An Ontology-based Framework for Managing Semantic Interoperability Issues in e-Health

*Pronab Ganguly, **Subhagata Chattopadhyay, *N. Paramesh, **Pradeep Ray
*School of Computer Science and Engineering
**Asia-Pacific ubiquitous Healthcare Research Center (APuHC), School of Information Systems, Technology and Management, Australian School of Business
University of New South Wales
Sydney, NSW 2052
Australia
E-mails: pronabganguly@iinet.net.au

subhagata@unsw.edu.au

nparamesh@cse.unsw.edu.au

p.ray@unsw.edu.au

Abstract

Ontology-based approaches may be tried in e-health scenario to solve some of the interoperability problems that are non-trivial, for e.g. 'Anemometer' and 'Sphygmomanometer' are two different terms but represent the blood pressure monitor in Europe and India, respectively. The semantic interoperability issue arises when similar things are meant differently to the remotely sitting agents. Given such scenario, this paper aims to describe a framework that combines A) dialogue game with B) a decision support system (DSS) leading to resolution of such types of ontological mismatches. Under 'Dialogue game' argumentation mechanisms are used for resolving certain types of ontological mismatches. While, the DSS decides on the initiation, termination, structuring and sequencing the dialogues, as these may not be properly adjudged by the 'Dialogue game'. The framework is therefore context driven in nature, where concepts can be substituted with the similar kind of concepts (may not be accurate by searching and choosing equivalent concepts. Such framework therefore offers potential for dynamic modification of participating Ontologies and provides explanation in context of diet management in diabetes with time in a more matured way.

Key Words- Semantic interoperability; Framework; e-Health; Ontology; DSS; Dialogue

1. Introduction

Ontologies provide mechanisms for sharing of implicit and hidden knowledge between heterogeneous systems. This paper describes development and evaluation of a framework that combines A) dialogue game with B) a model of rational ontology mapping DSS to generate automatic dialogue between software agents leading to resolution of certain types of ontological mismatches. The steps towards the development of the proposed framework are as follows,

Step-1: Adopting rules of dialogue games
Step-2: Adopting a decision mechanism (DSS) for ontology mapping
Step-3: Generating a dialogue model based on decision mechanism
Step-4: Defining the locutions in the dialogue game (syntax)
Step-5: Defining decision mechanisms that will invoke different types of locutions at different points, and
Step-6: Defining transitional rules governing linkage between dialogue locutions and decision making mechanism.

The authors are working to resolve semantic interoperability issues using various ontology-based mapping techniques. While our study in [6] provides a broad guideline for such dialogue and the study in [7] extends the framework with context reasoning, this paper provides a combination of dialogue game and DSS (with knowledge base - having rule base and database). The lay out of the paper is as follows. Section 2 describes the framework methodology; Section 3 provides a real-world scenario for the realization of the proposed framework, and Section 4 Draws some relevant conclusions and proposes future research.

2. Framework methodology

This paper aims to describe a framework that combines A) dialogue game with B) a decision support system (DSS) leading to resolution of semantic ontological mismatches. It involves six steps, mentioned in the previous section and is described here as follows.

2.1 Adopting rules of dialogue games

For diabetes management, diet is an important issue along with medication and regular exercise. Dietician consults with the patient to formulate various meals for the patient. Calorie intake is dependent on body weight and other factors while patient's preference for food is a criterion. Let us consider a scenario where a dietician agent is engaged with a patient agent to formulate breakfast. For this particular patient (which is dependent on various factors such as height, body weight and preferences), it is recommended that the calorie intake should be as below:

- Vegetable - 25 Cal
- Milk products – 90 Cal
- Protein 35 - Cal
- Fruit 60 - Cal
- Starch - 80 Cal

Here the dietician and the patient are collaboratively formulating the patient's breakfast. Let dietician's ontology is represented by ontology D and patient's ontology is represented by Ontology P. For simplicity we will consider

- Part of the Ontologies to highlight the issue
- They overlap to a large extent

It may be noted that dietician's ontology includes very lean protein but as the patient is vegetarian in nature it does not have that concept but contains kidney bean cooked under vegetable. Cooked kidney bean (1/3 cup) is equivalent to 1 small egg from protein content perspective. The high level problem is to plan the breakfast for the patient and the associated problem is the mapping between the Ontologies. For the sake of simplicity, let us assume:

- The patient is vegetarian – does not take either meat or egg
- The dietician agent is the expert agent and carries the relevant features (such as type of the calories) of all the elements in the ontology. Dietician's ontology is represented by D and patient's ontology is represented by P.

2.2 Adopting a decision mechanism (DSS) for ontology mapping

Ontology mapping through agent dialogue is a relatively new concept. However it is becoming increasingly recognized as a critical element because of its dynamic nature. For the sake of simplicity we assume that:

- Dietician agents (Adi or D) are more expert than the patient agent (Apj or P) & Apj accepts Adi's advice.
- Again the OWL Web ontology Language describes a language for Ontologies. This language is equipped with a formal semantics described in OWL Web Ontology Semantics and Abstract Syntax and is based on description logic.
- Using the reasoning mechanisms in description logic, inferences about Ontologies can be made. Explanation of reasoning process is a topic of research interest – reasoning for Ontologies based on description logic include following rules [7]:

Rule 1: The subclass is inferred due to subclass being used in existential quantification
Rule 2: The subclass is inferred due to a sub property assertion
Rule 3: Axioms are used to assert additional necessary information about a class
Rule 4: Axioms provide an interaction between an existential quantification and a universal quantification
Rule 5: Interaction between complete and partial definitions plus a universal quantification allow an inference about role filler.
Rule 6: Interaction between an inverse relationship and domain and range constraints on a property
Rule 7: Domain restriction gives additional information which then allows inferring more specific type.
Rule 8: The universal quantification then allows inferring about the role filler.
Rule 9: Union is distributive in existential, intersection is not.
Rule 10: Intersection is distributive in universals, union is not.

For our mapping through dialogue between agents we will be using those rules. As illustration of all the rules through dialogue is beyond the scope of this paper, we will concentrate on the rules associated with our problem. In the given example, the dietician agent D advises patient P that for breakfast, the following is the recommended diet:

- Vegetable - 25 Cal
- Milk – 90 Cal
- Protein - 35 Cal

- Fruit - 60 Cal
- Starch - 80 Cal

The agent P does the following:

1. Searches his ontology all the concepts.
2. Identifies the missing concept i.e. Protein
3. Engages in dialogue with dietician agent D
4. The dialogues between them invoke various decision mechanisms.
5. At the end of the dialogue Agent D advises that P can accept a substitute element (i.e. cooked kidney bean) as an alternative.
6. P agrees.
7. The dialogue terminates.

2.3 Generating a dialogue model based DSS

The high level models of dialogue between the agents are presented below [2] [4]:

1. Open Dialogue: The dialogue commences.
2. Inform: The expert agent Adi informs the need for a mapping and provides the context for mapping i.e. related information about mapping.
3. Form Mapping Criteria: The agents Adi & Apj generate the mapping criterion. The inform phase might be used for clarification.
4. Assess Mapping Criterion: Agent Adi & Apj assess mapping criterion based on the reasoning model in sec 2. They might use inform phase for clarification.
5. Negotiate: Agents negotiate over preferred option.
6. Confirm: The agents confirm the preferred option.
7. Close Dialogue: The dialogue terminates.

2.4 Defining the locutions in the dialogue game (syntax)

The locutions are accompanied by two stores shared by the agents – Information store and Confirmation store. During exchange of information, entries are inserted into an agent's information store by a locution uttered by that agent. Similarly the Commitment store records the confirmations by corresponding agents. Locution rules specifies following element:

- Precondition for the locution
- Meaning of the locution
- Response to that locution
- Effect on Information store
- Effect on Commitment store

Let the participating agents be denoted by Ax1, Ax2. etc., where X ϵ {Adi, Apj} represents the role of the agent as dietician or patient. We assume that each agent Axi has an information store denoted by IS(Axi) that contains entries such as (Ayj, ā) where Ayj is the other participant and ā is an option offered or under consideration.. The commitment store is denoted by CS (Axi) and contains entries such as (Ayj, ā) where Ayj is the other participant and ā is an option offered or under consideration. Let concepts be denoted by α, β etc

The locutions and associated rules in this context are given below:

Open Dialogue phase:

L1: open_dialogue locution

Locution: open_dialogue (Axi, α) where X ϵ {Adi, Apj}

L2: enter_dialogue

Locution: open_dialogue (Axj, α) where X ϵ {Adi, Apj}

Inform Phase:

For the following locations, there will be a general precondition that the participating agent has already uttered either open_dialogue() or enter_dialogue. Hence this common precondition is not included in the following locutions. We will mention additional precondition specific to that locution.

L3: seek_info

Locution: seek_info (Axi, p) where X ϵ { Adi, Apj } where p is a proposition

L4: provide_info

Locution: provide_info(Axj, Z) where the elements of Z satisfy the constraint imposed by p.

Negotiate phase

L5: prefer

Locution: prefer (Axj, Z, K) for Axj where Z and K are two sets of options that will be included in provide_info locution..

L6: willing_to_accept

Locution: agree_to_accept (Axi, Z) where Z is a non empty set of options.

L7: refuse_to_accept

Locution: refuse_to_accept(Axj,K) where K is a set of options

Confirm phase:

L8 : agree_to_accept

Locution: agree_to_accept (Axj,K)

Close Dialogue phase:

L9: Withdraw_dialogue
Locution : withdraw_dialogue(Axi, α) for X ϵ { Adi, Apj }

2.5 Defining decision mechanisms that will invoke different types of locutions at different points

For automated dialogue, on top of syntactical rules, the agents must be equipped with decision making mechanism to generate relevant type of locution at the point of need. This mechanism is internal to the agent and is based on the rules reasoning in section 2. Thus each agent is equipped with its own mechanisms. These mechanisms will invoke particular locations at particular points in dialogue. Thus a generic function of the following mechanisms is to decide , if any, locution to utter taking the output of the corresponding mechanism or generate a null locution (do nothing). Before proceeding with individual mechanisms of each agent, let us describe some generic functions that will be part of every mechanism.

Do or Wait: A function that will determine if the decision mechanism will be implemented now or delayed for future.

Normalize: A function to classify the ontology with the help of an automated reasoner such as RACER to get rid of ambiguity and inconsistency.

Select Locution: A function to decide what locution will be uttered depending on the outcome of the associated decision mechanism.

The decision making mechanisms for patient agent P may include:

MP1: Recognize Need: Mechanism that enables P to identify the need for mapping a concept θ. The automated reasoner will be used for this purpose. The possible output for this mechanism there_is_need (θ) and there_is_no_need (θ)
MP2: Seek Mapping Information: Mechanism to determine the type of information to be sought from the Dietician agent regarding mapping of the concept. The possible outcome of this mechanism is seek_mapping_info (θ) or seek_no_mapping _info
MP3: Provide mapping Information: Mechanism to determine the type of information to be provided to the Dietician agent. The possible outcome of this mechanism is provide_mapping_*info*(θ) or provide_no_mapping_info.
MP4: Consider Option: Consider advice offered by the agent D. The possible outcome of this stage is accept (V (θ)) or reject (V (θ))
MP5: Consider Withdraw: A mechanism to consider withdrawal from the dialogue if it is not producing any effective outcome. The possible outcomes of these mechanisms are withdrawn (θ) or not_withdraw.

Similarly for dietician agent D, these mechanisms may include:

MD1: Recognize term of interest: A mechanism to determine on the concept needs to be mapped. Automated reasoner will be used for this purpose. The possible outcome of this mechanism wish_to_map(θ) wish_not_to_map(θ)
MD2: Seek Mapping Information: Mechanism to determine the type of information to be sought from the Patient agent regarding mapping of the concept. The possible outcome of this mechanism is seek_mapping_info (θ) or seek_no_mapping _info
MD3: Provide mapping Information: Mechanism to determine the type of information to be provided Dietician agent. The possible outcome of this mechanism is provide_mapping_*info*(θ) or provide_no_mapping_info.
MD4: Form Consideration set: A Mechanism to form criteria set with the reasoning rules spelt out in sec 2. The output of this mechanism is a non empty set V(θ) or seek_mapping _info(θ)
MD5: Select Options: A Mechanism for selecting the option as per reasoning rules. The possible outcomes for this mechanism are an empty set α or a non empty set V (θ)
MD6: Consider_withdraw: A mechanism to consider withdrawal from the dialogue if it is not producing any effective outcome. The possible outcomes of these mechanisms are withdrawn(θ) or not_withdraw.

Some of the major mechanisms have been summarized above. Various types of mismatch may need other types of mechanisms as well.

2.6 Defining transitional rules

As the agents are equipped with decision mechanism, the outcomes of these mechanisms need to be associated with utterances of the locutions and vice versa. The locutions uttered will cause the transitions between the states of the mechanism. In order to define the links, ordered 3 {Pxi, K,s} is used to denote the mechanism with number K and with an output s of participant PXI. Where a transition is invoked by or invokes a particular output of a mechanism

K this is denoted by the specific output s in the third place of the triple. Where no specific output is invoked, a "." is used in the third place. When transitions occur between mechanisms through locutions, they are denoted by relevant locution number from section. The transitions between decision mechanisms of the single agent are represented by unlabeled arrow. These transition rules might include:

Transition Rule 1

This rule indicates that a patient agent with a current need for mapping a concept A will initiate a dialogue by means of a locution L1 i.e open_dialogue in the case where such a dialogue is not already initiated or will enter such a dialogue by means of locution l2 i.e enter_dialogue in the case where it has already been initiated.
TR1: {P, MP1, there_is_need} > L1, L2 > {D, MD1,}
Transition Rule 2
TR2 {P, MD1, wish_to_map } > L2 > {D,MP2,.}
Transition Rule 3:
TR3 {P, MP2, seek_mapping_info} > L3 > {D, MD2,}
Transition Rule 4
TR4 {D, MD2, seek_mapping_info} > L3 > {P, MP3}
Transition Rule 5
TR5 {P, MP3, provide_mapping_info} > L4 > {D, MD4,.)
Transition Rule 6
TR6 {D, MD4,.} > {PDI, MD5,.}
Transition Rule 7
TR7 {D, MD5, select_option} > L5> {P, MP4,}
Transition Rule 8
TR8 {P, MP4, consider_option} > L6 > {D, MD6,}
Transition Rule 9
TR9 {D, MD6, withdraw} > {P, MP6,}
Transition Rule 10
TR10 {P, MP4, reject} > L7 > {D, MD4,.}
Transition Rule 11
The output of withdraw in MP6 of patient along with locution L9 will terminate the dialogue.
TR11 {P, MP6, withdraw} > L9 > {D,}

We have provided the main transition rules. Depending on the size and complexity of the associated Ontologies, other rules may be needed. This framework supports argumentation based automated dialogue. Though computational mechanism for automated negotiation does exist, they are not argumentation based. A close of our framework will reveal that it supports generation of dialogue automatically [4].
Figure 1 has shown the steps of the ontology-based framework that incorporates both the DSS and the dialogue games as follows.

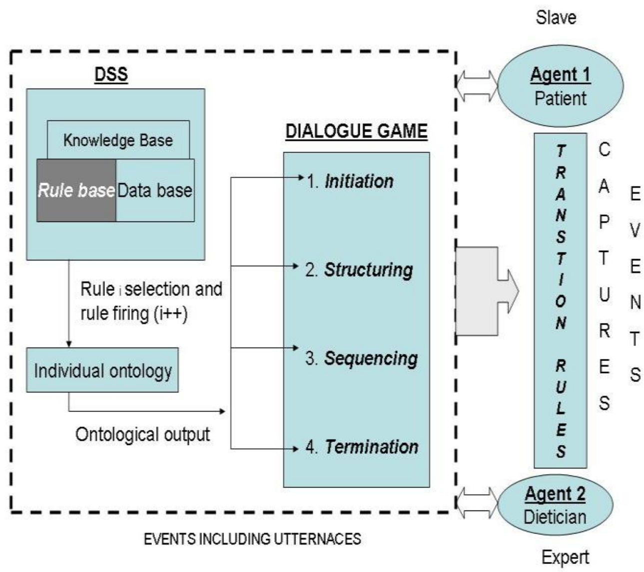

Figure 1. The proposed framework for ontological mismatch resolution.

3. Realization of the framework and proof of concept

In our diabetes management Ontologies, suppose the dietician has advised the patient to eggs for Protein. . But that concept is missing from patient's ontology. Thus the mapping dialogue starts as below:

Step 1: As per TR1, the patient agent opens dialogue.
Step 2: As per TR2, the dietician agent participates in the mapping.
Step 3: As per TR3, the patent agent seek mapping information about eggs.
Step 4: As per TR4, the dietician agent asks for existence of Very Lean Protein in patient's ontology. (Rule 1 & 2 in section 4)
Step 5: As per TR5, the patient agent answers negative.
Step 6: As per TR4, the dietician agent asks for existence of Food in patient's ontology.
Step 7: As per TR5, the patient answers positive.
Step 8: As per TR6, locates cooked bean as an alternative (Rule 3 & 4 of section 4) & calculates required amount .
Step 9: As per TR7, selects cooked bean as the viable option and provide that information to patient agent.
Step 10: As per TR8, patient agent considers the option and agrees.
Step 11: As per TR 9 & 11, the dialogue terminates.

Though the example is very simple in nature, it does highlight the essential features of the dialogue-based framework.

The Ontologies are built with ontology editor Protege. The **Protégé-OWL** editor enables users to build Ontologies for the *Semantic Web*, in particular in the W3C's Web Ontology Language (OWL). An OWL ontology may include descriptions of classes, properties and their instances. Given such an ontology, the OWL formal semantics specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but entailed by the semantics. These entailments may be based on a single document or multiple distributed documents that have been combined using defined OWL mechanisms.

4. Conclusions and future directions

We have drawn an outline of dialogue framework to resolve ontological mismatch. The mapping algorithm has been used for illustration purpose only. There are similar works for ontology negotiation mechanisms such as extended ONP (ontology negotiation protocol) [6] and game board rules [3]. A brief comparison among various ontological negotiation protocols are shown in the following table.

Features	Argumentation (SIS)	ONP	Game Board
Automatic generation of Dialogue	Explicit	Not Clear	Not Clear
Basis of Dialogue Model	Derived from Complex human Dialogue	Simplistic	Simplistic
Dynamic up gradation of Ontologies	Possible	Not Possible	Not possible.
Cooperation between agents	Very tight	Loosely coupled	Loosely coupled.

As a varieties of OWL Lite Ontologies are available in Semantic web environment, full implementation of our framework and associated benchmarking and validation will contribute to automated reasoning between agents. Further challenges include reasoning between various other versions of OWL type Ontologies.

References

[1] Beun R. J., van Eijk R. M (2004) 'Conceptual Discrepancies and Feedback in Human-Computer Interaction'. *Proc. of Dutch directions in HCI*, ACM Press, New York.

[2] Beun R. J., van Eijk R. M (2004) 'A Cooperative Dialogue Game for Resolving Ontological Discrepancies' In Dignum, F. (ed.) *Advances in Agent Communication*. volume 2922 of LNCS. Berlin: Springer.

[3] Beun R. J., van Eijk R. M (2004) 'Feedback in Human Computer Interaction: Resolving Ontological discrepancies'. In L. Breure, A. Dillon and H. van Oostendorp (eds.) *Creation, Use and Deployment of Digital Information*. London: Erlbaum, 2004.

[4] McBurney P., van Eijk R. M, Parsons S., Amgoud L (2003) 'A Dialogue Game Protocol for Agent Purchase Negotiations' *Autonomous Agents and Multi-Agent Systems* 7(3), pp. 235-273.

[5] Beun R.J. van Eijk R.M (2003) 'Dialogue Games for Ontological Commitment'. In J.S. Rosenschein, T. Sandholm, M. Wooldridge and M. Yakoo (eds.) *Proc. of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems* pp. 938 - 939, ACM press.

[6] Ganguly P., Parameswaran N., Ray P (2006) 'Inter-Agent Dialogue for Semantic Mapping between Domain Ontologies in Healthcare Informatics'. *Proc. of HealthCom 2006*, New Delhi.

[7] Orgun B., Dras M., Cassidy S. & Nayak A (2005). 'DASMAS – Dialogue based Automation of semantic Interoperability in Multi Agent System'. *Proc. of Australasian Ontology Workshop (AOW 2005).*

[8] Ganguly P., Parameswaran N., Ray P (2006) 'Semantic Interoperability of e-Health Systems using Dialog based Mapping of Ontologies'. Proc. of 1st Asian Semantic Web Conference (ASWC2006) Workshop on Semantic e-Science, Beijing.

[9] OWL Reasoning Examples. http://owl/man/ac/uk/2003/why/latest accessed on 02/02/2005