Using OWL Ontologies for Selective Waste Sorting and Recycling

Arnab Sinha, Paul Couderc

Using OWL Ontologies for Selective Waste Sorting and Recycling

Arnab Sinha and Paul Couderc

INRIA, Rennes-Bretagne Atlantique,
Campus Universitaire de Beaulieu
35042 Rennes Cedex, France
{arnab.sinha,paul.couderc}@inria.fr
http://www.inria.fr/en/en/teams/aces

Abstract. We are seeing the emergence of the Internet of Things, where digitally enabled objects can interact in smart environments. RFID can play an important role in linking common objects to the digital world. In this paper, we focus on efficient processing of collective waste items. These are considered to be smart by tagging them with RFID which bears the description of its properties. We have demonstrated a model using OWL ontology to sort these smart waste items for better recycling of materials. Our motive for using ontologies is for representing and reasoning of the domain knowledge to be autonomous, reducing the need for frequent references and updates for knowledge definitions from external sources in real time.

Keywords: OWL ontology, selective sorting of waste, RFID, recyclable materials, N-ary relations

1 Introduction

Today Pervasive computing is gradually entering people's everyday life. Commonly used objects are made smarter. They are able to adapt and integrate with the environment which might have capabilities to perform computation and information processing. The use of RFIDs and sensors have an important contribution in imparting intelligence. RFID chips are generally used to connect physical objects in the digital environment. Presently, RFID tags are widely used by retailers and manufacturers for inventory management. We can assume that in future they will tag every product. This would also support for better waste management and recycling. Considering this future trend, we consider to have "self describing" smart waste items. Efforts have been made in the past to perform efficient waste management using RFIDs [1].

The generation of waste items in recent times has been growing by leaps and bounds [2]. If these are not disposed and treated properly, it can be detrimental to the environment. The sorting of these items should be done at the earliest for maximizing the amount of valuable recyclable materials contained in it and reduce their contamination by unwanted materials. Making these waste items smarter, we could be able to handle such problems in a much efficient way. The tags on the waste items contain the information about the amount of valuable recyclable materials it contains. This information is read and decisions made during sorting process which requires knowledge. In this paper, we have demonstrated the use of OWL ontology to represent this knowledge. The reasoner is also used to make inferences when smart waste items are added. It has the advantage that the knowledge representation would be self sufficient and could be easily shared when required. In the following sections we describe a model using OWL to build an ontology for these problems in the domain of waste management.

2 Background

2.1 Selective Sorting

Every category of the selective sorting process would have some criteria and conditions like the amount and quality of recyclable materials recovered. In addition to this there may be some hazardous or unwanted waste items that might cause contamination to this category [5, 6]. Within the perspective of smart waste items we take into consideration that they contain the information regarding the recyclable material contents.

Taking examples from everyday life, some of the recyclable materials are different types of glass, paper/cardboard, plastic, metal etc. We would take examples for our model that elucidate the selective sorting of some of these materials. Hence our ontology contains classes such as *Glass*, *Paper*, *Plastic* and *Metal*. Being distinct types of materials, they are represented as disjoint classes in the ontology. They represent the generics and they contain individuals as members to represent specific categories. Specific types of *glass* like *SodaLimeGlass*, *FoamGlass* etc. are listed as indivduals in the ontology. Our model that accepts or rejects items depending on the preconditions that are set regarding the categories and amount of recyclable materials it wants to maximize.

2.2 N-ary Relations

The W3C group provides the ontology design method for N-ary relations in an ontology [3]. Since we have made use of these relations extensively in our model, a brief overview is necessary about the types of knowledge it can be used to represent.

Formally, an N-ary relation forms a mapping of an individual with other individuals or data values. The relations are all interconnected in some way. It is used when additional information need to be added to a binary relation.

The W3C lists four types of cases where N-ary relations are useful. The first case demonstrates the use of N-ary relations when additional attributes need to be provided with a relation. The classical example they have stated is *"Christine has breast tumor with high probability"*.

2.3 Modifying N-ary Relations with Numeric Values

The N-ary relations referred above have provisions for specifying the probabilities for the diagnosis as discrete values. This use case suggested by W3C could be modified to represent some specific situations. The attribute that relates the probability on the diagnosis consists of a discretized value. It would be more realistic if the probability values could be represented with numeric values. Hence we have incorporated this along with some more changes to build a model that exhibits useful features. We will show the usefulness of this model in practical applications which would be discussed in subsequent sections.

Modifying the first use case of W3C, we have defined the N-ary relations linking the classes as shown in figure 1. The class *Diagnosis_Relation_20* interconnects all the related classes we are interested to form relationships with. Its relation with the class *Disease* using the property *diagnosis_value* expresses a diagnosis with *at least one* disease. Similarly the other property *diagnosis_probability* holds *all* relations with integers having any value ≥ 20. This is one of the modifications we had stated earlier which makes the numeric representation of the probability value more realistic. We have defined both *diagnosis_value* and *diagnosis_probability* as functional properties, thus requiring that each instance of *Diagnosis_Relation_20* has exactly one value each for *Disease* and its probability.

Figure 1 has another property *has_diagnosis* that represents relation between the class *SickPerson* and *Diagnosis_Relation_20*. The existential quantifier "someValuesFrom" requires to have *at least one* instance of the class *Diagnosis_Relation_20*. So this completes the entire relation to represent a *SickPerson diagnosed with a disease with probability* $\geq$ *20*.

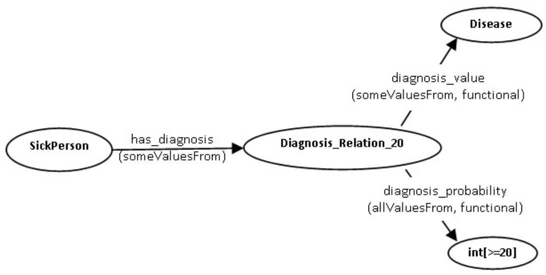

Fig. 1. Representing N-ary relation use case after the modification

Now we will define another similar N-ary relation class *Diagnosis_Relation_40* having probability value ≥ 40 and make both these classes *equivalentClass*. We are putting down the complete definitions for these two classes below:

```
Class: Diagnosis_Relation_20
    EquivalentTo:
        (diagnosis_value some Disease)
         and (diagnosis_probability only xsd:int[>= 20])
```

```
Class: Diagnosis_Relation_40
    EquivalentTo:
        (diagnosis_value some Disease)
         and (diagnosis_probability only xsd:int[>= 40])
```

We would observe an interesting feature with these classes. The class *Diagnosis_Relation_20* would be inferred as superclass of *Diagnosis_Relation_40* by the reasoner. Hence the individuals in the class *Diagnosis_Relation_40* satisfies both the class conditions. Figure 2 represents the sets of classes using a Venn diagram.

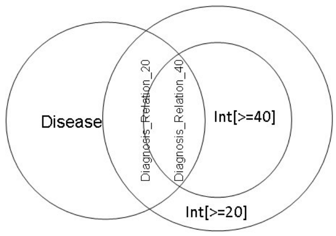

Fig. 2. Explanation for subclasses of the class Diagnosis_Relation_20

3 OWL Information Model

Reiterating on our problem described earlier, we intend to perform the reasoning of this selective sorting based on the percentage of recyclable materials it contains as well as maximize the extraction of the valuable recyclable materials? In this section we will explain how our implemented model performs this specific task. This model makes extensive use of the modified N-ary relation using numeric values as explained in section 2.3.

The system performing selective recovery should have sufficient knowledge for making decisions to accept or reject a waste item. We have chosen to embed this knowledge using ontologies in the systems. Ontologies can formally represent a set of concepts within a domain along with the relationships between them [9]. Among all the available ontology editors for OWL, we have used Protégé to develop the ontology for it [4].

Before proceeding with description of the various categories using ontologies, we would introduce the used N-ary relations. An N-ary relation such as *Glass_30* in figure 3 would represent *Glass* with quantity more than 30%. There are two properties *material_value* and *material_percentage* that links *Glass_30* with the recyclable material class *Glass* and the numeric value 30 as percentage. *Plastic_50*, *Glass_80*, *Metal_60* and *Paper_70* are the other similar relations used in our ontology. We have used consistent naming convention for all such N-ary relations in our ontology.

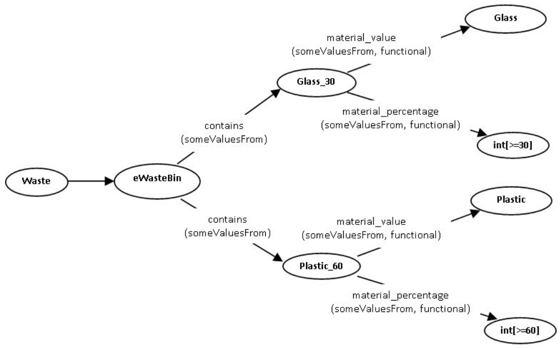

Fig. 3. Representing the using N-ary relation

3.1 Representing E-wastes

Electronic waste items or simply e-wastes constitutes any type of electronic item like computers, telephones, televisions and so on. These contain various types of materials both useful as well as hazardous. Materials like plastic, glass etc are contained in them which can be used for recycling. To identify and perform selective sorting of these types of waste items using the ontology we have defined an equivalent class *eWasteBin* that considers any item as e-waste which contains more than 60% of *plastic* combined with more than 30% of *glass*. Capturing these conditions, the axioms can be stated using Manchester syntax as:

```
Class: eWasteBin
  EquivalentTo:
    Waste and(contains some Glass_30) and(contains some Plastic_60)
```

Figure 3 shows the relations between the classes. The property *contains* establishes the relation between the class *ewasteBin* and the N-ary relations. It's OWL existential quantifier *someValuesFrom* implies that it should have one of these items each.

Suppose we have an electronic waste item that contains atleast 62% *Plastic* along with 35% *Glass*. We can represent the item as:

```
Individual: ItemX
  Types:
    GenericProducts, Waste
    and (contains some((material_value some Glass)
              and (material_percentage value 35)))
    and (contains some ((material_value some Plastic)
                 and (material_percentage value 62)))
```

This item would be classified as an electronic waste item satisfying the conditions of *eWasteBin*.

3.2 Capturing Categories like Glassbin

What happens if we are interested in sorting of items that contains only one major type of recyclable material. For instance, *GlassBins* would like collect items selectively that contains *Glass* of more than 80%. The following class represented below would be able to capture such types of items:

```
Class: GlassBin
  EquivalentTo:
    Waste and (contains some Glass_80) and (contains only Glass_80)
```

For the representation of this category there exists one and only one N-ary relation. So we have expressed the same using both the OWL existential restrictions *allValuesFrom* and *someValuesFrom*.

A typical example containing high amount of *Glass* would be *coke bottle*. The axioms for such an individual would be:

```
Individual: CokeBottle
  Types:
    contains exactly 1 Rel_Material_Qty, GenericProducts,
    contains some ((material_value value SodaLimeGlass)
                    and (material_percentage value 94))
```

There is a subtle addition while stating the properties of the individual *CokeBottle*. Due to the open world assumptions of OWL ontology, it needs to be stated explicitly to contain only one type of material. So the axiom "*contains exactly 1 Rel_Material_Qty*" is added to assure the OWL reasoner that this individual contains only one type of material. As we are using the model and add items through automatically generated programs, it would not be difficult ensuring to close off this information.

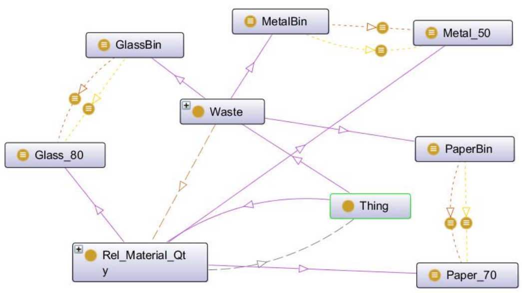

Fig. 4. Different Categories and Generic Items represented in Ontology

Classes for other categories can be created similarly to the *GlassBin*. Figure 4 shows the visual representations of *MetalBin* and *PaperBin* in the ontology. They use the modified N-ary classes *Metal_50* and *Paper_70* respectively.

These various categories can also be described by class expressions without using the N-ary classes such as *Metal_50, Paper_70* etc. The quantities for the recyclable materials are standardized for each category and won't be quite alot in number. So using these named classes would make the description of categories simple and easy to understand. We can also reuse them for defining newer categories as well as utilize them in DL queries.

4 Implementing the Sorting Application

Until this point, we have discussed our model for representing knowledge for selective sorting of collective waste items using ontology. The discussion also includes ways to represent items as individuals to be classified in the proper category. Our application has used the Information model in the backend. Ontologies are increasingly being used in Pervasive Computing Environments as some of their features are very well suited for the purpose [7, 8]. The Java API provided by Protégé manipulates the ontology programmatically and interfaces with the Information model in our stand-alone application [10].

The smart waste items are tagged with RFID tags that contain the details about the percentage contents of the various recyclable materials. For example, the individual *ItemX* in section 3.1 would have the data containing in its tag as *Glass-35,Plastic-62* as plain text. Our intelligent bin would have a RFID reader to detect the smart waste items and read the information contained in its tag. The application running on the bin would use the ontology based knowledge to infer the item's category. If found suitable, the bin would accept the item. The ontology based model can also be used to sort items in waste processing plants. In our project *Bin That Thinks*, we are developing such applications using the discussed model in consultation with a major waste management company [11].

We can use an alternative method for tagging the generic smart waste items. For example, a generic item like *Monitor* can have an entry in the ontology as below:

```
Class: Monitor
  SubClassOf:
    GenericProducts,(contains some ((material_value some Glass)
                           and (material_percentage value 31)))
    and (contains some ((material_value some Plastic)
                 and (material_percentage value 70)))
```

Hence while tagging monitors, it would be sufficient to write this classname *Monitor* into the tag. Our application would be able to detect such tags for proper categorization.

5 Conclusion

In this paper, we have proposed a model that can classify waste items based on the recyclable materials they are made up of. Our model utilizes the concept of N-ary relations with some modifications. It is very useful in the context of selective sorting. Waste processing plants have automated selective recovery systems to maximize the collection of recyclable materials with least contamination of hazardous or unwanted waste items. Currently, our ongoing work proposes the composition of recyclable materials to be indicated on the RFID tags affixed on the items [11]. The ontology can also contain the composition of generic items. So in the case of these generic items, the RFID tag just contains the generic name written in it. Since we are encoding the keywords from the ontology, integration and reasoning would easily be possible. This makes the system autonomous which is a forte without making frequent external references. This is an important property for systems working in real-time.

In our future work we propose to utilize the full benefits of ontology by performing the domain knowledge sharing among the autonomous sorting systems in the entire infrastructure.

Acknowledgments. We would like to thank the Protege user community for valuable suggestions while developing the ontology model.

References

1. B. Chowdhury and M.U. Chowdhury, "RFID-based real-time smart waste management system", Telecommunication Networks and Applications Conference, 2007. ATNAC 2007. Australasian , vol., no., pp.175-180, 2-5 Dec. 2007.
2. "The EUs approach to waste management", `http://ec.europa.eu/environment/waste/index.htm` [accessed 22 March 2012,10h30]
3. World Wide Web Consortium (W3C), `http://www.w3.org/TR/swbp-n-aryRelations/`
4. N. F. Noy and D. L. Mcguinness, "Ontology Development 101: A Guide to Creating Your First Ontology", Technical Report KSL-01-05, Stanford Knowledge Systems Laboratory, 2001.
5. Veolia : Research and Development, `http://environmentalpassion.com/Research_and_development`
6. Better Sorting for Better Recycling, `http://environmentalpassion.com/resource.php?id=2239`
7. G. R. Librelotto, L. O. Freitas, A. Fiorin, B. A. Mozzaquatro, L. Pasetto, R. G. Martini, R. P. de Azevedo, R. T. Pereira, "OntoHealth: A System to Process Ontologies Applied to Health Pervasive Environment", International Conference on Ubi-Media Computing, pp. 59-64, Fourth International Conference on Ubi-Media Computing, 2011
8. A. Ranganathan, R. E. McGrath, R. H. Campbell, M. D. Mickunas,"Ontologies in a pervasive computing environment", Workshop on Ontologies in Distributed Systems, IJCAI, 2003, pp. 10-14, ACM Press
9. Berners-Lee T., Hendler J., Lassila O.:The Semantic Web. Scientific American, pp. 284-285, 35-43. 2001
10. H. Knublauch et al., "The Protégé OWL Experience", Proc. OWL: Experiences and Directions Workshop 2005.
11. Bin That Thinks, `http://binthatthink.inria.fr/` [accessed 06 January 2012,11h35].