but knowledge sharing and historic use of meta-data is not feasible with this method. Ontology model has proven to be highly promising due to concept representation and their relationships in a very expressive manner [8], thus allowing for a sophisticated reasoning over the complexity of context-aware data.

Despite the proves for the robustness of this approach to modeling CA systems, there are also shortcomings like higher time consumption than user-define rule base reasoning which is attributed to the inbuilt large set of rule used for reasoning and this might have an adverse effect on time-critical dependent applications. We made use of the Ontology approach due to its intense support for knowledge sharing by the use of meta-data and reasoning capacity amongst other reasons.

III. RELATED WORK

Pervasive computing has become a rich area for research as several researchers had come up with different frameworks and methodology for creating CA systems. SOUPA (Standard Ontology for Ubiquitous and Pervasive Application) [9] was designed using Web Ontology Language OWL, it includes vocabulary component that represent beliefs, desire intentions, profiles, actions and security policies.

A Broker-Centric agent architecture CoBrA was used for development of smart meeting room in [4]. CoBrA was designed to reduce developer's effort in creating location context and also addressed the issue of context

reasoning, knowledge sharing with a feature of user protection policies for CA systems.

A further description was given in [10] on how SOUPA can be used in CoBrA and MoGATU (also a framework used for the management of peer-to-peer data for pervasive) frameworks in developing context systems. Although the intention of SOUPA was to create a standard ontology for systems of similar features, it could not fulfill the requirements of a proposed system in the aspect of conceptualization and context managements.

Context Ontology (CONON) [11] also used the Web Ontology Language for context reasoning which was designed in two folds; this includes checking context consistence at the upper level, and differentiating high level of implicit contexts to low-level explicit contexts. CONON concentrated more on knowledge sharing and learning for context interpretation and hybrid reasoning.

Context Management Framework (CMF) [12] is a generic framework that allows extension for application specific CA systems. It uses OWL_DL reasoner to process and exchange contextual information within the context domain. CMF is very generic that it accommodates an extension for another context provider; however, more work might have to be done in order to make use of the system because of its non-specificity as it will be time consuming developing a framework for just integration purpose.

The Service Oriented Context Aware Middleware SOCAM also uses the Web Ontology Language for context reasoning [13]. The context Interpreter component of the architecture takes the responsibility of querying the context knowledgebase, maintaining the consistency of context data and resolution of the context conflicts in the knowledgebase. This framework allows developers to write pre-defined rules in a file which can be loaded at application runtime or preload into the context reasoner. SOCAM context reasoning performance was highlighted in [14]. The results show that reasoning is computationally intensive and could result in a performance bottle-neck, although can be acceptable for a non-time critical application.

IV. DEVELOPMENT METHODOLOGY

The functional and non-functional requirements of the proposed system has made time-criticality an essential requirement to fulfill due to continuous change of context information and data adequacy maintenance.

It was also observed less concentration and emphasis are laid on the maintenance and a continuous update of the knowledgebase for a time-critical application, so we propose SDCAS following the methodology below.

A. Knowledge elicitation

This is referred to as the process of knowledge gathering from the domain experts in order to identify

system requirements and acquire a thorough understanding of the required concept for the target domain. As noted in [15], there has always been challenges in the engagement of domain experts (usually non-technical) when developing ontology for a non-technical domain. There are various approaches to knowledge elicitation which includes formal interview (structured or unstructured), observation, questionnaire (direct or indirect questions), concept mapping etc. The idea of formal interview and concept mapping was adopted for adequate understanding. Cmaps tool [16] was used for concept interaction and knowledge sharing with the domain experts.

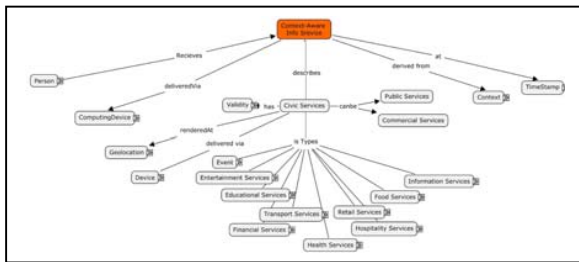


Fig.1 Concept map deduced from knowledge elicitation.

The User (Person) gets recommendation of the service on the ComputingDevice which could either be a mobile or static device. The Context-Aware Info Service (CAIS) comprises of the knowledge management of contextual information and wireless framework for the information delivery. Matching of the context which requires underlying human-like understanding is formalized within the Knowledge management of the CAIS.

B. Ontology development

We could not find ontology that matches the requirement of the project; so the system ontology was developed from scratch to fulfill the requirements gathered from the knowledge elicitation phase. The system ontology was developed using a protégé editor [33] for hierarchical declaration of taxonomies and their relationships. The ontology primarily consists of Subject, Events, Services, SpecificGeolocation, having several hierarchical classification and inter-relationships with further entities like, Profile, Devices Validity and Interest etc. SDCAS ontology consists of 168 classes, 54 object properties, and 38 data properties.

C. Constructing the knowledgebase

After developing the ontology, the challenge was the choice of repository for concept representation, which is to populate the developed ontology with the relevant instances i.e. the A-box. We also had to take into consideration the repository's reasoning engine (forward

or backward chaining), repository's efficiency in response to query, loading, update etc. and the effects in system development.

Many semantic repositories have been designed by various vendors to fulfill the role of semantic data storage and reasoning. Unlike relational databases, there has not been a standard for benchmarking semantics repositories like TPC [17]. Thakkar et.al in [18] performed a rigorous experiment on benchmarking some popular semantic repositories. Result shows BIGOWLIM [19] on the average gives the best query response time using UOBM [20] dataset. Allegrograph [21] and JENA [22] are fast with deletion of triples and Virtuoso [23] was confirmed to be fast with SPARQL/UPDATE query, but has the worst recall as it poor with giving answers to query from the API, but it does performs well with Virtuoso/SPARQL; although this has been attributed to a bug in API implementation rather than the repository capability itself [18].

In line with justify any findings in our research, we opted for BIGOWLIM and JENA/TDB repositories for the following reasons; BIGOWLIM gives more response query although, it is slow in data loading and deletion due to feature attributed to forward-chaining which the repository utilizes. Positive response to query is essential to the effectiveness and fulfillment of the requirements of the proposed framework. Also JENA/TDB supports data persistence, SPARQL/UPDATE query and its performance in terms of query response time is good.

D. Data aggregation, conversion and loading

A major challenge that has to be addressed was the population of the ontology as most of the required, relevant and available instances are not in RDF (Resource Description Framework) format. One of the required data is Open-Streetmap [24] data which contain the GPS (Global Positioning System) details of specific-locations. This data is not available in RDF as required, so a conversion into RDF format has to be done. The major challenge is to be sure the corresponding GPS data and their named locations are not mismatched. Open-Streetmap data can be downloaded at <http://www.openstreetmap.org>.

SDCAS ontology has been designed to have a class of SpecificGeolocation relating to a GPSCordinate class via and object relationship *hasGps*; therefore, the GPS data needs to be independent of one another for easier accessibility by some other classes such as the Device class etc.

Having had to separate the GPS data from the name of place data to meet the designed ontology specifications was a big challenge. Byrne in [25] discussed the mapping of RDBMS data in separate tables within the same database for RDF conversion and also how to deal with

joins using the UNION clause by ensuring the links between the data are found.

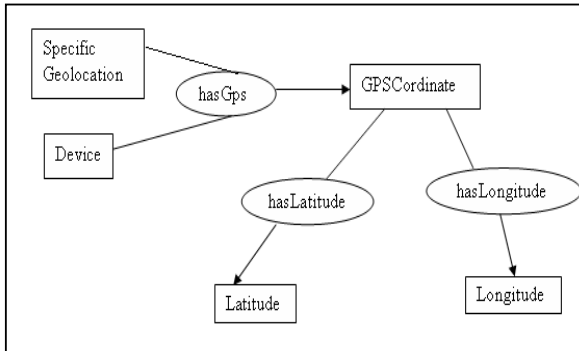


Fig. 2 A representation of concepts and the relationships within the system ontology.

Data conversion from RDBMS format was performed using the D2R tool [26] which can be an essential tool for publishing relational database on the semantic web; it allows a browser to navigate the content of the semantic web. The tool work very well with MySQL database, so we uploaded Open-Streetmap data into MySQL 5.5 database prior data conversion

Obtaining data from several sources that use different data formats/types posed a major challenge to our ontology design as data has a distinctive conceptualization for the required domain and this has to be done in mapping instances to our ontology (i.e. the A-box and T-box), due to instances having different concepts. We obtained data from Open-Streetmap [21], Uk-Ordinance Survey [27], dbpedia [28] etc. to build a content-rich system.

Researches on ontology mapping have largely concentrated on the automation of ontology mapping [29]. hence, ontology mapping tools like PROMPT [30] Chimaera[31], OWL Lite Alignment OLA [32] and few other ontology mapping tools, do not fulfill the requirements of the system in having 100% data accuracy or close as a minor mistake will give irrelevant recommendation, also some of the data obtained are richer only when using multiple ontologies as noted in [18]. Most automated mapping tools use NLP and other Language software to check class similarities, but this will not work with ontologies non-similar class nomenclatures.

Following the above reasons we opted for a one to one mapping using SPARQL/CONSTRUCT query. The query is used to extract relevant instance need for our ontology from the other richer dataset to generate a clean context for the A-box and T-box.

```

PREFIX owl:
<http://www.w3.org/2002/07/owl#>
PREFIX dc:
<http://purl.org/dc/elements/1.1/>
PREFIX xsd:
<http://www.w3.org/2001/XMLSchema#>
PREFIX owl2xml:
<http://www.w3.org/2006/12/owl2-xml#>
PREFIX rdfs:
<http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf:
<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX services:
<http://www.contextaware.com/onotologies/services.owl#>
PREFIX db:
<http://localhost:2020/resource/>

CONSTRUCT {
?inst a services:GPSCordinate.
?atmgps_gpsid a services:label.
?inst services:label ?atmgps_gpsid.
?atmgps_lat a services:hasLatitude.
?inst services:hasLatitude
?atmgps_lat.
?atmgps_lon a services:hasLongitude.
?inst services:hasLongitude
?atmgps_lon.
?atmgps_created a services:created.
?inst services:created
?atmgps_created .
}
WHERE {
$inst a vocab:atmgps.
$inst vocab:atmgps_gpsid
?atmgps_gpsid.
$inst vocab:atmgps_lat ?atmgps_lat .
$inst vocab:atmgps_lon ?atmgps_lon
.
$inst vocab:atmgps_created
?atmgps_created .
}
  
```

Fig. 3 An example of a Sparql query for individual class-mapping.

Extraction of data in figure 3.will be ideal for a scenario of few classes but this will generate a rigorous manual labour in a situation like 168 classes of ontology. This is an important contribution of this paper researching into a regular update to a semantic repository where an automatic mapping is not feasible and one-to-one mapping has to be used.

```

PREFIX xsd:
<http://www.w3.org/2001/XMLSchema#>
PREFIX owlxml:
<http://www.w3.org/2006/12/owl2-xml#>
PREFIX rdfs:
<http://www.w3.org/2000/01/rdf-
schema#>
PREFIX rdf:
<http://www.w3.org/1999/02/22-rdf-
syntax-ns#>
PREFIX services:
<http://www.contextaware.com/onotologi
es/services.owl#>
PREFIX db:
http://localhost:2020/resource/
PREFIX vocab:
http://localhost:2020/vocab/resource/

CONSTRUCT {
?inst a services:GPSCoordinate.
?" + gpsName + "_gpsid  a
services:label.
?inst services:label      ?" + gpsName +
" _gpsid.
?" + gpsName + "_lat      a
services:hasLatitude.
?inst services:hasLatitude  ?" +
gpsName + "_lat.
?" + gpsName + "_lon      a
services:hasLongitude.
?inst services:hasLongitude ?" +
gpsName + "_lon.
?" + gpsName + "_created  a
services:created.
?inst services:created     ?" + gpsName +
"_created .
}
WHERE {"
$inst a      vocab:" + gpsName + ".
$inst      vocab:" + gpsName +
"_gpsid  ?" + gpsName + "_gpsid.
$inst      vocab:" + gpsName + " lat
?" + gpsName + "_lat .
$inst      vocab:" + gpsName + " lon
?" + gpsName + "_lon .
$inst      vocab:" + gpsName +
"_created  ?" + gpsName + "_created .
};

```

Fig. 4 Sparql query for loading several classes

A programming interface developed to map instances to the relevant classes (i.e. writing a CONSTRUCT query for each class). The names of the classes are written in a file and loading could be done by matching the class

names in the file with the one in repository. A CONSTRUCT query that matches the name class on the file and the one in the ontology is the generated as show in fig. 4. We feel loading individual class CONSTRUCT query manually into the repository might have a performance effect on the repository, although this is yet to be tested at the point of writing this paper.

V. CONCLUSION AND FUTURE WORK

Ontology models have demonstrated a robust way of modeling context aware systems despite the few shortcomings. The major aim of this paper is to propose methodology for the deployment of semantic web that require continuous update of datasets from different repositories. We have been able to show critically the process of managing a semantic repository with different data type/format and data source.

Major contributions in using semantic technology to build CA system has been focused on the context interpretation and delivery, but less concentration had been done on the maintenance of this context information .

We were able to identify semantic approach as suitable for CA systems modeling. Also contribute to process of identifying specific methodology for semantic application that requires regular update of triples into the knowledgebase in temporal application development. SPARQL/UPDATE now include the INSERT and DELETE keywords which can be used for regular update of triple in similar to the traditional relational database, although this is not being supported by several repository but Jena supports it. Emphasis will have to be laid on the effect of update on the triple's relationships in giving adequate and correct response to user's query.

Our future work is to concentrate on the effect of regular update (insertion and deletion of relating concepts) in situation such as continuous data retrieval within the repository, identifying a robust maintenance methodology for a semantic repository, highlighting the performance issues of repository on data update and benchmarking of semantic repository in response to data modification. Further investigation are been currently been done on method of update such as SPARQL, RDOL, Jena etc.

REFERENCES

- [1] P. Prekop and M. Burnett, "Activities, context and ubiquitous computing," *Comput. Commun.*, vol. 26, pp. 1168-1176, 2003.
- [2] M. Baldauf, S. Dustdar and F. Rosenberg, "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, pp. 263-277, 2007.

- [3] A. K. Dey and G. D. Abowd, "Towards a better understanding of context and context-awareness," in *CHI 2000 Workshop on the what, Who, Where, when, and how of Context-Awareness*, 2000, pp. 1-6.
- [4] H. Chen, T. Finin, A. Joshi, L. Kagal, F. Perich and D. Chakraborty, "Intelligent agents meet the semantic web in smart spaces," *Internet Computing, IEEE*, vol. 8, pp. 69-79, 2004.
- [5] Q. Z. Sheng and B. Benatallah, "ContextUML: A UML-based modeling language for model-driven development of context-aware web services," in *Mobile Business, 2005. ICMB 2005. International Conference on*, 2005, pp. 206-212.
- [6] J. Zheng and J. Sun, "Building graphical models from relational databases for context-aware querying," in *Information Engineering, 2009. ICIE'09. WASE International Conference on*, 2009, pp. 626-630.
- [7] T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann and W. Retschitzegger, "Context-awareness on mobile devices-the hydrogen approach," in *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, 2003, pp. 10.
- [8] M. Angermann, P. Robertson and T. Strang, "Issues and requirements for bayesian approaches in context aware systems," *Location-and Context-Awareness*, pp. 235-243, 2005.
- [9] H. Chen, T. Finin and A. Joshi, "The SOUPA ontology for pervasive computing," *Ontologies for Agents: Theory and Experiences*, pp. 233-258, 2005.
- [10] H. Chen, F. Perich, T. Finin and A. Joshi, "SOUPA: Standard ontology for ubiquitous and pervasive applications," in *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. the First Annual International Conference on*, 2004, pp. 258-267.
- [11] X. H. Wang, T. G. Da Qing Zhang and H. K. Pung, "Ontology based context modeling and reasoning using OWL," 2004.
- [12] H. van Kranenburg, M. S. Bargh, S. Iacob and A. Peddemors, "A context management framework for supporting context-aware distributed applications," *Communications Magazine, IEEE*, vol. 44, pp. 67-74, 2006.
- [13] T. Gu, H. K. Pung and D. Q. Zhang, "A service-oriented middleware for building context-aware services," *Journal of Network and Computer Applications*, vol. 28, pp. 1-18, 2005.
- [14] T. Gu and H. K. Pung, "A middleware for building context-aware mobile services," in *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, 2005, pp. 2656-2660.
- [15] J. W. Coffey, R. R. Hoffman, A. J. Cañas and K. M. Ford, "A concept map-based knowledge modeling approach to expert knowledge sharing," *Proceedings of IKS*, pp. 212-217, 2002.
- [16] CMAPS TOOL. Available: <http://cmap.ihmc.us>.
- [17] TPCbenchmarks. Available: <http://www.tpc.org/information/benchmarks.asp>.
- [18] D. Thakker, T. Osman, S. Gohil and P. Lakin, "A Pragmatic Approach to Semantic Repositories Benchmarking," *The Semantic Web: Research and Applications*, pp. 379-393, 2010.
- [19] BIGOWLIM. Available: <http://www.ontotext.com/owlim/big>.
- [20] L. Ma, Y. Yang, Z. Qiu, G. Xie, Y. Pan and S. Liu, "Towards a complete OWL ontology benchmark," *The Semantic Web: Research and Applications*, pp. 125-139, 2006.
- [21] Allegrograph. Available: <http://www.franz.com/agraph/allegrograph>.
- [22] JENA. Available: <http://jena.sourceforge.net>.
- [23] Open link virtuoso. Available: <http://virtuoso.openlinks.com>.
- [24] OpenStreetMap. Available: www.openstreetmap.org.
- [25] K. Byrne, "Relational Database to RDF Translation in the Cultural Heritage Domain," *Internet, may*, 2008.
- [26] C. Bizer and R. Cyganiak, "D2r server-publishing relational databases on the semantic web," in *5th International Semantic Web Conference*, 2006,
- [27] Maps of Britain from Ordnance Survey.
- [28] Wiki.dbpedia.org .
- [29] N. F. Noy, "Semantic integration: a survey of ontology-based approaches," *ACM Sigmod Record*, vol. 33, pp. 65-70, 2004.
- [30] N. F. Noy and M. A. Musen, "The PROMPT suite: interactive tools for ontology merging and mapping," *International Journal of Human-Computer Studies*, vol. 59, pp. 983-1024, 2003.
- [31] D. L. McGuinness, R. Fikes, J. Rice and S. Wilder, "The chimaera ontology environment," in *Proceedings of the National Conference on Artificial Intelligence*, 2000, pp. 1123-1124.
- [32] P. Lambrix and H. Tan, "A tool for evaluating ontology alignment strategies," in *Journal on Data Semantics VIII*, 2007, pp. 182-202.
- [33] The Protégé Ontology Editor and Knowledge Acquisition System, Available protege.stanford.edu