An Evaluation Method for Ontology Complexity Analysis in Ontology Evolution*

Dalu Zhang[1], Chuan Ye[2], and Zhe Yang[3]

Department of Computer Science and Technology, Tongji University, Shanghai 200092, China
daluz@ieee.org, shtjjsjyjx@126.com, hatasen@163.com

Abstract. Ontology evolution becomes extremely important with the tremendous application of ontology. Ontology's size and complexity change a lot during its evolution. Thus it's important for ontology developers to analyze and try to control ontology's complexity to ensure the ontology is useable. In this paper, an evaluation method for analyzing ontology complexity is suggested. First, we sort all the concepts of an ontology according to their *importance degree* (a definition we will give below), then by using a well-defined metrics suite which mainly examines the concepts and their hierarchy and the quantity, ratio of concepts and relationships, we analyze the evolution and distribution of ontology complexity. In the study, we analyzed different versions of GO ontology by using our evaluation method and found it works well. The results indicate that the majority of GO's complexity is distributed on the minority of GO's concepts, which we call "important concepts" and the time when GO's complexity changed greatly is also the time when its "important concepts" changed greatly.

1 Introduction

Ontology construction and development are necessarily an iterative, dynamic and parallel process [1]. The change of domain, the adaptation for different application background, the re-realization of the domain and the change of the conceptual world, all these would influence the ontology construction and evolution [2]. During ontology evolution, the size and complexity of ontology change a lot, which bring difficulties to the management and maintenance of ontology. Actually, it is for the reason of complexity, formal ontologies which are focused on in the field of ontology research have in pratice, yielded little value in real-world application [3].

Though ontology complexity analysis is important in ontology development and evolution, we find few effective methods or metrics related with ontology complexity. In this paper, we suggest an evaluation method for ontology complexity analysis. First, we make a definition of "importance degree", then we present a well-defined complexity metrics suite, which mainly examine the quantity, ratio and correlativity of concepts and relations. After that we sort all the concepts of GO[4] according to their importance degree and by using the formerly defined metrics suite we analyze the trend of GO's complexity evolution and distribution. The reason we choose GO as

* Supported by National Natural Science Foundation of China under Grant No. 90204010.

S. Staab and V. Svatek (Eds.): EKAW 2006, LNAI 4248, pp. 214 – 221, 2006.

our experimental object is: GO is an ontology that has tremendous application in the domain of gene and it has a full collection of varied versions since 2002.

The rest of this paper is structured as follows: section 2 presents related works about ontology evaluation metrics. In section 3 we suggest our evaluation method. Section 4 presents the complexity analysis results of GO. Section 5 presents the conclusion and outlook for future works.

2 Related Works

As far as now, most existing ontology evaluation metrics are used to analyze the description ability of ontology, few of them are focused on complexity issue.

In literature [5], authors present a structure complexity measure for the UML class diagram based on entropy distance. It considers complexity of both classes and relations between classes and presents rules for transforming complexity value of classes and different kinds of relations into a weighted class dependence graph. This method can measure the structure complexity of class diagrams objectively. Idris studied two conceptual integrity metrics based on graph theory in his PhD thesis [6], which are conceptual coherence and conceptual complexity, but these metrics are all characteristics of single concept. Chris Mungall researched the increased complexity of Gene Ontology [7]. He measured the average number of paths-to-top of a term and used the path-to-term ratio to measure the complexity in an ontology. This metric is simple and only shows the evolution of ontology complexity. Though evaluation methods in [5], [6] and [7] are for ontology complexity analysis, the correctness and soundness of their methods are not well verified.

In our evaluation method we analyze both ontology's complexity evolution trend and distribution trend. Before analysis, we sort all the concepts in ontology according to their importance degree, it is our consideration that different concepts have different contribution to the complexity of the whole ontology, concepts which have more relations with others may be more "important", thus have a higher contribution.

Literature [8] and [9] research methods of excavating important concepts in an ontology. Idris, author of literature [8] suggests several metrics that measure the core degree of concepts based on graph theory and according to the experimental result the author suggests that "betweenness centrality" is comparatively reasonable. But his experimental ontology is small, further more, according to the definition of "betweenness centrality", the core degree of all the leaf concepts is 0, this is not reasonable. The method in [9] is mainly used in extracting ontology from text and not appropriate for constructed ontology. Thus, we gave our definition of importance degree, which is the basis of our evaluation method for ontology complexity analysis.

3 Proposed Evaluation Method

3.1 Measurement of Concept's Importance Degree

The following are some definitions related to concept's importance degree:

$\boldsymbol{C} = \{c_1, c_2, \ldots, c_m\}$: the set of m concepts defined explicitly in an ontology.

$\boldsymbol{R} = \{r_1, r_2, \ldots, r_m\}$: the set of number of relations each concept has, equals to the outdegree of a concept. Here we only consider those inherited relations that reflect the hierarchy of concepts, such as "is a", "part of", etc.

In an ontology, concepts hierarchy is typically expressed in DAG(directed acyclic graph) showed in Fig 1. Each node represents a concept and each directed arc represents a subtype relation to present the hierarchical structure between concepts in ontologies.

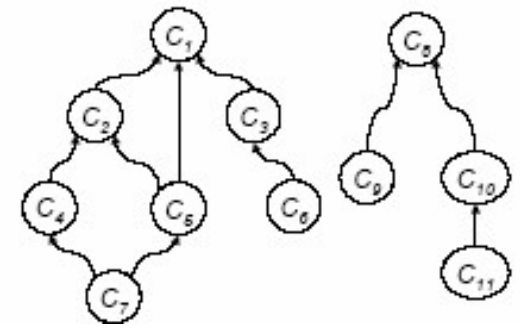

Fig. 1. Ontology in DAG

Path : A distinct trace in DAG from a specific particular concept to the most general concept in the ontology, which is the concept without any parent or superclass(e.g. c_7—c_5—c_2—c_1 in Fig 1). In Fig 1, there are 11 concepts(m=11), and c_1,c_8 are the two most general concepts.

$\boldsymbol{N} = \{n_1, n_2, \ldots n_m\}$: the set of number of Paths each concept in an ontology has.

$\boldsymbol{P_i} = \{p_{i,1}, p_{i,2}, \ldots p_{i,ni}\}$: the set of Paths a particular concept c_i has.

$\boldsymbol{PL_i} = \{pl_{i,1}, pl_{i,2}, \ldots, pl_{i,ni}\}$: the set of path length of a particular concept c_i, length of a particular path is the number of edges that appear in the path.

$\boldsymbol{AvgPl_i} = \sum_{j=1}^{n_i} pl_{i,j} / n_i$: the average path length of a particular concept c_i.

$\boldsymbol{P} = \{\{p_{1,1}, p_{1,2}, \ldots p_{1,n1}\}, \ldots \{p_{m,1}, p_{m,2}, \ldots p_{m,nm}\}\}$:the set of the set of paths of each concept in an ontology.

$\boldsymbol{IsIn}(\boldsymbol{c}_i, \boldsymbol{p}_{j,k}) = 0$（if $\boldsymbol{c}_i$ doesn't appear in $\boldsymbol{p}_{j,k}$）or 1（if $\boldsymbol{c}_i$ appears in $\boldsymbol{p}_{j,k}$）.

$\boldsymbol{IDC_i}$（Importance Degree of Concept）$= \sum_{p_{j,k}}^{P} IsIn(ci, p_{j,k})$:a particular concept c_i's importance degree $\boldsymbol{IDC_i}$ is the number of times c_i appears in all the paths of all the concepts in an ontology.

After calculating all the concepts' IDC, we can sort the concepts according to their IDC by an descending order, concepts with the same IDC should be sorted according to their AvgPl by an ascending order.

See Fig 2, by using the sorting method above we have the result as table 1.

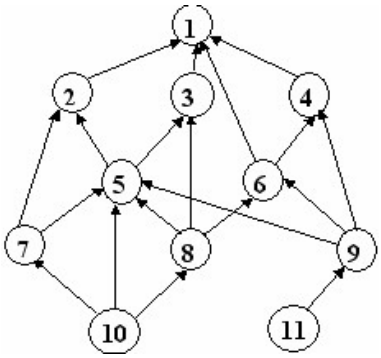

Fig. 2. An Example

Table 1. Sorting Result

id	1	5	2	3	6	8	9	10	4	7	11
IDC	35	16	11	11	10	10	10	10	8	6	5

From this example we can see: concept with id=1 is the most important, because it is the most general concept and appears in all the paths of all the concepts; concept with both higher outdegree and indegree may be more important because it has a higher chance to appear in other concepts' path set(see concept with id=5); concept that locates at a higher hierarchy may be more important, because it has a higher chance to appear in other concepts' path set, but this is not absolutely right (see the concept with id=4); leaf node concept that locates at the lowest hierarchy may be less important, because it can't appear in other concepts' path set, but this is not absolutely right (see the concept with id=10, its IDC is not the lowest because it has an outdegree of 3).

Compared with the betweenness centrality metric introduced in literature [8], which set all the leaf node concepts' betweenness centrality as 0, this importance degree measurement method is more reasonable.

3.2 Ontology Complexity Metrics

On the basis of 3.1, we present the definitions of our ontology complexity metrics.

TNOC(Total Number Of Concepts)=|C|=m: the number of concepts in the set C.

TNOR(Total Number Of Relations)= $\sum_{i=1}^{m} r_i$: the sum of the number of relations of each concept in an ontology.

TNOP(Total Number Of Paths)= $\sum_{i=1}^{m} n_i$: is the sum of paths of each concept.

As ontology consists of concepts and relations, TNOC and TNOR are the two basic attributes of ontology, we can see the change of basic size of an ontology by analyzing these two attributes. As path consists of relations and can reflect the inner structure and hierarchy of ontology, TNOP represents an ontology's hierarchical

complexity and its value is proportionate to difficulties in navigating and visualizing the ontology[7].

μ =TNOR/TNOC: the average relations per concept in an ontology.

ρ =TNOP/TNOC: the average paths per concept in an ontology.

μ indicates the average connectivity degree of a concept. ρ must be equal to or greater than 1(each concept must have a parent except for the most general concept). If ρ =1, then the ontology is a tree, multi-relation concepts(higher μ ratio) result in higher ρ ratio for an ontology.

The analysis object of the metrics defined above is the whole ontology, the following are definitions of complexity metrics that are used for single concept in ontology.

d_i: the degree of a particular concept c_i(equals to the sum of c_i's indegree and outdegree). It represents the connectivity degree of a particular concept c_i.

n_i: the number of paths of a particular concept ci. It represents a particular concept ci's hierarchical complexity and its value is proportionate to the difficulty in visualizing the concept and its relations with other concepts in an ontology.

3.3 Our Evaluation Method

The main steps of our method are as follows:

1.Get all the evolution versions of an ontology, transform all of them into DAG (Each node represents a concept and each directed arc represents a subtype relation to present the hierarchical structure between concepts in an ontology.)

2. Sort the concepts of every single ontology by our sorting method. Calculate the d_i and n_i value of every concept c_i .

3. For all the evolution versions of the ontology, calculate their *TNOC*, *TNOR*, *TNOP*, μ and ρ value.

4. By using the result of step 2 we can analyze the complexity distribution of a single ontology, by using the result of step 3 we can analyze the complexity evolution of varied versions of ontology, and by analyzing both the results of step 2 and step 3 contrastively we can research deep into the internal cause of ontology complexity evolution and find the relations between complexity evolution and distribution.

4 Experimental Results and Conclusions

We measured and analyzed the growing complexity of different versions of GO [7] from Dec. 2002 to Sep. 2005, in which the obsolete terms are eliminated from our statistics.

4.1 GO's Complexity Evolution Statistics and Analysis

As GO evolves, its TNOC, TNOR and TNOP increases. Fig 3 is the quantity evolution of TNOC, TNOR and TNOP. The left Y-axis shows the increase of

TNOP. The right Y-axis shows the increase of TNOC and TNOR. The lines of TNOC and TNOR indicate they increase at a steady but slow rate. The line of TNOP indicates that it has a steady but rapid growth. Some enormous ladderlike increases take place at the time of Mar. 2003, Dec. 2004, Jan. 2005 and Apr. 2005, which shows that at these times GO went through some larger changes and became more complex.

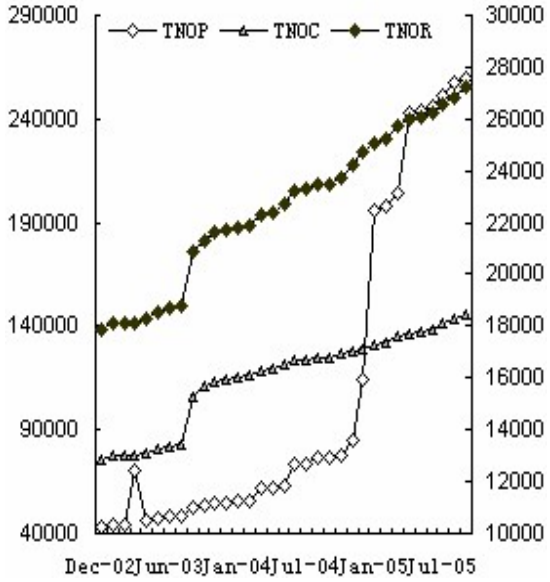

Fig. 3. Concept&Relation&Path

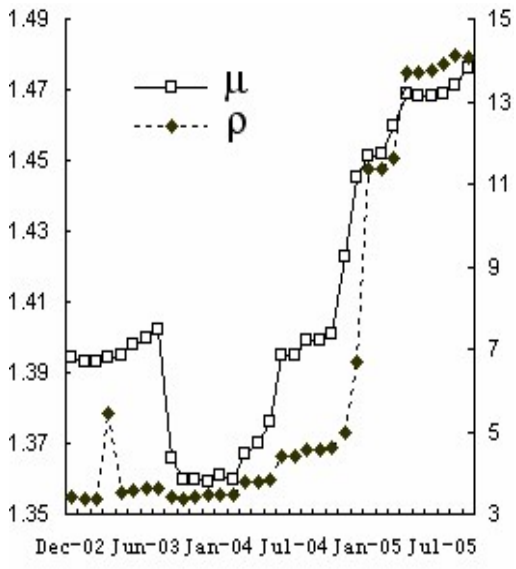

Fig. 4. μ & ρ

Fig 4 shows the changes of GO's μ and ρ from Dec. 2002 to Sep 2005. The left Y-axis shows the change of μ and the right Y-axis shows the change of ρ. In Fig 4, the line of μ indicates that μ has a relatively small change, basically between 1.35 and 1.50. The line of ρ indicates that ρ increases at a steady and rapid rate. Some enormous ladderlike increases occur at the time of Mar. 2003, Dec. 2004, Jan.2005 and April 2005. If we compare the two lines of TNOP in Fig 3 and ρ in Fig 4, we can find that they look almost the same and their enormous ladderlike increases occur at the same time. This is mainly because that *TNOP* is the result of $\rho \times TNOC$, while TNOC increases at a steady and a slow rate.

From the above analysis we conclude that the size of GO grew at a slow and steady rate, its complexity grew at a slow and fast rate with some sharp increasing points.

4.2 GO's Complexity Distribution Statistics and Analysis

We sort all the concepts according to their importance degree in all the GO versions from Dec. 2002 to Sep. 2005 and calculated their 70% number of paths distribution. Fig 5 is the result.

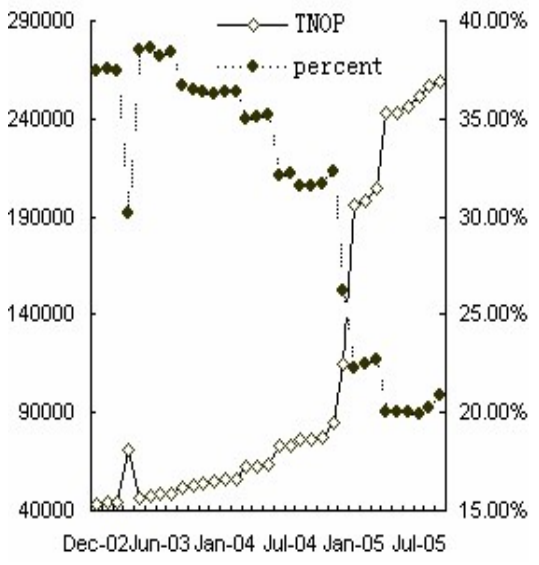

Fig. 5. Path distribution statistics

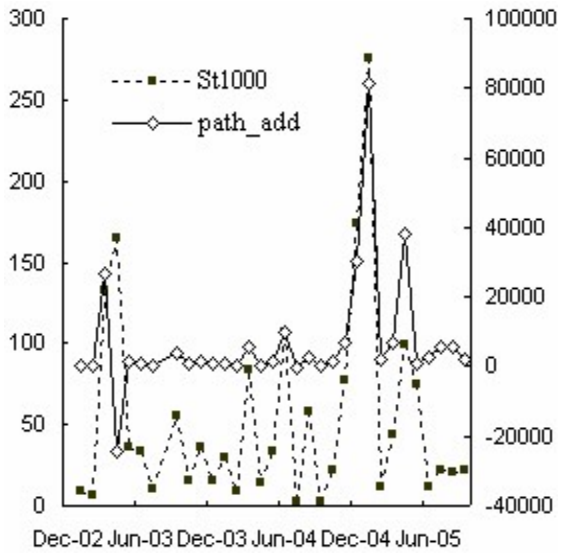

Fig. 6. Concepts&Paths Change

In Fig 5, the left Y-axis shows the increase of TNOP, the right Y-axis shows the change of "percent". The meaning of "percent" is: 70% of the total number of paths of a specific GO version is distributed on the first "percent" sorted concepts. It shows that percent decreases from 37.44% in Dec. 2002 to 20.85% in Sep. 2005, the line of "percent" indicates that it decreases steadily with some sharp decreasing points. Examine the two lines of TNOP and "percent", we see that they changed synchronously and the sharp increasing time of TNOP is also the sharp decreasing time of percent. The time points are: Mar. 2003, on which TNOP suddenly increased from 43776 to 70582 while percent suddenly decreased from 37.46% to 30.17%, Apr. 2003, Dec. 2004 and Jan. 2005. In some relatively smaller changes of TNOP and percent the time is also the same. The above analysis shows that every sharp increase of GO's TNOP leads to further concentration of GO's paths on important concepts.

We conclude that the majority of GO's complexity is distributed on the minority of its concepts which we call "important concepts" and as GO evolves, this trend becomes more and more evident.

In Fig 3, increasing forms of TNOC and TNOR can't explain increasing form of TNOP, so we have Fig 6. In Fig 6, the left Y-axis shows the change of St1000, the right Y-axis shows the change of path_add. We examine the first 1000 concepts of all the sorted GO versions, and St1000 represents the number of concepts that appear in a specific version of GO but not appear in the GO version that is one month before, path_add represents the change of TNOP of a specific GO version comparing with the GO version that is one month before, path_add below 0 indicates that there is a decrease of TNOP. From Fig 6 we can see that when the absolute value of path_add is small, St1000 is small too, and the time when path_add changes greatly is also the time when St1000 changes greatly. Some relatively sharp points of path_add occur at Mar. 2003(path_add=26806), Apr. 2003(path_add=-24637), Jun. 2004(path_add= 10122), Dec.2004(path_add=30065),Jan. 2005(path_add=81441),Apr. 2005(path_add =38317), and the corresponding values of St1000 are relatively large too, they are 132, 165, 107, 174, 276, 99(values of other times are much smaller, mostly no more

than 50). There is only one exception, in May 2005, path_add is small(616), but St1000 is large(75).

From the analysis we have the conclusion that the time when GO's complexity changed greatly is also the time when its "important concepts" changed greatly. The great change of GO's complexity may be caused by the appearance of new important concepts, another possibility is that because of the re-realization of domain, the whole ontology structure changes, so the order of important concepts changes greatly. In the process of ontology engineering, we shall consider more on those important concepts, which is a major cause of the great change of ontology complexity.

5 Summary and Future Works

With the tremendous use of ontology, its size and complexity change a lot during its evolution. So it becomes very necessary to set up a suite of metrics for developers to understand the complexity evolution and distribution of ontologies in order to improve the quality, estimate cost and reduce future maintenance. In this study, an evaluation method for analyzing ontology's complexity is presented. By using this method, we had a detailed statistics and analysis of GO's complexity evolution and distribution. In the future, we will continue to work on the ontology complexity metrics and other ontology metrics and the improvement of measuring concept's important degree (e.g. consider more on ontology's hierarchy).

References

1. A.Das, W. Wu, D.McGuinness. Industrial Strength Ontology Management. The Emerging Semantic Web, IOS Press, 2002.
2. E Daniel,O Leary. Impediments in the use of explicit ontologies for KBS development. International Journal of Human-Computer Studies,1997,46(2-3):327-337.
3. Amit Sheth and Cartic Ramakrishnan. Semantic (Web) Technology In Action:Ontology Driven Information Systems for Search, Integration and Analysis. IEEE Data Engineering Bulletin, Special issue on Making the Semantic Web Real, December 2003, pp. 40-48.
4. The Gene Ontology Homepage, http://www.geneontology.org/
5. Dazhou Kang, Baowen Xu, Jianjiang Lu,William C.Chu. A Complexity Measure for Ontology Based on UML. 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04)pp.222-228
6. IdrisHis. Analyzing the Conceptual Coherence of Computing Applications Through Ontological Excavation. PhD Thesis Proposal, May 13, 2004.
7. Chris Mungall, BDGP / GO Consortium, Increased complexity in GO. http://www. fruitfly.org/~cjm/obol/doc/go- complexity.html
8. Idris His, Colin Potts, Melody Moore. Ontological Excavation: Unearthing the core concepts of the application. Proceedings of WCRE2003, November 13-16,2003,pp. 345-352.
9. Mustapha Baziz,Mohand Boughanem,Nathalie Aussenac-Gilles,Claude Chrisment. Semantic Cores for Representing Documents in IR. SAC'2005- 20th ACM Symposium on Applied Computing, Santa Fe, New Mexico, 13-17 mars 2005. p. 1020-1026, ACM Press ISBN: 1-58113-964-0, USA