Chapter 4

Ontology-Based Negotiation of Dental Therapy Options

Mahsa Kiani, Michael Francis, Yassaman Zand-Moghaddam, Palash Verma

Abstract Negotiation is a decision making process through which several individuals, with differing points of view, attempt to interactively reach an agreement. It involves the discussion and evaluation of the problem as well as the consideration of alternatives. Therefore, a large number of possibilities should be evaluated by the decision makers, which is time-consuming and difficult (Chiu, Hung, Cheung and Leung). A negotiation support system can be used to facilitate interactions between different individuals. In this chapter, we propose using semantic web concepts such as ontologies and semantic reasoning to implement a system to facilitate negotiation. The target domain of this project is negotiation between dental experts over the treatment of wisdom teeth.

INTRODUCTION

This chapter proposes a negotiation facilitator system to detect conflicts and agreements in opinions of different individuals. System gives a list of agreements and disagreements between reports of different individuals. The inputs to the system are formal or semi-formal documents that represent the points of view of these individuals. These point-of-view reports would be annotated by a domain expert using predefined concepts from an ontology; therefore, each report can be mapped to several instances of the concepts in the ontology. Some concepts are shared explicitly between different individuals (global concepts) and could be considered directly in the negotiation process; however, other concepts (local concepts) should be interpreted into global concepts using rules. Negotiation is facilitated by performing reasoning on the instances of these shared concepts. Conflicts found through reasoning indicate disagreements in the points of view of the individuals, which can be output for the users to view. An intuitive representation of the discovered conflicts and agreements will be developed that will facilitate negotiation. The remainder of this chapter is organized as follows: At first, background

* University of New Brunswick
  mahsa.kiani@unb.ca
  michael.francis@unb.ca
  yassaman.zandmoghaddam@unb.ca
  palash.verma@unb.ca
researches related to negotiation are investigated; then, domain and also architecture of the proposed system are explored. In the next section, implementation issues including ontology, reasoner, and implementation tools are described. Finally, the process of the system for a specific example on the area of wisdom teeth treatment is investigated.

BACKGROUND

Negotiation is a process in which two or more parties try to reach an agreement or compromise by arguing (Jian and Wei 2009). Negotiation could be cooperative or competitive. In cooperative negotiation, the objectives of the participants do not conflict and the participants could have their own preferences on different issues. Therefore, it is possible for the two parties to obtain satisfactory results. In competitive negotiation, on the other hand, the objectives of both sides are conflicting or at least not compatible. The involved parties have different preferences on the same issues. Hence, one side loses some benefits when another side gains some (Guttman and Maes 1998).

Automated negotiation has been the focus of interest in different domains such as e-commerce, networking, and production planning (Jian, and Wei 2009; Zhu, Guan, Wang, Zhou, Liao 2005; Guttman, Maes 1998; Bejarano 2009), as it can requires less time and effort from the individuals involved. In multi-agent systems, software agents can negotiate with other software agents on behalf of their owners. To do this, some rules of encounter need to be defined between the negotiation participants. These rules, along with the criteria for reaching an agreement and agreement deal (in e-commerce), form the protocol for negotiation (Guttman, Maes 1998; Sierra, Faratin and Jennings 1997; Tamma, Wooldridge, Dickinson 2002; Kim, Cho, Yoon, Kwak, Seo 2005; Kraus 2001).

In one of the proposed approaches, a combination of Case Based Reasoning (CBR), agent reasoning, and a nearest-neighbour case matching method with commercial logic is used in order to automate negotiation (Zhu, Guan, Wang, Zhou, Liao 2005). Also, automatic negotiation could be modelled on fuzzy evaluation. This approach simplifies multi-issue negotiation and also resolves the problem of incomplete or unknown restrictions to negotiation (Jian and Wei 2009). In another approach, Fuzzy Repertory Table Technique has been used to capture the buyer’s role in a personalized, semi-autonomous, and automatic way (Castro-Schez, Castro, Zurita 2004). Genetic-based Algorithms (GA) have also been used to automate two kinds of negotiation called mutual negotiation and third party negotiation (Tsai, Fu, Chou 2005). Bayesian rules have been applied to update the environmental information in negotiation. In this approach, Q-learning algorithm of reinforcement learning is used to generate the negotiation proposals (Hue and Jing 2008). Fuzzy logic and fuzzy neural network have been employed to model the uncertainty of relations among the parameters of a negotiation procedure (Sakas, Vlachos, Simos 2007).

In our system, instead of agents, individuals want to reach an agreement to solve a specific problem and a shared ontology is used to map the local concepts of each individual to global concepts in order to facilitate negotiation. Therefore, individuals could use their local concepts while negotiating over a common problem. The purpose of the individuals participating in negotiation is solving a specific problem. The output of our system is a list of agreements and conflicts between those individuals. Using this output, individuals can negotiate without knowing local concepts of the other parties which could save time and effort in the negotiation process. This approach has been used to facilitate negotiation over various therapy options for the treatment of wisdom teeth.
MAIN FOCUS OF THE CHAPTER

Domain of the Project: Wisdom Tooth Treatment Options

The selected scope of the system is the treatment of wisdom teeth. Adult humans normally have 32 teeth evenly distributed over each quadrant. Each quadrant of 8 teeth consists of the following types of teeth: central incisor, lateral incisor, canine, first premolar, second premolar, first molar, second molar, and third molar. The third molar is commonly referred to as a wisdom tooth and may or may not erupt. In most adult humans these teeth erupt in their late teens or early twenties.

These teeth present potential problems when they are misaligned - they can position themselves horizontally, be angled toward or away from the second molars, or be angled inward or outward. Poor alignment of wisdom teeth can crowd or damage adjacent teeth, the jawbone, or nerves. Wisdom teeth that lean toward the second molars make those teeth more vulnerable to decay by entrapping plaque and debris. In addition, wisdom teeth can be entrapped completely within the soft tissue and/or the jawbone, or only partially break through, or erupt through the gum. Teeth that remain partially or completely entrapped within the soft tissue and/or the jawbone are termed "impacted". Wisdom teeth that are only partially erupted allow an opening for bacteria to enter around the tooth and cause an infection, which results in pain, swelling, jaw stiffness, and general illness. Partially erupted teeth are also more prone to tooth decay and gum disease because their hard-to-reach location and awkward positioning makes brushing and flossing difficult. In Figure 1, some possible positions of wisdom teeth that can lead to problems for the patient are illustrated.

![Figure 1. Some Possible Positions of Wisdom Teeth](image-url)

These teeth can be a valuable asset to the mouth when healthy and properly aligned; however, they are often misaligned and require removal. The decision whether to remove or keep a wisdom tooth has
to be made in day-to-day dental practice. However, this complicated judgment must be learned from either mentoring from other dentists, long practice, or evidence-based learning. The decision is not exclusively based on what the dentist thinks, but involves several factors. These factors could be classified into personal, general health, and dental health issues.

- Personal factors include emotional esthetic, financial, and related factors.
- General health issues affecting tooth treatments, such as drug allergy, HIV, diabetes, blood related problems, etc.
- Dental health issues such as crowding other teeth, sticking as it tries to emerge, infection (infected swelling), impaction, dry socket, damaging roots of other teeth, damaging important nerves such as lower jaw or ear, damaging other teeth and dental work, etc.

Possible treatment options for a wisdom tooth are:
1. Remove the tooth. The removal option is classified based on time and type of procedure. Removing the wisdom teeth could happen immediately (as an emergency measure), after a short period of time (necessary treatment but not an emergency; necessity of treatment of the infection by antibiotics before removing the tooth), or after a long period of time. Extraction can be performed either in the dentist’s office or in a hospital under general anesthesia. In some cases, the tooth decay is so severe that it cannot be removed as a whole and requires removal in small pieces.
2. Postpone the decision to a later time. As the jaw of girls before 11 and boys before 14 years old could grow, the problem would be solved automatically or the removal procedure would be easier because the tooth has grown.
3. Keep the tooth and do some minor treatments (to be used in bridging procedure).

Each treatment has some risks. For example, the risks and disadvantages of removing the wisdom teeth are pain and swelling in the gums and tooth socket, bleeding, difficulty with or pain from opening the jaw (trismus), damage to dental work, painful inflammation (dry socket), numbness in mouth and lips because of injury inflammation of nerves in the jaw, an opening into the sinus cavity, and small risk of death or other complications because of general anesthesia, etc. The risks and disadvantages of not removing the wisdom teeth could be impaction, infection, cavities and gum disease, affecting orthodontic, forming fluid sac, and permanent damage to nearby teeth jaw and bones, etc.
Architecture of the system

In figure 2, the architecture of the system is illustrated.

![Architecture of the system](image)

Figure 2. Architecture of the system

Implementation Issues

Ontology

For this system we designed and created an OWL ontology in Protégé 3.4.1. The ontology contains concepts of the domain in the form of classes and their relationships via properties, which are Object or Datatype properties. It provides the domain experts, who are in charge of annotating the reports, with a set of predefined concepts and relations. Then the domain experts will be able to create instances of those concepts, which are called individuals, based on the given reports.

Ontology Overview:

Classes: As it was mentioned earlier, classes in an ontology represent concepts in the domain. Therefore in this section when we use the term class it means that there is a concept for that class in the domain. In ontologies that are designed and created in Protégé all the classes are subclasses of the class Owl:Thing. In this project’s ontology we have 11 main classes which are subclasses of the class Owl:Thing and can also be divided into subclasses. However, the ontology can also be designed to have 8 main classes; the last main class, called Clinical Facts, is further subdivided into 4 subclasses which are Clinical Examination Results, Influential Factors, Radiograph Facts, and Anaesthesiologist Facts. From the main classes, we first describe the three most important concepts of the project’s domain, which are the Individuals, Symptoms, and Tooth classes in the ontology (Figure 3).

The Individuals class, as can be inferred from its name, represents all the individuals who participate in negotiation. Individuals class is divided into two subclasses Patient and Experts. Experts are Dentist(s), Family Doctor(s), Radiologist(s), and, Anaesthesiologist(s). Therefore the Experts class is divided into four subclasses: Dentist, Family Doctor, Radiologist, and Anesthesiologist (Figure 4).
The Symptoms class is the superclass of 18 subclasses which are representations of different symptoms in the domain like: pain, infection, bleeding, etc. Figure 5 shows some of these subclasses. All the subclasses of Symptoms are common or global concepts, vocabulary, between individuals. In other words, all the individuals may use those concepts in their reports, and they can be understood by each of the individuals.
Some of the subclasses of Symptoms, such as Caries and No Caries, are disjointed. This means an instance of class Symptoms cannot be a member of both Caries and No Caries.

Since the target domain of this system is negotiation over the treatment of wisdom teeth, two other main concepts of the domain are Tooth and Treatment. Therefore, Tooth and Treatment classes are defined in the ontology.

In our system, the Tooth class has no subclasses but has some relations with other classes like Region, which are going to be explained in the next section. The Treatment class consists of three subclasses, which are Minor Treatment, Removal, and Type of Procedure. Minor Treatment and Removal are disjointed. The Type of Procedure concept determines the type of procedure that is required or has been mentioned in the reports and its class has 7 subclasses. Figure 6 shows the Treatment subclasses. Subclasses Surgery and No Surgery, Completed Removal and Removal in Pieces, In Office and In Hospital are disjointed subclasses No Procedure Done, Surgery, Complete Removal and Removal in Pieces are also disjointed. As mentioned above, Type of Procedure has been considered as a class in our system; however, it could be designed as a property.
Each individual who participates in a negotiation has a report; therefore, Report is also a concept in the domain and its equivalent class in the ontology is Report. Furthermore, in addition to common concepts or vocabulary between experts which are defined as class Symptoms and its subclasses, experts also have their specialized concepts or vocabulary which may not be understood by other experts and can appear in their reports. In the ontology we design them as Clinical Examination Results, Influential Factors, Radiograph Facts, and Anaestheologist Facts classes, which are related to Dentist, Family Doctor, Radiologist, and Anaesthesiologist, respectively, via object properties. These classes may also have some subclasses. The last class of the ontology is Radiograph, which represents the radiograph concept. A radiograph can be observed by radiologists and has other properties like date, type as Panoramic or OPG, etc.

In the next section, properties of the ontology that are conceptualizations of relationships between concepts in the domain are described.

Properties: OWL Properties represent relationships between two instances of the classes. Instances of the classes are also called individuals (note that this is different from our predefined class Individuals in the ontology). There are two main types of properties, Object properties and Datatype properties. Object properties link an individual to an individual. Datatype properties link an individual to an XML Schema Datatype value or an rdf literal (Horridge, Knublauch, Rector, Stevens, and Wroe 2004). Each property has a domain and a range. Domain and range of an Object property are defined by classes. Domain of a Datatype property is also defined by class(s) but its range is an XML Schema Datatype value or an rdf literal. For Datatype properties allowed values can be determined. For further information about properties of OWL ontologies please refer to Horridge, Knublauch, Rector, Stevens, and Wroe 2004.

In the designed ontology, both types of properties exist. The total number of properties is 33, consisting of 15 Object properties and 18 Datatype properties. Table 1 presents some of the properties in the system.
Table 1. Some of the defined properties in the ontology

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Type</th>
<th>Domain</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>observesSymptoms</td>
<td>Object</td>
<td>Individuals</td>
<td>Symptoms</td>
</tr>
<tr>
<td>advisesTreatments</td>
<td>Object</td>
<td>Experts</td>
<td>Treatments</td>
</tr>
<tr>
<td>seesTooth</td>
<td>Object</td>
<td>Dentist</td>
<td>Tooth</td>
</tr>
<tr>
<td>hasInfluentialFacts</td>
<td>Object</td>
<td>Patient</td>
<td>Emotional, Financial, Health</td>
</tr>
<tr>
<td>seesRadiograph</td>
<td>Object</td>
<td>Radiologist</td>
<td>RadiographFacts</td>
</tr>
<tr>
<td>angleIs</td>
<td>Datatype</td>
<td>Tooth</td>
<td>(Right, Wrong)</td>
</tr>
<tr>
<td>hasDate</td>
<td>Datatype</td>
<td>Report</td>
<td>date</td>
</tr>
<tr>
<td>hasType</td>
<td>Datatype</td>
<td>Radiograph</td>
<td>String (Panaromic, OPG)</td>
</tr>
</tbody>
</table>

Rules: As mentioned before, in addition to a set of common vocabularies that may appear in reports of the experts and the patient, each of the experts in the domain may use specialized concepts in their reports that are specific to their domain of knowledge. To facilitate the negotiation between the individuals, we can extract some rules from the domain. The rules, based on the relationships between the domain’s concepts, can map the specialized concepts to their equivalent common concepts where it is possible. There are also common concepts that have the same meaning but different terms. Rules can help individuals to find these similar concepts as well. Another use of rules is when a combination of some symptoms may create one or more conditions for the treatment that should be considered. Rules can also find some of those conditions for the individuals.

Once the rules were found in the domain knowledge, they were formulated using the Semantic Web Rule Language (SWRL), which is a rule-based ontology language that allows individuals to benefit from inferring new knowledge from an existing OWL knowledge base. Using SWRL, Horn-like rules can be formulated in terms of OWL concepts (Park and Kim 2006). SWRL Rules take the form of an implication between an antecedent (body) and consequent (head). Both the antecedent and consequent consist of zero or more atoms (Horrocks, Patel-Schneider, Boley, Tabet, Grosof, and Dean 2004). SWRL rules are written in terms of OWL classes, properties, individuals, and data values. A detailed description of SWRL and its properties can be found in Horrocks, Patel-Schneider, Boley, Tabet, Grosof, and Dean 2004.

Protégé has a SWRL tab that provides users a development environment to work with SWRL rules. The SWRL rules can be executed by the Jess rule engine via SWRL Jess tab. Both the SWRL and SWRL Jess tabs need to be activated. Directions on how to activate them can be found here: SWRL tab: http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTab SWRL Jess tab: http://protege.cim3.net/cgi-bin/wiki.pl?SWRLJessTab. After execution of the rules, new inferred Jess facts can be added to the OWL knowledge base.
For our ontology, we found 14 rules. Table 2 shows four of them:

**Table 2.** Four examples of the rules

<table>
<thead>
<tr>
<th>No.</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dentist(?d) ^ seeClinicalExamination(?d, ?e) ^ FacialCellulities(?e) -&gt; Perocornites(?e)</td>
</tr>
<tr>
<td>2</td>
<td>Radiologist(?r) ^ seesRadiograph(?r, ?f) ^ DarkSpot(?f) -&gt; Infection(?f)</td>
</tr>
<tr>
<td>3</td>
<td>Decay(?s) -&gt; Caries(?s)</td>
</tr>
<tr>
<td>4</td>
<td>FamilyDoctor(?f) ^ hasTreatment(?f,?t) ^ seesHealthIssues(?f, ?p) ^ Pregnancy(?p) -&gt; NoSurgery(?t) ^ adviseTreatment(?f, ?t)</td>
</tr>
</tbody>
</table>

From Table 2, rule 1 is an example of rules that can map specialized concepts to their existing common concepts. Rule 1 indicates that if an instance of class Dentist has a relationship with an instance of class Facial Cellulities, which is one of the dentist's specialized concepts, via the “see Clinical Examination” Object property, then that instance of class Facial Cellulities can be added to class Precornites which is a common concept. Rule 2 is similar to rule 1, but in this rule, Dark Spot is one of the radiologist's specialized concepts that can be added to the common concept Infection class. Rule 3 is an example of similar concepts with different terms. It infers that an instance of class Decay, which is one of the subclasses of Symptoms, can be added to class Caries, another subclass of Symptom. Rule 4 can infer conditions for treatments based on the existing facts. It indicates if an instance of class Family Doctor, which has an Object property with an instance of class Treatment via “has Treatment” property and another Object property “sees Health Issues”, exists between that instance of class Family Doctor and an instance of class Pregnancy, which is a subclass of Health Issues; then add that instance of class Treatment to class No Surgery and add “advise Treatment” property to all individuals of OWL who are family doctors, have treatments, and have patients who are pregnant.

**Reasoner**

The reasoner executes the final step of the negotiation-facilitation system. Its function is to compare two ontology instances and report any similarities and differences between them. It then stores the results of this comparison, which the graphical user interface (GUI) can display in an appropriate manner. A simplistic GUI was created for demonstration purposes; it is expected that a production level system would utilize a much more useful GUI.

**Reasoner Overview:** As mentioned above, the reasoner compares two instances of the ontology. Each instance is a directed graph, where the nodes either represent individuals (i.e., an instance of a class in our ontology) or contain literal data, and edges represent properties. Individuals may appear multiple times in the graph; each occurrence of the individual in the graph is caused by that individual being the object of some property. It is thus possible for a graph to contain only one individual, yet have many nodes. It is assumed the graph contains no cycles (thus, it is a Directed Acyclic Graph) and that the ontology instances on which the reasoning is being performed are connected for some node \( \alpha \) (i.e., there is a path from \( \alpha \) to every node in the graph). Furthermore, \( \alpha \) plays a special role in the graphs compared. The individual \( \alpha \) is assumed to be the "person" whose negotiation position is represented by the graph. Therefore, the role of the reasoner is to compare the graphs centred on these individuals.

In the current implementation, \( \alpha \) must be specified by the user. This ensures there is no ambiguity about which individuals represent the people participating in the negotiation. Once specified, the reasoner begins to compare the two graphs, starting with the specified individuals. The comparison algorithm functions as follows. It begins by checking which classes the two individuals belong to. If they belong to mutually exclusive classes, then that is recorded as a difference. The datatype properties
are then compared. A datatype property is only considered if it exists for both individuals. If the literal values are different, then that difference is recorded. Finally, the algorithm considers the remaining properties of the two individuals. For every property in common between the two individuals, the algorithm calls itself recursively on the object of the property. As previously mentioned, the graph is assumed to have no cycles, so the current implementation does not keep track of nodes previously visited, although that would be a simple extension.

The results of the comparison at each node are stored themselves in a graph, which is equivalent in size and construction to the ‘intersection’ of the two input graphs. The nodes in the intersection graph contain the results recorded from the comparison of one node to another. The edges indicate how each node connects together. The graph can then be traversed by the GUI to output the results in a human-readable manner.

The reasoner is also designed to perform comparisons between more than two people. In this case, it considers every individual belonging to a user specified class (in the same way the user would specify $\alpha$ when comparing just two individuals). It then calls the comparison algorithm on every pair of users, thus producing $n(n-1)/2$ comparison results.

Technical Details: The reasoner is currently implemented in Java 1.6. It uses the Jena 2.6.2 library to interact with the ontology instance. The program begins by creating a simple GUI, primarily designed for demonstration. The guide to the GUI follows this section. The input to the program is any OWL/XML file specifying the instances of the ontology. Once the input file has been selected, the GUI starts the reasoner, which uses the Jena to create a model of the ontology using the ModelFactory class. Once the model has been created, any information pertaining to the instance can be collected from it. The reasoner generates a list of all classes which contain at least one individual by using the Jena function model.listClasses() and checking each class for individuals with hasIndividual(c). Other information from the model can be obtained in a similar manner.

A number of classes are used to help store information results or add additional functionality to the results of Jena. The results of the comparison are stored in the class CompareResult. It contains references to the two individuals being compared, as well as a reference to the property that connects the object to its parent. The results of the comparison are stored in two separate ways. An OntClass[][] stores the mutually exclusive classes that the individuals belong to, OntClass[0][i] stores the results for the first individual compared to the second, and OntClass[1][i] stores the results of the second individual compared to the first. It also stores a DataResult[] containing the results of the datatype comparison. CompareResult also contains an ArrayList of the children of this comparison result, that is, all nodes that the current node connects to via a Property. Finally, CompareResult implements the MutableTreeNode interface, and all appropriate methods, so that the results can be displayed as a JTree.

The DataResult class stores the result of a datatype comparison. It stores references to the two individuals in the comparison, the property being compared, and an ImprovedRDFNode object for each individual indicating what the literal's value and type is. ImprovedRDFNode is a wrapper class for Jena's RDFNode object. When initiated, it extracts the value and type of the literal from the RDFNode's toString() (as there are no appropriate methods to obtain this information), and records them, and provides methods to obtain that information.

The App, Pair, GuiFrame and Type classes contain classes and methods that facilitate other classes, and which control the GUI. The T5Reasoner and T5ComplexReasoner perform the reasoning on the instances of the ontology. T5Reasoner only performs one type of comparison between individuals - it checks to see if they belong to any mutually exclusive classes, and stores the result in an OntClass[][] array. T5ComplexReasoner extends the T5Reasoner class, and adds the additional functionality required by the reasoner. The T5ComplexReasoner requires the name of the class containing the individuals to compare (String indClass), this is supplied by the user through the GUI.
User Guide: As previously mentioned, the GUI supplied with the tool is fairly simplistic. When the program is run, it prompts the user to select the file containing the instance of the ontology. It then presents a dropdown list of classes, from which the user selects the classes containing the individuals the user wishes to compare. The reasoner can be executed by selecting the Reason menu and selecting Compare All. Two tabs display the results of the comparison. The Output tab contains a list of all compared individuals, along with details of the comparison. The Compare Tree tab displays a JTree of the results, which presents the results in an easier to view manner.

The output and tree tabs can be cleared by selecting Edit - Clear Output and Edit - Clear Tree. Additionally, the name space prefix can be hidden or displayed by selecting Edit - Remove/Add NS prefix. The user can also change the selection of the individuals class by selecting Reason - Select Individuals Class.

Results: The results produced by the reasoner do not form an ideal solution to the problem being solved. As previously discussed, rules are used to translate local concepts to general concepts. The reasoner works on an open-world assumption which means that local concepts are generally not returned as differences, as conflicts only exist explicitly between general concepts. Therefore, the last step of the negotiation facilitator program should be to take the results of the reasoner, and backtrack any rules that led to those results to obtain the original local concepts. These are the things that will best facilitate negotiation, as they most closely represent the opinions of the individuals involved in negotiation. These local concepts are mapped directly from the individual's original statements; thus, these are the concepts which should be considered in negotiation.

It is not possible to keep track of rules that are applied in the current implementation. Therefore, the java tool which currently only performs reasoning after the rules have been applied should be extended to apply the rules themselves. It would then be able to keep track of the rules that have been applied, and, thus, could display that information to the user.

The graphical user interface is another aspect of the tool which should be improved. The current implementation produces the correct results, but the primary use of a negotiation facilitator system is to facilitate communication between people; thus, it requires an intuitive user interface that effectively displays the agreements and conflicts between them. Ideally, such a user interface would be web-based, allowing physically separate individuals to participate in the system. The reasoner would then respond to requests from clients in a server-client manner.

Other Tools: Pellet was originally considered to perform the role of the reasoner. In that system, two different individuals would be asserted as being the same by the program. Pellet would then check the consistency of the resulting ontology instance. Any inconsistencies found would indicate conflicts, which would then be reported. Unfortunately, this approach did not work since Pellet would only find the first inconsistency and then stop. Therefore, this tool was developed to perform the reasoning.

Implementation Tools
For implementing this system, four tools/languages have been used: Protege, OWL-DL, SWRL, and Java. The ontology of the concepts has been implemented using Protege. Protege is an open-source platform that provides users with a suite of tools to construct domain models and knowledge-based applications with ontologies. OWL-DL is one of the three increasingly expressive sublanguages of OWL. The OWL Web Ontology Language is designed for use by applications that need to process the content of information instead of just presenting information to humans. This language provides additional vocabulary along with a formal semantics. Rules for mapping local concepts to global concepts have been coded using SWRL. SWRL is a Semantic Web Rule Language. Horn-like rules can be defined by the SWRL using the terms of OWL concepts. SWRL is fully compatible with OWL. A reasoner for finding agreements and disagreements has been implemented using Java.
Example

In the next section we will explain one case of wisdom tooth extraction as an example. We have taken this example of Wisdom teeth extraction from general day-to-day cases. In the later sections we will explain how to annotate various reports and prescriptions from various experts. We will also explain how this example gets processed in our system with the help of ontology sub-trees and the SWRL rules.

Reports (Input)

- **Patient**: The patient goes to the dentist with complaints of general weakness and pain around his right lower wisdom tooth and right ear. Patient has surgery phobia.
- **Dentist**: Clinical examination shows multiple restorations, missing maxillary left and right first bicuspids, and pericoronitis with facial cellulitis around right third molar. In the mandible, left and right first molars are missing and the patient has generalized periodontal defects. It is suggested that the patient get a panoramic radiograph. Minor infection has been seen in the gums.
- **Radiologist**: The panoramic radiograph revealed inverted impaction of the mandible right third molar and disto-angular impaction of the left third molar. There was a generalized horizontal bone resorption of both arches. No dark spots in the gums have been observed.
- **Dentist**: Radiograph shows inverted impaction of mandible right third molar and disto-angular impaction of left third molar. There was a generalized horizontal bone resorption of both arches. Patient is advised to have two third molars extracted. Periodontal Surgery is necessary under general anesthesia.
- **Family Doctor**: Patient has type II diabetes. He also had a myocardial infarction recently.
- **Anesthesiologist**: Based on results from the current tests, the blood sugar level of the patient is high. Patient is allergic to anaesthetic agents. The patient has added risk for general anesthesia.

Annotation

- **Patient**
  
The patient (Figure 7) goes to the dentist with complaints of general weakness and pain around his right lower wisdom tooth and right ear. Patient has surgery phobia.

  Note that in the figures below Patient1 represents instances.

  So we will annotate the patient’s report as:

  1. General weakness (Figure 7, Figure 8)
  2. Pain
  3. Surgery phobia
Ontology-Based Negotiation of Dental Therapy Options

Figure 7. Getting an instance of Patient class

Figure 8. Getting instances of Pain and Weakness classes

- Dentist
Clinical examination shows multiple restorations, missing maxillary left and right first bicuspid, and pericoronitis with facial cellulitis around right third molar. In the mandible, left and right first molars are missing and the patient has generalized periodontal defects. It is suggested that the patient get a panoramic radiograph. Minor infection has been seen in the gums.

So we will annotate the dentist’s report as (Figure 9, Figure 10):

1. Multiple restorations
2. Pericoronitis
3. Facial cellulites
4. Generalized periodontal defects
5. Infection
Figure 9. Getting instances of Patient and Dentist classes

Figure 10. Getting instances of Dentist’s specialized concepts based on his report
"Radiologist"

The panoramic radiograph revealed \textit{inverted impaction} of the mandible right third molar and disto-angular impaction of left third molar. There was a generalized horizontal \textit{bone resorption} of both arches. \textit{No dark spots} in the gums have been observed.

So we will annotate the radiologist’s report as (Figure 11, Figure 12):
1. Impactions
2. Bone resorption
3. No dark spots

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure11.png}
\caption{Getting instances of Anaesthesiologist}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure12.png}
\caption{Getting instances of Radiologist’s specialized concepts based on his report}
\end{figure}
**Dentist**

Radiograph shows *complete-bone impaction* of mandible left third molar. Patient is advised to have one third molar extracted. *Surgery is necessary under general anesthesia.*

So we will annotate the dentist’s report as (Figure 13):

1. Impactions
2. Advised extraction
3. Surgery under GA

---

**Figure 13.** Getting instances of Treatment
- **Family Doctor**

  Patient has *type II diabetes*. He also had a *myocardial infarction* recently. So we will annotate the family doctor’s report as (Figure 14):
  
  1. Diabetes
  2. Myocardial Infarction

*Figure 14. Getting instances of Influential Factors*
- Anesthesiologist
Based on results from the current tests, the blood sugar level of the patient is high. Patient is allergic to anesthetic agents. The patient has added risk for general anesthesia.
So we will annotate the anaesthesiologist’s report as (Figure 15):
1. High blood sugar
2. Allergic to anaesthetic agents

Figure 15. Getting instances of Health which represent health issues of the patient and are determined from Anesthesiologist’s report

Annotations of the reports are used as input to the mapping step. In this step, using predefined rules, local concepts are mapped to global concepts. Then these concepts along with original global concepts in the reports are considered as instances of the ontology classes. Finally, the reasoner checks these
instances in order to find possible inconsistencies. Inconsistency is the sign of conflict. These conflicts are then output to the users as disagreements.

All the four figures below are screenshots of running the program in Protégé 3.4.1. To create instances of the concepts for all the examples below, we used Protégé 3.4.1 forms.

Example: First we should create an instance of class Patient for the patient whose name is James (Patient1). He is 62 years old and has a report and a radiograph. After creating instances of Report and Radiograph (ReportP, Radiograph1) for Patient1 and setting their properties next step is to add the instances of his symptoms (Pain_1 and Weakness_3). Figure 16 shows form of the Patient1 in Protégé.

Other instances of the reports’ concepts were created in the way that Patient1 was created. Figure 17 and Figure 18 are screenshots of the instances of Dentist and Anesthesiologist (Dentist1, Anaesthesiologist_42). The dentist advises Surgery (Surgery_14) as treatment while the
anesthesiologist advises surgery should not be done based on patient’s health conditions (NoSurgery_13).

Figure 17. Form of the Dentist1 in Protégé
After creating all the instances of the concepts based on the reports from the domain experts and patient, rules should be executed and new inferred facts (if they exist) should be added to the OWL knowledge base. This step was done using the SWRL and Jess tabs in Protégé. Figure 19 is the screenshot of the Protégé SWRL tab.
The result of running the reasoner with new OWL knowledge is shown in Figure 20 and Figure 21.
Comparing Anaesthesiologist_42 to Dentist1
-------------------------
  Mutually Exclusive classes which Anaesthesiologist_42 belongs to
   Anaesthesiologist
  Mutually Exclusive classes which Dentist1 belongs to
   Dentist
Literal Differences:
  Anaesthesiologist_42 differs from Dentist1 on property hasWorkHistory with values: 8:9
  Anaesthesiologist_42 differs from Dentist1 on property hasName with values: Kate:John

Comparing ReportA to ReportD
-------------------------
  Connected to parent via: hasReport
  No differences found

Comparing NoSurgery_13 to Surgery_14
-------------------------
  Connected to parent via: advisesTreatment
  Mutually Exclusive classes which NoSurgery_13 belongs to
   NoSurgery
  Mutually Exclusive classes which Surgery_14 belongs to
   Surgery
  Literal Differences:

Comparing Anaesthesiologist_42 to Patient1
-------------------------
  Mutually Exclusive classes which Anaesthesiologist_42 belongs to
   Anaesthesiologist
  Mutually Exclusive classes which Patient1 belongs to
   Patient
Literal Differences:
  Anaesthesiologist_42 differs from Patient1 on property hasName with values: Kate:James

Comparing ReportA to ReportP
-------------------------
  Connected to parent via: hasReport
  No differences found

Figure 20. Output of the Reasoner with OWL File of Example1 as Input
Figure 21. Output of the Reasoner with OWL File of Example1 as Input (Tree format)
CONCLUSION

In this chapter, a negotiation system has been proposed that uses semantic web concepts, ontology, and semantic reasoning to facilitate the negotiation process. Using the ontology and predefined rules, the system maps local concepts of each individual to global concepts and, therefore, provides a common layer for negotiation. The output of the system is a list of conflicts and agreements between reports of different individuals. The proposed system has been tested for decision making about wisdom teeth treatment options. Improvements that could be made to the system include an ability to keep track of rules and ideally a web-based interface.

References


