

Automatic Construction of RSM Based on XML

Lei He

Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences China
helei@kg.ict.ac.cn

Abstract — A Resource Space Model is a semantic model to organize, locate and operate Web resources in a multi-dimensional resource space. It's easy for users to understand the resource space and locate resources in it because a resource space is constructed based on the classification semantics. In general, resource spaces models are manually designed and constructed based on domain knowledge and resource analysis. Human factors, such as personal opinions, knowledge level and design skill, will influence the design result of a resource space. To reduce the difficulties of the manual design and ease the designing process, this work studies the issues of automated creation of a resource space and proposes a general method to automatically construct resource spaces from XML files.

I. INTRODUCTION

A. The Basic Concepts of Resource Space Model

A Resource Space Model (RSM) is a semantic data model for specifying, storing, managing and locating contents of Web resources based on a appropriate classification on the contents of resources [1][2][3]. Users can query and locate the resources in a resource space based on the classification coordinates of the resource space. It's easy for users to understand and manipulate a resource space based on the classification semantics since the classification is not only an approach to efficiently managing resources but also a basic method for human to percept the real world [7][8].

A resource space is a multi-dimensional space in which each point has its corresponding coordinates that map into a relevant resource set (maybe an empty set). In an n-dimensional resource space, each axis has a number of coordinates that can be flat or can have a certain hierarchy. Axes and the coordinates reflect the classification semantics of the resources. The distribution of resources depends on the classification information. Resources with the same classification semantics are located in the same point according to their classification coordinates in the space.

A Resource Space Schema is a five-tuple $\{RS, A, C, S, dom\}$ that defines the structure of the space [3]:

- (1) RS is the name of the resource space;
- (2) $A = \{X_i | 1 \leq i \leq n\}$ is the set of the axes;
- (3) $C = \{C_{ij} | C_{ij} \in X_i, 1 \leq i \leq n\}$ is the set of the coordinates;
- (4) S is the power set of the domain ontology;
- (5) dom is the mapping from the axes set A and the coordinates set C to S . $dom: A * C \rightarrow S$, for any axis $X_i = \{C_{i1}, C_{i2}, \dots, C_{ip}\}$, $dom(X_i, C_{ij}) = V_{ij}$, $V_{ij} \in S$, where $1 \leq i \leq n$, $1 \leq j \leq p$.

B. Motivation of automatic construction

The approach to design and create the resource space is as following steps [16]:

- (1) *Resource analysis*. It is to determine the application scope, to explore the resources to be managed, and then to specify the resources in a Resource Dictionary.
- (2) *Top-down resource partition*. Different designers can have different opinions of the resource partition, so we need a unified method on the resource partition on the top level.
- (3) *Design a low dimensional resource space* (e.g., a 2-dimensional resources space). A low-dimensional space is easier to handle. So, we first design low dimensional resource spaces, then integrate low-dimensional spaces into a higher dimensional resource space.
- (4) *Join resource spaces*. In order to obtain an unified resource view, we need to join the low-dimensional resource spaces into a higher dimensional resource space.

Due to the large number of the resources, it is important to explore the approach to automatically classifying the resources and uploading the resources into the resource space. There are some limitations in designing and creating the resource space manually, such as heavy workload, personal knowledge level and design skills.

To ease the resource space design process, this work studies the issues of automated creation of resource spaces and proposes a general method to automatically construct resource spaces from XML files and resource entities.

II. AUTOMATIC CONSTRUCTION OF RSM

A. Task Introduction

The construction of the resource space is based on domain knowledge. In manual work of resource space building, we can classify the resource and design the resource space according to certain domain knowledge. To make automatically build a resource space, we need an objective method to construct the resource space which can correctly reflect the domain knowledge and the classification semantics of the resources.

A resource space includes three parts: the structure of the space including axes and their coordinates, the specification of the content (including identity, path and semantic description) of resources and the resource entities. The task of the automatic construction the RSM includes the following two aspects:

1. Design a better method to construct the structure of the resource space according to the existing files that describe the contents of the resource.
2. Insert the resources into the resource space, that is, mapping the resources to each dimension and find the corresponding relations between the resources and the coordinates.

The input of the system includes the resource entity and

their content description, and the output is the automatically generated.

B. Basic Idea

The resource space model is based on the classification semantics of the resources and the resource partition related to certain characteristics of the resources. This section presents a method to get the classification rules from XML files. Through parsing the XML files we can obtain the resource elements (resources in the resource space corresponds to the element tag in the XML file), the child elements of the resource element and the attributes of the resource element which describe the classification features of the resources.

XML is a set of rules for encoding documents in machine-readable form. The design goals of XML emphasize simplicity, generality, and usability over the Internet. It is a simple self-described data storage language using a series of simple tags to describe data. It can not only describe the content of the data but also highlight the data structure which reflects the relationship among the data. The construction method of the resource space based on the XML files is shown in Fig. 1.

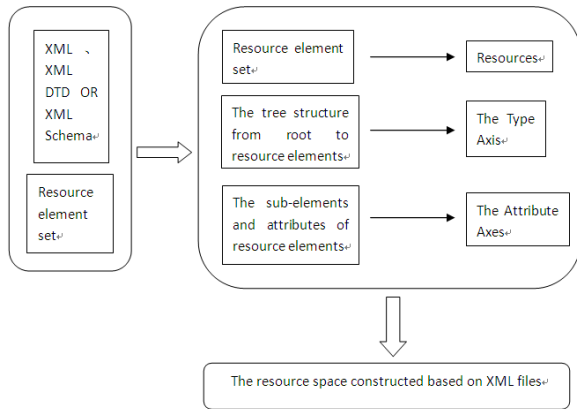


Fig. 1 The resource space constructed based on XML files

C. Resources Determination in RSM

As the description of the resources, XML files can describe the resource characteristics by nested elements. When we construct the resource space based on XML files, some elements in XML files can be regarded as resources, while other elements may be used to describe the resource features or to define the specific objects that objectively exist in the domain area, but these objects are not the resources which will be organized and managed in the resource space.

For example, “inproceedings” is an element describing the conference paper in the XML file of the DBLP project and it is regarded as a resource element which means it will be converted into a resource in the resource space. The “inproceedings” record described in the XML file as follows:

```

<inproceedings>
  <author>Alan M. McIvor</author>
  <title>Calibration of a Laser Stripe Profiler.</title>
  <booktitle>3DIM</booktitle>
  <year>1999</year>
  <pages>92-98</pages>

```

```

<url>db/conf/3dim/3dim1999.html#McIvor99</url>
</inproceedings>

```

The statements above describe an “inproceedings” record. As the “inproceeding” is one kind of resources that will be organized and managed in the resource space, so the “inproceeding” element is in the resource elements set as the input of the method. The “inproceedings” element includes six child elements: author, title, booktitle, year, pages, url, these child elements describe the resource features from different aspects.

“proceedings” is another element in the XML file of the DBLP project and it defines the conference in the domain but because the conference is not the resource that will be organized and managed in the resource space, so, the “proceedings” element is not in the resource elements set.

To determine which elements in the XML file is the resource in the resource space, one feasible method is mapping the resources in the resource space into the elements in the XML file and the set of the resource elements is the input parameter of the automatic construction system of RSM, which can ensure the accuracy of resource selection.

D. Construction of the Type Axis

The resource type expresses the classification of the resource itself, so, the hierarchy of resource type can be considered as a resource classification rules and can be converted in to the type axis.

If there are a variety of resources to be managed in the resource space, then the resource element set as the input will contain more than one resource element, which is one of the partitions of the resources from the perspective of the resource type.

We can parse an XML file to find the classification types. If a document conforms to XML syntax specification, it is a “well-structured” document. A well-structured XML document can form a tree structure that expands from the “root” node to the “branches” nodes. The root node of this tree corresponds to root element of the XML document. The intermediate nodes of the tree correspond to the sub-elements and attributes and all elements can have child elements and attributes. The bottom is the leaf nodes of the tree, corresponding to the content of the elements and the value of the attributes.

For each resource element in the set, we begin depth-first search from the root until we find a resource element node, then save all intermediate nodes on the path from root to that resource element node. The retained nodes form a hierarchy structure which expresses the resource classification clearly from the perspective of the resource type according to the domain knowledge.

However, if the existing resources cannot cover all types of resources, the type axis of the resource space automatically constructed from the XML file cannot express the resource classification completely and accurately. Therefore, we need to use XML DTD or XML Schema to obtain the partition of resources. By parsing XML DTD or XML Schema, the tree

structure of the XML document can be obtained, which can still be used as the resource type hierarchy.

We design different methods to get hierarchical structure from XML DTD and XML Schema. The hierarchical tree structure based on dblp.dtd or dblp.xsd file is shown below in Fig.3.

The hierarchy structure can be considered as the type axis. As the top of the structure contains only one root node and it doesn't work to classification, so it should not appear as a coordinate in the type axis. When constructing the resource space, the root node should be discarded. The type axis is shown in Fig.4:

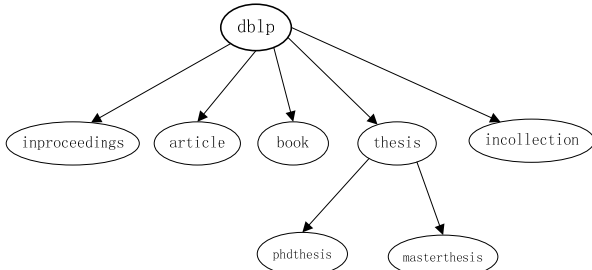


Fig. 3 The tree structure of resource types in DBLP

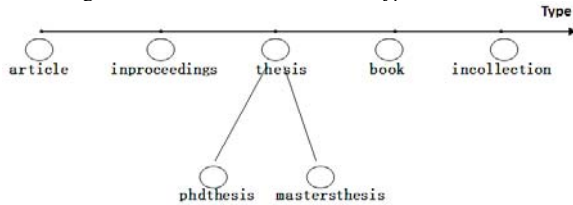


Fig. 4 The type axis of the resource space.

E. Construction of the Attribute Axes

The XML file can describe the resource characteristics by using the child elements and attributes of resource elements, so it is feasible to construct the attribute axes and set up the classification rules according to the attribute values and the element content.

To automatically constructing the attribute axes, we obtain the resource elements' attributes and sub-elements by parsing the XML DTD or XML Schema file. If the resource space needs to manage a variety of resources, we should obtain the sub-elements, attributes, type and constraints for each resource element respectively.

XML DTD and XML Schema take different mechanisms to constrain the number of occurrences of elements and attributes, for example, XML DTD using the following special tags to declare the number of the element occurrences. Tag "*" indicates the elements that can occur zero or more times; Tag "+" indicates the elements that occur at least once; Tag "?" indicates the elements that can occur zero or one time. XML Schema can also control the number of occurrences of elements by "minOccurs" indicator and "maxOccurs" indicator. In addition, XML DTD and XML Schema support the optional attribute by different methods.

When constructing the attribute axes, the required sub-elements and attributes of resource elements need to be retained, giving up those sub-elements that may not appear

and optional attributes. This is because the third normal form of RSM requires that all axes must be reasonable, that is, all resources in a resource space can be projected on any axis finding the corresponding coordinates so that the resource set that each axis represents is equal.

For example, the element "article" in dblp.dtd is described as follow:

```
<!ENTITY % article.fields
"author,title,booktitle,year,volume?,number?,pages?,m
onth?,note?">
<!ELEMENT article (%article.fields;)>
<!ATTLIST article
key CDATA #REQUIRED
mdate CDATA #IMPLIED
>
```

The resource element article includes the required sub-elements and optional sub-elements, the required sub-elements such as "author", "title", "booktitle" and "year", the optional sub-elements such as "volume", "number", "pages", "month" and "note". The element article has a required attribute "key" and an optional attribute "mdate".

After getting the sub-element set and attribute set, we will find the intersection of them to select the common features of various types of resources. Then, we will construct the attribute axes based on these common features to classify resources and the axes, which will be named after the sub-element name or attribute name.

The next step is to generate the coordinates on the attributes axes automatically. The coordinate express the content of sub-elements or the value of attributes. The coordinate value of a resource projected on the attribute axis can be obtained from the content of sub-elements or the value of attributes.

XML Schema offers a variety of data types for elements and attributes and allows users create their own data types derived from the standard types. A data type may contain a finite number of values or contain an infinite number of values. As the RSM requires the number of coordinates on any axis be finite, so, we need an appropriate treatment if the axis contains an infinite number of values. Two methods are proposed to generate the coordinates on attribute axes:

1. Design a division scheme for each built-in data type respectively. For example, we can divide xsd:string into 27 fields according to the first letter: a, b, ..., z and other; For the type xsd:integer, we can divide it into 3 fields according to the value magnitude: >0, =0, <0, and so on.
2. According to the distribution of resource classification property values by parsing the XML files, classify the resources automatically by making use of the existing classification or clustering methods in the fields such as pattern recognition or knowledge discovery.

XML Schema supports the restriction mechanism to further define the data type for the elements and attributes. Restriction can also provide reference to the value division of data types. Sometimes it is possible to construct the coordinates on attribute axes exactly according to the restriction. XML Schema provides the restriction types partially shown in Table.

For example, we use restrictions "maxInclusive" and "minInclusive" to limit the element year indicating the XML file only stores the resources published from 2000 to 2010. The relevant statements in XML Schema are as follows:

```
<xsd:element name="year">
  <xsd:simpleType>
    <xsd:restriction base="xs:integer">
      <xsd:minInclusive value="2000"/>
      <xsd:maxInclusive value="2010"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

The attribute axis constructed based on the year element with restrictions contains eleven coordinates 2000, 2001, ..., 2010. Thus, it meets the RSM requirement of the reasonable design of axes.

The attribute axes have been constructed through this method. Although this process has selected the sub-elements and attributes, but in practice not all retained child elements and attributes are suitable for building attribute axes. For example, each resource element contains a title child element to express the name of a resource. If we convert it to an attribute axis, then each coordinate on that axis will map exclusively to a resource, which is meaningless to the resource classification. Therefore, it is necessary to determine whether the property is suitable for converting the axis according to its ability to express the classification semantics. To determine whether a property is suitable is difficult, we can take the following method to deal with it: *calculate the ratio of the number of coordinates to the number of the resources, if the ratio is close to 1, the property is of little importance to the classification and therefore it is not suitable for the conversion* [4].

III. EXPERIMENTAL ANALYSIS

We take the DBLP dataset as input and produce a 3-dimensional resource space to organize and manage the paper resources in the computer science based on the dblp.dtd and dblp.xsd file. The result is shown in Fig.5. The 3-dimensional resource space includes a type axis and two attribute axes. After inserting the resources into the resource space, the resource classification management system is achieved. The resource space allows users to search the resources from multi-faceted such as *Year*, *Author* and *Type* dimension. Users can have a comprehensive understanding of the distribution of resources through the holistic view of the resource space.

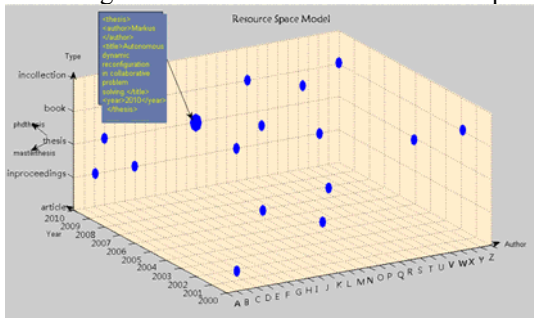


Fig. 5 The automatically constructed 3-dimensional resource space.

IV. CONCLUSION

This paper presents a framework that automatically constructs a resource space from XML resource data. The proposed method can extract the classification rules, the coordinates, the attributions to build a multi-dimensional resource space and insert resource entities into corresponding points in the resource space. We use the DBLP XML dataset to test the proposed method and the results show that the proposed method can easily construct a proper resource space for managing literature resources in DBLP dataset. Future work includes the generation of the complex semantic space integrating the resource space and semantic link network deployed on P2P network towards the ideal of the future interconnection environment [7-19].

REFERENCES

- [1] H.Zhuge, The Knowledge Grid, World Scientific Publishing Co., Singapore, 2004.
- [2] H.Zhuge, Resource Space Model, Its Design Method and Applications, Journal of Systems and Software, 72 (1) (2004) 71-81.
- [3] H.Zhuge, The Web Resource Space Model, Springer, 2007.
- [4] H.Zhuge, Y.Xing and P.Shi, Resource Space Model, OWL and Database: Mapping and Integration, ACM Transactions on Internet Technology, 8/4, 2008.
- [5] H.Zhuge and Y.Xing, Probabilistic Resource Space Model for Managing Resources in Cyber-Physical Society, IEEE Transactions on Service Computing, http://doi.ieeecomputersociety.org/10.1109/TSC.2011.12.
- [6] H.Zhuge, The Complex Semantic Space Model, Keynote at 20th IEEE International Conference on Collaboration Technologies and Infrastructures, June 27th-29th, 2011, Paris, France.
- [7] H.Zhuge, Interactive Semantics, Artificial Intelligence, 174(2010)190-204.
- [8] H.Zhuge, Semantic linking through spaces for cyber-physical-socio intelligence: A methodology, Artificial Intelligence, 175(2011)988-1019.
- [9] H.Zhuge, Communities and Emerging Semantics in Semantic Link Network: Discovery and Learning, IEEE Transactions on Knowledge and Data Engineering, vol.21, no.6, 2009, pp. 785-799.
- [10] H.Zhuge and X.Sun, A Virtual Ring Method for Building Small-World Structured P2P Overlays, IEEE Transactions on Knowledge and Data Engineering, 20 (12) (2008)1712-1725.
- [11] H.Zhuge and J.Zhang, Topological Centrality and Its Applications, Journal of the American Society for Information Science and Technology, 61(9)(2010)1824-1841.
- [12] H.Zhuge and L.Feng, Distributed Suffix Tree Overlay for Peer-to-Peer Search, IEEE Transactions on Knowledge and Data Engineering, 20 (2) (2008) 276-285.
- [13] H.Zhuge and X.Li, Peer-to-Peer in Metric Space and Semantic Space, IEEE Transactions on Knowledge and Data Engineering, 19 (6) (2007) 759-771.
- [14] H.Zhuge, X.Chen, X.Sun and E.Yao, HRing: A Structured P2P Overlay Based on Harmonic Series, IEEE Transactions on Parallel and Distributed Systems, 19 (2) (2008) 145-158.
- [15] H.Zhuge, et al, A Scalable P2P Platform for the Knowledge Grid, IEEE Transactions on Knowledge and Data Engineering, 17 (12) (2005) 1721-1736.
- [16] H. Zhuge, Resource space model, its design method and applications. Journal of Systems and Software, 72(1)(2004)71-81.
- [17] H. Zhuge, Fuzzy resource space model and platform. Journal of Systems and Software, 73(3) (2004)389-396.
- [18] H. Zhuge and X. Li, RSM-Based Gossip on P2P Network. ICA³PP 2007: 1-12.
- [19] H.Zhuge, The Future Interconnection Environment, IEEE Computer, 38 (4) (2005) 27-33.