AN OBJECT SELECTION MECHANISM FOR SCHEMA INTEGRATION OF AGENT'S KNOWLEDGE STRUCTURE IN VIRTUAL REALITY

Dong-Hoon Kim and Jong-Hee Park
E10-506, Electronics Dept, Kyungpook Nat'l Univ., Sankyuk-dong, Pukgu, Daegu, South Korea

Keywords: Knowledge structure, Schema integration, Object selection mechanism.

Abstract: Similar to human knowledge, the knowledge of agents should be able to express various and vast information in the virtual reality. In order to represent the numerous information we should construct the lots of schemas. For such a reason, the schemas are represented redundantly bringing about the problems such as update and insertion anomalies. In order to solve these problems we should consider the method of schema integration. In this paper, we propose the methods of selecting object which are suitable for the schema integration.

1 INTRODUCTION

An ontology is an explicit specification of a conceptualization (Gruber, T. R., 1993). Everything that can compose a situation is abstracted in an ontology. The concepts, the topmost elements of our ontology, encompass the entities and the logical concepts (Park, J, 2004). Based on the ontology, a schema that is the abstraction of the real world symbolizes all objects, and is organized by their relations.

A schema captures the skeletal semantics of a domain in terms of concepts. Thus the knowledge of the agents should be structured in advance so that agents in the virtual real world are able to act as in the reality. The knowledge structure of an agent consists of three abstraction layers, schema layer, instance layer, and situation layer (Ji, S.J., 2007). There could be many partial according to specific situation in reality. Those partial schemas need to be integrated into a global schema to simulate the reality as a whole or the cosmos. Since those schemas have many discrepancies and redundancies causing such problems as systematic management, update anomalies, and insertion anomalies (Ramakrishnan and Gehrke, 2003), their integration requires selection of appropriate grafting points before diverse resolution associated with those discrepancies and redundancies. The methodologies for database schema integration have much common with the knowledge schema integration we are pursuing. Schema integration is considered in two contexts such as database integration (Carlo Batini. 1986., Parent, C., Spaccapietra, S. 1998) and view integration (Navathe,s.b., Gadgil,s.g, 1982). Those studies are performed on targets of integration namely, object types and connector types, and focus only on partial schemas.

In this paper, we introduce methods of selecting suitable object for schema integration, to expand into a global schema. The comparison methods are conducted in three factors: names, properties and hierarchy structure.

2 KNOWLEDGE STRUCTURE OF AGENT

An entity class is characterized by its definitional and characteristic properties. Definitional property is used to express specialization of entity in class hierarchy (Sun Mi NOH, 2005). A class is defined by its definitional properties. Classes may have common properties and those classes organize into class hierarchy.

Definitional attributes of a class are specified on its associated specialization link in the class hierarchy. For example in Figure.1 as its specialization link (i.e. +life, mobility, +intelligence), the Living thing class has action as characteristic properties such as

Kim D. and Park J. (2008).
AN OBJECT SELECTION MECHANISM FOR SCHEMA INTEGRATION OF AGENT'S KNOWLEDGE STRUCTURE IN VIRTUAL REALITY.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - AIDSS*, pages 411-415
DOI: 10.5220/0001680004110415
Copyright © SciTePress

'reproduce()' & 'breathe()' and the Animal class has a characteristic property, i.e. 'perceive()'. And the Human class has a characteristic property, i.e. 'beget()'. Because the class hierarchy structures are complicatedly constructed with many entities, activities, and attributes, it is important for each instance and class to build on characteristic identity in itself for an organization of linking formation. Identity of an object class is recognized by its characteristic properties uniquely.

The characteristic attributes which are connected with subclasses may be omitted on link, because subclasses inherit the characteristic attributes of the ancestor class. Still those properties represent characteristic of each entity, those properties are important to select suitable object in order to integrate schema.

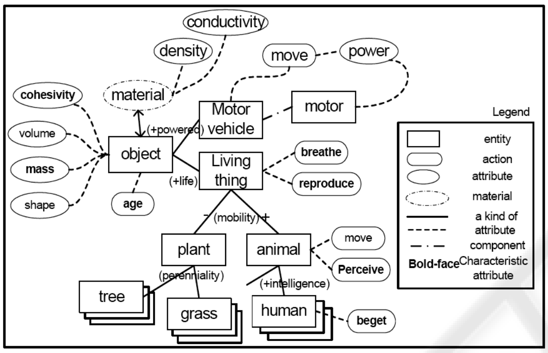

Figure 1: Structure of knowledge.

3 METHODS OF OBJECT SELECTION

In this chapter, we introduce the three comparison methods which can select the suitable entities for integration. That is, we describe the comparison method by the name of entity, the comparison of the property and the comparison method based on hierarchy structure.

3.1 Comparison by Name

This method finds identical nodes by comparing their names. The lexical difference in the names of two semantically identical nodes would be resolved by considering their synonymy (Miller, George A., Richard Beckwith, Christiane Fellbaum, Derek Gross and Katherine J. Miller, 1990). We start the comparison of the two schemas at the root node of the schema with the lower depth. Two nodes are judged to be identical if their synsets overlap by more than fifty percent.

This is the algorithm for object selection through the comparison of names.

```
/* Name compare*/
Function Same_name(Schema1, Schema2)
 while(Schema2_node !=NULL)
 // Starting the root node of schema1
  Point the root node of Schema1
  // Searching every node of schema1
  while(Schema1_node !=NULL)
    CompareString(Schema1_nodename,
                  Schema2_nodename)
    // if node name is same, then select the node
    if same the nodename
     then choose the node and break
    // compare to the node synset
    else CompareString(
    Schema1_nSynset,Schema2_nSynset)
    // if the node synset has similar name
         if similarity of the
           nodeSynsetname is found then
           choose the node and break
       else next the Schema1_node
    end while
  next the Schema2_node
 end while
end Function
```

We present an example on the basis of this algorithm. As shown in Figure.3, Woman entity in Figure.2(b) is selected as the start node for comparison with the entities in Figure.2(a). The Woman entity and Being entity are compared to each other in terms of their names and synsets. If the name does not match, then comparing continues with the next entity. In Figure.2(a), the Woman entity is chosen a suitable entity for schema integration by comparing with the synset of Female entity.

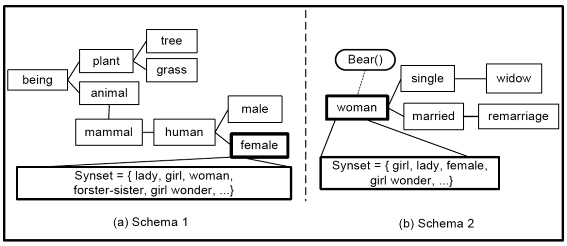

Figure 2: Comparison of Object name.

3.2 Comparison by Properties and Attributes

An entity class is characterized by the attributes and activities connected to it. We can judge the similar nodes by comparing their characteristic properties from the two schemas being evaluated. The characteristic properties are represented in terms of their associated attributes and activities which in turn are represented by their associated attributes. Each of these attributes is compared pair-wisely to find similar entities. Although they don't have the same characteristic attribute, if their property sets overlap by more than seventy percent then two nodes would be judged to be identical. This is the algorithm for object selection through comparison by attributes.

```
/* Attribute compare*/
Function Same_attribute(Schema1,Schema2)
  // Searching every node of schema2
  while(Schema2_node !=NULL)
    // Starting the root node of schema1
    Point the root node of Schema1
    while(Schema1_node !=NULL)
      CompareAttribute(Schema1_nattribute,S
                   chema2_nattribute)
       if same the characteristic attribute
       then choose the node and break
    // if property set of the node has similar objects
    else if similarity of the property
    set is found
    then choose the node and break
     // compare to the next node of schema1
     else next the Schema1_node
    end while
     next the Schema2_node
  end while
end Function
```

We will give an example based on this algorithm. In Figure.3(b), the schema of the Woman entity has an attribute set, i.e. attribute = {size, shape}, and characteristic attribute, i.e. {bear()}. Comparison of this entity starts from the root entity of the target schema Figure.3(a), in terms of the actions and attributes. If any attribute does not match for an entity, then comparison continues with its child entity's attributes. In the Figure.3, for example we may judge the Animal entity is similar to Woman entity because both entities include the same attributes such as 'size'& 'shape'. Since their characteristic attributes do not match however, further comparison with a child entity of Animal entity is attempted. Since Lady entity inherits all the other ancestor entities including Human entity, the system would estimate that the two entities have the same characteristic attribute i.e., 'bear()'. Therefore the two entities, Lady and Woman entities are selected as suitable entity pair for integration.

3.3 Comparison by Similar Construction and Hierarchy

We also can select the similar entity by means of comparing their hierarchical similarity such as class hierarchy. Figure.4 is the example of integrating the

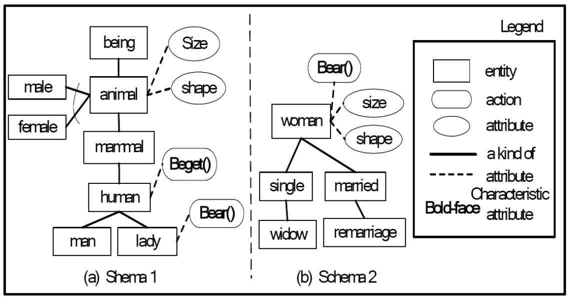

Figure 3: Comparison of Property.

two schemas. The knowledge structure of Figure.4(a) and Figure.4(b) encompasses concept and links of objects with their own information. However, Figure.4(a) depicts the hierarchy from 'Physical Object' with relations, on the other hand, Figure.4(b) shows the hierarchy of move() of animal. The comparison of hierarchical similarity of the two schemas starts from the root node based on their names and characteristic properties. If not similar, the comparison moves to the next entities until either reaching the end node or finding the similarity in the top-down fashion. We in particular consider links which have a meaning, i.e. '+life', 'mobility'.
The algorithm for object selection through the comparison by structure is following.

```
/* Structure compare*/
Function Same_construction(Schema1,
Schema2)
 Check the depth the two schemas
//if Schema1 has long depth
  while(Schema1_node !=NULL)
    // Starting the root node of schema1,schema2
    Point the root node of Schema1,
      Schema2
    // part of structure compare
    ComparetheLink
    (Schema1_node,Schema2_node)
     If same the number of Link the
      two schemas
    //use the attribute algorithm
    then call Same_attribute(Schema1,
   Schema2)
      If same the definitional attribute
       then select the node and break
      else next to Schema1_node
       next to Schema2_node
    else asking the comparison
       process continue
      if approval then next to
        Schema1_node
         next to Schema2_node
      else return abort
   end while
end Function
```

We present the example on the basis of this algorithm. In Figure.4, Living thing and Animal entities are intuitively not the similar entity because of different characteristic attribute despite of both having similar structure, that is, they are in the middle of the hierarchy and are connected by the same information of the link. There is special information for the comparison, link information. Animal entity, in Figure.4(b) has a characteristic attribute, i.e., move() and it can infer that Animal entity inherits 'mobility' property because 'mobility' should be in the upper class as the condition of the existence of 'move()'. Therefore, we can directly search the link of '+mobility' in Living thing entity, then follow the link. At the end of the link, there is Animal entity, as a result, the two entities are selected as suitable entity pair for integration.

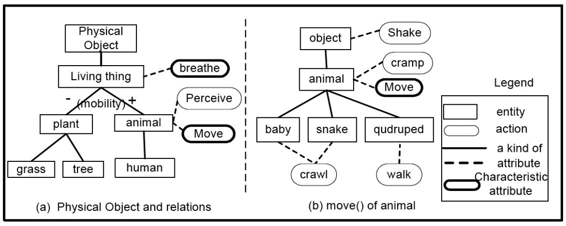

Figure 4: Comparison of similar construction and hierarchy.

3.4 Overall Flow of the Mechanism

The Figure.5 shows the flowchart of the overall mechanism by comparing their names and characteristic properties studied so far. In order to select a suitable node for integration, two schemas are selected, and then, estimated to the possibility of the integration by comparing their hierarchy structure. The algorithm in the following is based on three algorithms above. If a suitable selection of a node is not completed, another schema will be an input.

4 EXAMPLE

The Figure.6 shows the example about choosing the suitable nodes for integration. The Figure.6(a) and 6(b) are satisfactory to the condition of integration which has similarity class hierarchy structure. We can estimate the sequence of entities which are constructed by the two schemas through the comparison of names and links of upper and lower nodes and connection with characteristic attributes. The Animal entity is selected as the same entity by comparing their names. Then it is integrated like shown in Figure.6(c). The properties which are linked by the entities must operate the addition and

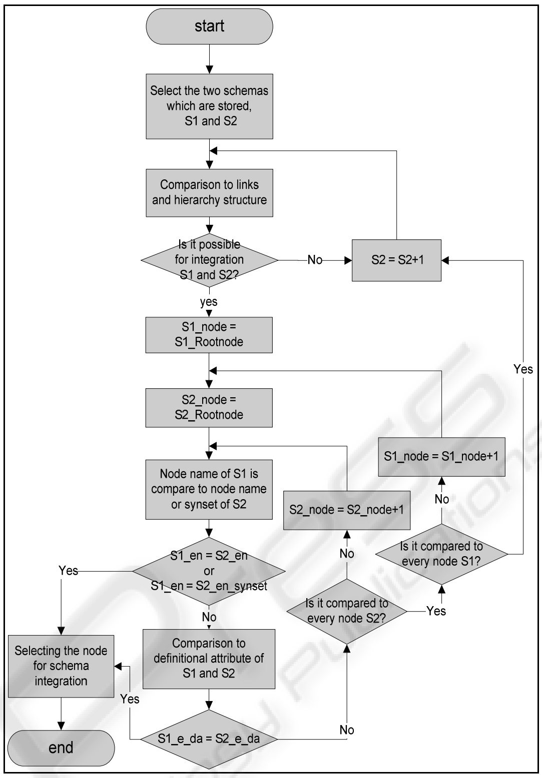

Figure 5: The flowchart of object selection.

deletion in order to avoid representing redundancy or omission. The Figure.6 indicates the problem of sequence of the entities which are generated when Figure.6(a) and Figure.6(b) are integrated. In Figure.6(c), the Human entity has the Baby entity as child entity because Human entity has a characteristic attribute such as 'beget()' and Baby entity has actions such as 'beget() & crawl()'. We can understand that child entities do not represent the properties which are inherited from ancestor entities.

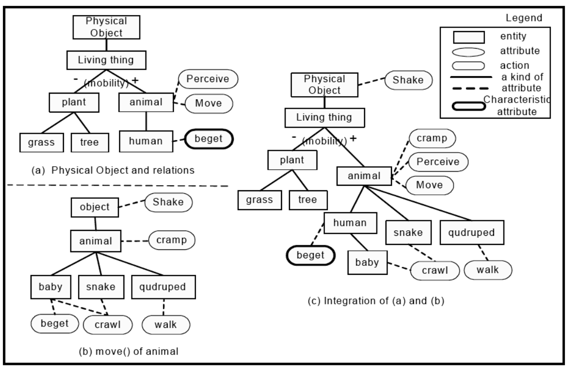

Figure 6: Example of object selection.

5 CONCLUSIONS AND FURTHER STUDIES

In this paper, we introduced the knowledge structure of agent which is constructed with class hierarchy and the methods of selecting suitable entities. We also proposed the methods for selecting suitable object by comparing their characteristic properties, similar names, and similar structure hierarchy. These methods of selecting entities for schema integration can solve the problems such as update anomalies, insertion anomalies.

In our further studies, we will delve constraints of diverse links and develop a schema integration tool.

REFERENCES

Gruber, T. R., 1993. A translation approach to potable ontology specification, *Knowledge Acquistion, vol.5, no.2, pp.199-200*

Park, J, 2004. *Ontology about the microcosm*, Tech. report, AIMM Lab., Kyungpook Nat'l Univ.,Feb.

Ramakrishnan, Gehrke, 2003. *Database Management Systems*, McGRAW-HILL Co., 3rd ed., pp606-607

Carlo Batini, Maurizio Lenzerini, Shamkant B. Navathe, 1986. A Comparative Analysis of Methodologies for Database Schema Integration. *ACM Comput. Surv. 18(4): 323-364 BibTeX*

Parent, C., Spaccapietra, S, 1998. Issues and Approaches of Database Integration. *Communication of the ACM. Vol. 41, No.5 166-178*

Navathe,s.b., Gadgil,s.g, 1982. A methodology for view integration in logical data base design. In Proceedings of the 8th International Conference on very Large Data Bases. *VLDB Endowment, Saratoga, Calif*

Sun Mi NOH, 2005. *A Knowledge Structure of Cyber-Microcosm Ontology for Efficient Representation*, Masters Thesis, Kyoungpook Nat'l Univ.

Miller, George A., Richard Beckwith, Christiane Fellbaum, Derek Gross and Katherine J. Miller, 1990. Introduction to WordNet : an on-line lexical database, *In : International Journal of Lexicography 3 (4).*

Ji, S.J., 2007. *A knowledge Model for simulating Human-like Behavior of virtual Inhabitant*, Tech. report, AIMM Lab., Kyungpook Nat'l Univ.,Dec.