Formalizing Ontology-Based Hierarchical Modeling Process of Physical World

Nan Wang[1,2], Dantong OuYang[1,*], and Shanwu Sun[3]

[1] Department of Computer Science and Technology, Jilin University, Changchun 130012
[2] Department of Information, Jilin University of Finance and Economics, Changchun, 130117
[3] Network and Laboratorial Center, Jilin University of Finance and Economics, Changchun, 130117
ouyangdantong@163.com

Abstract. The important step of model-based reasoning is to construct the abstraction model of physical world. In this paper, we introduce the concept of hierarchical ontology classes within the G-KRA model framework to realize the modeling knowledge sharing and reuse. Three kinds of ontology operators are defined and the formal process of them working is partly given. We also point out that constructing hierarchical ontology classes can be independent of the modeling process. Moreover we show how to automatically build up a hierarchical model of a world W based on the hierarchical ontology classes which is as well updated during the modeling. We expect that this work will rich the way to construct hierarchical model of physical world automatically and efficiently.

Keywords: G-KRA Model Hierarchical Ontology Classes Hierarchical Modeling Knowledge Sharing and Reuse.

1 Introduction

In resent years, researchers have paid a lot of attention to the abstraction of the physical world. The KRA model [1,2,3] is proposed to help both the conceptualization phase of a problem and the automatic application of abstraction operators. The extended model, called G-KRA model (General KRA model) [4] can represent the world from different abstraction granularities. Ontology has been proposed as a specification mechanism to enhance knowledge sharing and reuse across different applications [5] and many researchers has defined ontology in different ways [6,7,8,9,10].

In the second section of this paper, we explore the levels included in the KRA model, particularly the perception level, the language level and the theory level, so that the representing framework is simplified by knowledge sharing and reuse. We introduce the concept of hierarchical ontology class. The abstraction operators acting on the ontology classes as well as the mappings between them are also defined. In the following section, we discuss constructing process of hierarchical ontology classes and formalizing the ontology-based hierarchical modeling process of the physical world .We will make a short conclusion to present work and next-step research in the last section.

* Corresponding author.

F.L. Wang et al. (Eds.): AICI 2010, Part I, LNAI 6319, pp. 18–24, 2010.

2 Introducing Ontology in the KRA Model

The KRA model represents the world from four levels: Perception, Structure, Language and Theory (see [1] for details). In the G-KRA model, the abstraction model can be automatically constructed by using the abstraction objects database and the abstraction mapping defined manually. We introduce the concept of ontology and extend the abstraction objects database to Ontology Class (OC) which can be defined as OC=(E,L,T), such that E={OBJ,ATT,FUNC,REL}. The elements of E and the rest of OC mean the same as the ones in [3]. OC exists like database to realize knowledge sharing and reuse. There are two kinds of ontology class according to the constructing process: Fundamental Ontology Class (FOC), Abstraction Ontology Class (AOC). The former is generated based-on the physical world W, whereas the latter through operating on another different ontology class.

The ontology-based KRA model framework can be represented as follows: $R=(P^*,S,L^*,T^*)$, in which the three parts of P^*,L^*,T^* represent not the actual contents but the mapping to some OC, whereas OBS and S remain for being involved with the real world.

We define three kinds of ontology abstraction operators (sets): fundamental operator Ψ, entity operator set $\Omega=\{\omega_E,\omega_L,\omega_T\}$ and connection operator set $\Phi=\{\lambda_E,\lambda_L,\lambda_T\}$, to create or extend the ontology class. The operation of all the operators can be independent of modeling process. Operator Ψ applies to some physical world W and FOC, which specifies or updates FOC, that is $\Psi(W,FOC_1)=FOC_2$. The arguments of the operator set Ω are an existing ontology class and in particular, Ω combines a grouping of indistinguishable entity types of an ontology class OC_1 into one entity type in an ontology class OC_2, $\Omega(OC_1)=OC_2$. The connection operator set also acts on an ontology class and it aggregates a set of entity types of OC_1 and the relations between them to form a new compound entity type of OC_2, $\Phi(OC_1)=OC_2$.

```
----------------------------------------------------------------------------------------------------
Procedure FundamentalOperatorΨ WORLD W, ONTOLOGY-CLASS FOC
For every entity e of W
  For every e* of FOC{
  if SMP(e, e*) and BMP(e, e*) exit;// SMP and BMP are two procedures to do the
structure matching and behavior matching , see details in [11];
  type=TYPE(e);//Define a new entity type
  att=ATT(e); //Define the attribute set of e
  func=FUNC(e);//Define the behavior set of e
  rel=REL(e); //Check the entities related to e and construct the relation rel;
  Add (type,att,func,rel) to FOC.E; //Add a new entity type to FOC.E
  Add Language(e) to FOC.L;
  Add Theory(e) to FOC.T;// Add a new reasoning ontology of FOC about e
}
```

Fig. 1. Constructing the Fundamental Ontology Class

```
---------------------------------------------------------------------------------------------
Procedure EntityOperatorSetΩ ONTOLOGY-CLASS OC1, ONTOLOGY-CLASS OC2 //OC2=Ω(OC1)
   EntityOperatorωE(OC1.E, OC2);
   EntityOperatorωL (OC1.L, OC2);
   EntityOperatorωT (OC1.T, OC2);

Procedure EntityOperatorSetΦ ONTOLOGY-CLASS OC1, ONTOLOGY-CLASS OC2//OC2=Φ(OC1)
   EntityOperatorλE (OC1.E, OC2);
   EntityOperatorλL (OC1.L, OC2);
   EntityOperatorλT (OC1.T, OC2);

Procedure EntityOperatorωE ONTOLOGY-CLASS_E E, ONTOLOGY-CLASS OC2
//The operation process of operator ωE
Do while there exist(s) unchecked entity(ies) in E{
  Entities={}; // The set of indistinguishable entity types.
   For every unchecked entity E1 of E{
   Entities={E1};
      For every unchecked entity E2 of E-{E1}
        if SMP(E1,E2) and BMP(E1,E2) Entities= Entities∪{E2};
   }
   //Search the indistinguishable entity types of E.

   if |Entities|>1 {
      Replicate E1 to a new entity E*;
      If there is no entity type E* in OC2, add E* to OC2;
      Update RELs of OC2.E using the RELs of E and replace every entity type E′ of
OC2.E with E* if E′∈Entities;
      Update the entity mapping EM from OC2.E to E;
   }
     //If there exit more than one indistinguishable entity types, do the combining.

   if |Entities|=1{
      Update OC2.E using E;
      Update the entity mapping EM from OC2.E to E;
   }
     //If there is no other entity type indistinguishable with E1, do the updating.
 }
---------------------------------------------------------------------------------------------
```

Fig. 2. Partial Process of Operators sets Ω and Φ

There are two relationships between ontology classes: entity mapping(EM) and connection mapping(CM), which make one(/more) object(/s) of ontology class OC_1 relating to one(and only one) object of OC_2. The mapping relationships can be constructed during the hierarchical process of ontology classes.

3 The Hierarchical Modeling Process of Physical World

The very important step of model-based reasoning is to construct the model of the real world. And the hierarchical model can help to reduce the complexity of the reasoning. The ontology-based KRA model framework makes it possible to automatically generate such a hierarchical model of the world W.

3.1 The Hierarchical Process of Ontology Classes

Fundamental operator Ψ acquires knowledge directly from some world W and it perceives all entities in W, including their types, attributes, behaviors, and the relations between them. See Fig 1 for details.

Both entity operator set $\Omega=\{\omega_E,\omega_L,\omega_T\}$ and connection operator set $\Phi=\{\lambda_E,\lambda_L,\lambda_T\}$ act on an existing ontology class. Fig 2 shows the partial process. We only give the algorithm that the first operator ω_E of Ω, because λ_E operates just like ω_E only not searching a set of indistinguishable entity types but finding out a set of interrelated ones. The rest operators, based on the results of ω_E *and* λ_E , are just the process which deals with variables replacing and expressions aggregating of one-order logic.

We note that while using operator λ_E, the connection mapping CM from a compound entity type to a set of interrelated entities is constructed or updated. Through alternate operating of the ontology abstraction operators (sets), the hierarchical ontology classes can be built up automatically, Fig 3 shows the process.

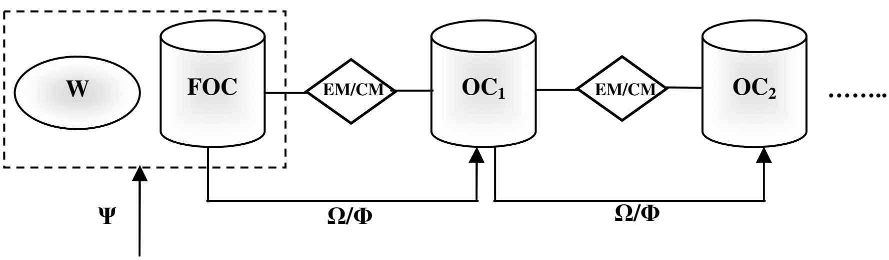

Fig. 3. Constructing the Hierarchical Ontology Classes

3.2 Hierarchical Modeling of Physical World Based-on Ontology Classes

The hierarchical ontology classes are generated or extended independently of the modeling process of the physical world. The more kinds of worlds are input, the more knowledge ontology classes can be achieved. Actually, while modeling a world W within the simplified G-KRA framework, the ontology classes can also be updated dynamically. Fig 4 shows the hierarchical modeling process of physical world based-on ontology classes.

It is noted that the definition of condition C can specify the abstraction degree of the hierarchical model. We can set a threshold value like the number of the entity types in some ontology class and if the number is lower than the threshold value, the procedure should be stopped. Fig 5 shows the hierarchical modeling process of W based-on hierarchical ontology classes.

```
--------------------------------------------------------------------------------------------------
Procedure HierarchicalModeling WORLD W, ONTOLOGY-CLASS-SET
{FOC,OC_1,…,OC_n}
  //Construct fundamental model R_g=(P_g*, S_g, L_g*,T_g*) by using FOC;
For every entity type e of W {
   if (∃ e_1 ∈ FOC.E and e_1 is indistinguishable with e)
   Create a mapping M_e from P_g* to the entity type e_1 of FOC;
   else Add the entity type e to FOC and Create a mapping M_e from P_g* to it;
   }
    //Create the mapping from the entity types of W to the entity types of FOC

For every type of relations r in W {
   if (∃ REL_1 ∈ FOC.E and REL_1 is the same type of relation as r and no mapping on
REL_1)
   Create a mapping M_r in P_g* to REL_1;
   else Add the relation type r to P_g* and Create a mapping M_r in P_g* to it;
 }
    //Create the mapping from the relation types of W to the relation types of FOC

Manually add new language and theory involved with W to FOC.L and FOC.T if need
be and create mappings from L_g*, T_g* to FOC.L, FOC.T;
//Constructing the fundamental model of W

//Generate hierarchical models R_OC1,…R_OCn, and what abstraction degree of the model
//is our satisfaction can be specified by a condition C defined in advance.
i=1;
do while C is satisfied {
   ROC_i=ROC_i-1; //define ROC_0 as Rg
   Adjust all mappings in R_OCi to OC_i according to the entity mapping EM or the
connection mapping CM between R_OCi and R_OCi-1;
   i=i+1;
}
--------------------------------------------------------------------------------------------------
```

Fig. 4. Hierarchical Modeling Process of Physical World Based-On Ontology Classes

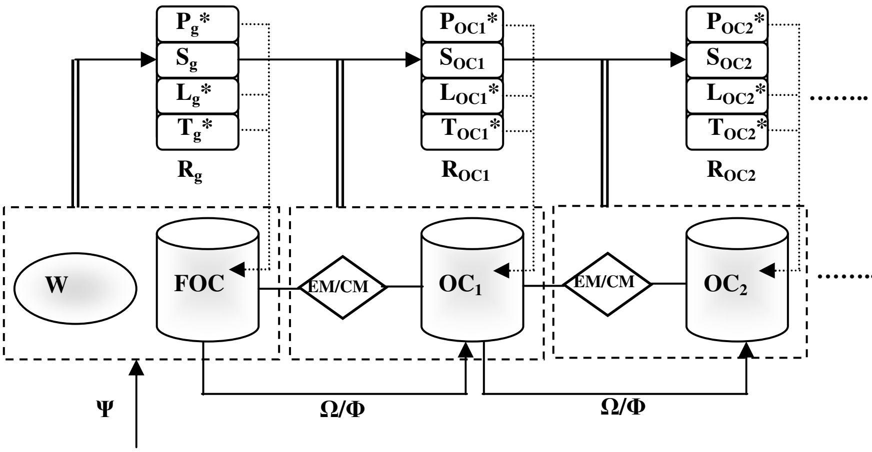

Fig. 5. Hierarchical Modeling Process of W Based-On Hierarchical Ontology Classes

4 Conclusion

This work explores how to use the concept of hierarchical ontology classes to realize the modeling knowledge sharing and reuse and then to automatically construct the hierarchical model of physical world. The KRA model framework is be simplified by replacing actual contents with the mapping to some ontology class. At the same time, the hierarchical ontology classes can be built up independently of the modeling process. Here we only concern about the static world and will continue to work on the modeling of the dynamic world in future.

Acknowledgment

The authors are grateful to the support of NSFC Major Research Program under Grant Nos. 60496320 and 60496321, Basic Theory and Core Techniques of Non Canonical Knowledge; NSFC under Grant Nos. 60973089,60773097 and 60873148; Jilin Province Science and Technology Development Plan under Grant Nos.20060532 and 20080107.Erasmus Mundus External Cooperation Window's Project (EMECW): Bridging the Gap, 155776-EM-1-2009-1-IT-ERAMUNDUS-ECW-L12.

References

1. Saitta, L., Zucker, J.: Semantic Abstraction for Concept Representation and Learning. In: Proc. SARA, pp. 103–120 (1998)
2. Saitta, L., Zucker, J.-D.: A Model of Abstraction in Visual Perception. Applied Artificial Intelligence 15(8), 761–776 (2001)
3. Saitta, L., Torasso, P., Torta, G.: Formalizing the abstraction process in model-based diagnosis. In: Tr cs, 34. Univ. of Torino, Italy (2006)

4. Shan-wu, S., Nan, W., Dan-tong, O.: General KRA Abstraction Model. Journal of Jilin University (Science Edition) 47(3), 537–542 (2009)
5. Neches, R., Fikes, R., Finin, T., Gruber, T., Senator, T., Swartout, W.: Enabling technology for knowledge sharing. AI Magazine 12, 36–56 (1991)
6. Uschold, M.: Knowledge level modelling: concepts and terminology. The Knowledge Engineering Review 13(1), 5–29 (1998)
7. Guarino, N., Giaretta, P.: Ontologies and Knowledge Bases: Towards a Terminological Clarification. In: Mars, N. (ed.) Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, pp. 25–32. IOS Press, Amsterdam (1995)
8. Gruber, T.: Towards principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies 43(5/6), 907–928 (1995)
9. William, S., Austin, T.: Ontologies. IEEE Intelligent Systems, 18–19 (January/February 1999)
10. Chandrasekaran, B., Josephson, J.R., Benjamins, V.R.: What Are Ontologies, and Why Do We Need Them?, pp. 20–25 (January/February 1999)
11. Wang, N., OuYang, D., Sun, S., Zhao, C.: Formalizing the Modeling Process of Physical Systems in MBD. In: Deng, H., Wang, L., Wang, F.L., Lei, J. (eds.) AICI 2009. LNCS, vol. 5855, pp. 685–695. Springer, Heidelberg (2009)