A Data Modeling Example of File Permission Management Using the Cellular Data System

TOSHIO KODAMA[1], TOSIYASU L. KUNII[2], YOICHI SEKI[3]
[1] CDS Business Dept., Advanced Computer Systems, Inc. and Maeda Corporation, Tokyo, JAPAN
kodama@lab.acs-jp.com, https://www.cellulardatasystem.com/e/index.html
[2] Morpho, Inc., The University of Tokyo, Tokyo, JAPAN
kunii@ieee.org, kunii@acm.org, http://www.kunii.net/
[3]Software Consultant, Tokyo, JAPAN
gamataki61@mail.hinocatv.ne.jp

Abstract: In the era of cloud computing, data is processed within "the cloud", and data and its dependencies between systems or functions progress and change constantly within "the cloud", as user requirements change. Such information worlds are called cyberworlds. In designing cyberworlds, the Incrementally Modular Abstraction Hierarchy (IMAH) gives a most appropriate mathematical background to model dynamically changing cyberworlds by descending from the abstract level to the specific one, while preserving invariants. An attaching function is defined on the adjunction space level to model attachment of spaces by an equivalence relation in IMAH. In this paper, we have improved the attaching function to the Cellular Data System (CDS) that we developed based on IMAH. The function is quite effective in business application development when a system user recognizes an equivalence relation in business objects. We have also shown an example of the use of CDS for file permission information management under an unexpected situation, such as when an organizational structure or its staff assignments change, using CDS. In the example, we design and take advantage of spaces on three of the seven levels of IMAH, in order of abstractness: the set theoretical level, the topological space level and the adjunction space level.

Key-Words: incrementally modular abstraction hierarchy, formula expression, topological space, file permission management, attaching function

1 Introduction

Cyberworlds are more complicated and fluid than any other previous worlds in human history, and are constantly evolving and expanding. One of the features of cyberworlds is that data and its dependencies are constantly changing within them. For example, millions of users communicate with each other on the Web using mobile devices, which are considered one of the main elements of cyberworlds. At the same time, user requirements for cyberworlds also change and become more complicated as cyberworlds change. If a user analyzes data in business applications correctly under a dynamically changing situation using the existing technology, the schema designs of databases and application programs have to be modified whenever schemas or user requirements for output change. That leads to combinatorial explosion. To solve the problem, we need a more powerful mathematical foundation than what current computer science enjoys. As a possible candidate, we have introduced the Incrementally Modular Abstraction Hierarchy (IMAH), built by one of the authors (T. L. Kunii). IMAH seems to be the most suitable for reflecting cyberworlds, because it can model the architecture and the changes in cyberworlds and real worlds from a general level to a specific one, preserving invariants while preventing combinatorial explosion [1]. In our research, one of the authors (Y. Seki) has proposed an algebraic system called Formula Expression as one of finite automaton, and another (T. Kodama) has designed how to express the spaces and the maps on each level of IMAH and actually implemented IMAH as a data processing system using Formula Expression [6]. We call the system the Cellular Data System (CDS). CDS has already been applied to the development of several business application systems as a flexible system development tool. In this paper, we have applied an attaching function to file permission management. The attaching map was defined in an adjunction level in IMAH [1] and the attaching function is based on the map; terms as topological spaces are attached by

common factors found through the attaching function. If the function is used with the condition formula search that is the main data search function of CDS (2.3), it becomes a very effective means of analyzing data in cyberworlds without losing consistency in the entire system, since a user can get the data desired without changing application programs. In addition, we put emphasis on practical use by taking up an example of the development of a file permission information management system. First, we explain IMAH and Formula Expression briefly (Section 2). Second, we design the attaching function and implement it (Section 3). Next, we demonstrate the effectiveness of CDS by developing a file permission information management system. More flexible file permission management is shown to be possible (Section 4). Related works are mentioned (Section 5), and, finally, we conclude (Section 6).

2 The Cellular Data System

2.1 Incrementally Modular Abstraction Hierarchy

The following list constitutes the Incrementally Modular Abstraction Hierarchy to be used for defining the architecture of cyberworlds and their modeling:

1. the homotopy (including fiber bundles) level
2. the set theoretical level
3. the topological space level
4. the adjunction space level
5. the cellular space level
6. the presentation (including geometry) level
7. the view (also called projection) level

In modeling cyberworlds in cyberspaces, we define general properties of cyberworlds at the higher level and add more specific properties step by step while climbing down the incrementally modular abstraction hierarchy. The properties defined at the homotopy level are invariants of continuous changes of functions. The properties that do not change by continuous modifications in time and space are expressed at this level. At the set theoretical level, the elements of a cyberspace are defined, and a collection of elements constitutes a set with logical operations. When we define a function in a cyberspace, we need domains that guarantee continuity, such that neighbors are mapped to a nearby place. Therefore, a topology is introduced into a cyberspace through the concept of neighborhood. Cyberworlds are dynamic. Sometimes cyberspaces are attached together, an exclusive union of two cyberspaces where attached areas of two cyberspaces are equivalent. It may happen that an attached space is obtained. These attached spaces can be regarded as a set of equivalent spaces called a quotient space that is another invariant. At the cellular structured level, an inductive dimension is introduced into each cyberspace. At the presentation level, each space is represented in a form which may be imagined before designing cyberworlds. At the view level, the cyberworlds are projected onto view screens.

2.2 The definition of Formula Expression

Formula Expression in the alphabet is the result of finite times application of the following (1)-(7).

(1) a ($a \in \Sigma$) is Formula Expression
(2) unit element ε is Formula Expression
(3) zero element φ is Formula Expression
(4) when r and s are Formula Expression, addition of r+s is also Formula Expression
(5) when r and s are Formula Expression, multiplication of r×s is also Formula Expression
(6) when r is Formula Expression, (r) is also Formula Expression
(7) when r is Formula Expression, {r} is also Formula Expression

Strength of combination is the order of (4) and (5). If there is no confusion, ×, (), {} can be abbreviated. + means disjoint union and is expressed ⊔ as specifically and × is also expressed as Π. In short, you can say " a formula consists of an addition of terms, a term consists of a multiplication of factors, and if the () or {} bracket is added to a formula, it becomes recursively the factor". In Formula Expression, five maps (the expansion map, the bind map, the division map, the attachment map, the homotopy preservation map) are defined [9].

2.3 A Condition Formula Search

A function for specifying conditions defining a condition formula by Formula Expression is supported in CDS. This is one of the main functions. A formula created from these is called a condition formula. "!" is a special factor which means negation. Recursivity by () in Formula Expression is supported so that the recursive search condition of a user is expressed by a condition formula. A condition formula processing map *f* is a map that gets a disjoint union of terms which satisfies a condition formula from a formula. When condition formula processing is considered, the concept of a remainder of spaces is

inevitable. A remainder acquisition map *g* is a map that has a term that doesn't include a specified factor. Fig 2.3 shows each image by the condition formula processing map *f*.

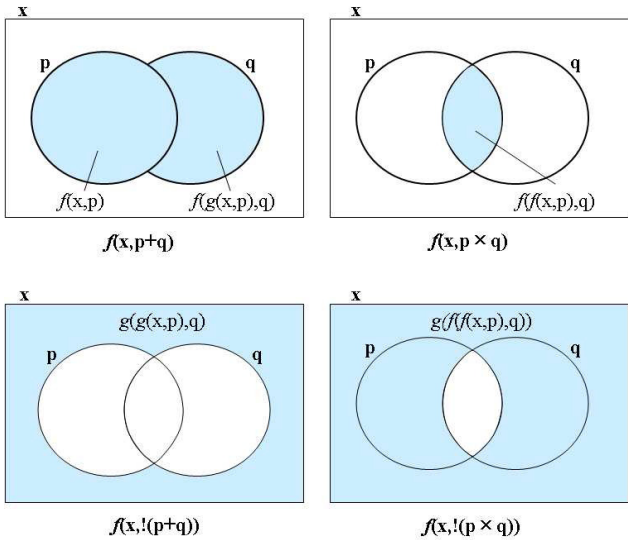

Figure 1 Images by the condition formula processing map f

3 The Attaching Function

3.1 Definition

An attaching map on the adjunction space level is a continuous and surjective mapping [1]. Given two disjoint topological spaces X and Y,

$$Y \sqcup_f X = Y \sqcup X / \sim$$

is an attaching space obtained by *attaching* X to Y by an *attaching map* f (or by identifying points $x \in X0 \mid X0 \subseteq X$ with their images $f(x) \in Y$, namely by a surjective map f)

$$f: X_0 \rightarrow Y.$$

The attaching function of CDS based on the attaching map is the function that attaches several terms in a formula by the common factor(s). Assume the map to be *g*, the entire set of formulas to be A, the entire set of terms to be B and the entire set of factors to be C, *g*: A → A. Arbitrary terms r, s, t, u, p, q (∈C) follow these rules:

g: r×p×s+t×p×u → {r+s}p{t+u}

g: r×p×s+t×q×u → φ (when there is no common factor)

In short, terms in a formula are attached by the common factor(s) between them through the attaching function, so that the formula is separated into attaching spaces and others. A simple example is shown below.

g:color×red(fruit×apple+rainbow)+human×blood×red+blue(sea+sky)+green(tree×leaf+seaweed)
→{color+human×blood}red{(fruit×apple+rainbow)+ε}

In this example, the common factor is "red".

3.3 Implementation

This application was developed using Java Servlet and Tomcat 5.0 as a Web server. The specifications of the server were:

OS: Red Hat Enterprise Linux 5
CPU: Intel Core2 Duo (1.8GHz)
RAM: 4GB
Web server: Apache 2.2.2
AP server: Tomcat 5.5.4
JAVA: JDK1.5.015
RDB: mysql5.1
HD: 240GB
The specifications of the client machine were:
OS: Windows XP
CPU: Intel Core2 Duo (3.00GHz)
RAM: 4GB

A quotient acquisition map is the main function of an attaching map. In this algorithm, the absolute position of the specified factor by the function of the language and the term including the factor are acquired first. Next, the nearest brackets of the term are acquired and, because the term becomes a factor, a recursive operation is done. Details are abbreviated due to the restriction on the number of pages.

4 Data Modeling of File Permission Management

4.1 Outline

We have developed a business application system for managing file permission information in an organization using CDS. In this system, file permission information is managed under the assumption that its organizational structure or relations among staff and their assignments will change. In this section, we simplify all data without losing generality. Firstly, we design sets of staff and files of the company. Secondly, we create a topological space for organizational information and file permissions, adding an organizational structure and file permission information to the set of staff and to the set of files, respectively. Thirdly, we use the maps such as the attaching map, when we output the

data according to user requirements. This system is currently being used by companies in Japan to manage file permission information.

4.2 The space design

We design formulas for two sets as follows:

1. A formula for a set of staff
   staff($\Sigma staffi_i$)
2. A formula for a set of files
   file($\Sigma filename_i$)

Next, we design a formula for the topological space based on the above sets as follows:

1. A formula for organization data for the company as a topological space
   OrgInfo($\Sigma month_i(\Sigma node_{i,1}(\Sigma node_{i,2}(\Sigma\ldots(\Sigma node_{i,j}(\Sigma staffi_{i,k})))))$
   node: a factor which expresses the node of an organizational structure
   month: a factor which expresses month data
2. A formula for file permission data for files as a topological space
   PermisInfo($\Sigma node_{i,1}(\Sigma node_{i,2}(\Sigma\ldots(\Sigma node_{i,j}(\Sigma filename_{i,j})))))$
   filename: a factor which expresses file name data

4.3 Data Input/Output

Here, data on employees of a company are to be managed. You assume that the company in September of a certain year has HQ. HQ has Dept.1 and Dept.2; Dept.1 has on its staff Mr.A and Mr.B, while Dept.2 has on its staff Mr.C and Mr.D. Then the organizational structure changes in October of the year: Dept.1 and Dept.2 are unified as Dept.3, which staffed by Mr.A, Mr.B and Mr.C. project1, which has Mr.C (who serves Dept.3 concurrently) and Mr.D, is created, as you see in Fig.2. If a user wants to input data on the organizational structure in September and October of the year, he/she creates formula 4.3-1 according to the space design.

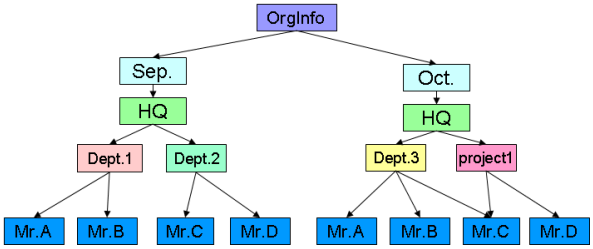

Fig 2 Data structure of an organization and its staff in September/October

formula 4.3-1:
OrgInfo(Sep.×(HQ(Dept.1(Mr.A+Mr.B)+Dept.2(Mr.C+Mr.D))+Oct.×HQ(Dept.3(Mr.A+Mr.B+Mr.C)+project1(Mr.C+Mr.D)))

Next, data on files and their permissions are to be managed. You assume that there are five files: file1 can be dealt with by Dept.1, file2 can be dealt with by Dept.1 and Dept.2, file3 can be dealt with by Dept.2 and Dept.3, file4 can be dealt with by Dept.3 and file5 can be dealt with by project1, as expressed in Fig 3. If a use wants to input data on the file permissions, he/she creates formula 4.3-2 according to the space design and adds it to the previous formula.

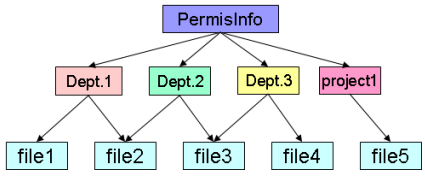

Fig 3 Data structure of permission of each files

formula 4.3-2:
(formula 4.3-1)+PermisInfo(Dept.1(file1+file2)+Dept.2(file2+file3)+Dept.3(file3+file4)+project1(file5))

If a user wants to answer the question "Which files can *Mr. A* deal with ?", he/she creates the condition formula "*Mr.C+PermisInfo*" and gets the image of formula 4.3-2 through the condition formula processing map f (2.3). He/She then attaches the result by the attaching map g (3.1).

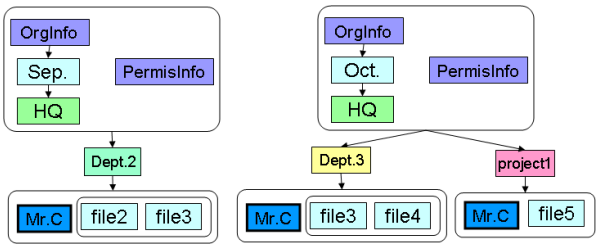

Fig 4 The attaching space by Mr.C and the permission space

g(f (formula 4.3-2, "Mr.C+PermisInfo"))
={OrgInfo×Sep.×HQ+PermisInfo}Dept.2{Mr.C+(file2+file3)}+{OrgInfo×Oct.×HQ+PermisInfo}(Dept.3{Mr.C+(file3+file4)}+project1{Mr.C+file5})

From the above, a user can know that Mr.C can deal with file2 and file3 in September, and that Mr.C can deal with file3, file4 and file5 in October.

Next, if a user wants to answer the question "Who can deal with *file2*?", in the same way he/she creates the condition formula "*OrgInfo+file2*" and gets the image of formula 4.3-2 through the maps *f* and *g*.

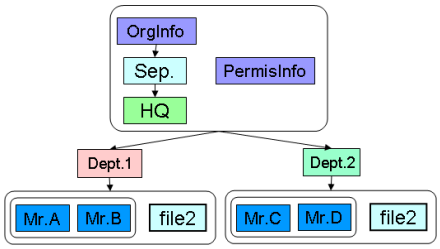

Fig 5 The attaching space by the organization space and file2

g(f (formula 4.3-2, "OrgInfo+file2"))
={OrgInfo×Sep.×HQ+PermisInfo}(Dept.1{(Mr.A+Mr.B)+file2}+Dept.2{(Mr.C+Mr.D)+file2})

From the above, a user can know that file2 can be dealt with by Mr.A, Mr.B, Mr.C and Mr.D in September.

Next, if a user wants to answer the question "Can *Mr. A* deal with *file1*?", in the same way he/she creates the condition formula "*Mr.A+file1*" and gets the image of formula 4.3-2 through the maps *f* and *g*.

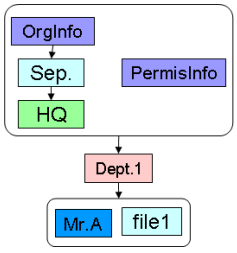

Fig 6 The attaching space by Mr.A and file1

g(f (formula 4.3-2, "Mr.A+file1"))
={OrgInfo×Sep.×HQ+PermisInfo}Dept.1{Mr.A+file1}

From the above, he/she knows that Mr.A can deal with file1 only in September.

4.4 Considerations

In existing development of file permission information management systems, when there are unexpected situations in personnel affairs such as staff changing posts, being transferred or participating in multiple temporary projects; or when the organizational structure changes, such as with the merging or dividing of departments, it is generally difficult to cope with the changes because the database design or application programs need to be modified, which can be costly. On the other hand, if CDS is employed in system development, the design of data structure as formulas is more adaptable to changes in data structure (such as those relating to organizational structure or changes in relations among staff and their assignments, file permissions and managerial positions) if you design space on the set theoretical level and on the topological space level in order of abstractness, and the formula for a state at a certain point in time can be expressed as the disjoint union of formulas up to that state. In the above examples, the formulas for the situation in October consist of the formulas for the situation in September and the formulas for the changes since that time. Therefore, only a few programs need to be modified, and a user can get the data he/she wants flexibly by using the maps of CDS. In other words, the business objects and their relations, and business logic are described directly and simply by CDS and the data that a user wants are outputted in parts from the inputted data through the maps of CDS, such as the condition formula processing map or the attaching map.

5 Related works

The distinctive features of our research are the application of the concept of topological processing, which deals with a subset as an element, and that the cellular space extends the topological space, as seen in Section 2. The conceptual model in [2] is based on an ER model and is the model where tree structure is applied. The approach in [3] aims at grouping data of a graph structure where each node has attributes. The ER model, graph structure and tree structure are expressed as special cases of topological space, and a node with attributes is expressed as one case of the cellular space. These models are included in the function of CDS. Many works dealing with XML schema have been done. The approach in [4] aims at introducing simple formalism into XML schema definition for its complexity. An equivalence relation of elements is supported in CDS, so that complexity and redundancy in schema definition are avoided if CDS is employed, and a homotopy preservation function is introduced into CDS in the model for preserving information. As a result, the problems

described in [4] do not need to be considered in CDS. Some works of inductive data processing have been done recently. CDS can also be considered as one of those inductive systems. The goal of research on the inductive database system of [7] is to develop a methodology for integrating a wide range of knowledge generation operators with a relational database and a knowledge base. The main achievement in [8] is a new inductive query language extending SQL, with the goal of supporting the whole knowledge discovery process, from pre-processing via data mining to post-processing. If you use the methods in [7], [8], the attributes according to users' interests have to be designed in advance. Therefore it is difficult to cope with changes in users' interests. If you use CDS, a formula for a topological space without an attribute as a dimension in database designing can be created so that the attributes of objects don't need to be designed in advance.

6 Conclusions

In this paper, we have applied the attaching function of CDS to the development of a file permission information management system and verified its effectiveness. Using the function with the condition formula search, a user can get the data he/she wants flexibly from formulas as data storage according to user requirements. The point we should emphasize is that the quality of the system using CDS is closely related to how the formulas for space are designed according to IMAH [1]. The design of formulas is fully various, because Formula Expression is very simple in describing business objects and their relations. Therefore, the creativity of a system developer who designs formulas becomes more important when he/she uses CDS.

References:

[1] T. L. Kunii and H. S. Kunii, "A Cellular Model for Information Systems on the Web - Integrating Local and Global Information", In Proceedings of DANTE'99, Kyoto, Japan, IEEE Computer Society Press, pp.19-24, 1999.

[2] Anand S. Kamble, "A conceptual model for multidimensional data", In Proceedings of APCC'08, Tokyo, Japan, Australian Computer Society, Inc., pp.29-38, 2008.

[3] Alexandr Savinov, "Grouping and Aggregation in the Concept-Oriented Data Model", In Proceedings of SAC'06, Dijon, France, ACM press, pp.482-486, 2006.

[4] Wim Martens, Frank Neven, Thomas Schwentick, Geert Jan Bex, "Expressiveness and complexity of XML Schema", ACM Transactions on Database System, pp.770-813, 2006.

[5] Denilson Barbosa, Juliana Freire, Alberto O. Mendelzon, "Designing Information-Preserving Mapping Schemes for XML", In Proceedings of VLDB'04, Trondheim, Norway, VLDB Endowment, pp.109-120, 2004.

[6] Toshio Kodama, Tosiyasu L. Kunii, Yoichi Seki, "A New Method for Developing Business Applications: The Cellular Data System", In Proceedings of CW'06, Lausanne, Switzerland, IEEE Computer Society Press, pp.64-74, 2006.

[7] Kenneth A. Kaufman1, Ryszard S. Michalsk, Jarroslaw Pietrzykowski, and Janusz Wojtusiak, "An Integrated Multi-task Inductive Database VINLEN: Initial Implementation", Knowledge Discovery in Inductive Database, LNCS 4747, pp.116-133, Springer-Verlag Berlin Heidelberg, 2007.

[8] S. Kramer,V. Aufschild, A. Hapfelmeier, A. Jarasch, K. Kessler, S. Reckow, J. Wicker and L. Richter "Inductive Databases in the Relational Model: The Data as the Bridge", Knowledge Discovery in Inductive Database, LNCS 3933, pp.124-138, Springer-Verlag Berlin Heidelberg, 2006.