An Ontology for Robotics: Implemented using a Mobile Android Application

Dr. Shankar
Spring 2011
COT6930-Advanced Android Projects

Brian Chamba, Joseph Gundel
Abstract

There is a lot of information available for robotics and robotic related material but with no clear cut easy way to search through it. When someone searches the web for a specific question they receive thousands of irrelevant links. The link just shares a word or two. To solve this problem there is a need to organize the data and relate the data in specific ways. There must be an intelligent way of searching. To begin to correct this problem this project will create and implement an Ontology for Robotics. We will create a mobile application to search the ontology and we will begin to populate the ontology with relevant information and relationships.
## Table of Contents

1. Introduction ................................................................................................................. 4  
   1.1 Problem .................................................................................................................. 4  
   1.2 Solution .................................................................................................................. 4  

2. Tool Usage ....................................................................................................................... 4  
   2.1 Protégé ..................................................................................................................... 4  
   2.2 Eclipse ..................................................................................................................... 5  
   2.3 Android SDK .......................................................................................................... 5  

3. Robotic Ontology ............................................................................................................ 5  
   3.1 Main Classes ......................................................................................................... 5  
      3.1.1 Education ........................................................................................................ 6  
      3.1.2 Components ..................................................................................................... 6  
   3.2 Sub Classes of Education ....................................................................................... 7  
      3.2.1 Kits .................................................................................................................. 7  
      3.2.2 Tutorials ......................................................................................................... 8  
   3.3 Sub Classes of Components ................................................................................... 9  
      3.3.1 Communication ............................................................................................. 9  
      3.3.2 Mechanics ....................................................................................................... 10  
      3.3.3 Microcontrollers ............................................................................................ 10  
      3.3.4 Sensors .......................................................................................................... 11  
      3.3.5 Software ......................................................................................................... 11  
   3.4 Future Ontology Expansion .................................................................................... 12  

4. Integration ....................................................................................................................... 12  
   4.1 JAVA Based Parser ............................................................................................... 12  
      4.1.1 Parser Class .................................................................................................... 13  
      4.1.2 Individual Class .............................................................................................. 13  
   4.2 Android Mobile Application ................................................................................... 14  
      4.2.1 Profile ............................................................................................................. 14  
      4.2.2 Google Search ............................................................................................... 15  
      4.2.3 Ontology Search ............................................................................................. 16  
      4.2.4 Future Application Expansion ....................................................................... 18  

5. Conclusion ..................................................................................................................... 18  
References .......................................................................................................................... 20
1. Introduction

1.1 Problem

Imagine a question like this: Tell me the best way to design a robot with wheels? You will get about 100,000 hits on Bing, and 72.4 million hits on Google. Most of the information replied to on this search is irrelevant. You will see various topics that do not have anything to do with the topic and just had one of the keywords in the title.

1.2 Solution

The goal of our Robot Ontology effort is to develop and begin to populate a neutral knowledge representation (the data structures) capturing relevant information about robot building which will assist in the development, testing, and implementing of robotic technologies. This knowledge representation must be specific enough to allow the user to filter out irrelevant information. As such, we have chosen to use an ontological approach to representing these requirements.

Part of the search is having a user profile which will include information such as level of education and training; interests; demographics; location; etc. We can then use that information to limit the search. Part of it is also having a dictionary of unambiguous terms that can be translated and sent on the web to find good fits and returning info on them to the user.

2. Tool Usage

2.1 Protégé

To create our Robotic Ontology we decided to use Protégé for many reasons. Protégé is a free, open source ontology editor and knowledge-base framework. Protégé is based on Java, is extensible, and provides a plug-and-play environment that makes it a flexible base for
rapid prototyping and application development. Protégé is supported by a strong community of developers and academic, government and corporate users, who are using Protégé for knowledge solutions in diverse areas.

2.2 Eclipse

To create our application we decided to use the Eclipse IDE for Java Developers for many reasons. Eclipse is an open source community, whose projects are focused on building an open development platform comprised of extensible frameworks, tools and run-times for building, deploying and managing software across the life-cycle. Also Eclipse has an Android plug-in which allows for easy Android development.

2.3 Android SDK

For our implementation of the Robotic Ontology we wanted to use an open source platform so we chose the Android Platform. Android is a software stack for mobile devices that includes an operating system, middle-ware, and key applications. The Android SDK provides the tools and libraries necessary to begin developing applications that run on Android-powered devices.

3. Robotic Ontology

3.1 Main Classes

We decided to start our ontology using two main classes: Education and Components [Figure 1]. All the information we found was classified under these two categories. Our ontology can be expanded in the future to have more main classes, but we felt these two main classes are the most relevant based on the information we found.
3.1.1 Education

The purpose of the Education class was to group pertinent information together which will be useful to users who need tutorials and learning tools. The information we collected under Education could be broken down further into two sub classes: Kits and Tutorials [Figure- 2]. Under Education more subclasses can be added but with the information we had collected only two subclasses where needed.

3.1.2 Components

All the information gathered that was not classified as an education resource was
classified as a component. The information we collected under Components could be broken down further into many sub classes, however we chose: Micro-controllers, Sensors, Mechanics, Communication, and Software [Figure 3]. Under Components many more subclasses can be added but with the information we had collected we chose five categories.

3.2 Sub Classes of Education

The information we collected under Education are broken down into two sub classes: Kits and Tutorials.

3.2.1 Kits

There are many different kits available for robotics. Everything from small beginner projects to large advanced kits. All the kits are broken down into two groups: Assembled and Non-Assembled as shown in [Figure 4].
For the purpose of our project we only populated the ontology with a few individual kits for each subclass. Each individual consists of a list of keywords and a list of URLs. The keywords are used for an external search linking the individuals to similar individuals. The URL's are links to websites that have information pertaining to the individuals.

### 3.2.2 Tutorials

Similarly to kits, there are many different tutorials available that relate to robotics and related fields. For the purpose we chose to limit the subclasses to six different categories: Communication Hardware Tutorials, Software Tutorials, Micro-controller Tutorials, Sensor Interfacing Tutorials, Mechanics Tutorials, and Resources. Resources and Software Tutorials are broken down further into smaller sub classes as shown in [Figure 5]. Similarly to Kits, each individual of Tutorials also consists of a list of keywords and a list of URLs. The keywords are used for an external search linking the individuals to similar individuals. The URLs are links to websites that have information pertaining to the individuals.
3.3 Sub Classes of Components

The information we collected under Components are broken down further into many sub classes. Micro-controllers, Sensors, Mechanics, Communication, and Software are subclasses of the Components class.

3.3.1 Communication

Communication is a major component of robotics. Due to the many different technologies available on the market and the vast amounts of information on the subject we created two subclasses: long range communication and short range communication as shown in [Figure 6].
These subclasses themselves can be further expanded into many more subclasses but for the purpose of our project we just populated these two subclasses with a few individuals. Each individual consists of a list of keywords, a list of URLs, and a list of Tutorials associated with it.

### 3.3.2 Mechanics

Mechanics is another major component of robotics if not the most important component. There exists a vast amount of information on mechanics and related topics. For the purpose of our project (as computer engineers and computer scientists) and due to lack of knowledge in mechanics we did not expand this component into other subclasses. Similarly to Communication each individual in the Mechanics subclass also consists of a list of keywords, a list of URLs, and a list of Tutorials associated with it.

### 3.3.3 Microcontrollers

Today micro-controllers can be found in almost any complex electronic device from portable music devices to washing machines to your car. They are programmable, cheap, small, can handle abuse, require almost zero power, and there are so many varieties to suit every need. This is what makes them so useful for robotics. They are like tiny affordable computers that you can put right onto your robot. For the purpose of this project we populated the micro-controller subclass with a few very common micro-controllers used for robotics.
Similarly to Communication each individual in the Microcontroller subclass also consists of a list of keywords, a list of URLs, and a list of Tutorials associated with it.

### 3.3.4 Sensors

Robot Sensors are one of the most important components of a functional robot; they are your robot's eyes, ears, sense of touch, balance, and only source of input from its environment. Robot switches to turn your bot on and off and detect collision, Accelerometers and Gyroscopes give feedback to maintain balance and orientation, Touch and Force sensors provide feedback on physical contact with objects, Robot Distance and Range Sensors to assist in navigation and avoid obstacles, Environmental sensors give temperature, humidity, and vibration input, and a variety of navigation sensors help your robot move through the world intelligently. For our project we classified all our information on sensors into two subclasses: Environmental Sensors and Location Sensors [Figure 7].

![Diagram](image)

[Figure 7]

Similarly to the other Components each individual in the Sensor subclasses also consists of a list of keywords, a list of URLs, and a list of Tutorials associated with it.

### 3.3.5 Software

Robot software is the coded commands that tell a robot what tasks to perform and control its actions. Programming robots is a non-trivial task. Many software systems and frameworks
have been proposed to make programming robots easier. The data gathered for our ontology can be further divided into subclasses: Low Level, Mid Level, and High Level [Figure 8].

Similarly to the other Components each individual in the Sensor subclasses also consists of a list of keywords, a list of URLs, and a list of Tutorials associated with it.

3.4 Future Ontology Expansion

The Robotic Ontology we created is only a start. There is much more information available. The ontology for robotics can be expanded to incorporate a lot more information. Also many more relationships can be added as needed. We did not have the resources available to expand on topics such as mechanics. If users continue to share information and update the ontology, it will become a great resource for people wanting to gather information on robotics and related material.

4. Integration

4.1 JAVA Based Parser

The first step in integrating the ontology into an Android application was converting it to an object that the application could search on. This is done in Java without any knowledge of Android specific classes or components using two main classes, Parser and Individual. Both
classes work together in order to extract the individuals from the ontology saved in Turtle format.

### 4.1.1 Parser Class

All the individuals listed in the ontology are first copied into a text document that is placed inside the project folder. The Parser class reads in every line of the text document and stores it as a String. Next, the Parser class splits the file into the separate individuals it contains. This is easily done using the Turtle format because each individual is preceded with the String “###”. This creates an ArrayList of Strings with each record in the ArrayList being the text for one individual. This is all done in the constructor of the Parser class which is shown below.

```java
public class Parser {
    public Parser(InputStream input) throws FileNotFoundException {
        String turtleFile="";
        String values[];
        Scanner in = new Scanner(input);
        while (in.hasNextLine()) {
            turtleFile+=in.nextLine();
        }
        values = turtleFile.split("###");
        for (int i = 0; i<values.length; i++) {
            values[i]+="###";
            individuals.add(new Individual(values[i]));
            Log.i("values", values[i]);
        }
    }
}
```

Before finishing, the Parser constructor uses the Strings (representing each individual) to create an ArrayList of Individual objects.

### 4.1.2 Individual Class

Each string passed to the constructor of the Individual class is parsed for the different properties it contains. These properties include the name and type of the object along with
data and object properties such as hasTutorials, hasKeywords, and hasURL. Each property parsed is stored into either a String or an ArrayList of Strings. The properties stored in the ArrayList are properties that can contain multiple values, such as hasTutorials. If an individual does not have a property that is being parsed the ArrayList for it is left empty. An example of this is when any individual of type Component is parsed the ArrayList for URLs is left empty since no components should have any URLs associated with them. All the parsing functions are listed below.

    private void parse()
    {
        parseName();
        parseType();
        parseKeywords();
        parseSuppliers();
        parseTutorials();
        parseTutorialFor();
        parseURL();
    }

When complete, the Parser object contains an ArrayList of Individual objects which can be used by the Android application to search through.

### 4.2 Android Mobile Application

The Android application has three main components to it: the user profile, the Google search, and the ontology search. Both of the searches rely on the user profile to determine which links and keywords to user when performing a search.

#### 4.2.1 Profile

The user profile is set up using four categories: Age, Education Level, Tech Skill Level, and Programming Skill Level as shown in [Figure 9]. Users must set up this profile before performing any search. Each category has three choices to choose from. Selecting one of these choices results in a value for that category of 0, 1, or 2. Once the user enters a response for each category and is ready to perform a search, the values of each category are
added together. This resulting total is then converted to a 0, 1, 2, or 3 depending on the range that the total falls into. The converted total represents the overall knowledge and skill level of the user in the field of robotics. 0 = no knowledge or skill, 1 = little knowledge or skill, 2 = advanced knowledge or skill, and 3 = expert knowledge or skill. At any time the user can modify their profile if they wish to obtain different results.

![Robot Ontology Search](image)

[Figure 9]

### 4.2.2 Google Search

One method of finding information using the ontology is with a Google search. The user enters their query and hits the Search Google button from the Search Activity as shown in [Figure 10].
A search is performed for each word in the query in all the individuals. If an individual contains one of the keywords, its list of keywords is added to the query.

In addition to keywords from the ontology being added to the query, words based on the user's profile are also added. A “beginner” profile would add ‘beginner’ or ‘first’ to the query whereas an “expert” profile would add ‘expert’ and ‘advanced’. Once the query is completed, a Google search is performed with the newly expanded query.

4.2.3 Ontology Search

The second search method implemented in the application is a search on the seed URLs placed in the ontology itself. With this method, the query is again broken up into its individual words and these words, along with words added to query because of the user’s
profile type, are searched in the keywords for individuals. When an individual contains one of the words in the query in its list of keywords, it gathers all the URLs from itself and any individual it has a relationship with. Once all the URLs have been obtained they are passed to the Results activity.

The Results activity is responsible for showing appropriate links to the user based on their profile. Each URL obtained from the ontology has a level of difficulty associated with it. A user will only see links that are equal to or less than their profile's knowledge level.

[Figure 11]
4.2.4 Future Application Expansion

The application has much room for improvement and expansion. One tool that could make the application more useful and accurate is Jena. Jena provides a way to search through the ontology without having to use the custom Java parser built for this project. It also is able to parse through more file types than just Turtle. Searching using Jena ensures accurate results as it is capable of parsing through the many relationships normally found in ontologies. The current parser would need to be modified every time a new relationship is created in the ontology.

Using Jena, however, does come at a cost to performance. Applications perform significantly slower when using Jena. Because the current robotics ontology is small, it may not have made a noticeable difference, but when searching through a much larger and complex ontology Jena could cause the application to hang up. This is a tradeoff to consider when deciding between performance and accuracy.

5. Conclusion

The goal of our Robot Ontology effort was to develop and begin to populate a neutral knowledge representation (the data structures) capturing relevant information about robot building. This knowledge representation must be specific enough to allow the user to filter out irrelevant information. As such, we have chosen to use an ontological approach to representing these requirements.

Using the ontology we created along with the user profile we were able to successfully decrease the amount of irrelevant hits. The query “How to build a robot with wheels for beginner” placed in Google returns 10.5 million links. Using Bing, the same query returns 1.97 million links. By translating the original query with the ontology, the amount of links returned is decreased in both search engines. The translated query using Google returns
only 997 thousand links and using Bing it returns 1.09 million. Not only is the amount of links minimized, but the most relevant information is still found at the top of the list.

While this is a good start to finding more relevant information about robotics, the eventual goal is to populate the ontology with enough links so that a search engine, like Google, does not have to be used.
References

1. http://protege.stanford.edu/
7. http://android.fau.edu