Chapter 3

A Resource Space Model

Classification is one of the most basic methods for humans to recognize and organize various resources. The Resource Space Model is a resource organization model based on multi-dimensional classifications.

3.1. Examples of Using Multi-Dimensional Classifications

One-dimensional classification is often used. For example, supermarkets classify goods and arrange classes in a certain order according to customers' purchasing habit so that customers can easily get necessary goods. File system is a one-dimensional classification (hierarchical directory) system that enables both human and computer to easily manage various digital files in computer's disk.

The following are examples of using multi-dimensional classifications:

(1) Maps use two-dimensional space (*latitude*, *longitude*) or three-dimensional space (*latitude*, *longitude*, *elevation*) to actually specify locations on the earth.
(2) Experimental results are displayed and compared in two- or three-dimensional charts, and the display devices are two- or three-dimensional.
(3) Faceted web browsing view a set of web pages from different facets of contents.
(4) Sensor grids are used to collect overall signals of the brain.
(5) Stock market uses three-dimensional classification space (*time*, *company*, *price*) to reflect the real-time situation of the market.

Keyword search is popular in general-purpose web search. Fig.3.1.1

shows an example of searching the map by keywords.

A map classifies surface objects according to geographical coordinate space. With continuous expansion of the scope of human activities, more and more surface objects have not only physical features but also socio features. For example, a city has population, culture, economy, and education. Classification on socio features provides a socio dimension for users to accurately locate objects.

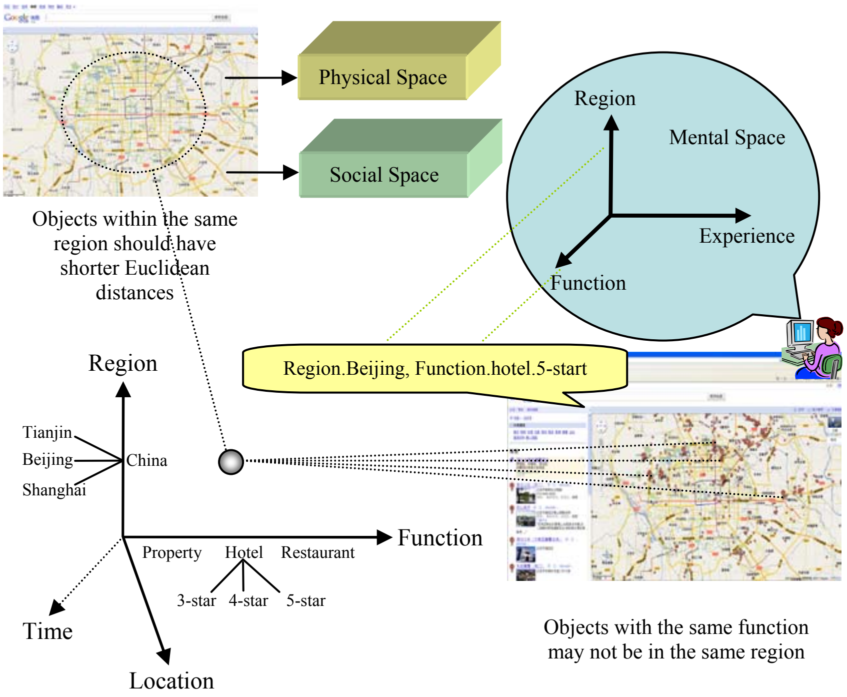

Physical Space
Social Space
Objects within the same region should have shorter Euclidean distances
Region
Mental Space
Experience
Function
Region.Beijing, Function.hotel.5-start
Region
Tianjin
Beijing
Shanghai
China
Function
Property
Hotel
Restaurant
3-star
4-star
5-star
Time
Location
Objects with the same function may not be in the same region

Fig. 3.1.1 Multi-dimensional classifications in map search.

Different spaces may use different distance measures such as Euclidean distance and semantic distance.

Two objects that are close in one space may not be close in the other space. The physical space is a metric space, where objects within the same region should have shorter Euclidean distances. Individuals within the same family or community are closer in the social space. Two individuals that are close in the social space (e.g., with the same social function) may not be close in the physical space (e.g., in the same region). A classification space with dimensions (*location*, *region, function*) can effectively organize various objects.

Map search concerns the mental space. Users have established classification spaces in minds through lifelong learning. They usually have consensus on classifications on social functions and physical regions. Querying a map is to find the points in the map that match the points in the physical space, the socio space and the mental space.

Users can search various objects by giving the keywords that indicate the coordinates in the dimensions or a part of the dimensions in mind. The social functions can be classified by physical regions, and physical regions can be classified by social functions.

More dimensions can be added to the classification space. For example, adding a time dimension enables users to accurately search historical maps.

The new version of Google map search enables users to select the classes of the search target, which can help narrow the search scope. To develop an underlying model to support efficient map search is a critical issue.

3.2. The Virtual Grid

The *Virtual Grid* is a platform independent interconnection environment, based on a resource organization model, which can effectively specify, share, use and manage a variety of resources (H. Zhuge, "Semantics, Resource and Grid", *Future Generation Computer Systems*, 20 (1)(2004)1-5). It has the following key parts:

(1) The *Resource Space Model* (RSM) — a resource space for effectively organizing and accurately locating resources. It can be a classification-based Resource Space Model. It can also be the integration of the semantic link network and the Resource Space Model. A resource space can be centralized or decentralized (H. Zhuge and X. Li, RSM-Based Gossip on P2P Network. ICA^{3}PP 2007: 1-12).

(2) The *Resource Using Mechanism* (RUM) — a *faceted resource browser*, a *resource management engine*, a *Resource Operation Language* (ROL), a *ROL interpreter*, and an *application development environment*.

The resource browser provides an easy-to-use interface that helps end-users select resources and operations, specify parameters, and then have operations carried out. The resource management engine accepts instructions and then has them carried out according to the types of the resources to be operated on.

The ROL enables end-users to carry out simple operations, and application developers to compose a program to apply complex operations to resources.

The ROL interpreter supports not only complex applications in their use of resources, but also the resource browser in its use of resources. End-users can either run an application system to carry out operations of a specific kind or use the resource browser to operate directly on resources.

Developers build programs with the support of the development environment, which is in turn supported partly by the Virtual Grid and partly by application development tools.

A resource space has three views as discussed in "Resource Space Grid: model, method and platform" (H. Zhuge, *Concurrency and Computation: Practice and Experience*, 16(3)(2004)):

(1) The *user view*, a two- or multi-dimensional resource space, is used by the resource browser to make it easy for end-users to locate and

use resources. The space will interact with the user's mental space during operation. It is hard for end-users to deal with multi-dimensional spaces without such help.

(2) The *universal view*, the entire *n*-dimensional resource space. A user view is a slice (or a subspace) of the universal view. Operating the same resource space, different users could have different user view.

(3) The *semantic view*, a semantic representation of a variety of resources to support effective interactions between human and machine as well as between machine and machine based on the semantic worldview, semantic image, and semantic lens mechanisms (H. Zhuge, "Interactive Semantics," *Artificial Intelligence*, 174(2010)190-204).

3.3 The Resource Space Model (RSM)

The Resource Space Model is the resource management mechanism using multi-dimensional classification space on resources.

3.3.1 *Resource spaces*

Definition 3.3.1 A resource space is an *n*-dimensional space in which every point has unique projection on every dimension.

The uniqueness implies that giving one coordinate at every dimension can uniquely determine one point, which contains a set of resources (possibly null). A resource space has a name, a type, a location (logical or physical), and an access privilege.

A dimension is classified by a set of coordinates. By establishing a good coordinate system for a resource space, we can precisely store and retrieve its resources by their coordinates. To distinguish coordinates or resources of the same name in different resource spaces, the coordinate or resource name can be used with the space name, for example: *coordinate-name***.***space-name*.

Further, we assume that an ontology service mechanism *Output* = *Ontology_Service* (*Input*, *k*) is available. The *Input* parameter is a word or phrase. The second parameter is a numeric variable used for controlling the output. If $k = 0$, the service puts out sets of words related to *Input* of the following kinds: synonym, abstract concept, specific concept, and instance. If $k = 1$, the ontology service puts out one more element — a quasi-synonym of the input word or phrase, if one exists.

In the discussion below, we use the following notations:

(1) A resource space is represented by $RS(X_1, X_2, \ldots, X_n)$, or *RS* for short, where *RS* is the name of the space and X_i is the name of an axis. $|RS|$ denotes the number of dimensions of *RS*.

(2) $X_i = \{C_{i1}, C_{i2}, \ldots, C_{im}\}$ represents an axis with its coordinates. Each element denotes a coordinate name in the form of a noun or a noun phrase. Any coordinate name must be defined in its domain ontology, as in the Word Net (www.cogsci.princeton.edu/~wn/).

Definition 3.3.2 Two axes are deemed the same if their names are the same and the names of the corresponding coordinates are the same based on a certain mapping. If two axes $X_1 = \{C_{11}, C_{12}, \ldots, C_{1i}\}$ and $X_2 = \{C_{21}, C_{22}, \ldots, C_{2j}\}$ have the same axis name and specify the same set of resources but have different coordinates, they can be joined into one: $X = X_1 \cup X_2$, denoted by $X_1 \cup X_2 \Rightarrow X$.

It is useful to form a relatively complete dimension when two persons classify a set of resources with different views. In the open domain applications, one name may be used to indicate different objects, and different names may be used to indicate the same thing, so the join operation should not just check the name of dimension.

Characteristic 3.3.1 An axis X can be split into two axes X' and X'' by dividing the coordinate set of X into two: the coordinate set of X' and that of X'', such that $X = X' \cup X''$.

A coordinate can be a hierarchy, with lower-level coordinates as subclasses of their common ancestor. We use $Sup(C)$ to denote the direct ancestor or immediate superior of C in a coordinate hierarchy. The name of each coordinate of a hierarchy can be differentiated from others of the same name by giving its name together with the names of all of its ancestors. The following definition shows the nature of coordinates in a hierarchy:

Definition 3.3.3 A coordinate C selects a class of resources (denoted by $R(C)$) such that if $C = Sup(C')$ then $R(C') \subseteq R(C)$.

From this definition, if $C = Sup(C')$ and $C' = Sup(C'')$ then $R(C'') \subseteq R(C)$.

For simplicity, an axis with hierarchical coordinates can be mapped onto an axis with flat coordinates by projecting its leaves onto the axis. But, this will lead to the lost of abstraction ability of the space.

Good resource space design should ensure correct resource sharing and management. Synonyms like “teacher”, “instructor” and “tutor” should not be used together as flat coordinates, because that could lead to resource operations making mistakes.

Definition 3.3.4 A coordinate C is dependent on coordinate C' if $C \in Output = Ontology\_Service(C', 1)$.

Definition 3.3.5 If $X = (C_1, C_2, \ldots, C_n)$ is an axis and C_i' is a coordinate of another axis X', we say that X is a fine classification on C_i' (denoted by C_i'/X) if and only if:

(1) $(R(C_k) \cap R(C_i')) \cap (R(C_p) \cap R(C_i')) = \phi$ ($k \neq p$, and $k, p \in [1, n]$); and,

(2) $(R(C_1) \cap R(C_i')) \cup (R(C_2) \cap R(C_i')) \cup \ldots \cup (R(C_n) \cap R(C_i')) = R(C_i')$.

As the result of the fine classification, $R(C')$ is classified into n categories: $R(C_i'/X) = \{R(C_1) \cap R(C_i'), R(C_2) \cap R(C_i'), \ldots, R(C_n) \cap R(C_i')\}$.

Definition 3.3.6 For two axes $X = \{C_1, C_2, \ldots, C_n\}$ and $X' = \{C_1', C_2', \ldots, C_m'\}$, we say that X is a fine classification on X' (denoted by X'/X) if and only if X is a fine classification on $C_1', C_2', \ldots$, and C_m'.

Characteristic 3.3.2 Fine classification is transitive, that is, if X''/X' and X'/X, then X''/X.

Fine classification is a basic mechanism of recognition and understanding.

Definition 3.3.7 Two axes X and X' are said to be orthogonal to each other (denoted by $X \perp X'$) if X is a fine classification on X' and vice versa, that is, both X'/X and X/X'.

For example, *KnowledgeLevel* = <*Concept*, *Axiom*, *Rule*, *Method*> ⊥ *Discipline* = <*ComputerScience*, *Mathematics*, *Physics*> because <*Concept*, *Axiom*, *Rule*, *Method*> is a fine classification of every coordinate of the *Discipline* axis and the <*ComputerScience*, *Mathematics*, *Physics*> is a fine classification of every coordinate of the *KnowledgeLevel* axis. Another example is *Gender*=<*Male*, *Female*> ⊥ *Student*=<*UndergraduateStudent*, *GraduateStudent*>.

Because fine classification is transitive, we can assert the following:

Characteristic 3.3.3 Orthogonality between axes is transitive, that is, if $X \perp X'$ and $X' \perp X''$, then $X \perp X''$.

3.3.2 *Normal forms*

To ensure a good design of a resource space, we need to define the following normal forms for the space.

Definition 3.3.8

(1) A resource space is in first normal form (1NF) if no coordinate names are duplicated within any axes.

(2) A space in 1NF is also in second normal form (2NF) if no two coordinates are dependent on each other.

(3) A space in 2NF is also in third normal form (3NF) if any two axes are orthogonal.

These normal forms provide designers with guidelines for designing a good resource space. The 1NF avoids explicit coordinate duplication. The 2NF avoids implicit coordinate duplication, and prevents one coordinate from semantically depending on another. The 3NF ensures that resources are properly used.

Characteristic 3.3.4 If two spaces RS_1 and RS_2 hold the same type of resources and they have n (≥ 1) axes in common, then they can be joined as one RS such that RS_1 and RS_2 share these n common axes and $|RS| = |RS_1| + |RS_2| - n$. RS is called the join of RS_1 and RS_2 and is denoted by $RS_1 \cdot RS_2 \Rightarrow RS$.

Characteristic 3.3.5 A space RS can be separated into two spaces RS_1 and RS_2 (denoted by $RS \Rightarrow RS_1 \cdot RS_2$) such that they have n ($1 \leq n \leq \mathit{minimum}\,(|RS_1|, |RS_2|)$) axes in common, $|RS| - n$ different axes, and $|RS_1| + |RS_2| = |RS| + n$.

The separation operation is also called disjoin in The Web Resource Space Model (H.Zhuge, Springer, 2008).

Characteristic 3.3.6 If two spaces RS_1 and RS_2 hold the same type of resources and satisfy: (1) $|RS_1| = |RS_2| = n$, (2) they have $n-1$ common axes, and (3) the distinct axes X_1 and X_2 satisfy the merge condition, then they can be merged into one RS by retaining the $n-1$ common axes and adding a new axis $X = X_1 \cup X_2$. RS is called the merge of RS_1 and RS_2, denoted by $RS_1 \cup RS_2 \Rightarrow RS$, and $|RS| = n$.

Characteristic 3.3.7 A resource space RS can be split into two spaces RS_1 and RS_2 that hold the same type of resources as that of RS and have $|RS| - 1$ common axes, by splitting an axis X into two axes X' and X'' such that $X = X' \cup X''$. This split operation is denoted by $RS \Rightarrow RS_1 \cup RS_2$.

Using these definitions and characteristics, we can prove the following three lemmas.

Lemma 3.3.1 Let $RS_1 \cdot RS_2 \Rightarrow RS$.
(1) RS is in 1NF if and only if both RS_1 and RS_2 are in 1NF.
(2) RS is in 2NF if and only if both RS_1 and RS_2 are in 2NF.
(3) RS is in 3NF if and only if both RS_1 and RS_2 are in 3NF.

Lemma 3.3.2 (1) $RS \Rightarrow RS_1 \cdot RS_2$ if and only if $RS_1 \cdot RS_2 \Rightarrow RS$; and, (2) $RS \Rightarrow RS_1 \cup RS_2$ if and only if $RS_1 \cup RS_2 \Rightarrow RS$.

The following lemma ensures that a resource space of many dimensions can be separated into several spaces of fewer dimensions that keep the same normal form as the original space. For instance, a five-dimensional space can be separated into two three-dimensional spaces that have an axis in common.

Lemma 3.3.3 If $RS \Rightarrow RS_1 \cdot RS_2$, we have:
RS is in 1/2/3NF if and only if both RS_1 and RS_2 are in that form.

From the definition of the normal forms, we have the following two lemmas about the merge and split operations.

Lemma 3.3.4 If $RS \Rightarrow RS_1 \cup RS_2$, and if RS is in 1/2/3NF, then RS_1 and RS_2 are in that form.

Lemma 3.3.5 If $RS_1 \cup RS_2 \Rightarrow RS$ and if RS_1 and RS_2 are in 3NF and RS is in 2NF, then RS is in 3NF.

Semantic overlaps may exist between resource definitions in some applications. In this case, the space designer must use quasi-synonyms as coordinates on an axis.

Lemma 3.3.6 Resources in a 3NF resource space can be accessed from any axis.

Proof. Let $X_i = \{C_{i1}, C_{i2}, \ldots, C_{ip}\}$, $1 \le i \le n$. If the *RS* is in 3NF, then for any two axes $X_k = \{C_{k1}, C_{k2}, \ldots, C_{kl}\}$ and $X_j = \{C_{j1}, C_{j2}, \ldots, C_{jm}\}$ $(1 \le k \ne j \le n)$, $X_k \perp X_j$. We have $C_{kq} \,/\, X_j$ for every C_{kq} $(1 \le q \le l)$.

From the definition of fine classification, we have:

$R(\mathrm{C}_{kq}) = (R(C_{kq}) \cap R(C_{j1})) \cup (R(C_{kq}) \cap R(C_{j2})) \cup \cdots \cup (R(C_{kq}) \cap R(C_{jm}))$

$= R(C_{kq}) \cap (R(C_{j1}) \cup R(C_{j2}) \cup \cdots \cup R(C_{jm}))$.

Then $R(C_{kq}) \subseteq (R(C_{j1}) \cup R(C_{j2}) \cup \cdots \cup R(C_{jm}))$ for $1 \le q \le l$.

Hence, we have $(R(C_{k1}) \cup R(C_{k2}) \cup \cdots \cup R(C_{kl})) \subseteq (R(C_{j1}) \cup R(C_{j2}) \cup \cdots \cup R(C_{jm}))$ for $1 \le k \ne j \le n$.

Similarly, we have: $(R(C_{j1}) \cup R(C_{j2}) \cup \cdots \cup R(C_{jm})) \subseteq (R(C_{k1}) \cup R(C_{k2}) \cup \cdots \cup R(C_{kl}))$ for $1 \le k \ne j \le n$.

And then we have: $(R(C_{j1}) \cup R(C_{j2}) \cup \cdots \cup R(C_{jm})) = (R(C_{k1}) \cup R(C_{k2}) \cup \cdots \cup R(\mathrm{C}_{kl}))$ for $1 \le k \ne j \le n$.

So, if $R = (R(C_{j1}) \cup R(C_{j2}) \cup \cdots \cup R(C_{jm}))$, then for every axis $X_i = \{C_{i1}, C_{i2}, \ldots, C_{ip}\}$, $1 \le i \le n$, $R = R(C_{i1}) \cup R(C_{i2}) \cup \cdots \cup R(C_{ip})$.□

This means that the resources that are accessible from any axis are the same. Hence a resource retrieval algorithm does not need to depend on the order of the axes.

Definition 3.3.9. A coordinate C is called weakly independent on another coordinate C' if $C \notin \mathit{Output} = \mathit{Ontology\_Service}\,(C', 0)$.

The second normal form has a weak analog.

Definition 3.3.10 The weak 2NF of a space is a 1NF, but in addition, for every one of its axes, any pair of coordinates are weakly independent on each other.

The inter-dependent coordinates can be merged into one coordinate to ensure the 2NF or semantic links can be established between coordinates (H.Zhuge and Y.Xing, Probabilistic Resource Space Model fro Managing Resources in Cyber-Physical Society, *IEEE Trans. on Service Computing*, http://doi.ieeecomputersociety.org/10.1109/TSC.2011.12). The third normal form also has a weak analog that can be useful in some applications.

Definition 3.3.11 The weak 3NF of a resource space is a weak 2NF, but in addition, all pairs of axes are orthogonal.

A 3NF resource space may contain points without resources, which lower the efficiency of resource management. We can further normalize a resource space by ruling out such empty points.

Definition 3.3.12 Let $X=(C_1, C_2, \ldots, C_n)$ be an axis and C_i' be a coordinate on another axis X'. We say that X is a regular and fine classification on C_i' (denoted by C_i'/X) if and only if

(1) $R(C_1) \cap R(C_i') = \phi$, $R(C_2) \cap R(C_i') \neq \phi$, $\ldots$, and $R(C_n)\ ) \cap R(C_i') \neq \phi$,

(2) $(R(C_k) \cap R(C_i')) \cap (R(C_p) \cap R(C_i')\ )= \phi$ $(k \neq p$ and $k, p \in [1, n])$, and $(R(C_1) \cap R(C_i')) \cup (R(C_2) \cap R(C_i')) \cup \ldots \cup (R(C_n) \cap R(C_i')) = R(C_i')$.

Definition 3.3.13 For two axes $X=\{C_1, C_2, \ldots, C_n\}$ and $X' = \{C_1', C_2', \ldots, C_m'\}$, we say that X is a regular and fine classification on X' if and only if X is a regular and fine classification on $C_1', C_2', \ldots$, and C_m'.

Definition 3.3.14 If two axes are regular and fine classification on each other, then the two axes are called *regularly orthogonal.*

Definition 3.3.15 The fourth normal form (4NF) of a resource space is a 3NF, but in addition, all pairs of axes are regularly orthogonal.

A generic *reference resource space* is the following four-dimensional resource space:

$RS = (category, level, location, time)$.

The category dimension is a classification of resources. Each coordinate on this axis represents a distinct category. A category coordinate can hold subcategories, and each subcategory can hold subcategories, and so on. A category together with all its lower subcategories forms a category hierarchy. Coordinates on the category axis are scalable because people usually consider resources across different levels. Except for certain basic subcategories, each coordinate of the category axis can be spread down onto a set of lower level coordinates, which can then be spread down again or collected back up to its higher level coordinates. Name duplication can be avoided by denoting a subcategory as a path, e.g., $category \cdot subcategory$.

A resource can be located by its category dimension and level dimension. The location dimension determines the locations of resources in the cyber space, physical space, socio space or mental space. The level dimension can not only classify the category dimension but also the location dimension and the time dimension.

The generic reference resource space can have different specializations on category, level, location and time dimensions depending on the types of resource that it contains.

3.4 Criteria for Designing Resource Spaces

The normal forms defined above can now be used to consider in what ways a resource space may be good or bad. The following are criteria for a resource in a good resource space.

(1) The *understandable* criterion. First, *any resource in a resource space should be understandable*. At least the classifications of the resources should be known. That is, a resource space should not be given a resource that cannot be described or explained. The simple reason is that only the understandable resources can help users.

This criterion also excludes the resources that are irrelevant to users' knowledge structures. Second, *any axis should be understandable*. At least, the classification of any axis should be clear. Third, any coordinate should be understandable. That is, the intention and extension of any coordinate should be clear.

(2) The *resource positioning* criterion. Any resource in a resource space should belong to a point in the resource space. A point has projection at all axes. Such a resource is said to be "well positioned" in the resource space.

(3) The *resource use* criterion. The use of any resource in a resource space should obey all restrictions on its usage that were defined when it was created. For example, some resources can only be operated by particular software.

(4) *Minimum number of axes*, a point can be accurately located by minimum number of axes.

A resource space has a logical representation layer and a physical storage layer. If the two layers are not consistent, for example, if a resource exists in one layer but not in the other, then the RUM will be unable to operate on it properly. To ensure consistency, we set the following criteria.

(1) The *existence consistency* criterion. Any resource in a resource space should ensure the consistency of its existence in the schemas of different levels. In other words, if a resource is in one schema it must be in another schema. Resource operations should maintain this consistency.

(2) The *pervasive residence* criterion. The resources of a resource space should be allowed to reside on various hardware and software platforms.

(3) The *operability* criterion. An RUM should support at least three basic resource operations: get a resource from a resource space, put a resource into a space, and remove a resource from a space. These operations must also be available for managing through any view of a resource space.

3.5 Designing Resource Spaces

The following is a reference design process for logical-level applications:

(1) *Resource analysis.* Resource analysis is to determine the application scope, to survey possible resources, and then to specify all the relevant resources by using a *Resource Dictionary* (RD), which records them as a local resource space for the application. These resources can be described in XML, and the RD can be managed using the ROL or any other XML query language.

(2) *Top-down resource partitioning*. Due to the differences in structure, different designers may partition resources differently, so a uniform approach to partitioning is needed. The first step is to unify the highest-level partition. Humans, information, and natural or artificial objects are key factors in human society, and the resources of human society may be clearly partitioned in this way. The top-level partitioning of a domain can be regarded as a special case of this partitioning of human society. For example, an institute's resources can be classified at the highest level within three exclusive categories: human resources, information resources, and service resources. This step and the preceding one are carried out on each category and its subcategories and so on until the resources in the lowest level category are few enough for the application being designed.

(3) *Design bidimensional resource spaces.* People can manage bidimensional spaces better than higher dimensional spaces. So, we can first design a set of bidimensional resource spaces, and then consider joining them to form higher dimensional spaces. This design process has the following steps.

- *Name the axes*. Each axis name should reflect one category of the top-level partition of resources.
- *Name the top-level coordinates*. Each coordinate should reflect one subcategory of the category of its axis.

- *Name the coordinate within each hierarchy*. For each top-level coordinate, name its lower level coordinates until all the coordinates at all levels have been named.
- *Remove any dependence between coordinates*. Look for dependence between coordinates at all levels. Where it occurs, redesign the partitioning at that level and then name any new coordinates.
- *Make all axes orthogonal*. If any pair of axes is not orthogonal, reselect axes.

(4) *Join spaces*. If two spaces can be joined into one space, carrying out join operation to form a single resource space.

Making use of abstraction, and analogy between the existing (or reference) resource spaces and the new resource space, are important techniques for designing a good resource space (H. Zhuge, "Resource Space Model, Its Design Method and Applications", *Journal of Systems and Software*, 72(1)(2004)71-81).

3.6 Representation of Resources

The semantics of a resource could be seen as a black box if the semantics come from those of related resources, or as a glass box if they come from the features and functions of the resource itself. Combining the two ways could be more effective than either alone.

A resource template represents the common features of a class of resources of the same type. Resources defined in a space need a set of templates, organized as a hierarchy, where the lower level templates are expansions of higher level ones. Some applications can use the following root template:

```
ResourceTemplate{
        Resource-name: <string> (domain name);
        Description: <abstract>;
         Related-materials: [
```

Relationships: [*LinkTo:<SemanticLinkType*$_1$, *Resource*$_1$>,
……,
LinkTo:<SemanticLinkType$_m$,
Resource$_m$>];
References: [*material-name*$_1$: <*address*$_1$>,
……,
material-name$_n$:<*address*$_n$>]
Others] }.

where an "address" can be a URL or a resource citation. For example, a book citation takes the form: <Name: *String*; Author: *String*; Publisher: *String*; PublisherAddress: *String* or URL>. A paper citation takes the form: (JournalName: *String*; Volume: *Number*; Issue: *Number*; PaperTitle: *String*; AuthorName: *String*; Publisher: *String*; PublisherAddress: *String* or URL).

"Related-materials" specifies relationships between resources and the references of the related resources. The relationship is a kind of semantic link that describes the relationship between resources.

"Others" can be the relationships such as "Peer-resources: <name-list>" and "Meta-resources: <name-list>".

To satisfy the pervasive residence criterion, the XML (www.w3.org/XML) or XML-based markup languages like RDF (Resource Description Framework) can be adopted to encode the worldwide resource space and local resource spaces. An example of representing a worldwide and a local knowledge space in XML is given in (H. Zhuge, Resource Space Grid: Model, Method and Platform, *Concurrency and Computation: Practice and Experience*, 16(13)(2004)).

In addition, the semantic link can be extended to the super link or the complex link to connect resources in the physical space, the socio space, and even the mental space in the future (H.Zhuge, Semantic linking through spaces for cyber-physical-socio intelligence: A methodology, *Artificial Intelligence*, 175(2011)988-1019).

3.7 The Resource Using Mechanism (RUM)

An RSM has two types of users: the end-users who use resources directly through an enabling interface or indirectly through an application system, and the application developers who build complex application systems for end-users with the support of the RUM.

The ROL defines a set of basic operations for creating resource spaces and for sharing and managing resources. Some of these operations allow the user to create a local space and to get resources for that space from the universal view of all the local resource spaces. Others allow the user to put a set of resources into a local space, to remove those resources if privileged to do so, to browse resources, to join a local space to the universal view or to separate it from that view, to open a local space to a specific set of users, and to join several spaces into one.

Thus the ROL includes operations called Create, Get, Place, Remove, Browse, Log, Open, Join, Separate, Merge and Split. These can be supplied with a list of resources and their locations, but at least one resource must be listed. Resources can be retrieved by specifying constraints in the condition portion of an ROL statement (discussed in detail in Chapter 4).

A resource browser has the ROL interpreter carrying out ROL statements, but also helps users to locate resources they need and display the content of those resources in a template form. A resource browser has the following main functions.

(1) Provide an easy-to-use interface for users to specify what they want to do.
(2) Provide either a local view or the universal view of all resources.
(3) Check the format and grammar of the ROL statements.
(4) Deliver operations to the ROL interpreter, and take results back.
(5) Show the results of those operations.

The browser allows a user to:

(1) choose an operation by clicking a button,
(2) choose the resources to be operated on by specifying their coordinates in a resource space, and
(3) put in or select parameters to refine the operation.

The RUM is responsible for carrying out the operations fed to it by the browser and feeding back the content of resources according to their type. Different types of resource may need specific RUM operations. An implementation of the RUM must

(1) ensure an appropriate granularity of operating object (for instance, rules should be used together because a single rule may not be meaningful and the deletion of a single rule may cause incompleteness of the component it belongs to),
(2) feed the resources back in a meaningful way, because the end-user may not be able to understand their original representation like formalism, and
(3) obtain any needed support from types of resource other than that of the resource being operated on.

The RUM can provide the following functions for experienced users:

(1) Apply reasoning, and supply explanation, based on the hierarchical structure of the space, the given rules and the experience extracted from the answers to users' queries.
(2) Make generalization and specialization on resources, classes and expressions.
(3) Convert raw resources into finely classified resources.
(4) Eliminate inconsistency and redundancy.

3.8 Comparisons

Viewing things from attributes and viewing things from classification represent different worldview of recognizing things. There are close relationship between the two views. The attribute view more focuses on individual and surface, while the classification view more concerns set and nature.

Identity is the foundation of the attribute view. Humans must assign every entity or attribute an identity. This is only feasible in close system as it is hard to establish consensus on identity in open system.

Abstraction on classifications takes the first priority of the classification view. The basic element of classification view is set, even though it can contain just one entity. A class can be assigned an identity, but a class can also be specified by its subclasses and super-class.

RSM is based on classifications, so the things managed by the RSM can be any form. The classification nature enables RSM to be suitable for managing contents by classification-based operations. This feature enables the RSM to manage semantics-rich Web resources. The classification nature enables the RSM to use various existing classification approaches. The classification nature also supports RSM to realize some automation, e.g., automatically generate the structure of resource space according to a set of given resources, and automatically upload resources into resource space. The automatic generation of resource space and adaptive resource space are research topics of RSM 2.0. The classification nature also enables RSM to manage resources in different spaces in the future cyber-physical society.

RDBM is based on identities and attributes of entities, so it is more suitable for managing atomic data and attribute-based operations. It basically does support abstraction on attributes.

Detailed comparisons are given in The Web Resource Space Model (H.Zhuge, Springer, 2008).

There are two commonalities between the RSM described here and the RDBM (R. Bocy, et al., "Specifying Queries as Relational

Expressions", *Communications of the ACM*, 18(11)(1975)621-628; E.F. Codd, "A Relational Model of Data for Large Shared Data Banks", *Communications of the ACM*, 13(6)(1970)377-387). The first is that the operations are separated from the objects to be managed. The second is the form of the operational languages: both SQL-like (ANSI, The Database Language SQL, Document ANSI X3.315, 1986; A.Eisenberg, and J.Melton, Sql:1999, Formerly Known as Sql3, *SIGMOD Record*, 28(1)(1999)131-138). This enables RDB users to easily understand the syntax and semantics of the ROL. The mapping between RSM and RDBM is introduced in (H. Zhuge, Y.Xing and P.Shi, Resource Space Model, OWL and Database: Mapping and Integration, *ACM Transactions on Internet Technology*, 8(4)(2008)article no.20).

The classification nature of RSM can also help users and the RSM systems (RSMS) to determine the classes of keywords in keyword-based search, e.g., in the example given in the beginning of this chapter.

Differences between OLTP and OLAP were compared in J. Han, and M. Kambr, *Data Mining: Concepts and Techniques* (Morgan Kaufmann Publishers, 2000). The multidimensional data model used for data warehousing and OLAP differs from the RSM in its foundation, its managed objects, its normalization, its operational features, and the basis for its exchange of data.

The object-oriented methodology (G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language: User Guide*. Reading, Mass.: Addison-Wesley, 1999) provides a method and mechanism for uniform domain modeling and system implementation. It reduces the complexity of systems of objects by using notions such as class and object to abstract a variety of entities, by encapsulating operations into each class, and by an inheritance mechanism. It supports reuse during software development. It does not have normal forms like RDBMS and RSM.

Object-relational databases (ORDBs) combine object-oriented and relational data base technologies (M. Stonebraker, P. Brown, and D. Moore, *Object-Relational DBMs: Tracking the Next Great Wave*, second ed., San Francisco: Morgan Kaufmann Publishers, 1999). In an

ORDB, a table need not be in the first normal form of the relational data base model, and tables can be nested. Various nested normal forms have been studied as extensions of traditional flat normal forms in RDBMs (W.Y. Mok, "A Comparative Study of Various Nested Normal Forms", *IEEE Trans. on Knowledge and Data Engineering*, 14(2)(2002); Z.M. Ozsoyoglu, and L.Y. Yuan, "A New Normal Form for Nested Relations", *ACM Trans. Database Systems*, 12(1)(1987)111-136; Z. Tari, J. Stokes, and S. Spaccapietra, "Object Normal Forms and Dependency Constraints for Object-Oriented Schemata", *ACM Trans. Database Systems*, 22(4)(1997)513-569), but these extensions are based on relational algebra and a relational data model. The RSM differs from the ORDB in its methodology, foundation, its managed objects, its data model, its normalization, and its operational features.

The resource browser helps users work on resources. Major differences between the resource browser and the typical current Web browser are as follows:

(1) the objects used by the resource browser are various resources, while the Web browser uses Web pages;
(2) the resource browser locates various resources by classification, while the Web browser locates Web pages by URL;
(3) the resource browser can locate resources and then operate on them by setting appropriate parameters, but the Web browser can only display Web pages according to URLs;
(4) the resource browser supports a uniform classification view on resources, while the Web browser only supports the viewing of a single Web page at a time; and
(5) the resource browser is supported by the RSM, while the current Web browser lacks the support of a coherent data model.

The design method for the RSM does not include a conceptual model, so a designer's experience and the reference model play key roles in designing a good resource space. The hierarchical resource organization approach is in line with top-down resource partitioning as described above, and with the "from general to specific" style of thought.

For application development, the ROL is not only an SQL-like language but also it uses XML syntax to support programming based on a semi-structured data model. The XML query language XQL is a concise language and is developed as an extension of the XSL pattern language. It builds upon the capability of identifying classes of nodes by applying Boolean logic, filters, and indexing to collections of nodes.

The ROL borrows its syntax and semantics from standard SQL. The statements of the ROL are SQL-like and have the SQL SELECT-FROM-WHERE pattern. The ROL can perform operations like those of the classical relational database, such as nested queries, aggregates, set operations, join and result ordering.

The ROL also borrows the following features from XML query languages:

(1) management of structured and semi-structured data;
(2) abstract data types;
(3) the XML-based data format and the result semantics;
(4) the skelom functions to associate a unique ID with a given resource space;
(5) document selection; and,
(6) partial path specification.

A detailed comparison is given in "Resource Space Grid: Model, Method and Platform" (H.Zhuge, *Concurrency and Computation*: *Practice and Experience*, 16(14)(2004) 1385-1413).

3.9 Dealing with Exponential Growth of Resources

Exponential growth of resources is an obstacle to effective management of resources. How to manage exponential expansion of resources becomes an important challenge. Some scientists like Jim Gray regarded the study of intensive data as a new science — the fourth paradigm for scientific exploration.

The RSM can solve the problem in theory by classifying resources and increasing the number of dimensions with the expansion of resources.

If expansion rate of resources is e^n, we could use a resource space with at least n-dimensions and each dimension having at least n coordinates to manage resource explosion since the following holds when $n \geq e$: $n^n \geq e^n$. Generally, if the expansion rate of resources is x^n, the following holds when $n \geq x$:

$$n^n \geq x^n.$$

If continuous expansion of resources accompanies appropriate classifications, this kind of expansion is controllable. For example, a 10-dimensional resource space with 10 coordinates at each dimension has 10^{10} points. If each point manages 100 resources in average, the resource space can contain 10^{12} resources. Increase one more dimension and one more coordinate at each dimension will enable the space to contain 11^{11} points. A 20-dimensional resource space with 20 coordinates at each dimension has 20^{20} points.

Furthermore, the increment of dimensions does not overload resource management. Any point can be accurately located by given its coordinates at each dimension.

Generally, resources are classified in different spaces. In each space, resources are classified by time, external feature, internal structure, and community. Further, resources are classified with the generation and evolution of dimensions.

Although people are facing unlimited expansion of resources, there are some limitations that can be used to deal with this expansion.

(1) Resources should have finite life spans, which could reduce expansion of resources to a certain extent.

(2) The expansion of dimensions is relatively slow compared with the rapid expansion of resources. That is, a new dimension may emerge when many resources are added.

(3) Users' interests, requirements and time of using resources are limited. That means users need the most interested resources and to get them rapidly.

Therefore, the following are ways to deal with the issue of rapid expansion of resources:

(1) Discovering self-organized communities of resources, and assigning constraints on the communities to regulate their evolution. This implies a dynamic data model. The problem is to find an appropriate schema for a community. Relational databases use data types to regulate data. Finding the schema for semantics-rich network is the basis for effectively managing the linked resources.
(2) Discovering the critical points to control the complex networks connecting resources.
(3) Linking appropriate resources to individuals according to their interests and requirements. This can be implemented by building a personal resource space that only includes resources matching the owner's interest and requirement. The personalized crawler searches those resources and uploads them into the personal resource space in time.
(4) Classifying resources according to resources' life span, reputation, and goodness to the society. The resources that are active, and have high reputation take the priority to be selected.
(5) Removing redundant resources.
(6) Recycling useless resources to meet the need of individuals if possible (e.g., editing text or video resources). Removing the useless and harmful resources is necessary in managing expanding resources. An example of the necessity is that people often remove junk emails to keep email box clean.

According to the report in *Nature* (May 12, 2011), MIT and Northeastern University have developed a computational model that can analyze any type of complex network and find the critical points that can be used to control the entire system. An algorithm has been designed to determine how many nodes in a network need to be controlled for total network control. The number of points needed depends on the network's degree distribution, which describes the number of connections per node. The researchers found that sparse networks require more controlled points than denser networks.

Data clean is a way to ensure the usefulness of data (V.Raman and J.M.Hellerstein, Potter's wheel: An interactive data cleaning system, *VLDB2001*, Roma, Italy).

Different from the other notions of dimension like that in SVM (C. Cortes and V.Vapnik, Support-Vector Networks. *Machine Learning*, 20(3)(1995) 273-297), dimension in RSM is a classification tree that can be mapped into an ontology hierarchy. The structure of the dimension in RSM reflects generalization and specialization on resources.

Harmony between the expansion of resources and the development of the environment is important for the future interconnection environment (H. Zhuge and X. Shi. Toward the eco-grid: a harmoniously evolved interconnection environment. *Communications of the ACM*, 47(9)(2004)78-83). A unified resource model called soft-device has been proposed for modeling resources in the future interconnection environment (H.Zhuge, Clustering Soft-Devices in Semantic Grid, *Computing in Science and Engineering*, 4(6)(2002)60-62).

A schema mechanism of the Semantic Link Network was suggested (H. Zhuge and Y. Sun, The schema theory for semantic link network. *Future Generation Computer Systems*, 26(3)(2010)408-420). If the physical characteristics of complex network such as the diameter and centrality can be considered, the schema can better reflect the nature of network. Further, it will be much better if the schema can adapt itself according to the change of network. We can image that a large-scale complex network of resources can be operated like database if an appropriate schema can be found. The classification-based Resource Space Model is likely to help implement this idea and support faced navigation on the network.

3.10 Extension of the Resource Space Model

3.10.1 *Formalizing resource space*

Let O be a domain terminology set, with a mapping from O onto the domain ontology that explains the domain's semantics. The resource space can be formalized as follows.

Definition 3.10.1 Let $S = 2^O$ be the power set of O. The resource space defined on O is represented as $RS(X_1, X_2, \ldots, X_n)$, where RS is the name of the space and $X_i = \{C_{i1}, C_{i2}, \ldots, C_{ip}\}$ is an axis, $1 \le i \le n$, C_{ij} is the root of the hierarchical structure of coordinates on X_i, $C_{ij} = \{<V_{ij}, E_{ij}> \mid V_{ij}$ is the set of coordinates, E_{ij} is the set of relations from coordinate $v_t \in V_{ij}$ to $v_s \in V_{ij}$ such that $R(v_t) \supseteq R(v_s)\}$, $1 \le j \le p$, where $R(v)$ is a class of resources represented by v. Every point in RS is an element of the Cartesian product $X_1 \times X_2 \times \ldots \times X_n$, represented as $p(x_1, x_2, \ldots, x_n)$.

Tuples of relational data models reflect the attributes of entities. In the RSM, x_i in a point $p(x_1, x_2, \ldots, x_n)$ reflects partitioning resources from one axis. Resources represented by a point $p(x_1, x_2, \ldots, x_n) \in RS$ can be represented as $R(p(x_1, x_2, \ldots, x_n)) = R(x_1) \cap R(x_2) \cap \ldots \cap R(x_n)$, where $R(x_i)$ is a class of resources represented by x_i, $1 \le i \le n$.

3.10.2 *Resource space schemas and normal forms*

A resource space schema formally describes a resource space. The major task in the logical design of a space is to specify its schema, and to define the axes and coordinates.

Application domains require that resources in a space schema satisfy certain integrity constraints. The schema should satisfy all these constraints, so it is defined as follows:

Definition 3.10.2 A resource space schema is a 5-tuple: $RS<A, C, S, dom>$, where

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

In applications, (4) and (5) should be determined before the schema is designed, so that the schema can be simplified as a triple: $RS{<}A, C{>}$.

The schema is relatively stable, while resource spaces can be dynamic due to the resource operations on the space. The design of a resource space is to determine its schema.

An axis with hierarchical coordinates can be transformed into an axis with flat coordinates if only the leaf nodes of each hierarchy are considered. Here we discuss only the flat case, and assume that an RS is always in 2NF. The equivalent definitions of the normal forms can be given.

For the space $RS(X_1, X_2, \ldots, X_n)$, we use $R(X_i)$ to denote resources represented by axis X_i, where $X_i = \{C_{i1}, C_{i2}, \ldots, C_{ip}\}$, $1 \le i \le n$. $R(X_i) = R(C_{i1}) \cup R(C_{i2}) \cup \ldots \cup R(C_{ip})$. First, we define fine classification.

Lemma 3.10.1 For two axes $X_i = \{C_{i1}, C_{i2}, \ldots, C_{ip}\}$ and $X_j = \{C_{j1}, C_{j2}, \ldots, C_{jq}\}$ of the space RS, if $R(X_\mathrm{i})=R(C_{i1})\cup R(C_{i2})\cup \ldots \cup R(C_{ip})$ and $R(X_\mathrm{j}) = R(C_{j1}) \cup R(C_{j2})\cup\ldots\cup R(C_{jp})$, $X_j/X_i \Leftrightarrow R(X_j) \subseteq R(X_i)$ hold.

From this we can redefine the notion of orthogonality.

Lemma 3.10.2 For two axes $X_i = \{C_{i1}, C_{i2}, \ldots, C_{ip}\}$ and $X_j = \{C_{j1}, C_{j2}, \ldots, C_{jq}\}$ in the space RS, $X_j \perp X_i \Leftrightarrow R(X_j) = R(X_i)$.

Proof: The lemma is approved as follows:

(1) If $X_j \perp X_i$, then we have X_j/X_i and X_i/X_j from the definition of orthogonality. From lemma 3.10.1 we have $R(X_j) \subseteq R(X_i)$ and $R(X_i) \subseteq R(X_j)$. So, $R(X_j) = R(X_i)$.

(2) If $R(X_j) = R(X_i)$, then $R(X_j) \subseteq R(X_i)$ and $R(X_i) \subseteq R(X_j)$. From lemma 3.10.1, we have X_j/X_i and X_i/X_j. That means $X_j \perp X_i$. From (1) and (2), we have $X_j \perp X_i \Leftrightarrow R(X_j) = R(X_i)$. □

The above lemma indicates that "two axes are orthogonal" is equivalent to that the expression ability of two axes is the same.

Clearly $X_i \perp X_j \Leftrightarrow X_j \perp X_i$, which means the orthogonal operation $\perp$ is symmetrical. From all this follows a new proof of the transitivity of *fine classification* and the *orthogonal* operation.

Theorem 3.10.1 The fine classification and orthogonal operations are transitive.

Proof: From Lemma 3.10.1 and Lemma 3.10.2, we can get $X_j/X_i \Leftrightarrow R(X_j) \subseteq R(X_i)$ and $X_j \perp X_i \Leftrightarrow R(X_j) = R(X_i)$. Because the set operations "$\subseteq$" and "=" are transitive, fine classification and the orthogonal operation is transitive. □

From this follows the definition of the third normal form.

Theorem 3.10.2 For space $RS(X_1, X_2, \ldots, X_n)$, RS is in 3NF $\Leftrightarrow R(X_1) = R(X_2) = \ldots = R(X_n)$, that is, every axis X_i can retrieve all the resources in RS.

Proof: Proof includes the following two aspects:

(1) If RS is in 3NF, then $X_1 \perp X_2 \perp \ldots \perp X_n$. From Lemma 3.10.2, $R(X_1) = R(X_2) = \ldots = R(X_n)$.

(2) Similarly, if $R(X_1) = R(X_2) = \ldots = R(X_n)$, we get $X_1 \perp X_2 \perp \ldots \perp X_n$, and because $\perp$ is both transitive and symmetrical, then, for any two axes X_i and X_j in RS, $X_i \perp X_j$. That means RS is in 3NF.

From (1) and (2), it follows that RS is in 3NF $\Leftrightarrow R(X_1) = R(X_2) = \ldots = R(X_n)$. □

Theorem 3.10.2 can be restated as a definition of the 3NF.

Beyond the three normal forms, other normal forms of the resource space schema can be defined for the convenience of partitioning and other operations.

Definition 3.10.3 (2^+NF) A space $RS(X_1, X_2, \ldots, X_n)$ is in 2^+NF, if it is in 2NF and $X_2/X_1, X_3/X_2, \ldots, X_n/X_{n\text{-}1}$.

The above definition means $R(X_1) \supseteq R(X_2) \supseteq \ldots \supseteq R(X_n)$ from Lemma 3.10.1. If *RS* is in 2^+NF, then, because fine classification / is transitive, we have: for every two axes X_i and X_j, $1 \le i \ne j \le n$, either X_i/X_j or X_j/X_i. So / is a full ordering on the set $\{X_1, X_2, \ldots, X_n\}$. On the other hand, it is obvious that if / constitutes a full ordering on the axes of *RS*, then *RS* is in 2^+NF. In the following, we discuss the properties of the 2^+NF under the operations on resource spaces.

Corollary 3.10.1 For two spaces RS_1 and RS_2, let $RS_1 \cdot RS_2 \Rightarrow RS$. Then, although both RS_1 and RS_2 are 2^+NF, *RS* needs not be in 2^+NF.

Proof: Suppose $RS_1 = \{X_1, X_2\}$ and $RS_2 = \{Y_1, Y_2\}$, where X_i and Y_i are axes and satisfy X_2/X_1, Y_2/Y_1 and $X_2 = Y_2$. Then, we can join RS_1 and RS_2. Let $RS_1 \cdot RS_2 \Rightarrow RS$, so that $RS = \{X_1, X_2, Y_1\}$.

(1) If either $R(X_1) \subseteq R(Y_1)$ or $R(Y_1) \subseteq R(X_1)$, either $R(X_2) \subseteq R(X_1) \subseteq R(Y_1)$ or $R(X_2) \subseteq R(Y_1) \subseteq R(X_1)$. From Definition 3.10.3, *RS* is in 2^+NF.

(2) Otherwise, if both $R(X_1) \subseteq R(Y_1)$ and $R(Y_1) \subseteq R(X_1)$ are false, then neither Y_1/X_1 nor X_1/Y_1. Since / is a full order on the axes of *RS,* if *RS* is in 2^+NF then *RS* is not in 2^+NF, a contradiction.

Therefore, from (1) and (2), *RS* is in 2^+NF. □

Corollary 3.10.1 tells us that 2^+NF does not persist under the Join operation. But if we add some conditions, 2^+NF will persist.

Corollary 3.10.2 (Join) Let $RS_1 = \{X_1, X_2, \ldots, X_n\}$ and $RS_2 = \{Y_1, Y_2, \ldots, Y_m\}$ be two 2^+NF resource spaces, and $RS_1 \cdot RS_2 \Rightarrow RS$. If $Y_1 = X_n$ or $X_1 = Y_m$, then RS is in 2^+NF.

Proof: (1) If $X_1 = Y_m$, then from $RS_1 \cdot RS_2 \Rightarrow RS$, we have $RS_2 = \{Y_1, Y_2, \ldots, Y_{m\text{-}1}, X_1, X_2, \ldots, X_n\}$. Since RS_1 and RS_2 are in 2^+NF, we have $X_n/X_{n\text{-}1}/\ldots/X_2/X_1$ and $Y_m/Y_{m\text{-}1}/\ldots/Y_2/Y_1$, then, from the transitivity of / we have: $X_n/X_{n\text{-}1}/\ldots/X_2/X_1 = Y_m/Y_{m\text{-}1}/\ldots/Y_2/Y_1$. From definition 3.10.3, RS is in 2^+NF. (2) Also, if $Y_1 = X_n$, RS is in 2^+NF for the same reason as (1). □

Corollary 3.10.3 If $RS \Rightarrow RS_1 \cdot RS_2$, and RS is in 2^+NF, then RS_1 and RS_2 are also in 2^+NF.

Proof: Suppose $RS = \{X_1, X_2, \ldots, X_n\}$. Because RS is in 2^+NF, and / is a full ordering on RS, and $RS \Rightarrow RS_1 \cdot RS_2$, then the axes of RS_1 are a subset of the axes RS, so / is also a full ordering on the axes of RS_1, and RS_1 is in 2^+NF. For the same reason, RS_2 is also in 2^+NF. □

Corollary 3.10.3 tells that 2^+NF persists under the operation Separate. From the definitions of Join and Separate, we can get $RS_1 \cdot RS_2 \Rightarrow RS$ if and only if $RS \Rightarrow RS_1 \cdot RS_2$. Then, from corollary 3.10.3, we have the following corollary.

Corollary 3.10.4 For resource spaces RS_1 and RS_2, let $RS_1 \cdot RS_2 \Rightarrow RS$. If either RS_1 or RS_2 is not in 2^+NF, then RS is not in 2^+NF.

From the above corollaries, we can get the following:

Corollary 3.10.5 If $RS \Rightarrow RS_1 \cdot RS_2$, and RS is not in 2^+NF, then either RS_1 or RS_2 or both could be in 2^+NF.

Corollary 3.10.6 For resource spaces RS_1 and RS_2, let $RS_1 \cup RS_2 \Rightarrow RS$. Then, if RS_1 and RS_2 are in 2^+NF, RS is in 2^+NF.

Proof: Suppose $RS_1 = \{X_1, X_2, \ldots, X_n\}$ satisfies $X_n/X_{n\text{-}1}/\ldots/X_2/X_1$, and $RS_2 = \{Y_1, Y_2, \ldots, Y_n\}$ satisfies $Y_n/Y_{n\text{-}1}/\ldots/Y_2/Y_1$. Since $RS_1 \cup RS_2 \Rightarrow RS$, RS_1 and RS_2 have n-1 common axes and one different axis. Suppose that $X_i = Y_i$, $1 \leq i \neq k \leq n$, and $X_k \neq Y_k$. Then, $RS = \{X_1, \ldots, (X_k \cup Y_k), \ldots, X_n\}$. Because $X_{k+1}/X_k/X_{k\text{-}1}$ and $X_{k+1} = Y_{k+1}/Y_k/Y_{k\text{-}1} = X_{k\text{-}1}$, from Lemma 3.10.1, we have: $R(X_{k\text{-}1}) \supseteq R(X_k) \supseteq R(X_{k+1})$ and $R(X_{k\text{-}1}) \supseteq R(Y_k) \supseteq R(X_{k+1})$.
So, $R(X_{k\text{-}1}) \supseteq (R(X_k) \cup R(Y_k)) \supseteq R(X_{k+1})$, which means $R(X_{k\text{-}1}) \supseteq R(X_k \cup Y_k) \supseteq R(X_{k+1})$. From Lemma 3.10.1 $X_{k+1}/(X_k \cup Y_k)/X_{k\text{-}1}$, so, $X_n/ \ldots /X_{k+1}/(X_k \cup Y_k)/X_{k\text{-}1}/ \ldots /X_1$, hence RS is in 2^+NF. □

This corollary tells us that 2^+NF persists under the Merge operation.

Corollary 3.10.7 Let $RS \Rightarrow RS_1 \cup RS_2$. Although RS is in 2^+NF, neither RS_1 nor RS_2 need be in 2^+NF.

Proof: Suppose $RS = \{X_1, X_2, X_3\}$, $X_3/X_2/X_1$ and $X_2 = X_2' \cup X_2''$. Then, $RS_1 = \{X_1, X_2', X_3\}$ and $RS_2 = \{X_1, X_2'', X_3\}$.

(1) If either $R(X_3) \subseteq R(X_2')$ or $R(X_2') \subseteq R(X_3)$, then, we have: either $R(X_3) \subseteq R(X_2') \subseteq R(X_1)$ or $R(X_2') \subseteq R(X_3) \subseteq R(X_1)$ respectively. From Definition 3.10.4, we have: RS_1 is in 2^+NF.

(2) Otherwise if neither $R(X_3) \subseteq R(X_2')$ nor $R(X_2') \subseteq R(X_3)$, then neither X_2'/X_3 nor X_3/X_2'. Since / is a full ordering on the axes of RS_1, if RS_1 is 2^+NF then RS_1 is not in 2^+NF.

According to (1) and (2), RS_1 need not be in 2^+NF. For the same reason, RS_2 needs not be in 2^+NF. □

Corollary 3.10.7 tells us that the 2^+NF does not persist under the Split operation. From Corollary 3.10.7 we have the next corollary.

Corollary 3.10.8 Let $RS \Rightarrow RS_1 \cup RS_2$, let RS be in 2^+NF, and let $RS = \{X_1, \ldots, X_{k\text{-}1}, X_k, X_{k+1}, \ldots, X_n\}$, $X_n/X_{n\text{-}1}/\ldots/X_2/X_1$, $X_k = X_k' \cup X_k''$, $RS_1 = \{X_1, \ldots, X_{k\text{-}1}, X_k', X_{k+1}, \ldots, X_n\}$, and $RS_2 = \{X_1, \ldots, X_{k\text{-}1}, X_k'', X_{k+1}, \ldots, X_n\}$, if $X_{k+1}/X_k'/X_{k\text{-}1}$, then RS_1 is in 2^+NF, and if $X_{k+1}/X_k''/X_{k\text{-}1}$, then RS_2 is in 2^+NF.

From these three corollaries, we have the following:

Corollary 3.10.9 For two spaces RS_1 and RS_2, let $RS_1 \cup RS_2 \Rightarrow RS$. Although neither RS_1 nor RS_2 is in 2^+NF, RS could be.

Corollary 3.10.10 Let $RS \Rightarrow RS_1 \cup RS_2$. Although RS is not in 2^+NF, either RS_1 or RS_2 or both could be.

The 2^+NF is the weakened form of the 3NF. We can also define a strengthened form of the 3NF as follows:

Definition 3.10.4 A space $RS(X_1, X_2, \ldots, X_n)$ is 4NF if it is a 3NF, and for any point $p(x_1, x_2, \ldots, x_n) \in RS$, $R(p(x_1, x_2, \ldots, x_n)) = R(x_1) \cap R(x_2) \cap \ldots \cap R(x_n) \neq \Phi$.

Because a 4NF space is also in a space 3NF, it has the same properties as 3NF in a space under resource space operations.

3.10.3 *Topological properties of resource spaces*

If we define a distance between two points in an n dimensional space $RS(X_1, X_2, \ldots, X_n)$, then the distance can be used to define a topological space. We focus on the 2NF space, and first define a distance d on axis X_i, $1 \leq i \leq n$, then construct from d a distance D on the whole space RS.

For a given set G, if there exists a function d: $G \times G \rightarrow \Re^+$, where $\Re^+$ represents the set of non-negative real numbers, then d is called a distance on G if it satisfies the following three axioms:

Axiom 1. $d(g_1, g_2) = 0 \Leftrightarrow g_1 = g_2$.
Axiom 2. $d(g_1, g_2) = d(g_2, g_1)$.
Axiom 3. $d(g_1, g_2) \leq d(g_1, g_3) + d(g_3, g_2)$, for any g_1, g_2 and $g_3 \in G$.

For an axis $X = \{C_1, C_2, \ldots, C_n\}$, C_i is a coordinate hierarchy denoted as $<V_i, E_i>$, where V_i is a set of sub-coordinates, and E_i is the subclass relation between sub-coordinates. We define the function d on X as follows.

Definition 3.10.5 For points x_1 and x_2 on axis X,
$d(x_1, x_2) =$

$$\begin{cases} 0, & \text{if } x_1 = x_2. \\ \infty, & \text{if } x_1 \in V_i, x_2 \in V_j \text{ and } i \neq j. \\ \min\{length(\Gamma) \mid \Gamma = (x_1, x_1', \cdots, x_m', x_2)\} & \text{if } x_1 \text{ and } x_2 \in V_i, \text{and } x_1 \neq x_2. \end{cases}$$

where $< x_1, x_1' >, < x_j', x_{j+1}' >, < x_m', x_2 > \in E_i, 1 \leq j \leq m-1$, $length(\Gamma)$ is the length of the path Γ with a weight on each link. And, we make a reasonable assumption: if x_1 and $x_2 \in V_i$ and $x_1 \neq x_2$, there is a path $\Gamma = (x_1, x_1', \ldots, x_k', x_2)$ from x_1 to x_2. So $d(x_1, x_2) < length(\Gamma) < \infty$.

Theorem 3.10.3 *d is a distance on axis X.*

In the following, we first give the definition of function D on RS, and then prove that it is a distance on RS.

Definition 3.10.6 For any two points $p_1(x_1, x_2, \ldots, x_n)$ and $p_2(y_1, y_2, \ldots, y_n)$ in the space $RS(X_1, X_2, \ldots, X_n)$, we define

$$D(p_1, p_2) = (\sum_{i=1}^{n} d^2(x_i, y_i))^{\frac{1}{2}},$$

where d is the distance on axis X_i, $1 \leq i \leq n$.

Theorem 3.10.4 *D is a distance on RS.*

So the space $RS(X_1, X_2, \ldots, X_n)$ is a metric space (RS, D) with distance D. The distance D in RS defines a discrete topological space (RS, ρ). The following discusses the properties of the topological space (RS, ρ).

According to the definition of distance d, we have $d(x_1, x_2) < \infty \Leftrightarrow x_1$ and x_2 belong to the same coordinate hierarchy.

Definition 3.10.7 For two points $p_1(x_1, x_2, \ldots, x_n)$ and $p_2(y_1, y_2, \ldots, y_n)$ in the resource space *RS*, p_1 is said to be linked to p_2 if $D(p_1, p_2) < \infty$. For a set of points P in *RS*, P is called a linked branch if for any two points p_i and p_j in P $(i \neq j)$, p_i is linked to p_j.

From Definition 3.10.7 comes the following corollary.

Corollary 3.10.11 In a space $RS(X_1, X_2, \ldots, X_n)$, if a set of points P constitutes a linked branch, then for any two points $p_1(x_1, x_2, \ldots, x_n)$ and $p_2(y_1, y_2, \ldots, y_n)$ in P, x_i and y_i $(1 \leq i \leq n)$ belong to the same coordinate hierarchy.

Proof: If P is a linked branch, then for any two points $p_1(x_1, x_2, \ldots, x_n)$ and $p_2(y_1, y_2, \ldots, y_n)$ in P, $D(p_1, p_2) < \infty$. Since

$$D(p_1, p_2) = (\sum_{i=1}^{n} d^2(x_i, y_i))^{\frac{1}{2}},$$

we can get $d(x_i, y_i) < \infty$, $1 \leq i \leq n$. Hence, x_i and y_i belong to the same coordinate hierarchy. □

Corollary 3.10.11 tells us the following rules:

If two points in a space are linked to each other, then their corresponding coordinates belong to the same coordinate hierarchy.

It is obvious that the connective relation (denoted by ~) is an equivalent relation on the topological space *RS*. So, *RS*/~ is a quotient space of *RS*. The next corollary describes the structure of the quotient space *RS*/~.

Corollary 3.10.12. The quotient space *RS*/~ =

{ $p^{'}(C_{i1}^{1}, C_{i2}^{2}, \cdots, C_{in}^{n})$ | C_{ik}^{k} is a root coordinate on axis X_k in $RS(X_1, X_2, \ldots, X_n)$, $1 \le k \le n$}, where $p'(x_1, x_2, \ldots, x_n)$ in $RS/\sim$ is the linked branch including point $p(x_1, x_2, \ldots, x_n)$ in RS.

Proof: Proof consists of the following two aspects:
(1) It is clear that any point $p^{'}(C_{i1}^{1}, C_{i2}^{2}, \cdots, C_{in}^{n})$ is in $RS/\sim$. So $RS/\sim \supseteq$ { $p^{'}(C_{i1}^{1}, C_{i2}^{2}, \cdots, C_{in}^{n})$ | C_{ik}^{k} is a root coordinate on axis X_k in RS}.
(2) For any point $p(x_1, x_2, \ldots, x_n)$ in $RS(X_1, X_2, \ldots, X_n)$, from Corollary 3.8.11, we get that there exists a root coordinate C_{i1}^{1} on axis $X_1, \ldots,$ and C_{in}^{n} on axis X_n, such that x_1 is in C_{i1}^{1}, ..., and x_n is in C_{in}^{n}. So $p(x_1, x_2, \ldots, x_n)$ is in the linked branch of $p^{'}(C_{i1}^{1}, C_{i2}^{2}, \cdots, C_{in}^{n})$, which means $p^{'}(x_1, x_2, \cdots, x_n) = p^{'}(C_{i1}^{1}, C_{i2}^{2}, \cdots, C_{in}^{n})$. Then, we have $RS/\sim \subseteq$ { $p^{'}(C_{i1}^{1}, C_{i2}^{2}, \cdots, C_{in}^{n})$ | C_{ik}^{k} is a root coordinate on axis X_k in RS}.

From (1) and (2), $RS/\sim =$ { $p^{'}(C_{i1}^{1}, C_{i2}^{2}, \cdots, C_{in}^{n})$ | C_{ik}^{k} is a root coordinate on axis X_k in $RS(X_1, X_2, \ldots, X_n)$, $1 \le k \le n$}. □

In the quotient space $RS/\sim$, we can define a distance $D_\sim$ on $RS/\sim$ as induced from the distance D on RS. $D_\sim(p_1', p_2') = min$ $\{D(p_1, p_2) \,|\, p_1 \in p_1'$ and $p_2 \in p_2'\}$, where p_1' and p_2' represent the linked branches including p_1 and p_2 respectively. Then, for any $p_1', p_2' \in RS/\sim$, $p_1' \neq p_2'$, $D_\sim(p_1', p_2') = \infty$, $D_\sim(p_1', p_1') = 0$, which means that $RS/\sim$ is a discrete topological space with the distance $D_\sim$ on it.

The resource space RS enables us to locate resources by coordinates. The quotient space $RS/\sim$ enables us to search in a more abstract space.

Theorem 3.10.5 A point exists in RS if and only if it belongs to a point of $RS/\sim$.

Proof: Proof consists of the following two aspects:
(1) For a $p(x_1, x_2, \ldots, x_n)$ in RS, from Corollary 3.10.12, there exists $p^{'}(C_{i1}^{1}, C_{i2}^{2}, \cdots, C_{in}^{n})$ in $RS/\sim$ such that $p(x_1, x_2, \ldots, x_n)$ is in the linked

branch of $p^{'}(C_{i1}^{1}, C_{i2}^{2}, \cdots, C_{in}^{n})$. So $p(x_1, x_2, \ldots, x_n)$ belongs to a point of *RS*/~.
(2) Suppose $p(x_1, x_2, \ldots, x_n)$ belongs to a point $p'(x_1, x_2, \ldots, x_n)$ in *RS*/~. From Corollary 3.10.12, all the points in the linked branch $p'(x_1, x_2, \ldots, x_n)$ are in *RS*, so $p(x_1, x_2, \ldots, x_n)$ exists in *RS*.

From (1) and (2), we can infer that a point is in *RS* if and only if it also belongs to a point in *RS*/~. □

This theorem provides a top-down refinement search strategy for a large-scale space:

From the quotient space down to the resource space.

The strategy also ensures that all resources in space *RS* can be found through *RS*/~.

3.11 Integrity Constraints for the Resource Space Model

The integrity constraints for the RSM are of four kinds: *entity*, *membership*, *referential* and *user-defined.* These work together so that the RSM can correctly and efficiently specify and manage resources.

3.11.1 Entity integrity constraints

In relational databases, keys play a fundamental role in the data model and in conceptual design. They enable tuples to refer to one another and ensure that operations can accurately locate tuples.

As a coordinate system, naturally the RSM supports precise resource location. However, it is not always necessary to have the user painstakingly specify all the coordinates of a point, especially when an axis is added. The RSM needs better resource location.

Definition 3.11.1. Let $p \cdot X_i$ be the coordinate of p at axis X_i in $RS(X_1, X_2, \ldots, X_n)$, that is, the projection of p on X_i. If $p_1 \cdot X_{\mathrm{i}} = p_2 \cdot X_i$ for $1 \leq i \leq n$, then we say that p_1 is equal to p_2, denoted by $p_1 =_p p_2$.

Using this definition, a *candidate key* of the RSM can be defined as follows.

Definition 3.11.2. Let CK be a subset of $(X_1, X_2, \ldots, X_n)$, and let p_1 and p_2 be non-null points in $RS(X_1, X_2, \ldots, X_n)$. CK is called a candidate key of RS if we can derive $p_1 =_p p_2$ from $p_1 \cdot X_i = p_2 \cdot X_i$, where $X_i \in CK$.

A candidate key is specific enough to identify non-null points of a given space.

The *primary key* is a candidate key specified by the designer of the space. The axes of the primary key are called *primary axes*.

Point constraint. If axis X is a primary axis of the space RS, then no X coordinate of any point in RS should be null.

This constraint is used to ensure that primary keys can distinguish non-null points in a given space. One type of null value is "at present unknown".

In the RSM one can infer some keys from the presence of others. This is of great importance in query optimization, especially when creating new spaces. Inference rules for candidate keys come from the following four theorems.

Theorem 3.11.1. If a set of axes CK is a candidate key of the space RS, then any axis set that includes CK is also a candidate key of RS.

From the definition of Join we have the following theorem.

Theorem 3.11.2. Let RS_1 and RS_2 be two spaces which can be joined to produce a new space RS. If CK_1 and CK_2 are candidate keys of RS_1 and RS_2 respectively, then $CK = CK_1 \cup CK_2$ is a candidate key of RS.

From the definition of Merge we have the following theorem.

Theorem 3.11.3. Let RS_1 and RS_2 be two spaces that can be merged into one space RS. Let X_1 and X_2 be two different axes of RS_1 and RS_2 respectively, and let $X_c = X_1 \cup X_2$. If CK_1 and CK_2 are candidate keys of RS_1 and RS_2 respectively, then $CK = (CK_1 - \{X_1\}) \cup (CK_2 - \{X_2\}) \cup \{X_c\}$ is a candidate key of RS.

From the definition of Split we have the following theorem.

Theorem 3.11.4. Let RS_1 and RS_2 be two spaces created by splitting the space RS. Suppose that the axis X_c of RS is split into X_1 and X_2 belonging to RS_1 and RS_2 respectively. Let CK be a candidate key of RS. If $X_c \not\in CK$, let $CK_1 = CK_2 = CK$, otherwise let $CK_1 = CK - \{X_c\} \cup \{X_1\}$ and $CK_2 = CK - \{X_c\} \cup \{X_2\}$. Then CK_1 and CK_2 are candidate keys of RS_1 and RS_2 respectively.

Proof. Let A be the set of all axes of RS and A_1 be the set of all axes of RS_1. Assuming that CK_1 is not a candidate key of RS_1, there must be two non-null points p_1 and p_2 in RS_1 which satisfy both $(\forall X \in CK_1)$ $(p_1 \cdot X = p_2 \cdot X)$ and $(\exists X^* \in A_1)$ $(p_1 \cdot X^* \neq p_2 \cdot X^*)$. Let p_1' and p_2' in RS have the same coordinate values as p_1 and p_2 respectively. Clearly, $(\forall X \in CK)$ $(p_1' \cdot X = p_2' \cdot X)$ if $CK_1 = CK$ or $CK_1 = CK - \{X_c\} \cup \{X_1\}$.

(1) When $CK_1 = CK$, if $X^* \neq X_1$, then $p_1' \cdot X^* \neq p_2' \cdot X^*$, otherwise $p_1' \cdot X_c \neq p_2' \cdot X_c$;

(2) When $CK_1 = CK - \{X_c\} \cup \{X_1\}$, then $X^* \neq X_1$. So $p_1' \cdot X^* \neq p_2' \cdot X^*$.

From (1) and (2), $p_1' \neq_p p_2'$. Clearly, this conclusion contradicts the assumption that CK is a candidate key of RS. So, CK_1 is a candidate key of RS_1. Similarly, we can prove that CK_2 is a candidate key of RS_2. □

In resource space systems, there are often spaces created by join, merge and split operations. Theorems 3.10.2, 3.10.3 and 3.10.4 provide an efficient means of deriving candidate keys of these spaces.

In the RSM, a resource entry denoted by a 3-tuple $Resource\_Entry <ID, Index, Description>$ is used to index into a resource representation layer. The *ID* field is used to specify the entries at a given point. Two entries at different points could have the same *ID*. The *Index* field is the index data linked to the representation layer. Description of resources concerns internal semantics and external semantics. To facilitate operations on resources, the *Description* can simply use a set of attributes to reflect a resource, while leave the detailed descriptions to the representation layer. In the following discussion, $re \cdot ID$, $re \cdot index$ and $re \cdot SD$ denote the *ID*, *index* and *Description* of entry *re* respectively.

Resource entry constraint 1. No ID should be null, and for any two entries re_1 and re_2 at the same non-null point, $re_1 \cdot ID \neq re_2 \cdot ID$.

This constraint requires that all entries in a given non-null point should have distinct *ID*s. This ensures that any operation can precisely locate its target entry.

Resource entry constraint 2. No index of an entry should be null, and for any two entries re_1 and re_2 at the same non-null point, $re_1 \cdot index \neq re_2 \cdot index$.

This constraint requires that:

(1) every entry should include index data linking to the representation layer, and
(2) no two entries at the same non-null point should have the same index data.

Otherwise, it will lead to information redundancy and unnecessary maintenance of consistency between resource entries at the same point.

The syntactic structure of the index data of entries depends on the implementation of the representation layer. For instance, an XML-based implementation of a representation layer commonly uses XPath expressions, whereas filenames are often used for file-based implementations.

To analyze the index of a resource entry, not only the syntactic structure but also the semantics should be considered. For example, an absolute path differs from a relative path syntactically. However, these two types of paths may indicate the same data.

Resource entry constraint 3. The semantic description *SD* of any entry should not be null, and no two entries re_1 and re_2 at the same non-null point should be the same or imply each other, that is, neither $re_1 \cdot SD \Rightarrow re_2 \cdot SD$ nor $re_2 \cdot SD \Rightarrow re_1 \cdot SD$.

This is the entity integrity constraint for the *Description* of an entry. It is optional but stricter than constraint 2. Since a $re \cdot SD$ embodies the semantic existence of an entry in a resource space, clearly $re \cdot SD$ should not be null. Furthermore, entries at a given non-null point should neither be the same nor imply each other. For example, a resource and its copies are allowed to coexist at a non-null point by constraint 2, but not by constraint 3.

3.11.2 The membership integrity constraint

In relational databases, a tuple can be inserted into a table only if all fields of the tuple satisfy the domain constraints of the table. So the relationship between the tuple and the table should be checked before insertion.

In the RSM, a resource space holds the classification of its resources. The existence of entry *re* at point *p* means that the resource indexed by *re* belongs to the type represented by *p*. An entry can be placed at a point by the following operation:

$$\textbf{PLACE}\ re <ID, Index,\ Description>\ \textbf{AT}\ p\ (C_{1,i1}, C_{2,i2}, \ldots, C_{n,in}).$$

If there were no restrictions, an entry could be placed at any point of the space. So, checking the memberships of resource entries plays an important role in the RSM.

$R_{\Delta}(RS)$, $R_{\Delta}(C)$ and $R_{\Delta}(p)$ denote the sets of resources currently stored by space *RS*, coordinate *C* and point *p* respectively. For any entry *re*, if *re* has been placed at the point *p*, then $re \in R_{\Delta}(p)$.

Membership constraint. Let *re* <*ID*, *Index*, *Description*> be a resource entry. For any point $p\,(C_{1,i1}, C_{2,i2}, \ldots, C_{n,in})$ in a given space, putting *re* into the space should be constrained by the definition of the point $R\,(p)$, that is, $re \in R_{\Delta}(p) \rightarrow re \in R\,(p)$.

An entry *re* can be placed at point *p* only if *re* belongs to the type that *p* represents. Constraining membership in this way can ensure correct resource classification. When a place or update operation is applied to an entry, this constraint should be checked.

3.11.3 Referential integrity constraints

In relational databases, it is often required that a value that appears in one relation for one set of attributes should also appear for another set of attributes in another relation. This condition is called a referential integrity constraint.

In the following discussion, three types of referential integrity constraints for the RSM are considered.

In the RSM, the basic function of an entry is to index a resource in the representation layer. For any entry *re*<*ID*, *index*, *Description*>, $re \cdot index$ is the index of the resource. Resource entry constraint 2 ensures that $re \cdot index$ is non-null. But it cannot ensure that $re \cdot index$ makes sense. This is mainly because modifications to entries or representation layers may cause the indices of entries to become dead links. The first referential integrity constraint is intended to eliminate dead links

Referential constraint 1. For every entry *re* in a resource space system, there exists a resource in the representation layer which is referred to by its index ($re \cdot index$).

The resource space layer refers to the representation layer. The above constraint ensures that $re \cdot index$ makes sense for any entry *re*. This constraint should be checked when a *re* is placed or a $re \cdot index$ updated.

When changes take place in a representation layer, this integrity should also be satisfied. The layer can be viewed as a Semantic Link Network (SLN). An SLN consists of semantic nodes and semantic links. A semantic node can be an atomic node (a piece of text or image) or complex node (another SLN). For most applications, the domain is a subset of the whole resource representation layer. This subset SLN is denoted by SLN^*.

In the resource representation layer, any resource denoted by a 3-tuple *Resource*(*ID*, *Description*, *Resource*-*Entry*-*List*) can be regarded as a semantic node. The *ID* is the identifier of a resource in a given SLN^*. It is helpful for locating the target resource. The *Description* is the detailed description of a resource, used to facilitate operations. The *Description* of a resource can be represented by an SLN. There may exist many indices (resource entries) in the resource space layer to a resource in the representation layer. The *Resource*-*Entry*-*List* of a resource is used to record all entries indexing this resource.

For any entry, an item of the *Resource*-*Entry*-*List* should include the name of its space, the coordinate of its point, and its *ID*. So, from the *Resource*-*Entry*-*List*, all corresponding entries can be obtained. In the following $res \cdot ID$, $res \cdot SD$ and $res \cdot REL$ denote the *ID*, *Description* and *Resource*-*Entry*-*List* of resource *res* respectively.

There exists a variety of relations between resources. The following discussion is about the relations of similarity and inclusion. Two functions, *Similarity* and *Inclusion,* are introduced to evaluate the similarity and inclusion between resources. $Similarity(res_1, res_2)$ returns a real number between 0 and 1 giving the similarity between resources

res_1 and res_2. $Inclusion\,(res_1, res_2)$ returns a real number between 0 and 1 giving the degree of inclusion of resource res_2 in resource res_1.

For a given threshold value δ, if $Similarity\,(res_1, res_2) \geq \delta$, then resources res_1 and res_2 are regarded as equal. And, if $Inclusion\,(res_1, res_2) \geq \delta$, then res_2 is viewed as a subset of res_1. Both equality and inclusion between resources lead to semantic redundancy and unnecessary maintenance of consistency between resources in a given SLN*. To eliminate this redundancy, the following constraint is introduced.

Resource redundancy constraint. Let res_1 and res_2 be two resources in the SLN*. For a given threshold value δ, both $Similarity\,(res_1, res_2) < \delta$ and $Inclusion\,(res_1, res_2) < \delta$.

Before placing a resource in the SLN* or after updating a resource, this constraint should be checked. In the case of placement, if the above constraint has been violated, the operation will be canceled. But suppose resource res_1 is to be updated to res_1'. If there already exists a resource res_2 such that either $Similarity\,(res_1', res_2) \geq \delta$ or $Inclusion\,(res_1', res_2) \geq \delta$, the alternative actions are:

(1) the update operation is canceled, or
(2) after the update of res_1, resource res_2 is deleted.

If the second action is taken, the alternative actions are:

(1) all resource entries indicated by $res_2 \cdot REL$ are deleted, or
(2) all resource entries indicated by $res_2 \cdot REL$ are redirected to res_1'.

Thus, dead links in the space layer will be avoided after the changes take place in the representation layer.

The first referential integrity constraint specifically between resource spaces relates to the join operation.

Referential constraint 2. If RS_1, RS_2 and RS are three spaces that satisfy $RS_1 \cdot RS_2 \Rightarrow RS$, then $R_\Delta(RS) \subseteq R_\Delta(RS_1) \cup R_\Delta(RS_2)$.

RS is derived from RS_1 and RS_2, and this referential constraint maintains the dependency of RS on RS_1 and RS_2. Thus, when an entry is placed in RS or removed from RS_1 or RS_2, this constraint should be checked.

The second type of referential integrity constraint applies to 3NF spaces. We first define the *foreign* key of the RSM.

Definition 3.11.3. Let S be a subset of axes of the space RS_1, but not the primary key of RS_1. If there exists another space RS_2 such that $R(RS_1) = R(RS_2)$ and S is the primary key of RS_2, then S is called the foreign key of RS_1, RS_1 is called the referencing space of RS_2 and RS_2 is called the referenced space of RS_1.

From this definition, we have the following theorem.

Theorem 3.11.5. Let $S = \{X_1, X_2, \ldots, X_m\}$ be the foreign key of the referencing space $RS_1(X_1, X_2, \ldots, X_m, X_{m+1}, \ldots, X_n)$, and $RS_2(X_1, X_2, \ldots, X_m, Y_{m+1}, \ldots, Y_t)$ be the corresponding referenced space. For two non-null points $p(C_1, C_2, \ldots, C_m, C_{m+1}, \ldots, C_n)$ and $p'(C_1, C_2, \ldots, C_m, C'_{m+1}, \ldots, C'_t)$ in RS_1 and RS_2 respectively, $R(p) \subseteq R(p')$.

This theorem indicates the inclusion relationship between points in the referencing space and their counterparts in the referenced space. The next constraint aims to maintain the legal referential relationship between the referencing space and its referenced space.

Referential constraint 3. Let $S = \{X_1, X_2, \ldots, X_m\}$ be the foreign key of the referencing space $RS_1(X_1, X_2, \ldots, X_m, X_{m+1}, \ldots, X_n)$, and $RS_2(X_1, X_2, \ldots, X_m, Y_{m+1}, \ldots, Y_t)$ be the corresponding referenced space. For two non-null points $p(C_1, C_2, \ldots, C_m, C_{m+1}, \ldots, C_n)$ and $p'(C_1, C_2, \ldots, C_m, C'_{m+1}, \ldots, C'_t)$ in RS_1 and RS_2 respectively, $R_\Delta(p) \subseteq R_\Delta(p')$.

This constraint ensures that if an entry *re* appears at a certain point *p* in the referencing space, then *re* must exist as the counterpart of *p* in the referenced space.

3.11.4 *User-defined integrity constraints*

Any resource space system should conform to the entity, membership and referential integrity constraints. In specific applications, different space systems should obey certain context-relevant constraints. These constraints are called user-defined integrity constraints. The following introduces three frequently used types of user-defined constraints. Two spaces shown in Fig. 3.11.4 are used to illustrate these constraints.

In Fig. 3.11.4 (a), the resource space *Salary-Post* is used to hold data about employees. Every point classifies these employees by their salary and post. In Fig. 3.11.4 (b), *Keeper-Warehouse* is used to hold data about goods. Each point of *Keeper-Warehouse* classifies these goods by their keeper and warehouse.

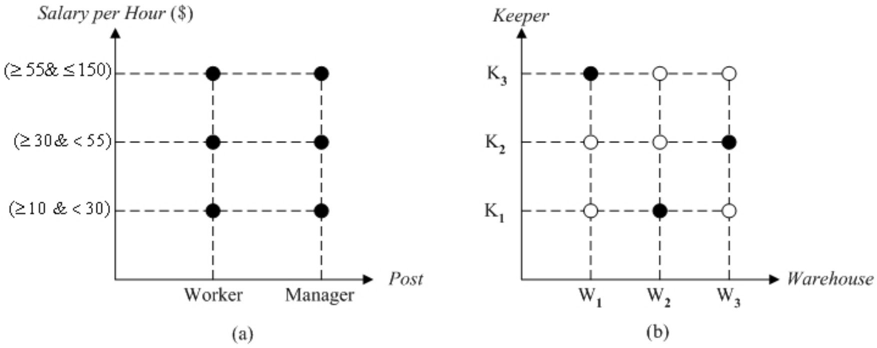


Fig. 3.11.4 Examples of bidimensional resource spaces.

User-defined constraints require the attribute values in the resource description to satisfy some rules. The function *GetAttribute* (*re*, *attr*) returns the value of attribute *attr* specified in the *Description* of the entry *re*. The Boolean function *JudgeRelation* (*operand*$_1$, *operand*$_2$, *relational-operator*) judges whether *operand*$_1$ and *operand*$_2$ satisfy the relation specified by *relational-operator*. This constraint can be described as follows:

<Constraint expression> ::=
JudgeRelation(*GetAttribute*(*re*, *attr*), user-defined-constant-value, <Relational-Op>) |
<Constraint expression> $\vee$ <Constraint expression> |
<Constraint expression> $\wedge$ <Constraint expression> |
$\neg$<Constraint expression>;
<Relational-Op> ::= < | > | = | ≤ | ≥ |≠.

Take Fig.3.11.4(a) for example. If the resource space designer requires of *Salary-Post* that the salary per hour of any worker should not be lower than \$12 and that the salary per hour of any manager should not be lower than \$50, then this user-defined constraint for entry *re* is:

(*JudgeRelation* (*GetAttribute* (*re*, *post*), "worker", =)
$\wedge$ *JudgeRelation* (*GetAttribute* (*re*, *salary per hour*), 12, ≥)) $\vee$
(*JudgeRelation* (*GetAttribute* (*re*, *post*), "manager", =)
$\wedge$*JudgeRelation* (*GetAttribute* (*re*, *salary per hour*), 50, ≥)).

Before the entry *re* can be placed in *Salary-Post* or updated, the system should check whether the above constraint has been violated.

In some applications, rich semantic relations among entries should be taken into consideration. Operations on an entry may require other operations on semantically relevant entries. This type of user-defined constraint is called a resource-entry-based constraint. For example, suppose *RS* is a space holding all the registration data about students of a school and *RS′* is another holding all the health data of the same students. Let *re* be the entry holding a particular student's registration data and *re′* be the entry holding his/her health data. The health data depend on the validity of the registration data, that is, $re' \in R_{\Delta}(RS') \rightarrow re \in R_{\Delta}(RS)$. So this constraint should be checked before *re′* is placed or after *re* is deleted.

As resource sets, points are often required to satisfy some application relevant rules from the viewpoint of set theory. Take Fig. 3.11.4 (b) for example. Suppose that a warehouse could have only one keeper in *Keeper-Warehouse* and that each keeper is in charge of only one warehouse.

For any K_i, there exists at most one W_j such that $R_{\Delta}(p(K_i, W_j)) \neq \varnothing$, and for any W_m there exists at most one K_n such that $R_{\Delta}(p(K_n, W_m)) \neq \varnothing$. We define the following function:

$$NotNull(\boldsymbol{p})=\begin{cases}1, & R_{\Delta}(\boldsymbol{p})\neq\varnothing\\ 0, & R_{\Delta}(\boldsymbol{p})=\varnothing\end{cases}$$

And, use p_{ij} to denote the point $p(K_i, W_j)$. Then, the formal description of this constraint is as follows:

$$\forall i(\sum_{j=1}^{3} NotNull(\boldsymbol{p}_{ij}) \leq 1) \wedge \forall j(\sum_{i=1}^{3} NotNull(\boldsymbol{p}_{ij}) \leq 1).$$

Thus, before any goods can be placed in *Keeper-Warehouse*, the system must check whether the above constraint is violated or not.

The effectiveness of resource use also depends on the users' classifications in their mental spaces and the semantic relationships between resources. A way is to establish the "Fuzzy Resource Space Model and Platform". Further discussion can be found in (H. Zhuge, *Journal of Systems and Software*, 73(3) (2004)389-396).

3.12 Storage for Resource Space and Adaptability

An efficient storage mechanism is critical for resource space systems. To store resource space is to find an appropriate way to index multidimensional and hierarchical structure so as to support various queries. There are multiple ways to implement the storage of a resource space. The following are several feasible ways.

Making use of the storage mechanism of relational database to store resource space. This just needs to transform resource space into relational tables. The underlying resource management mechanism is implemented by the database system. In this case, a resource space is just like a multi-dimensional classification view. However, a good resource space may not correspond to a good relational database due to their different normal form definitions. The mapping between the Resource Space Model and the relational database has been introduced in "Resource Space Model, OWL and Database: Mapping and Integration" (H.Zhuge, et al., *ACM Transactions on Internet Technology*, 8(4)(2008) article no.20).

Making use of XML and its operation mechanisms. The hierarchical structure supports the representation of resource space, while the efficiency of resource space operation will depend on the efficiency of XML operation mechanisms.

Making use of peer-to-peer mechanism to store resource space. This is suitable for decentralized applications. Several proposals have been proposed, e.g., in RSM-Based Gossip on P2P Network (H.Zhuge, *The 7th International Conference on Algorithms and Architectures for Parallel Computing*, IC^{3}PP07, Hangzhou, China, 2007). The structured and unstructured P2P resource space systems have been introduced in The Web Resource Space Model (H.Zhuge, Springer, 2008).

Developing new indexing mechanisms to support the characteristics of the resource space. Solutions such as CTree and C*Tree have been proposed (H.Zhuge, The Web Resource Space Model, Springer, 2008; Q.Zeng and H.Zhuge, C*-tree: A Multi-Dimensional Index Structure for Resource Space Model, SKG2010). The spatial indexes like R-tree (A.Guttman, R-trees: a dynamic index structure for spatial searching, SIGMOD'84, pp.47-57) are more suitable for indexing flat and continuous dimensions. Generally, a resource space with hierarchical coordinate structure can be transformed into a hierarchical structure of spaces where the spaces at each level only have flat axes, and each low-level space corresponds to a point in a high-level resource space. This structure keeps the semantics of the original resource space but may lead to high space complexity.

An ideal RSM storage mechanism should be adaptive to the following operations on the resource space: Add, delete, or modify an axis or coordinate. That is, the change of index should be minimal when changing the structure of the space.

Multi-dimensional indexes such as RTree and R*Tree are not suitable for storing resource space directly (N. Beckmann, et. al. The R* tree: an efficient and robust index method for points and rectangles. *SIGMOD*, 1990, pp322-331), because there may not be meaningful order on the coordinates at the same axis, therefore the Most Minimum Rectangle is meaningless.

The following issues should be considered to effectively store a resource space.

(1) How to store an axis, its coordinates, and the hierarchical relations? Given a coordinate (concept), all direct sub-coordinates, its direct super-coordinates, and all ancestor coordinates should be rapidly retrieved.
(2) How to quickly return all resources indicated by a given coordinate? How to support fast intersection of queries on multiple axes?
(3) How to efficiently support dynamic updating of coordinates in a resource space?
(4) How to index continuous and discrete axes and to support effective resource retrieval?
(5) How to enable the indexing mechanism to support efficient range query?

A way to implement the indexing mechanism is to integrate the following three approaches.

(1) Using RTree to index continuous axes.
(2) Encoding the axis with discrete coordinates by the hierarchical encoding mechanisms. ORDPATH is a hierarchical encoding mechanism based on prefix for labelling XML nodes (P.O'Neil, et. al., ORDPATHs: Insert-Friendly XML Node Labels, *SIGMOD*, 2004, pp.903-908). It is derived from the Dewey Order Encoding, a global hierarchical encoding method (C.D. Batty, An introduction to the Dewey decimal classification. Melbourne, Cheshire, 1966).
(3) Using inverted table to index resources indicated by each coordinate.

Much work has been done on indexing and querying on XML data. The approach to integrating structure indexes and inverted lists was introduced (R.Kaushik, et al., SIGMOD'04, France).

An important issue is the normal form checking of a changing resource space. A resource space may need changing with the change of resources and human cognition. Changing a resource space has the following influence.

(1) Removing an axis can be regarded as the result of the disjoin operation. The normal forms of the new space can be verified by Lemma 3.3.3. This operation actually generalizes the classification

of the space. As the consequence, resources will no longer be classified by the removed axis, and the granularity of classification is enlarged.

(2) Adding an axis may break 3NF since the new axis may not be orthogonal to existing axes. Adding an axis *X* to a space *S* can be regarded as the join of *S* and a bi-dimensional space consisting of *X* and one axis in *S*. The normal forms can be verified by Lemma 3.3.1.

(3) Removing a coordinate from a dimension will not break the 1NF, 2NF, 3NF and 4NF if the resources on the coordinates can be appropriately moved to the rest coordinates or put into an undefined coordinate.

(4) Adding a coordinate to a dimension implies that classification on the dimension needs revision. Adding a coordinate may lead to the increment of null points corresponding to the coordinate. So, this operation may break the 4NF. So, all the normal forms need verifying.

3.13 Application: Faceted Navigation

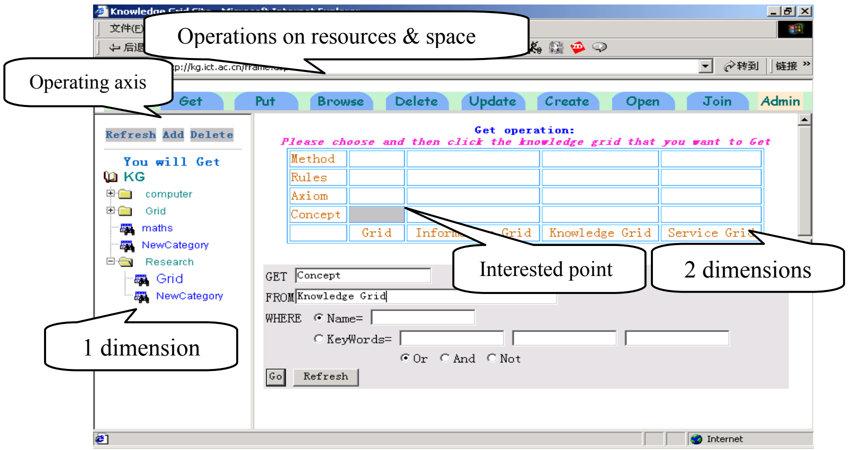
Traditional web browsing displays one web page for once operation: inputting the URL in browser or clicking the interested hyperlink in the current web page. Search engines display a list of hyperlinks according to the ranks of the pages by ranking the web pages sharing the same keywords.

Faceted navigation (also called faceted browsing or faceted search) is an important approach to improving traditional web searching and browsing by refining the search results through a process of multi-step facet selection. It allows users to view the contents of a set of resources at each step from a specific facet.

Although lacking fundamental theory and model on the underlying resource organization, some application systems have implemented faceted navigation through defining faceted metadata and traditional indexing approaches (K.-P. Kee, et. al., Faceted metadata for image search and browsing, CHI03' *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2003; E. Oren, et al., Extending Faceted Navigation for RDF Data, ISWC2006; D.F.Huynh and D.R.Karger, Parallax and companion: Set-based browsing for the data web, WWW09; Dakka, W., et al. Faceted Browsing over Large Databases of Text-Annotated Objects, *ICDE 2007*; Dachselt, R. et al. FacetZoom: a continuous multi-scale widget for navigating hierarchical metadata, CHI08, 2008).

The Resource Space Model naturally supports faceted navigation as the model and theory for resource organization due to its multi-dimensional classification characteristic. Scholar search shows that the work published in 2002 (H. Zhuge, A knowledge grid model and platform for global knowledge sharing. *Expert Systems Applications*, 22(4)(2002)313-320) is the first faceted navigation system. The work published in the first edition of this book (2004, chapter 3) is the first theoretical work on faceted browsing, navigation and search.

Fig.3.13.1 is the interface of the early faceted browsing system developed in 2001. It is based on three dimensions: the left hand column is one dimension, and the middle grid contains two dimensions. Users can accurately locate a set of resources by selecting the interested point. Users can also operate the axes and space by the operation buttons.



Get operation:

Please choose and then click the knowledge grid that you want to Get

Method				
Rules				
Axiom				
Concept				
	Grid	Inform Grid	Knowledge Grid	Service Grid

GET Concept

FROM Knowledge Grid

WHERE ● Name=

○ KeyWords=

● Or ○ And ○ Not

Fig. 3.13.1 The initial faceted browsing system developed in 2001.

Fig. 3.13.2 shows an example of using a three dimensional resource space (*Topic*, *Language*, *Time*) to organize web pages. Users don’t need to remember the URLs of the web pages. Once the interested point is located, all the required web pages are in the point.

The faceted navigation system based on the Resource Space Model consists of the following three main components:

(1) A Resource Space Management System RSMS including the storage mechanism, management mechanism, constraint mechanism, and query language.

(2) A crawler is responsible for searching relevant web pages according to one point or several points in the space and uploading the obtained web pages into the point or points.

(3) A friendly user interface that helps users to conveniently locate the point by selecting the coordinates at each axis and to view the search result.

Users can get all of the required web pages by giving coordinates on the dimensions determining the point in the space. For example, using

(*Language.Chinese*, *Topic.Stock*, *Time.2006-12-13*) can get all the web pages on stock market in Chinese on December 13 in 2006. There are two major advantages of the RSM-based faceted navigation systems: the crawler has clear searching target, the user can get all required web pages by just one time operation, and the user can specialize or generalize the search result.

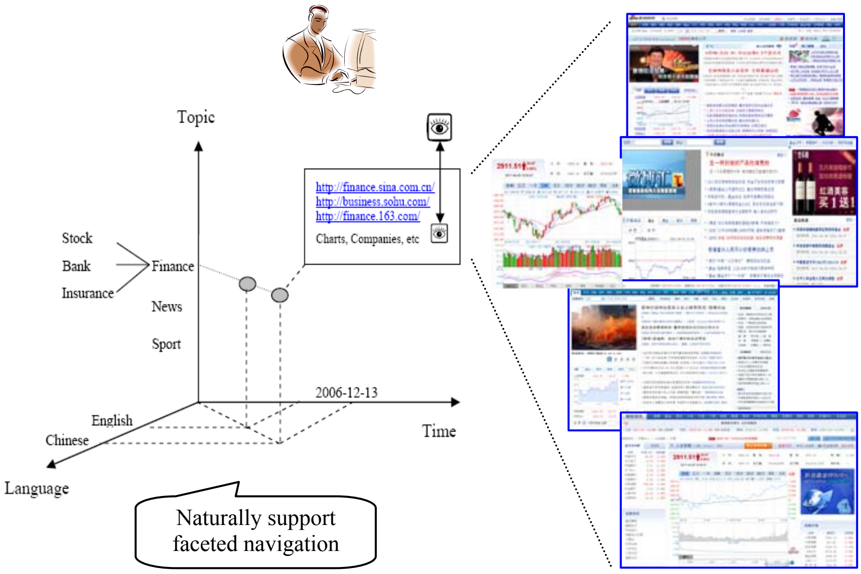


Fig.3.13.2. Underlying resource organization model for faceted navigation of web pages.

Furthermore, decentralized systems can be developed to support decentralized faceted navigation.

Fig. 3.13.3 shows two interface examples. The right hand side is three dimensional. Users can locate a small cube by rotating the cube and making separation from any dimension. The six sides of one basic cube represent six different facets of the same set of resources. The left hand side is suitable for more than three dimensions. Users can select the coordinates in the dimensions in the lower left area by clicking the dimensions and moving the pointers along the selected dimensions.

Faceted browsing, navigation, and search are not only significant in the cyber space but also useful in the other spaces in the future cyber-physical society (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). The study of the Resource Space Model is fundamental for faceted browsing, navigation and search.

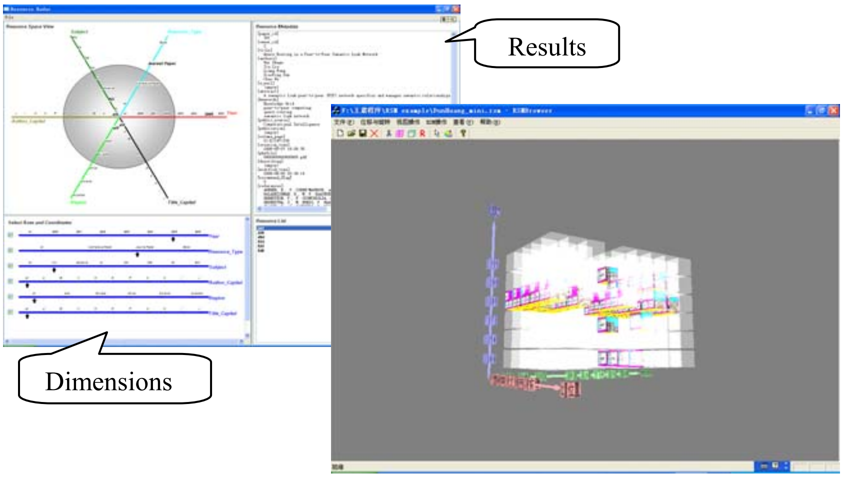


Fig.3.13.3. Interfaces for faceted navigation.

In depth research concerns the issue of how humans establish the dimension.

3.14 Application: Personal Resource Space

3.14.1 *The idea*

An important way to solve the issue of expanding ocean data is to establish personal resource space, which can provide the necessary resources for particular user. The reason is that the number of resources needed by an individual is limited due to limited personal time and energy.

Current operating systems are based on file system, which provides resource management mechanism like the My Computer in MS Windows for users to mange personal files. However, the file system is a one-dimensional directory system, which classifies files top-down into pre-named folders. Once user goes into one folder, he/she can only see the inside files and folders. The classifications (directory) cannot be refined from the other dimensions. Although the interface can be improved, to create an appropriate underlying data model is the key issue.

Fig.3.14.1 shows a multi-dimensional personal resource space. The core space has three dimensions: (*Publication*, *Area*, *Time*), which forms a 3NF resource space. If the user wants to find the references in WWW published in 2009, he/she can quickly locate a set of resources in the point determined by (*Publication*: *Reference*, *Area*: *WWW*, *Time*: *2009*). The user can also refine his search by selecting some sub-coordinates: (*Publication*: *Reference.technical_report*, *Area*: *WWW.search*, *Time*: *2009*). A interface should be able to display the dimensions and enable users to easily select the coordinates.

The user may be also interested in the work of particular researchers. In this case, one more dimension *Researcher* can be added. The *Researcher* dimension can take the following form: (*A*, *B*, … , *Z*), which classifies researchers by the first alphabet of surname. The dimension can also be classified by the academic ranks, for example, (*Turing_Award_Winner*, *ACM_Awards_Winner*, *IEEE_Awards_Winner*, *ACM_Fellow*, *IEEE_Fellow*, *Professor*, *Others*).

If the user is also interested in the work of particular region, one more dimension *Region* can be added. The first level coordinates of the

Region dimension are the names of the countries he/she is interested in. However, adding the *Researcher* dimension will break the 3NF because some researchers have no publications in some areas.

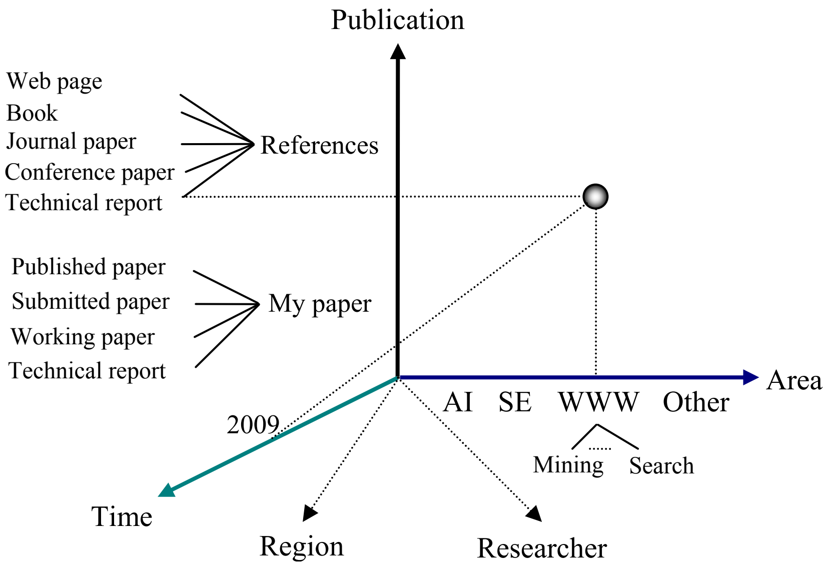


Fig.3.14.1. A researcher's personal resource space.

The following issue is important:

Can the names of researchers be the coordinates?

The answer is yes, because the name of a researcher here represents a set of his/her publications rather than an individual. Similarly, the name of a country represents the set of its publications rather than its entity. It is important for designers to distinguish entity and set in RSM. Even the 2009 at the time dimension, it represents a set of resources. The following are rules:

(1) The names of dimensions and coordinates represent sets.
(2) The name of any resource represents the entity of the resource.

With this personal resource space, the user can put the resources in the right point, and retrieve the interested resources accurately according to the interested dimensions and coordinates.

To create an intelligent personal resource space, the following issue needs to be solved:

Can machines know the meaning of dimensions?

The dimensions and coordinates are named by users. There will be no problem for users to know the relations between a coordinate and the resources it represents. How to enable machines to automatically put resources into and retrieve them from the right points is an important issue. The key of the solution is the representation of resources and the representation of coordinates. If machines know the representations, they can calculate the semantic distance between representations, then put resources into or retrieve resources from the right points.

The candidate representation approaches include the SVM (Support Vector Machine), LSA (Latent Semantic Analysis), ontology or metadata based approaches, and the classification based on Wikipedia and ODP (Open Directory Project).

In specific domain, e.g., in bioinformatics, it is easy for users to establish consensus on some form of simple representations. But, it is hard in open domain applications.

The following are informal approaches to represent dimension and its coordinates:

(1) Define the pattern or pattern tree of the dimension. A dimension can be represented as $X(pattern)[C_1(pattern), \ldots, C_n(pattern)]$, where a *pattern* can be a *pattern tree* as discussed in the Probabilistic Resource Space Model for Managing Resources in Cyber-Physical Society (H.Zhuge et. al., *IEEE Trans. on Service Computing*, doi.ieeecomputersociety.org/10.1109/TSC.2011.12). For example, a set of sequentially appeared words ($w_1, \ldots , w_n$) can be the pattern of coordinate indicating the textual resources. It can be extended to the

following forms: (w_1|w$_1$', … , w_n|w_n') and (w_1→w$_1$', … , w_n→w_n'), where w_1 and w_1' are words, symbol "|" represents "or", and symbol "→" represents implication, that is, one word semantically implies the other. A semantic link network of small set of words can also be a pattern.

(2) Give a set of rules to describe the features of the resources. So, a dimension should be represented as X(*Constraint*)[C_1(*Constraint*), …, C_n(*Constraint*)], where *Constraint* consists of a set of constraints for regulating resources. This is useful when the patterns are difficult to express.

(3) Give a process to select the appropriate resources. A dimension can be represented as X(*Process*)[C_1(*Process*), …, C_n(*Process*)], where a *Process* maps input resources into [yes, no]. If the output is yes, the input resource can be put into the coordinate. This approach can be used to classify complex resources.

The meaning of coordinate depends not only on the dimension but also on the other dimension. For example, the name of a researcher indicates a set of resources that are also regulated by the publication dimension. So, a point can provide more semantics for the crawler than a coordinate.

In many cases, classification is inexact, even for humans. A probabilistic value can be assigned to each resources belonging to a coordinate, e.g., (R∈C, *conf*), where *conf*∈[0, 1] is a confidence degree. It is important to give a higher priority to the resource with higher confidence degree.

3.14.2 *Uploading linked resources*

Resources are not isolated. There are explicit and implicit relations between resources. These relations represent the external semantics of resources. For example, if paper R cites paper R' and R has been put into coordinate C, then R' can be put into C. That is, $(R\longrightarrow R')\wedge R \in C \Rightarrow R' \in C$. The following reference increases the confidence of putting R' into C:

$(R \longrightarrow R') \wedge (R'' \longrightarrow R') \wedge R \in C \wedge R'' \in C \Rightarrow R' \in C.$

The following are two rules to include resources in a network into a coordinate:

Rule1: If all the neighbors of resource *r* have been put into coordinate *C*, *r* should be put into *C* with a certain confidence. The more neighbors, the higher the confidence degree.

A resource has a certain centrality in the semantic link network of resources (H.Zhuge and J.Zhang, Topological Centrality and Its e-Science Applications, *Journal of the American Society for Information Science and Technology*, 61(9)(2010)1824-1841). The resource with higher centrality is more important in the network. The following rule can be derived from Rule1.

Rule2: If all the neighbors of resource *r* and the neighbors of *r*' have been put into coordinate *C*, and the centrality of *r* is higher than *r*', then putting *r* into *C* should have higher confidence than putting *r*' into *C*.

Fig.3.14.2.1 explains the rules of putting the linked resources into coordinate. The resource networks consist of nodes and solid lines. The dotted lines denote that the linked nodes have been put into coordinate *C*. The dark nodes in different networks have different confidences: the confidence of the node in the right hand side network is higher than that of the middle network, which is higher than the confidence of the node in the left hand side network.

The rules are useful in determining the classes of the resources when the semantics of the linked resources have been determined.

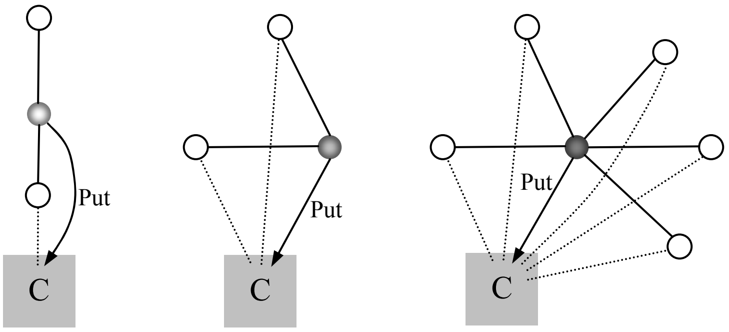


Fig.3.14.2.1. Putting the linked resources into coordinate *C*.

As shown in Fig. 3.14.2.2, if all the neighbors of a resource have been put into coordinate *C*, it should be put into *C* rather than the other coordinates. Evaluation should be made when some of its neighbors or the neighbors of some of its neighbors have been put into another coordinate. One way is to put the resource into the coordinate that contains the most of its neighbors. The other way is to put the resources into all the relevant coordinates with different confidences, which can be calculated by *the number of neighbors belonging to C* / *total number of neighbors*. This is reasonable since recognition of classification is usually incomplete. With the change of the network, the confidence degrees should be changed.

The confidence can also be evaluated by fuzzy and probabilistic approaches. Fuzzy Resource Space Model and Probabilistic Resource Space Model have been proposed to deal with uncertain resources (H. Zhuge, Fuzzy resource space model and platform, *Journal of Systems and Software*, 73(3)(2004)389-396; H.Zhuge and Y.Xing, Probabilistic Resource Space Model for Managing Resources in Cyber-Physical Society, *IEEE Transactions on Service Computing*, doi.ieeecomputersociety.org/10.1109/TSC.2011.12).

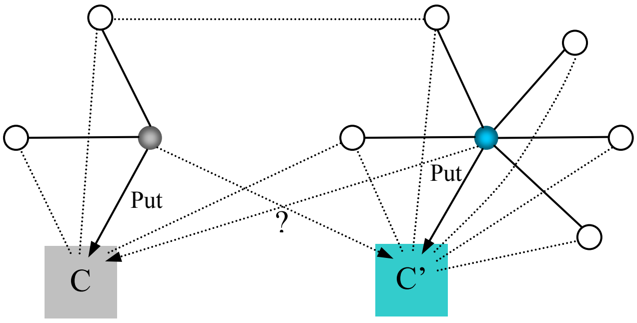


Fig.3.14.2.2. Putting the linked resources into different coordinates.

From social development point of view, human understanding and representation abilities are established in a process of experience and learning. It is natural to transfer the problem of representation into an evolution process. So, we can create a process to enable machines and users to interact with each other. For example, the process can start from the current classification in the user's file directory, and make adjustment according to the increasing resources during using. The difficulty is that adaptation depends on the tracing of user's interest during long-term use of the space.

3.15 The Dimension

Recognition of classification and attributes are two basic ways for humans to recognize the world. For example, resources in the cyber space concern the following attributes:

(1) *Name* — the identifier differentiating one resource from another.
(2) *Author* — the name(s) of the creator(s).

(3) *Abstract* — a general description of the content or function of a resource. It could be a set of keywords, natural language description, formal description, semantic link network, or template.
(4) *Version* — the number that identifies the evolution of the resource.
(5) *Location* — the addresses in the cyber space.
(6) *Privilege* — it has three possible values: a) *public*, any user can access the resource; b) *group*, only group members can access the resource; and, c) *private*, only the author(s) can access the resource.
(7) *Access-approach* — the valid operations on the resource.
(8) *Duration* — the life span of the resource.

Some attributes like abstract cannot be dimension if it is hard to determine the relations between dimensions.

Recognition of classification usually concerns a top-down process while the recognition of attributes usually concerns a bottom-up process. *A dimension is meaningful only when it is related to a space and its resources*. An attribute is meaningful only when it is related to the resource entity and the attribute value. The two ways reflect two levels of a unified cognition process.

A common attribute of a set of resources is a candidate of being a dimension or a coordinate of a dimension when a space is concerned. For example, *gender* can be a dimension of the human resource space although it is used as an attribute of a human individual. When *gender* is a dimension, it represents all the resources in the space. When *gender* is an attribute, an attribute value will be linked to the individual, e.g., *gender=male* for Zhuge. When people use attribute to describe a set of resources, they concern a resource space, and the attribute is actually a dimension if it can cooperate with the other dimensions. An individual has many attributes, while a resource space only needs a small number of dimensions, because *a space should use a minimum number of dimensions to locate resources*.

Usually, the dimensions of a resource space are determined by the creator (designer) who is familiar with the resources and the requirements of users. Humans have established many consensuses on classifications through continuous learning and experience. These consensuses are the basis for the generation of dimensions in the cyber space and in the socio space.

A critical issue is: *Can machines automatically discover the dimensions on a given set of resources*? If they can, resource spaces can be automatically generated. That will be a breakthrough in information management. Unfortunately, it is hard if machines do not know the resources and the users.

However, the following approaches can help machines to automatically discover the dimensions in resources.

(1) Using a small set of resources with known dimensions that reflect the user's viewpoint to train machines.
(2) Making use of online massive classifications such as Wikipedia and ODP (Open Directory Project). One advantage is that the online classifications keep evolving with the development of the society.
(3) Making use of domain classification like ACM Computing Classification System.
(4) Making use of domain ontology mechanisms.
(5) Making use of the current classification and clustering techniques.
(6) Analyzing the external links of resources.
(7) Making use of community discovery techniques.
(8) Tracing user browsing and searching behaviors, including used keywords, selection of hyperlinks, downloaded resources, and the folders selected for the resources.

An important issue in generating the dimensions of a resource space is to *leverage the weight of different dimensions*.

A good resource space should ensure that its resources are evenly distributed so that the search efficiency and efficacy can be guaranteed. Otherwise, the following case may occur: some points are overloaded, while others are empty. Therefore, the depth and width of the coordinate hierarchies should be largely balanced. One over length or over width coordinate tree should be maintained by the separation or merge operations. A resource space is appropriate if the length and width of a coordinate hierarchy are less than nine.

Another important issue is the orthogonal relations between dimensions. *Humans are able to determine the orthogonal relation between dimensions because they continuously classify things and sharing their opinions in lifetime*. Classification may change in an open system. For example, a new dimension *color* emerges to classify swan when black swans appear. Lemma 3.10.2 provides an approach to check the orthogonal relation between dimensions according to their ability of specifying resources.

Chapter 1 has discussed the relation between dimension and space. The generation of dimension is also relevant to the motion of resources and their self-organization. Human behaviors are the original force of the motion and self-organization in the cyber space and socio space. Human behaviors drive the formation of dimensions in the cyber space and in the socio space.

Generally, the Resource Space Model concerns the issue of how to establish an appropriate classification space for the resources of an application domain, to normalize the space, and to enable users to easily operate the space to manage the contents of various resources, no matter what forms they have.

So far, the Resource Space Model has a complete theory, model and method.

Bibliography

Aberer, K., Cudre-Mauroux, P., Hauswirth, M. and Pelt. T.V. (2004). GridVine: Building Internet-Scale Semantic Overlay Networks, *The 3rd International Semantic Web Conference*, LNCS 3298, pp.107-121.

Adamic, L. A. and Huberman, B. A. (2000). Power-law distribution of the World Wide Web, *Science*, 287, No.24, pp.2115.

Aiello, W., Chung, F. and Lu, L. (2000). A random graph model for massive graphs, *Proc. 32nd ACM Symposium on the Theory of Computing*, New York, pp.171-180.

Albert, R., Jeong, H. and Barabási, A. L. (1999). Diameter of the World-Wide Web, *Nature*, 401, pp.130-131.

Albert, R. and Barabási, A. L.(2000). Topology of evolving networks: local events and universality, *Phys. Rev. Lett.*, 85, pp.5234-5237.

Albert, R. and Barabási, A. L. (2002). Statistical mechanics of complex networks, *Rev. Mod. Phys.*, 74, pp.48-94.

Allan, J. (1997). Building hypertext using information retrieval, *Information Processing and Management*, 33, No.2, pp.145-159.

Amaral, L.A.N. et al. (2000). Classes of small-world networks, *Proc. Nat. Acad. Sci.*, 97, No.21, pp.11149-11152.

Androutsellis-Theotokis, S. and Spinellis, D. (2004). A Survey of Peer-to-Peer Content Distribution Technologies, *ACM Computing Surveys*, 36, no. 4, pp. 335-371.

Anthonisse, J. (1971). The Rush in a Graph, *Amsterdam: University of Amsterdam Mathematical Centre*.

Bachman, C. (1972). The Evolution of Storage Structures, Communications of the ACM, 15, No.7, pp.628-634.

Bachman, C. (1973) The Programmer as Navigator. *Communications of the ACM*, 16(11) 653-658.

Bailey. N. T. J. (1975) The Mathematics Theory of Infectious Diseases and Its Applications, second ed. Hafner Press.

Balakrishnan, H. et al. (2003). Looking Up Data in P2P Systems, *Communications of the ACM*, 46, no. 2, pp. 43-48.

Barabási, A. L. and Albert, R. (1999a). Emergence of scaling in random networks, *Science*, 286, pp. 509-512.

Barabási, A. L., Albert, R. and Jeong, H. (1999b). Mean-field theory for scale-free random networks, *Physica A*, 272, pp.173-187.

Barabási, A. L., Jeong, H., Neda, Z., Ravasz, E., Schubert, A. and Vicsek, T. (2002). Evolution of the social network of scientific collaborations, *Physica A*, 311, pp.590-614.

Barabási, A. L. and Bonabeau, E. (2003). Scale-free networks, *Scientific American*, May, pp.50-59.

Barrat, A. and Weigt, M. (2000). On the properties of small-world networks, *Eur. Phys. J. B*, 13, pp.547-560.

Barabasi, A., Jeong, H., Neda, Z., Ravasz, E.. Schubert A., and Vicsek T. (2002). Evolution of the social network of scientific collaborations, *Physica A: Statistical Mechanics and its Applications*, 311, no. 3-4, pp. 590–614.

Batagelj, V. and Mrvar, A. (2000). Some analyses of the erdös collaboration graph, *Social Networks*, 22, pp.173-186.

Batty, C. D. (1966) *An introduction to the Dewey decimal classification. Melbourne*, Cheshire.

Beckmann, N., Kriegel, H. P., Schneider, R. and Seeger, B. (1990) The R* tree: an efficient and robust index method for points and rectangles. In *SIGMOD*, pp322-331.

Beckwith, R., Fellbaum, C., Gross, C. and Miller, G. (1991). WordNet: a lexical database organized on psycholinguistic principles, lexical acquisition: exploiting on-line resources to build a lexicon, U.Zernik, ed., *Lawrence Erlbaum*, pp.211-231.

Berners-Lee, T., Hendler, J. and Lassila, O. (2001). Semantic Web, *Scientific American*, 284, No.5, pp.34-43.

T. Berners-Lee, et al. (2006). A Framework for Web Science, Now Publishers Inc.

Berry, M.W. et al. (1999). Matrices, Vector Spaces, and Information Retrieval, *Society for Industrial and Applied Mathematics Review*, 41, pp.335-362.

Bianconi, G. and Barabási, A. L. (2001a). Bose-Einstein condensation in complex networks, *Phys. Rev. Lett.*, 86, pp.5632-5635.

Bianconi, G. and Barabási, A. L. (2001b). Competition and multiscaling in evolving networks, *Europhys. Lett.*, 54, pp.436-442.

Bocy, R. et al. (1975). Specifying queries as relational expressions, *Communications of the ACM*, 18, No.11, pp.621-628.

Booch, G., Rumbaugh, J. and Jacobson, I. (1999). The unified modeling language: user guide. Reading, Mass. *Addison-Wesley*.

Bornholdt, S. and Ebel, H. (2001). World Wide Web scaling exponent from Simon's 1955 model, *Phys. Rev. E*, 64, 035104(R).

Bonacich, P. (1972). Factoring and weighting approaches to status scores and clique identification, *Journal of Mathematical Sociology*, 2, no. 1, pp. 113–120.

Bradshaw, J. M., Greaves, M. and Holmback, H. (1999). Agents for the masses? *IEEE Intelligent Systems*, 14, No.2, Mar./Apr., pp.53-63.

Broder, A. et al. (2000). Graph structure of the Web, *Proc. of the 9th WWW Conf.*, Amsterdam, May, pp.309-320.

Broekstra, J. et al, (2001). Enabling knowledge representation on the Web by extending RDF schema, *Proc. 10th International WWW Conf.*, Hong Kong, pp.467-478, available at http://www.cs.vu.nl/~frankh/abstracts/www01.html.

Bush, V. (1945). As we may think. *The Atlantic Monthly*, 176, No.1, pp.101-108.

Callaway, D. S. et al. (2001). Are randomly grown graphs really random? *Phys. Rev. E.*, 64, cond-mat/0104546.

Cascia, M. L., Sethi, S. and Sclaroff, S. (1998). Combining textual and visual cues for content-based image retrieval on the worldwide web. *Proc. IEEE Workshop on Content-based Access of Image and Video Libraries*, pp.24-28.

Cass, S. (2004). A fountain of knowledge. *IEEE Spectrum*, 41, No.1, pp.60-67.

Chakrabarti, S., Dom, B., Gibson, D., Kleinberg, J., Raghavan, P. and Rajagopalan, S. (1998). Automatic resource list compilation by analyzing hyperlink structure and associated text, *Proc. 7th International WWW Conf.*, pp.65-74.

Chen, P. (1976) The Entity-Relationship Model: Toward a Unified View of Data, *ACM Transactions on Database Systems*, 1, No.1, pp.9-36.

Chen, Q. et al. (2002). The origin of power laws in Internet topologies revisited, *Proc. of the IEEE INFOCOM Conf.*, pp.608-617.

Clauset, A., Newman, M., and Moore, C. (2004). Finding community structure in very large networks, *Physical Review E*, 70, no. 6, p. 66111.

Codd, E. F. (1970). A relational model of data for large shared data banks, *Communications of the ACM*, 13, No.6, pp.377-387.

Cohen, E., Fiat, A. and Kaplan, H. (2007). Associative search in peer to peer networks: Harnessing latent semantics, *Computer Networks*, 51, no. 8, pp. 1861-1881.

Colizza, V., Flammini, A., Serrano, M., and Vespignani, A. (2006). Detecting rich-club ordering in complex networks, *Arxiv preprint physics/0602134*.

Cortes, C. and Vapnik, V. (1995), Support-Vector Networks. *Machine Learning*, 20(3)(1995) 273-297.

Crespo, A. and Garcia-Molina, H. (2002). Routing Indices for Peer-to-peer Systems, *Proceedings of the 22nd International Conference on Distributed Computing Systems*, pp. 23-32.

Dachselt, R. Frisch, M. Weiland, and M. (2007) FacetZoom: a continuous multi-scale widget for navigating hierarchical metadata, CHI08, 2008, pp.1353-1356.

Dakka, W., Ipeirotis, P.G., and Wood, K.R. (2007) Faceted Browsing over Large Databases of Text-Annotated Objects, ICDE 2007, pp.1489 – 1490.

Davenport, T. H., Jarvenpaa, S. L. and Beer, M. C. (1996). Improving knowledge work process, *Sloan Management Review*, 34, No.4, pp. 53-65.

Davis, G. F. and Greve, H. R. (1997). Corporate élite networks and governance changes in the 1980s, *Am. J. Sociol.*, 103, pp.1-37.

Dean, J. and Henzinger, M. R. (1999). Finding related pages in the World Wide Web, *Proc. 8th International WWW Conf.*, pp.1467-1476.

Decker, S., et al. (2000). The Semantic Web: the roles of XML and RDF, *IEEE Internet Computing*, 4, No. 5, pp.63-74.

Desouza, K. C. (2003). Facilitating tacit knowledge exchange, *Communications of the ACM*, 46, No.6, pp.85-88.

Dieng, R. (2000). Knowledge management and the Internet, *IEEE Intelligent Systems*, 15, No.3, May/June, pp.14-17.

Dorogovtsev, S. N. and Mendes, J. F. F. (2000a). Scaling behavior of developing and decaying networks, *Europhys. Lett.*, 52, pp.33-39.

Dorogovtsev, S. N. and Mendes, J. F. F. (2000b). Evolution of reference networks with aging, *Phys. Rev. E.*, 62, pp.1842-1845.

Dorogovtsev, S. N., Mendes, J. F. F. and Samukhin, A. N. (2000c). Structure of growing networks with preferential linking, *Phys. Rev. Lett.*, 85, pp.4633-4636.

Dorogovtsev, S. N., Mendes, J. F. F. and Samukhin, A. N. (2000d). WWW and Internet models from 1955 till our days and the "popularity is attractive" principle, *Condensed Matter Archive*, cond-mat/0009090.

Dorogovtsev, S. N. and Mendes, J. F. F. (2001). Language as an evolving word Web, *Proc. Royal Soc. London B*, 268, No.2603, cond-mat/0105093.

Drucker, P. F. (1998). Harvard business review on knowledge management, Boston, *Harvard Business School Press*.

Faloutsos, M., Faloutsos, P. and Faloutsos, C. (1999). On power law relationships of the Internet topology, *Comput. Commun. Rev.*, 29, pp.251-262.

Fensel, D., et al. (2001). OIL: an ontology infrastructure for the Semantic Web, *IEEE Intelligent Systems*, 16, No.2, Mar./Apr., pp.38-45.

Ferrer, R., Janssen, C. and Sole, R. V. (2001b). The topology of technology graphs: small-world patterns in electronic circuits, *Phys. Rev.E.*, 64, No.32767.

Ferrer, R. and Sole, R. V. (2001a). The small-world of human language, *Working Papers of Santa Fe Institute*, http://www.santafe.edu/sfi/publications/Abstracts/01-03-004abs.htm.

Franz, T. et al. (2009) TripleRank: Ranking Semantic Web Data by Tensor Decomposition. In *Proceedings of the International Semantic Web Conference*, 2009, pp.213-228.

Freeman, L. (1979), Centrality in social networks: Conceptual clarification, *Social Networks*, 1, no. 3, pp. 215–239.

Freeman, L. (1977). A Set of Measures of Centrality Based on Betweenness, *Sociometry*, 40, no. 1, pp. 35–41.

Fikes, R. and Farquhar, A. (1999). Distributed repositories of highly expressive reusable ontologies, *IEEE Intelligent Systems*, 14, No. 2, Mar/April, pp.73-79.

Fortunato, S., Latora, V., and Marchiori, M. (2004). Method to find community structures based on information centrality, *Physica Review E*, 70, no. 5, p. 56104.

Foster, I. (2000). Internet computing and the emerging Grid, 408, No.6815, *Nature*, http://www.nature.com/nature/webmatters/grid/grid.html.

Foster, I., Kesselman, C., Nick, J. M. and Tuecke, S. (2002). Grid services for distributed system integration, *Computer*, 35, pp.37-46.

Ganesan, P. Bawa, M. and Garcia-Molina, H. (2004) Online Balancing of Range-Partitioned Data with Applications to Peer-to-Peer Systems. *VLDB '04*. pp.444-455.

Gevers, T. and Smeulders, A. W. M. (1999). The PicToSeek WWW image search system, *Proc. IEEE International Conf. on Multimedia Computing and Systems*, pp.264-269.

Girvan, M. and Newman, M. (2002). Community structure in social and biological networks, *Proceedings of the National Academy of Sciences*, 99, no. 12, p. 7821.

Goldberg, A. (1984). SMALLTALK-80: the interactive programming environment, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

Gong, L. (2001). JXTA: a network programming environment, *IEEE Internet Computing*, 5, no.3, pp.88-95.

Gorlitz, O., Sizov, S. and Staab, S. (2008). PINTS: Peer-to-Peer Infrastructure for Tagging Systems, *The 7th International Workshop on Peer-to-Peer Systems*.

Govindan, R. and Tangmunarunkit, H. (2000). Heuristics for Internet map discovery, *Proc. of the IEEE INFOCOM Conf.*, Tel Aviv, Israel, March, pp.1371-1380.

Gray, J. (2003). What next? A dozen information-technology research goals, *Journal of the ACM*, 50, No.1, pp.41-57.

Green, S. J. (1999). Building hypertext links by computing semantic similarity, *IEEE Trans. On Knowledge and Data Engineering*, 11, No.5, pp.713-730.

Grosz, B. J. and Kraus, S. (1996). Collaborative plans for complex group action, *Artificial Intelligence*, 86, pp.269-357.

Gupta, A. and Jain, R. (1997). Visual information retrieval, *Communications of the ACM*, 40, No.5, pp.71-79.

Gyssens, M .et al., (1994). A graph-oriented object database model, *IEEE Transactions on Knowledge and Data Engineering*, 6, No.4, pp.572-586.

Han, J. and Kambr, M. (2000). Data mining: concepts and techniques, *Morgan Kaufmann Publishers*.

Haridasan, M. and Renesse, R. (2008). Gossip-based Distribution Estimation in Peer-to-Peer Networks, *the 7th International Workshop on Peer-to-Peer Systems*.

Harmandas, V., Sanderson, M. and Dunlop, M. D. (1997). Image retrieval by hypertext links. *Proc. 20th Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval*, pp.296-303.

Hammer, M. and McLeod, D. (1978) The Semantic Data Model: A Modeling Mechanism for Database Applications, *SIGMOD*78, pp.26-36.

Havil, J. and Dyson, F. (2003). The Harmonic Series, Chapter.2 in Gamma: Exploring Euler's Constant. *Princeton, NJ: Princeton University Press.*

Heflin, J. and Hendler, J. (2001). A portrait of the Semantic Web in action, *IEEE Intelligent Systems*, 16, No.2, Mar./Apr., pp.54-59.

Hendler, J and McGuinness, D. (2000). The DARPA agent markup language, *IEEE Intelligent Systems*, 15, No.6, Nov./Dec., pp.72-73.

Hendler, J. (2001). Agents and the Semantic Web, *IEEE Intelligent Systems*, 16, No.2, Mar./Apr., pp.30-37.

Henzinger, M. R. (2001). Hyperlink analysis for the Web, *IEEE Internet Computing*, 5, No.1, Jan./Feb., pp.45-50.

Heylighen, F. (1992). A Cognitive-systemic reconstruction of maslow's theory of self-actualization, *Behavioral Science*, 37, pp.39-58.

Huberman, B. A. and Adamic, L. A. (1999). Growth dynamics of the World-Wide Web, *Nature*, 401, 131p.

Huynh D.F. and Karger D.R. (2009). Parallax and companion: Set-based browsing for the data web, WWW09, pp.145-156.

Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., Barabási, A. L. (2000). The large-scale organization of metabolic networks, *Nature*, 407, pp.651-654.

Jeong, H. Neda, Z. and Barabási, A. L. (2003). Measuring preferential attachment for evolving networks, *Europhysics letters*, 61, pp.567-572.

Kanth, K., Agrawal, D. and Singh, A. (1998). Dimensionality reduction for similarity searching in dynamic databases, *Proc. ACM SIGMOD International Conf. on Management of Data*, 75, pp.166-176.

Kaashoek, M. F. and Karger and Koorde, D. R. (2003). A Simple Degree-Optimal Distributed Hash Table. *Proceeding of International Workshop on Peer-to-Peer Systems (IPTPS)*. (2003)98-107.

Kee, K.-P., Swearingen, K., Li, K. and Hearst, M. (2003). Faceted metadata for image search and browsing, CHI03' *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2003, April 5-10, 2003, Florida, USA.

Kermarrec, A.M., Massoulié, L. and Ganesh, A.J. (2003). Probabilistic Reliable Dissemination in Large-scale Systems, *IEEE Transactions on Parallel and Distributed Systems*, 14, no. 3, pp. 248-258.

Klein, M. (2001). XML, RDF, and relatives, *IEEE Internet Computing*, 16, No.2, Mar./Apr., pp.26-28.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46, No.5, pp.604-632.

Kleinberg, J and Lawrence, S. (2001). The structure of the Web, *Science*, 294, No.5548, pp.1849-1850.

Kleinberg, J. (2001) Small-World Phenomena and the Dynamics of Information, *Proceeding of Advances in Neural Information Processing Systems* (NIPS) 14, pp.431-415.

Krapivsky, P. L. Redner, S. and Leyvraz, F. (2000). Connectivity of growing random networks, *Phys. Rev. Lett.*, l.85, pp.4629-4632.

Krapivsky, P. L. and Redner, S. (2001a). Organization of growing random networks, *Phys. Rev. E.*, 63, cond-mat/0011094.

Krapivsky, P. L., Rodgers, G. J. and Redner, S. (2001b). Degree distributions of growing networks, *Phys. Rev. Lett.*, 86, pp.5401-5404.

Krapivsky, P. L. and Redner, S. (2002). A statistical physics perspective on Web growth, *Computer Networks*, 39, pp.261-276.

Kumar, R. et al. (1999). Extracting large-scale knowledge bases from the Web, *Proc. 25th VLDB Conf.*, Edinburgh, Scotland, September, pp.639-650.

Larry, P., Sergey, B., Motwani, R. *et al.* (1998). The PageRank citation ranking: Bringing order to the web, *Online: http://citeseer. nj.Nec com/page98pagerank. html* [*04.06. 2003*].

Latora, V. and Marchiori, M. (2004). A measure of centrality based on the network efficiency, *Arxiv preprint cond-mat/0402050*.

Lawrence, S. and Giles, C. L. (1998). Searching the World Wide Web, *Science*, 280, No.5360, pp.98-100.

Lawrence, S. and Giles, C. L. (1999). Accessibility of information on the Web, *Nature*, 400, No.8, pp.107-109.

Lazer, D. et al. (2009), Computational social science, *Science*, 323, no.5915, pp.721-723.

Lee, S., Yook, S., and Kim, Y. (2009). Centrality measure of complex networks using biased random walks, *The European Physical Journal B*, 68, no. 2, pp. 277–281.

Leland, W. E. et al. (1994). On the self-similar nature of Ethernet traffic, *IEEE/ACM Transactions on Networking*, 2, no.1, pp.1-15.

Lempel, R. and Moran, S. (2000). The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Proc. 9th International WWW Conf.*, pp.387-401.

Lempel, R. and Soffer, A. (2001). PicASHOW: pictorial authority search by hyperlinks on the Web, *Proc. 10th International WWW Conf.*, pp.438-448.

Levene, M., Fenner, T., Loizou, G. and Wheeldon, R. (2002). A stochastic model for the evolution of the Web, *Computer Networks*, 39, pp.277-287.

Leymann, F. and Roller, D. (1997). Workflow-based applications, *IBM Systems Journa*l, 36, No.1, pp.102-122.

Licklider, J.C.R. (1960). Man-Computer Symbiosis, Transactions on Human Factors in Electronics, vol. HFE-1, pp.4–11, March.

Lieberman, H. (2008). A Creative Programming Environment, in HCI Remixed, T.Erickson and D.W.McDonald (Ed).

B. Libet, Mind Time, Harvard University Press, 2005.

Liu, Z. Q. and Satur, R. (1999). Contextual fuzzy cognitive maps for decision support in geographic information systems, *IEEE Transactions on Fuzzy Systems*, 7, No.10, pp.495-502.

Mack, R., Ravin, Y. and Byrd, R. J. (2001). Knowledge portals and the emerging knowledge workplace, *IBM Systems Journal*, 40, No. 4, pp. 925-955.

Maedche, A. and Staab, S. (2001). Ontology learning for the Semantic Web, *IEEE Intelligent Systems*, 16, No.2, Mar./Apr., pp.72-79.

Malkhi, D., Naor, M. and Ratajczak, D. (2002). Viceroy: A Scalable and Dynamic Emulation of the Butterfly. *Proceeding of the 21st ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing* (PODC ’02), pp.183-192.

Manku, G. Bawa, M. and Raghavan, P. Symphony, (2003), Distributed Hashing in a Small World. *Proceeding of the 4th USENIX Symposium on Internet Technologies and Systems*, pp.10-22.

Marchionini, G., Dwiggins, S., Katz, A. and Lin, X. (1990). Information roles of domain and search expertise, *Library and Information Science Research*, 15, no.1, pp.391-407.

Martin, P. and Eklund, P. W. (2000). Knowledge retrieval and the World Wide Web, *IEEE Intelligent Systems*, 15, no.3, May/June, pp. 18-25.

Martinez, N. D. (1991). Artifacts or attributes? Effects of resolution on the Little Rock Lake food web, *Ecological Monographs*, 61, pp.367-392.

Maybury, M. (2001). Collaborative virtual environments for analysis and decision support, *Communication of the ACM*, 14, no.12, pp. 51-54.

McCreight, E. M. (1976) A Space-Economical Suffix Tree Construction Algorithm. *Journal of ACM*, 23, no.2, pp. 262-272.

McHraith, S. A., Son, T. C. and Zeng, H. (2001). Semantic Web services, *IEEE Intelligent Systems*, 16, No.2, Mar./Apr., pp. 46-53.

Miao, Y., et al. (2001). Dynamic cognitive network, *IEEE Transactions on Fuzzy System*, 9, no.5, pp. 760-770.

Milgram, S. (1967) The Small World Problem. *Psychology Today*, 1, no.61.

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002). Network Motifs: Simple Building Blocks of Complex Networks, pp. 824–827.

Mitri, M. (2003). Applying tacit knowledge management techniques for performance assessment, *Computers & Education*, 4, pp.173-189.

Mok, W.Y. (2002). A comparative study of various nested normal forms, *IEEE Trans. on Knowledge and Data Engineering*, 14, no.2, pp. 369-385.

Nasukawa, T. and Nagano, T. (2001). Text analysis and knowledge mining system, *IBM Systems Journa*l, 40, no. 4, pp. 967-984.

Nejdl, W. et al. (2002). Edutella: a P2P Networking Infrastructure Based on RDF, *Proceedings of the 11th International Conference on World Wide Web*, pp. 604-615.

Newman, M. E. J. et al.(2001a). Random graphs with arbitrary degree distribution and their applications”, *Phys. Rev. E*, 64, no.026118, cond-mat/0007235.

Newman, M. E. J. (2001b). The structure of scientific collaboration networks, *Proc. Nat. Acad. Sci. U.S.A.*, 98, pp.404-409.

Newman, M. E. J. (2001c). Scientific collaboration networks: I. Network construction and fundamental results, *Phys. Rev. E*, 64, No.016131.

Newman, M. E. J. (2001d). Scientific collaboration networks: II. Shortest paths, weighted networks, and centrality, *Phys. Rev. E*, 64, No.016132.

Newman, M. E. J. (2001e). Who is the best connected scientist? A study of scientific coauthorship networks, *Phys. Rev. E,* 64, No.016131.

Newman, M.(2004). Analysis of weighted networks, *Physical Review E*, 70, no. 5.

Ng, W.S. et al. (2003). PeerDB: A P2P-Based system for distributed data sharing, *Proc. of International Conf. on Data Engineering*, ICDE2003, Bangalore, India.

Nieminen, J. (1974). On the centrality in a graph, *Scandinavian Journal of Psychology*, 15, no. 1, pp. 332–336, 1974.

Nissen, M. E. (2002). An extended model for knowledge-flow dynamics, *Communications of the Associations for Information Systems*, 8, pp. 251-266.

Nonaka, I. (1994). A dynamic theory of organizational knowledge creation, *Organization Science*, 5, No.1, pp. 14-37.

Nowak, M. A. Komarova, N. L., and Niyogi, P., Computational and Evolutionary Aspects of Language, *Nature*, 417(2002)611-617.

Oida, K. (2002). The birth and death process of hypercycle spirals, in: R.K. Standish, M.A. Bedau, H.A. Abbass, edd., Artificial Life VIII, *MIT Press*.

Oren, E., Delbru R. and Decker, S. (2006). Extending Faceted Navigation for RDF Data, ISWC2006, pp.559-572.

O'Leary, D. E. (1998). Enterprise knowledge management, *Computer*, 31, no.3, pp.54-61.

O'Neil, P. O'Neil, E. Pal, S. Cseri, E. and Schaller, G. (2004), ORDPATHs: Insert-Friendly XML Node Labels, *SIGMOD*, pp. 903-908.

Ozsoyoglu, Z. M. and Yuan, L. Y. (1987). A new normal form for nested relations, *ACM Trans. Database Systems*, 12, no.1, pp.111-136.

Pansiot, J. J. and Grad, D. (1998). On routes and multicast trees in the Internet, *Computer Communications Review*, 28, pp.41-50.

Parise, S. and Henderson, J. C. (2001). Knowledge resource exchange in strategic alliances, *IBM Systems Journal*, 40, no. 4, pp.908-924.

Peltz, C. (2003). Web Services Orchestration and Choreography, *Computer*, 36, no.10, pp. 46-52

Perra, N. and Fortunato, S. (2008). Spectral centrality measures in complex networks, *Physical Review E*, vol. 78, no. 3, pp. 036107.

Pittel, B. On Spreading a Rumor, (1987). *SIAM Journal of Applied Mathematics*, 47, no. 1, pp. 213-223.

Raman, V. and Hellerstein (2001), J. M. Potter's wheel: An interactive data cleaning system, *VLDB2001*, Roma, Italy.

Ratnasamy, S. et al. (2001). A Scalable Content-addressable Network, *ACM SIGCOMM Symposium on Communication, Architecture, and Protocol*, pp.161-172.

Redner, S. (1998). How popular is your paper? An empirical study of the citation distribution, *European Physics Journal B*, 4, pp.131-134.

Renesse, R. V., Minsky, Y. and Hayden, M. (1998). A Gossip-style Failure Detection Service, *Middleware98: IFIP International Conference, Distributed Systems and Platforms and Open Distributed Processing*, pp. 55-70.

Ritter, F. E., Baxter, G. E., Jones, G and Young, R.M. (2000). Supporting cognitive models as users, *ACM Trans. on Computer-Human Interaction*, 7, no.2, pp.141-173.

Robertson, S., Walker, S., and Beaulieu, M. (1998). Okapi at TREC-7: Automatic ad hoc, filtering, VLC and interactive. *Proceedings of the 7th text retrieval conference (TREC-7),* NIST special publication, pages 500–242.

Rowstron, A. and Druschel, P. (2001). Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-Peer Systems, *International Conference on Distributed Systems Platforms*, pp. 329-350.

Sabidussi, G.(1966). The centrality index of a graph, *Psychometrika*, 31, no. 4, pp. 581–603, 1966.

Simon, H. A. (1955). On a class of skew distribution functions, *Biometrika*, 42, pp.425-440.

Simon, H. A. and Van Wormer, T. (1963). Some Monte Carlo estimates of the Yule distribution, *Behavioral Sci.*, 8, pp.203-210.

Srihari, R. K., Zhang, Z. and Rao, A. (2000). Intelligent indexing and semantic retrieval of multimodel documents, *Information Retrieval*, 2, no.2, pp. 245-275.

Stoica, I. et al., (2003). Chord: A scalable Peer-to-Peer lookup protocol for Internet applications, *IEEE/ACM Transactions on Networking*, 11, pp. 17–32.

Sun, X. (2010), OSLN: An Object-Oriented Semantic Link Network language for complex object description and operation. *Future Generation Computer Systems*, 26, no.3, pp.389-399

Tari, Z., Stokes, J. and Spaccapietra, S. (1997). Object normal forms and dependency constraints for object-oriented schemata, *ACM Trans. Database Systems*, 22, no.4, pp.513-569.

Tatarinov, I. et al. (2003). The Piazza Peer Data Management Project, *SIGMOD Record*, 32, no. 3, pp. 47-52.

Tempich, C., Staab, S. and Wranik. A. (2004). REMINDIN': Semantic Query Routing in Peer-to-Peer Networks Based on Social Metaphors, *Proceedings of the 13th international conference on World Wide Web*, pp. 640-649.

Thistlewaite, P. (1997). Automatic construction and management of large open Webs, *Information Processing and Management*, 33, no.2, pp. 145-159.

Thomas, J. C., Kellogg, W. A. and Erickson, T. (2001). The knowledge management puzzle: Human and social factors in knowledge management, *IBM Systems Journal*, 40, no. 4, pp.863-884.

Tudhope, D. and Taylor, C. (1997). Navigation via similarity: automatic linking based on semantic closeness, *Information Processing and Management*, 33, no.2, pp.233-242.

Turing, A.M. (1950). Computing machinery and intelligence, *Mind*, vol.59, no.236, pp.433-460.

Ullman, J. D. (1988). Principles of database and knowledge-base systems, *Computer Science Press, Inc.*, Valverde, S., Cancho, R. F. and Sole, R. V. (2002). Scale-free networks from optimal design, *Europhysics Letters*, 60, pp.512-517.

W. Vogels, R. V. Renesse, and K. Birman. (2003) The Power of Epidemics: Robust Communication for Large-Scale Distributed Systems. *ACM SIGCOMM Computer Communication Review*. 33, no.1, pp.131-135.

Warshall, S.(1962). A theorem on boolean matrices, *J. ACM*, 9, no. 1, pp. 11–12.

Wasserman, S. and Faust, K. (1994). *Social Network Analysis: Methods and Applications*. Cambridge University Press.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of "Small-world" networks, *Nature*, 393, pp.440-442.

Watts, D. J. (1999). Networks, dynamics, and the small-world phenomenon, *Am. J. Sociol.*, 105, pp.493-592.

Wilks, Y. Is There Progress on Talking Sensibly to Machines? *Science*, 5852(318)927-928.

Yang, B. and Garcia-Molina, H. (2002). Efficient Search in Peer-to-peer Networks, *International Conference on Distributed Computing Systems*, pp.5-14.

Yook, S. H. et al. (2002). Modeling the Internet's large-scale topology, *Proc. the National Academy of Sciences*, pp. 13382-13386, cond-mat/0107417.

Zeng Q., Jiang X. and Zhuge, H. (2012). Adding logical operators to tree pattern queries on graph-structured data, *VLDB* 2012.

Zhao, B. Y., Kubiatowicz, J.D. and Joseph, A.D. (2001). Tapestry: An Infrastructure for Fault-resilient Wide-area Location and Routing, Technical Report UCB/CSD-01-1141, U.C.Berkeley.

Zhu, Y. W., Wang, H. and Hu, Y. M. (2003). Integrating Semantic-Based Access Mechanisms with Peer-to-Peer File System, *Proceedings of the 3rd International Conference on Peer-to-Peer Computing*, pp. 118-125.

Zhuge, H. (1998). Inheritance rules for flexible model retrieval, *Decision Support Systems*, 22, no.4, pp. 383-394.

Zhuge, H. (2000a). Conflict decision training through multi-space co-operation, *Decision Support Systems*, 29, no. 5, pp.111-123.

Zhuge, H., Cheung, T. Y. and Pung, H. K. (2000b). A timed workflow process model, *Journal of Systems and Software*, 55, no.3, pp.231-243.

Zhuge, H. (2002a). A Knowledge Grid model and platform for global knowledge sharing, *Expert Systems with Applications*, 22, no.4, pp.313-320.

Zhuge, H. (2002b). A knowledge flow model for peer-to-peer team knowledge sharing and management, *Expert Systems with Applications*, 23, no.1, pp.23-30.

Zhuge, H. (2002c). Clustering soft-device in the Semantic Grid, *IEEE Computing in Science and Engineering*, 4, no.6, Nov./Dec., pp.60-62.

Zhuge, H. (2003b). China e-science Knowledge Grid Environment, *IEEE Intelligent Systems*, 19, No.1, Sep./Oct., pp. 13-17.

Zhuge, H. (2003c). Workflow-based cognitive flow management for distributed team cooperation, *Information and Management*, 40, no.5, pp.419-429.

Zhuge, H. (2003d). Active e-Document Framework ADF: model and tool, *Information and Management*, 41, no.1, pp. 87-97.

Zhuge, H. and Shi, X. (2003e). Fighting epistemology in knowledge and information age, *Computer*, 36, no.10, pp.114-116.

Zhuge, H. (2004a). Resource Space Model, its design method and applications, *Journal of Systems and Software*, 72, no.1, pp. 71-81.

Zhuge, H. (2004b). Semantics, resources and Grid, *Future Generation Computer Systems*, Special issue on *Semantic Grid and Knowledge Grid: the Next-generation Web*, editorial, 20, no.1, pp.1-5.

Zhuge, H. (2004c). Fuzzy resource space model and platform, *Journal of Systems and Software*, 73, no.3, pp.389-396.

Zhuge, H. (2004d). Resource Space Grid: model, method and platform, *Concurrency and Computation: Practice and Experience*, 16, no.13.

Zhuge, H. (2004e). Retrieve images by understanding semantic links and clustering image fragments, *Journal of Systems and Software*, 73, no.3, pp.455-466.

Zhuge, H. (2004f). Toward the eco-grid: a harmoniously evolved interconnection environment, *Communications of the ACM*, 47, no.9, pp.78-83.

Zhuge, H. (2004g). Semantics, Resource and Grid, *Future Generation Computer Systems*, 20, no.1, pp.1-5.

Zhuge, H. (2005a). The Future Interconnection Environment, *IEEE Computer*, 38, no.4, pp.27-33.

Zhuge, H. (2005b). Exploring an Epidemic in an E-Science Environment, *Communications of the ACM*, 48, no.9, pp.109-114.

Zhuge, H. et al. (2005c), Query Routing in a Peer-to-Peer Semantic Link Network *Computational Intelligence*, 21, no. 2, pp. 197–216.

Zhuge, H. (2005d). Semantic Grid: Scientific Issues, Infrastructure, and Methodology. *Communications of the ACM*, 48, no.4, pp.117-119.

Zhuge, H., Sun, X., Liu, J., Yao, E. and Chen, X., (2005). A Scalable P2P Platform for the Knowledge Grid, *IEEE Transactions on Knowledge and Data Engineering*, 17, no.12, pp.1721-1736.

Zhuge, H. (2006). Discovery of Knowledge Flow in Science, *Communications of the ACM*, 49, no. 5, pp. 101-107.

Zhuge, H. (2007). Autonomous Semantic Link Networking Model for the Knowledge Grid, *Concurrency and Computation: Practice and Experience*, no.19, pp.1065–1085.

Zhuge H. and Li, X. (2007), Peer-to-Peer in Metric Space and Semantic Space, *IEEE Transactions on Knowledge and Data Engineering*, 19, no. 6, pp. 759–771.

Zhuge H. and Feng, L. (2008), Distributed Suffix Tree Overlay for Peer-to-Peer Search, *IEEE Transactions on Knowledge and Data Engineering*, 20, no.2, pp.276–285.

Zhuge, H. (2008). The Knowledge Grid Environment, *IEEE Intelligent Systems*, 23, no. 6, pp.63-71.

Zhuge, H. and Sun, X. (2008). A Virtual Ring Method for Building Small-World Structured P2P Overlays, *IEEE Transactions on Knowledge and Data Engineering*, 20, no.12, pp.1712-1725.

Zhuge, H. Xing, Y.and Shi, P. (2008). Resource Space Model, OWL and Database: Mapping and Integration, *ACM Transactions on Internet Technology*, 8, no.4.

Zhuge, H., Chen, X., Sun, X. and Yao, E. (2008), HRing: A Structured P2P Overlay Based on Harmonic Series, *IEEE Transactions on Parallel and Distributed Systems*, 19, no.2, pp.145-158.

Zhuge, H. and Feng, L. (2008). Distributed Suffix Tree Overlay for Peer-to-Peer Search, *IEEE Transactions on Knowledge and Data Engineering*, 20, no.2, pp.276-285.

Zhuge, H. (2009), Communities and Emerging Semantics in Semantic Link Network: Discovery and Learning, *IEEE Transactions on Knowledge and Data Engineering*, 21, no.6, pp.785-799.

Zhuge, H. and Sun, Y. (2010) The schema theory for semantic link network. *Future Generation Computer Systems*, 26, no.3, pp. 408-420.

Zhuge, H. (2010) Interactive Semantics, *Artificial Intelligence*, 174, no.2, pp.190-204.

Zhuge, H. (2011) Semantic linking through spaces for cyber-physical-socio intelligence: A methodology, *Artificial Intelligence*, 175, pp.988-1019.