

Ontology Modelling Methodology for Temporal and Interdependent Applications

Sakirulai Isiaq

Department of Computer and Information Technology
Nottingham Trent University
Nottingham, United Kingdom
sakirulai.isiaq@ntu.ac.uk

Taha Osman

Department of Computer and Information Technology
Nottingham Trent University
Nottingham, United Kingdom
taha.osman@ntu.ac.uk

Abstract—The increasing adoption of Semantic Web technology by several classes of applications in recent years, has made ontology engineering a crucial part of application development. Nowadays, the abundant accessibility of interdependent information from multiple resources and representing various fields such as health, transport, and banking etc., further evidence the growing need for utilising ontology for the development of Web applications. While there have been several advances in the adoption of the ontology for application development, less emphasis is being made on the modelling methodologies for representing modern-day application that are characterised by the temporal nature of the data they process, which is captured from multiple sources. Taking into account the benefits of a methodology in the system development, we propose a novel methodology for modelling ontologies representing Context-Aware Temporal and Interdependent Systems (CATIS). CATIS is an ontology development methodology for modelling temporal interdependent applications in order to achieve the desired results when modelling sophisticated applications with temporal and interdependent attributes to suit today's application requirements.

Keywords- *semantic knowledgebase; supplementary reasoning; temporal and Interdependent knowledge; dynamic Context; ontology modelling methodology*

I. INTRODUCTION

In recent years, access to information has been widely available with the tremendous advancement in networking and smart device technologies. Users are easily able to access dynamic information anywhere, at any time with the use of their smart devices.

Accessing colossal amount of information with these devices is, however, prone to resource limitations, thus prompting the necessity for information personalisation from the pool of widely available information sources. Context-aware systems (CAS) are considered to be robust for accomplishing personalised services [1]. CAS allows software agents to autonomously relate to the users by integrating the user's situational conditions with their surrounding information, thus facilitating the personalisation of resource and information as required by the user.

Although several contexts-aware modelling approaches have been considered by different system developers,

ontology-based models (a concept based on Semantic Web technology) have proven to be most suitable for representing context-aware systems [2]. Within the context of Semantic Web, not only does ontology allow extensive knowledge expressivity through facts representation, i.e., concept definitions with their respective relationships, but it also permits reasoning over the defined facts through the use of their meta-data to generate new and interesting facts.

Despite the benefits associated with Semantic Web technology, such as knowledge-sharing and inference are undeniable, the increasing need for user applications that meet the user's situational needs has necessitated the shift from static Web application development to temporal Web applications. Therefore, Semantic Web technologies must also cater for the requirements of today's applications of processing temporal personalized information. For example, it is not uncommon nowadays, that enterprise applications ranging from health, transport, banking etc., heavily rely on analysing publicly published knowledge in conjunction with the user's continuously changing situational conditions in order to optimise business decisions in the provision of services. Therefore, the supporting technologies for such enterprise applications should not only satisfy the requirements of automation, transparency, Knowledge expressivity and sharing, but also capable of accomplishing temporal knowledge reasoning and interdependent knowledge reasoning. While applications relying on Semantic Web technologies will easily satisfy most of these modern-day application demanded requirements, the aspects of the temporal and interdependent knowledge reasoning is not readily achievable with the current standards and tools of Semantic Web technology.

Adapting the Semantic Web technologies to satisfy temporal and interdependent reasoning requirements without intrusively obstructing the operational dynamics of the technology framework necessitates the alteration of the conventional ontology modelling approaches. Therefore, a robust ontology modelling methodology is required in order to maintain output consistency in attaining temporal and interdependent reasoning by the system. We hereby present an ontology modelling methodology for Context-Aware for Temporal and Interdependent Systems (CATIS). The methodology presents a robust approach of ontology development for systems utilising the Semantic Web. The

methodology is aimed at assisting ontology developers in defining ontology for modern-day sophisticated systems such as context-aware systems with features that include temporal and interdependent knowledge reasoning. The following sections of the paper include section 2, which discusses temporal and interdependent application example by illustrating the application requirements, section 3 that illustrates temporal and interdependent knowledge definitions in preparation for formal facts representation (ontology modelling), section 4 that presents CATIS methodology for modelling temporal and interdependent systems, and sections 5 and 6, which present the related ontology modelling methodologies and conclusions respectively.

II. TEMPORAL AND INTERDEPENDENT APPLICATION EXAMPLE

As mentioned in section one, the advancement in smart device and the wide availability of colossal amount of information has driven the evolvement of present day applications into satisfying temporal and interdependent reasoning requirements so as to meet today's desire of human day-to-day activities. We take into account a context-aware service recommendation system that recommends services to nomadic users as an example of present-day application with temporal and interdependent functionalities.

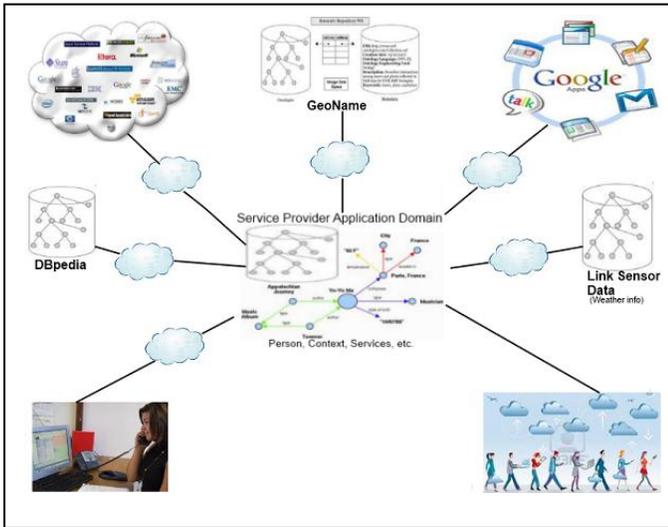


Figure 1. Example of Temporal Interdependent Context-Aware Application.

In order to advise on suitable services and their availabilities in such system as shown in figure 1, the recommender system will not only have to consider the user's temporal condition in making decision about recommended service(s) to users, but also have to extract and

make reasoning over extracted related facts or resources from the multifaceted domains available to the system. In achieving such goal, the system will need to fulfil the following criteria:

A. Automation

Context identification and reasoning, especially, in a large scale knowledgebase necessitates process automation. The system's ability to automatically identify contextual data and make decisions about such data is critical to making the system cope with user ever-changing situations in relation to available services. The importance of functionalities such as automation cannot be undermined in developing a system that will especially, handle scenarios where the change of one context determines the behaviour or derivation of another. We consider the level automation adopted by various context modelling approaches in terms of context discovery, context resolution, and method of execution and delivery etc. It is expected that the integral components and subcomponents of a system that will handle such complex scenarios should without obtrusiveness, relate with optimum level of communication or understanding.

B. Knowledge Expressivity and Sharing

In a distributed environment where integral components automatically relate, a high level of expressiveness is important in order to aid thorough understanding amongst these components. This level of expressiveness can be in terms of knowledge representation i.e., standard of knowledge description, which will in-turn promotes effective information exchange amongst the system components. The sharing of historical data within integral system component is crucial for the derivation of newer knowledge from existing one in dynamic environments especially, with large scale systems.

C. Transparency

In order to maintain system continuity and standards, extending exiting models is not an uncommon practice of system development. For more complicated scenarios such as the handling of knowledge interdependencies, developers and knowledge engineers in most cases are obliged to reuse, or extend existing context models to suit the requirements of the future system. To successfully achieve such extendibility, the transparency of the existing context models (i.e., the level of accessibility and adaptability of the model composition such as the processes of context discovery, matchmaking of contextual data and context execution) is crucial to promoting reusability and extendibility of these models for the development of dynamic context-aware systems that are capable of handling the complexity of dynamic contexts.

D. Temporal Knowledge Reasoning

Service recommendation in-line with the user's contextual information and processing such information,

which can include updating temporal contextual information such as user's profile will automatically necessitate the pre-computation of the user's current information. Pre-computing the user's information such as information about user's current location will be required for the determining user's profile category to determine appropriate service(s) for the user. Thus, designing a mechanism to reason over this ever-changing information on the fly is not only a necessity but also a requirement.

E. Reasoning of interdependent knowledge

Taking into consideration the aspect contextual data reasoning on the fly, the reasoning outputs can be interdependent of one another to derive most adequate results. Therefore the interdependent nature of this contextual information for multi-faceted domain scenario will not only required an intelligent system that discovers interdependent knowledge, but also perform reasoning over such knowledge by averting knowledge mismatching (i.e., contexts and services mismatch in this example).

By taking cognisance of the complexity and the aforementioned requirements of recent time applications, the utilisation of Semantic Web tools for the realisation of such complex scenario requires a clear methodology. Such methodology will help in maintaining consistency in the developmental process of this application class. We define in the following section the CATIS ontology development methodology to help achieve the fulfilment of such complex scenario.

III. ONTOLOGY MODELLING OF TEMPORAL INTERDEPENDENT APPLICATIONS

Semantic Web technologies have no doubt demonstrated their suitability for the design of multi-domain applications. However, the handling of temporal and interdependent knowledge has been a challenging aspect of the technologies due to the inherent limitations of the underlying format that the technology relies on.

Highlights from literature in the field of context-aware system management reflected that emphasis had been put on issues such as context modelling, context reasoning, knowledge sharing, and privacy protection etc., by employing several existing methodologies. However, in most cases, the limitations of the adopted methodologies are pragmatically being carried on to the proposed context management solutions. Hence, we aimed to pursue a logical vision by contributing toward the efforts of managing dynamic interdependent knowledge from the point of discovery, reasoning and retrieval as patent as possible. Although our approach to modelling adopted some techniques of the existing modelling methodologies, we

cautiously do not allow the limitations of these existing methodologies affect our proposed solutions. The methodology aimed at encouraging participants (users, intelligent agents and developers) of dynamic interdependent context-aware applications to better concentrate on the high level details of their developments and worry less about underneath technological details.

Although ontology modelling methodology for CATIS is applicable to other context application modelling, it majorly targets temporal interdependent information modelling. Therefore, enabling the current Semantic Web engine to cope with the processing of information with temporal and interdependent features requires additional support that supplements the reasoning functionality in order to complement the technology limitations. Such support can be presented as a framework, which will then focus on temporal and interdependent aspect of the presented application data.

In order to effectively construct a suitable supplementary framework that supports the Semantic Web engine in the handling of the described application class, a robust ontology modelling methodology is required, which should be structured enough to accommodate the semantic representation and reasoning of temporal and interdependent knowledge.

In constructing an ontology that satisfies the temporal and interdependent requirements of the aforementioned application class, we considered temporal representation techniques from a recent temporal representation approach [3]. The modelling approach uses logic-based approach for representing validity of time in RDF and OWL. In doing so, we logically distinguish static and dynamic concepts by annotating dynamic concepts with the *Validity* attribute at the modelling stage for efficient handling and validation of the temporal and dependency aspects of the contextual data.

As we have described in [2], a substantial challenge to be considered in the modelling of temporal event with Semantic Web tools is the definition of relationship between event times such as beginning, end, continuous, instantaneous. This challenge is solely due to the use of binary predicates to define the technology formats (OWL and RDF amongst others), which make provision for only first order semantic syntax[4]. The same limitation impacts the efficient handling of interdependent data. In order to efficiently model information with temporal and interdependent state, we define a new methodology in the following section.

IV. ONTOLOGY DEVELOPMENT METHODOLOGY FOR CONTEXT-AWARE, TEMPORAL AND INTERDEPENDENT SYSTEMS

Our proposed methodology adopts aspects of various existing methodologies in conjunction with new techniques for the development of ontologies for Context-Aware Temporal Information System (CATIS). We further segregate our ontology development methodology into five major phases as below:

A. *Defining Motivating Scenario*

Like many other development methodologies, identifying the system requirements, which is usually drawn from a storyboard problems or application examples, is the starting point of development. The problem is thoroughly analysed at this stage to precede the knowledge acquisition process in the next phase. Therefore, requirement analysis for the motivating scenario will be performed at this stage of the methodology. For instance, in analysing (identifying system requirements and tools) the application example shown in section 2, indicated temporal and interdependent requirements as core in fulfilling the modelling of such system.

B. *Tools Evaluation for Knowledge Acquisition and Generic Domain Identification*

This phase involves completing the following processes:

- 1) Knowledge elicitation and identification of Key generic domains in-line with the proposed systems
- 2) Evaluation and adoption of tools and languages for development.
- 3) Concept identification and conceptualisation of generic domains
- 4) Derive generic competency questions for upper-level domains.

Following the motivating scenario is the knowledge acquisition and definition phase. This phase of the methodology involves the knowledge elicitation, declaration of concepts and identification of generic domains of the system, relevant tools and languages for ontology development. The knowledge elicitation stage helps with the identification of the systems concepts and any knowledge elicitation technique can be adopted. Knowledge elicitation techniques as elucidated in [5] vary from concept-mapping, interviews (structured, unstructured, semi-structured) and reclassification amongst others. Consequently, the justification of the adopted tool and language can further be enhanced with sufficient understanding of the knowledge representation techniques, which is extensively analysed in [1].

Following, the identified generic domains represent system general domains from which upper-level and

lower-level (specific) domains are derived. The identified generic domains is conceptualised at this stage of development and can be verified using the derived generic competency questions, which is derived for the complete systems ontology. These generated competency questions are to be used in checking provision of solution to the entire system ontology at the later stage.

C. *Domain Knowledge Simplification*

This phase subsequently follows the competency question derivation stage of the knowledge acquisition phase. This is simplified as follows:

- 1) Identify specific domains (upper and lower-level domains) from generic domains and derive concepts for identified individual domains.
- 2) Define specific competency questions for lower-level domains.
- 3) Conceptualisation and formalisation of concepts, relationships and sub-domains of lower-level domains
- 4) Perform completeness theorem by evaluating specific competency questions against formalised concepts of the lower-level domains.
- 5) Repeat the step one (C1) if there are unanswered competency questions.

An important part of the CATIS development methodology is the process of domain identification. A survey of the existing methodologies indicated that none of these methodologies gave a clearer technique of identifying domains for system ontology developments. Unlike the development methodologies of other engineering exercises such as software engineering where, for instance, the process of identifying or describing classes, object and methods etc., are clearly defined; there have been no clear and define measures to ascertain the correctness of the identified domains in the ontology engineering process of system development. Thus, this is obviously causing the lack of uniformity in the process of the domain identification for different application class. While we identified domain concepts and defined the domains' scope using an iterative knowledge elicitation process, we defined a procedure for understanding how upper-level and low-level domains are to be classified from the use-case elicited knowledge. Since the low-level domains are the subsequent inheritance or dependants of the upper level domains that operates within the jurisdiction of the upper-level domain they have been identified with, the focus will then be a major task of identifying the upper level domains.

We propose a subsumption rule-base with the use of situational and action pair approach in setting a marker for identifying domain levels as below in figure 2.

```

if
  <domain operates within the scope of a single
  domain>
then
  <set as lower-level domain>
else if
  <domain operates within the scope of multiple upper-
  level domains>
then
  <set domain as upper level domain>
else
  <do not consider as active domain>

```

Figure 2. Subsumption rule for domain identification

For instance, concept declaration for the aforementioned application example in section 2 includes Context, Entity, Profile, Services, Event, Location, GPS, GeoArea amongst others. Domain categorisation of these concepts is achieved by subsumption rule as shown in figure 2 below.

The domain identification rule is when a concept subsumes other or fellow concepts, or a concept that stands alone without being subsumed will be termed to be a domain. For example, *Location* concept will subsume the *GPS* and *GeoArea* concepts as these concepts cannot stand-alone and only rely on the *Location* concepts when take into account the requirements from the system storyboard. Therefore, concepts such as *Context*, *Entity* and *Service* become domains based on the concepts they subsumed. Expanding further on subsumption rules, the *Profile* and *Device* domains will only operate within the jurisdiction of *Entity* domain; therefore, such two domains will be classified as lower-level domains in the *Entity* domain (upper-level). The *Location* domain on the other hand, connects multiple upper-level domains, including the *Entity*, *Services*, and even *Context* domains, thus, *Location* is classified as an upper-level domain. The upper-level domain for CATIS ontology is schematically represented in the figure 2 below.

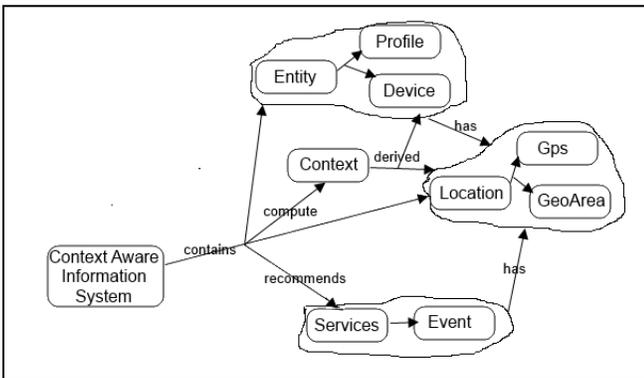


Figure 3. Derived Upper-level Domains Using CATIS Methodology.

The subsumption rule allows concepts to be subsumed as domains and upper level domains to subsume the lower level domain until the all upper level domains are finally identified. The concept behind the subsumption rule-based technique for domain identification is to create a standard as well as uniformity in the process of identifying domains for ontology development. A clear identification of the upper level domains gives a clearer insight for the derivation of the generic competency questions.

D. Integration of Upper-level Domains

Integration upper-level domain involves the following processes:

- Definition and formalisation of generic domains relationships
- Perform completeness theorem on generic domains by evaluating generic competency questions in 2 above (Knowledge acquisition and Generic domain identification) against the formalised concepts.
- Repeat step three of B. (B3) if there are any unanswered questions.

The integration of the upper-level domain is simply achieved by defining top-level relationships for the identify of upper-level domains. Subsequently, completeness theorem is used to evaluate the defined generic competency questions against the developed ontology (entire system Ontology). For instance, the *Services* and the *Location* domains, which are both upper-level domains are integrated using the *renderedService* top-level relationship. This top-level relationship can then be declared with any required restrictions, which are automatically transferred to the low-level relationships and domain members.

E. Defining Motivating Scenario

Finally, the process of development needs to be adequately documented for future maintenance of the developed ontology. We believe the documentation phase of ontology development should detail the major generic domain (upper level domains), lower-level domains, key domains integration relationships, generic and specific competency questions as shown so far. One of the distinctive features of our methodology is the ability to accommodate the unforeseen circumstance, where the process of accomplishing solution to a certain aspect of an enterprise model ontology is subject to change and cannot be guaranteed (i.e. propose plan might change) without affecting other aspect of the system ontologies

V. RELATED WORK

Gruninger and Fox [6] proposed a methodology based on first-order logic. This methodology uses the refined experiences of ontology development for enterprise

applications to formalise a process for ontology development for enterprises. It utilises a logical reasoning approach, and thus has the advantage of classical logic robustness to easily transform an informal scenario into computer models. Nevertheless, since our proposed system is required to cater for knowledge dependency reasoning and continuous concept extensibility, the methodology is not extensive enough to be adopted as its suitability for handling the required level of complexity of such systems remain questionable.

Similarly, the approach presented in the development of TOVE (Toronto Virtual Enterprise) [7] created ontology requirements as informal competency questions, which are derived from the initially defined motivating scenario. These questions then guide developers in ensuring the developed ontology can provide answers. Subsequently, the objects, attributes and their relationships are formally and axiomatically expressed using specific terms with their respective constraints in first order logic.

Also evaluated methodology is the approach presented by METHODOLOGY [8], a framework that supports the creation of new ontology from scratch and the reusing of already created ontology or even the reengineering of old ontology to meet newly defined specifications. METHODOLOGY development activities involve starting the processes with an initial specification definition. The implementation of the METHODOLOGY in the development of ontology clearly focuses on provision future ontology maintenance. Although many of these methodologies have quite a few similarities such as phases of development that include defining the motivation scenario, defining competency questions etc., however, each of them tend to be more distinctive in their focuses. While METHODOLOGY is directed at a goal of comprehensively resolving the surrounded issue of maintenance of ontology development life cycle, the approach in TOVE and few others focuses on utilising a formal technique.

We opted to combine multiple approaches by majorly extracting few techniques from all methodologies and extended them to suit a complex application class and many other complex systems. Other published ontology development methodologies we analysed in arriving at the CATIS methodology include IDEF5 methodology in KBSI[9], Ontolingua [10], SENSUS describe in [11] and Cys KB approach amongst others.

VI. CONCLUSIONS

Whilst ontology-based approach and tools has proven to be a robust means of modelling context-aware

knowledgebase systems, the development methodology play significant role in accomplishing the purpose of design for various application classes. That said, the challenge of employing suitable methodology for ontology engineering process has become an inevitable hurdle for developers. Although various ontology development methodologies have been proposed for enterprise ontology modelling, the majority are still at the infancy stage when compared to other engineering processes such as software engineering or knowledge engineering. Employing an approach of identifying temporal knowledge at the modelling stage permits the Framework to automatically perform a non-intrusive reasoning about temporal knowledge. Our future work involves extending the CATIS methodology for the development of a supplementary framework, which will assist the standard semantic web tools in handling the reasoning of temporal and interdependent knowledge.

REFERENCES

- [1] [1] S. O. Isiaq, "AN INTELLIGENT MULTIPHASE FRAMEWORK FOR TEMPORAL AND INTERDEPENDENT KNOWLEDGE REASONING IN THE SEMANTIC WEB," September, 2013.
- [2] [2] S. O. Isiaq and T. Osman, "Multi-phase reasoning model for temporal semantic knowledgebase," in *Proceedings of the International Workshop on Intelligent Exploration of Semantic Data (IESD)*, Galway City, Ireland, October 9, 2012, 2012, .
- [3] [3] B. Motik, "Representing and querying validity time in RDF and OWL: A logic-based approach," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 12, pp. 3-21, 2012.
- [4] [4] C. Welty, R. Fikes and S. Makarios, "A reusable ontology for fluents in OWL," *Frontiers in Artificial Intelligence and Applications*, vol. 150, pp. 226, 2006.
- [5] [5] K. G. Coffman and A. M. Odlyzko, "The size and growth rate of the Internet," *First Monday*, vol. 3, pp. 1-25, 1998.
- [6] [6] M. Gruninger and M. S. Fox, "Methodology for the design and evaluation of ontologies," in *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI*, 1995, .
- [7] [7] M. S. Fox, J. F. Chionglo and F. G. Fadel, "A common-sense model of the enterprise," in *Proceedings of the 2nd Industrial Engineering Research Conference*, 1993, pp. 425-429.
- [8] [8] A. Gómez-Pérez, "Ontological engineering: A state of the art," *Expert Update: Knowledge Based Systems and Applied Artificial Intelligence*, vol. 2, pp. 33-43, 1999.
- [9] [9] P. C. Benjamin, C. Menzel, R. Mayer, F. Fillion, M. Futrell, P. de Witte and M. Lingineni, "Idef5 method report," *Knowledge Based Systems, Inc*, 1994.
- [10] [10] T. R. Gruber, *Ontolingua: A Mechanism to Support Portable Ontologies*. Stanford University, Knowledge Systems Laboratory, 1992.
- [11] [11] B. Swartout, R. Patil, K. Knight and T. Russ, "Toward distributed use of large-scale ontologies," in *Proc. of the Tenth Workshop on Knowledge Acquisition for Knowledge-Based Systems*, 1996, .