Smart Space System Interoperability

Abdur Rakib

Department of Computer Science and Creative Technologies
The University of the West of England, Bristol, United Kingdom
Rakib.Abdur@uwe.ac.uk

Abstract. This paper presents our research approach which uses and integrates the terminologies and inference mechanism towards the development of functionally correct systems for smart spaces, considering the recent advances in the area of ontology-based modelling, agent-based technology, formal verification using model checking, and Android application development techniques. The overall goal is to arrive at a point where it is possible to achieve computer-supported knowledge exchange between technical/non-technical stakeholders in a distributed fashion for informed decision-making, for example, in eHealth systems.

Key words: Ontology, Interoperability, Resource-bounded agents, Rule-based reasoning

1 Introduction

In the health care domain, semantic interoperability plays an important role, especially when the action and/or decision making process requires sharing of information between the participant entities in smart spaces. In such an environment, a good knowledge representation technique helps to coordinate and interact between the agents. Knowledge representation is a method by which a knowledge engineer can model the facts and relationships of the domain knowledge, and it is of major importance in knowledge-based systems. The overall aim of this method is to design computer systems that reason about human-understandable and machine-interpretable representation of real-world domains [1]. A knowledge-based system creates a computational model of a given domain of interest using knowledge representation techniques to support human decision making and solving various problems. It is often designed for a certain type of knowledge representation based on inference rules. In the literature, several languages have been proposed for knowledge representation, each with syntax and semantics designed for a particular application [1, 2, 3]. However, there does not exist any single general formalism suitable to represent knowledge for all purposes [4]. Although the existing formalisms vary in their approach, they mostly share a common goal that describes the concepts and their relationships. Amongst others, the ontology-based approach is one of the most popular approaches used in the field of knowledge-representation and reasoning [5]. It has been realised that ontologies have become important tools in developing knowledge-based systems which provide a formal conceptual and computational model of a particular domain of interest, specifically in biomedicine, health care and biomedical research [6, 7]. This paper provides an overview of our research focussing on ontology-based knowledge representation and its use in resource-bounded multi-agent reasoning systems. The basic idea is to formalise and capture the agreed domain knowledge, describe the shared environment, define shared vocabulary for facilitating knowledge communication between the agents, to reason and process the shared information, and eventually provide a solution to informed decision-making in eHealth systems.

The rest of the paper is structured as follows. In Section 2, we briefly review background concepts. In Section 3, we discuss rationale for using multi-agent architecture and ontologies. In Section 4, we present our system development framework, and conclude in Section 5.

2 Background Concepts

This section presents some background on ontologies within computing, and ontology-driven multi-agent reasoning framework.

2.1 Ontologies and Description Logic

The mostly cited definition of an ontology is: *"an ontology is a formal explicit specification of a shared conceptualization"* [5]. In the context of multi-agent systems, this explicit specification could be interpreted as a common conceptualisation for knowledge sharing among distributed agents, which prevents ambiguity when interpreting messages from agents. Ontologies have been used widely to address the semantic interoperability and information sharing in distributed computing systems [5, 8, 9]. In the Semantic Web community various ontology representation language profiles are widely used for different purposes [10], and various tools exist to create and manage ontologies, including the Protégé ontology editor and knowledge acquisition system [11]. The Web Ontology Language (OWL) [10] is the recommended standard ontology language for the Semantic Web, and its different versions and profiles are build upon RDF and RDF schema (RDFS) to address the limitations of RDF and RDFS [12]. OWL axioms are formed by combining vocabulary entities using language constructors for axioms and other expressions, entity types are Individual (individual object), Class (sets of objects), and Property (binary relationships between objects). OWL has three sub-languages OWL Lite, OWL DL and OWL Full, and the tradeoff is between expressibility and reasoning efficiency. The W3C declared two different standardizations for OWL, namely OWL 1 and OWL 2. Both OWL 1 and OWL 2 are based on description logic, a decidable fragment of first order logic. OWL 2 has three sub-languages known as profiles, namely, OWL 2 EL, OWL 2 QL, and OWL 2 RL; each of them is useful in different application scenarios. The expressive power of OWL is strictly limited to certain tree structure-like axioms. For instance, a simple rule: *hasStructuredTestResult(?p,?r), indicatesDisease(?r,?d)* $\rightarrow$ *suffersFrom(?p,?d)* can not be modelled using OWL axioms [13]. This rule involves the properties *hasStructuredTestResult*, *indicatesDisease*, and *suffersFrom*, and the rule specifies the combination of the first two properties implies the third one. Function-free Horn clause rules can remove such restrictions while being decidable but they are restricted to universal quantification and no negation. A combination of OWL 2 with rules offers a more expressive formalism for building Semantic Web applications [14]. Several proposals have been made to combine rules with ontologies. We use one of them, the SWRL that extends OWL 2 by adding new axioms, namely Horn clause rules. In our work, we use the OWL 2 RL and SWRL [15] languages for defining ontologies and rules. An OWL 2 RL ontology can be translated into a set of rules and it is suitable for the design and development of rule-based reasoning systems [14].

2.2 Ontology-design

In the literature various approaches exist in designing ontologies for a domain of interest [16, 17]. In the bottom-up approach, ontology for smaller parts are constructed first, then using high-level abstract classes the desired ontology is developed. In this approach, ontology development starts with the leaves of the hierarchy that defines the most specific classes first, and subsequently groups these classes into more general concepts. For example, we can start by defining classes for *Nurse* and *Doctor*, then create a common superclass for these two classes as *Formal* in turn is a subclass of *CareGiver* and so on. In contrast to the bottom-up approach, a top-down approach designs the upper classes first and then develops the small parts of the hierarchy.

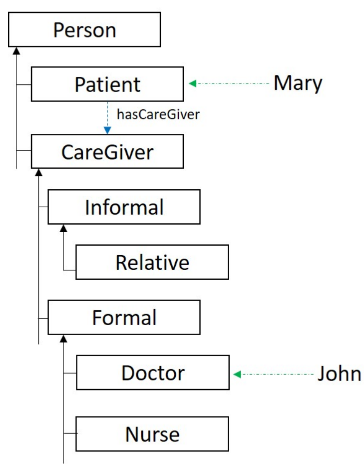

Fig. 1: An ontology class hierarchy and object taxonomy

While developing ontologies for a given domain, e.g., healthcare domain, we start with the requirement analysis phase where concepts, attributes, relationships and axioms are identified. Since health care is a complex domain and it is always hard to design it as a single ontology, we can create ontologies considering multiple and/or hybrid approaches [9]. After finding appropriate concepts, taxonomy formation is applied to provide a structure for the ontology for human understanding and the integration of other ontologies. Appropriate relationships then indicate the interaction among the concepts in the domain. Fig. 1 depicts a simple ontology class hierarchy and object taxonomy.

2.3 Ontology-Driven Multi-Agent System

In artificial intelligence (AI), various definitions exist for software agent [18, 19]. Wooldridge and Jennings [18] define—*an agent as a piece of software that requires to be reactive, pro-active, and that is capable of autonomous action in its environment to meet its design objectives*. The term autonomous means that an agent encapsulates its behaviour and internal state. That is an agent itself has high degree of control over its own actions and behaviour. The set of actions which are available to be performed by an agent are called a behaviour. When a system is composed of multiple interacting agents it is called a multi-agent system (MAS). In an MAS, agents are typically communicate via message passing and co-operate with other agents in order to achieve common goals. In our research, we address software reasoning agents which are capable of reasoning about their behaviour and interactions. That is our agents are primarily viewed as doing some kind of inference over a knowledge base (KB), e.g., using forward chaining rules (an agent starts with the initial facts and derives new facts whenever a rule matches with the current working memory facts). We model a multi-agent system consisting of n_{Ag} (≥ 1) individual *agents* $A_g = \{1, 2, \ldots, n_{Ag}\}$. Each agent $i \in A_g$ has a program, consisting of Horn clause rules of the form $P_1, P_2, \ldots, P_n \rightarrow P$ (derived from OWL 2 RL and SWRL), and a working memory, which contains ground atomic facts representing the initial state of the system. In our model, agents share a common ontology (or multiple ontologies) and communication mechanism. To model communication between agents, we assume that agents have two special communication primitives $Ask(i, j, P)$ and $Tell(i, j, P)$ in their language, where i and j are agents and P is a concept/role not containing an Ask or a $Tell$. $Ask(i, j, P)$ means 'i asks j whether P is the case' and $Tell(i, j, P)$ means 'i tells j that P' ($i \neq j$). Note that OWL 2 is limited to unary and binary predicates and it is function-free. Therefore, in the Protégé editor all the arguments of Ask and $Tell$ are represented using constant symbols. Using a Protégé plugin [20] we can extract Horn-clause rules from ontologies. The extracted rules are then used to design our rule-based agents.

3 Rationale for using Multi-Agent Architecture and Ontologies

In many circumstances building centralized systems are quite impractical or undesirable such as ubiquitous systems, where MAS technology appears to be a primary choice in producing distributed information systems. The aim of such systems is modularity, scalability, flexibility, robustness and distributed computing. In particular, MAS architecture could logically be divided in standalone agents offering certain services, capable of interacting with other agents. In the MAS community, there exists a considerable body of literature on modelling multi-agent systems [18, 21]. However, all these works are based on the classical approach of knowledge representation, they do not model resources such as time, space and communication restrictions on the agents ability to derive consequences of its beliefs. In contrast, our approach uses syntactic belief to avoid logical omniscience problem [22]. This is because the reasoning process of an agent requires resources in terms of time, space and perhaps communication in any implemented agent system. The use of ontology-driven rules provides a more natural way to think about and model real world rules. In addition, existing tools, including Protégé, support the design of OWL 2 RL and SWRL based ontologies, making it easier to model rule-based agents using semantic rules. Furthermore, the communication between distributed agents requires a common understanding of the exchanged knowledge. Ontologies play a crucial role for domain knowledge modelling, interaction specifications, and agent's behavioural aspect representation.

4 System Development Framework

In our system development framework, we apply the semantic web technologies namely OWL 2 RL and SWRL to design our agents. Fig. 2 shows our overall system development steps. The first task involves identifying the application domain and its boundaries for representation. Among others, Gruninger and Fox [23] have discussed a methodology for the design and evaluation of ontologies using the competency questions for extracting ontological concepts and axioms. For example, suppose we want to describe a domain named smart clinic (patient centric); we need to consider important parts including for example, *possible locations*, *furniture equipment*, *technical equipment*, and *physical devices* with which people can interact. By observing the scenario, some questions can be raised into the system designer's mind, for example, *who are the actors?*, *what role(s) does an actor or a group of actors play in the clinic?*, *how many rooms are there in the clinic?*, *what is the location of blood test unit?*, *which devices are located at what places and how they interact with each other?*, and so on. Based on these kind of questions we can identify the major concepts that can be used in building the ontology. Moreover, various existing ontologies can also be used while building the ontologies, for example, in healthcare domain a set of standard terms could be obtained from SNOMED-CT [24]. We also need to design SWRL rules where actionable knowledge is required as well as conditions which cannot be expressed as OWL axioms.

In order to design our ontology-driven rule-based agents, first we use the DLP framework [14] to translate the ontologies to a set of Horn clause rules [20]. In our framework agents are resource-bounded in terms of time, space, and communication [22]. A rule-based agent consists of a set of rules and working memory facts, it can update its working memory by performing one of the three possible actions:

Rule firing: firing a matching rule instance in the current state (possibly overwriting a fact from the previous state);

Comm: if agent i has an $Ask(i, j, P)$ (or a $Tell(i, j, P)$) in its current state, then agent j can copy it to its next state (possibly overwriting a fact from the previous state); and

Idle: which leaves its state unchanged.

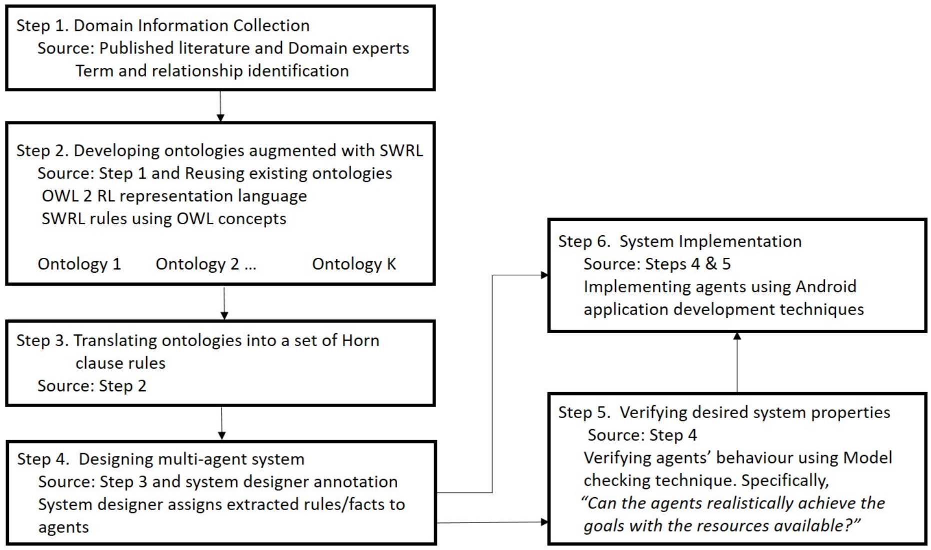

Fig. 2: System Development Steps

That is, each transition (result of an action) corresponds to a single execution step and takes an agent from one state to another. Once the translated rules are assigned to the agents and the system is designed as composed of multiple interacting agents, it can be encoded and verified using Model checking technique [22]. The verified design can then be implemented using Android application development techniques [25]. We chose the Google Android SDK to implement resource-bounded context-aware applications because the Google Android has the major user base. However, this choice does not restrict the research objective to Android only, and in the future we aim to develop a context-aware implementation framework that can be used to run application programs on multiple platforms seamlessly.

5 Conclusion and Future Work

In this paper, we discussed an ontology-driven approach towards the development of functionally correct systems for smart spaces. In our research work, ontology is used as a formal declarative knowledge-representation technique to specify the application domain of multi-agent systems. The advantages of using ontologies for representation include standardization of terms, knowledge sharing, support for automated reasoning, and to maintain consistency within a heterogeneous environment. From our experience, we would like to emphasize that OWL 2 RL augmented with SWRL is an excellent combination for the domain modelling and reasoning in complex systems using rule-based technique. The research on smart space systems, specifically on decision support still in its early stages, many challenges still remain in this area. For example, many real-world systems are inherently probabilistic. In the future, we would like to explore formal verification technique for modelling and analysis of systems for smart spaces with probabilistic behaviour.

Acknowledgements

I am grateful to have received the travel grant from the Western Norway University of Applied Sciences and UWE to attend the MMHS'18 workshop. The research highlighted in this paper was undertaken with funding received through the Ministry of Science, Technology and Innovation (MOSTI), Govt. of Malaysia as project number 01-02-12-SF0269. I am also indebted to my former students at the University of Nottingham Malaysia Campus, including Hafiz Mahfooz Ul Haque and Ijaz Uddin.

References

1. J. F. Sowa Knowledge representation: logical, philosophical and computational foundations, Brooks Cole Publishing Co. Pacific Grove, CA, USA, 2000
2. N. J. Nilsson. Principles of Artificial Intelligence.Springer-Verlag Berlin Heidelberg, 1982.
3. E. Turban. Expert Systems and Applied Artificial Intelligence. Macmillan Publishing Company; New York, 1992.
4. H. Reichgelt and F. van Harmelen. Criteria for Choosing Representation Languages and Control Regimes for Expert Systems. The Knowledge Engineering Review, Volume 1(4), pp. 2-17, 1984.
5. T.R. Gruber. A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 6(2):199221, 1993.
6. K. W. Fung and O. Bodenreider. Knowledge representation and ontologies. Richesson RL, Andrews JE (Eds.). Clinical Research Informatics, Health Informatics, Springer-Verlag London Limited Publishing, pp. 255-275, 2012.
7. O. Bodenreider. Biomedical ontologies in action: role in knowledge management, data integration and decision support. Yearb Med Inform, 67-79, 2008.
8. V. Kashyap and A. Sheth. Semantic heterogeneity in global information systems: The role of metadata, context and ontologies. In Cooperative information systems: Current trends and directions, pp. 139-178, 1998
9. H. Yang and W. Li. An ontology-based approach for data integration in regionally interoperable healthcare systems. In: 11th International Conference on Informatics and Semiotics in Organisations, pp. 93-96, 2009.
10. D. Calvanese et al. OWL 2 Web Ontology Language: Profiles (Second Edition). W3C Recommendation, 11 December 2012.
11. The Protégé Ontology Editor and Knowledge-base Framework (Version 4.1), 2011, http://protege.stanford.edu/
12. D. Brickley and R.V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema W3C Working Draft 30 April 2002
13. V. Kashyap, C. Bussler, and M. Moran. The Semantic Web– Semantics for Data and Services on the Web. Springer-Verlag Berlin Heidelberg, 2008.
14. B.N. Grosof et al. Description logic programs: Combining logic programs with description logic. In Proceedings of the 12th international conference on World Wide Web, pp 48-57, 2003.

15. I. Horrocks et al. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. Acknowledged W3C Submission, Standards Proposal Research Report: Version 0.6, 2004.
16. M. Uschold and M. Gruninger. Ontologies: Principles, Methods and Applications. Knowledge Engineering Review 11(2) (1996)
17. R. Klischewski Top down or bottom up? How to Establish a Common Ground for Semantic Interoperability within e-Government Communities. In Proceedings of 1st International Workshop on E-Government at ICAIL 2003, Bologna, Italy, 2003.
18. M. Wooldridge and N. R. Jennings. Intelligent Agents: Theory and Practice. In Knowledge Engineering Review, Vol 10(2), pp. 115-152, 1995.
19. J. M. Bradshaw. An introduction to software agents. MIT Press, 1997.
20. I. Uddin et al. Modeling and Reasoning about Preference-Based Context-Aware Agents over Heterogeneous Knowledge Sources. Mobile Networks and Applications, Volume 23, Issue 1, pp 1326, 2018.
21. A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In: Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, pp. 473484, 1991.
22. A. Rakib and H. M. U. Haque. A Logic for Context-Aware Non-monotonic Reasoning Agents. In: Gelbukh A., Espinoza F.C., Galicia-Haro S.N. (eds) Human-Inspired Computing and Its Applications. MICAI 2014. Lecture Notes in Computer Science, Vol 8856, pp. pp 453-471. Springe, 2014.
23. M. Gruninger and M.S. Fox. Methodology for the design and evaluation of ontologies. In Proceedings of Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada, pp. 110, 1995.
24. SNOMED-CT Systematized Nomenclature of Medicine-Clinical Terms. http://www.ihtsdo.org/snomed-ct/ (2007)
25. I. Uddin,A. Rakib, and H. M. U. Haque, A Framework for Implementing Formally Verified Resource-bounded Smart Space Systems. In Journal of Mobile Networks and Applications@Springer, 22 (2). pp. 289-304, 2016.