Staff Assignment Information Management Using the Cellular Data System

Toshio Kodama[1], Tosiyasu L. Kunii[2], Yoichi Seki[3]
[1]CDS Business Department
Advanced Computer Systems Inc. and Maeda Corporation
kodama@lab.acs-jp.com

[2]Morpho, Inc., The University of Tokyo
kunii@ieee.org, kunii@acm.org

[3]Software Consultant
gamataki61@mail.hinocatv.ne.jp

ABSTRACT: *In the era of cloud computing, cyberworlds evolve constantly and dynamically where data and data dependencies constantly change. In system development, correspondence to frequent changes in user requirement has become more important. But existing data models are not able to model such dynamically changing information worlds precisely. We consider the Incrementally Modular Abstraction Hierarchy (IMAH) to offer the most appropriate mathematical background to model such information worlds and real worlds, as well, by descending from the abstract level to the specific, while preserving invariants. To validate the mechanism, we designed spaces and maps on each level of IMAH by means of Formula Expression, which is a simple finite automaton, and implemented them. We call the data processing system the Cellular Data System (CDS). In this paper, we have applied CDS to the core logic development of a business application that can be adaptable to unexpected situations: a staff assignment information management system. We then verified that use of CDS simplifies business application development.*

Keywords: Cyberworlds, Incrementally modular abstraction hierarchy, Formula expression, Topological space, Staff assignment information management

Received: 2 March 2010, Revised 4 April 2010, Accepted 8 April 2010

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 a finite automaton called Formula Expression, 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). The condition formula processing map defined on the set theoretical level was developed as the main data search function of CDS (2.3.). When CDS is used to construct cyberworlds, the map is a very effective means of outputting data without losing consistency in the entire system, since a user can get the data desired without changing application programs. 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 CDS to the core logic development of staff assignment information management system of a company.

In this paper, first, we explain IMAH and Formula Expression and its implementation (Section 2). Next, we demonstrate the effectiveness of CDS by developing core logic of a staff assignment information management system (Section 3). Related works are mentioned (Section 5), and, finally, we conclude (Section 5).

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 (Figure1)
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 each other, 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, which 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.

The Spaces and functions of IMAH are designed by the following Formula Expression. The details are explained in [6].

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 $r \times s$ 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".

2.3 Condition Formula Search

If you assume x to be a formula and p, $!p$, $p+q$, $p \times q$, $!(p+q)$, $!(p \times q)$ to be condition formulas, the images of (x, $p+q$), (x, $p \times q$), (x, $!(p+q)$), (x, $!(p \times q)$) by f, g are the following:

$f(x, p) = g(x, !p)$

$f(x, !p) = g(x, p)$

$f(x, p+q) = f(x, p) + f(g(x, p), q)$

$f(x, p\times q) = f(f(x, p), q)$

$f(x, !(p+q)) = g(g(x, p), q)$

$f(x, !(p\times q)) = g(f(f(x, p), q)$

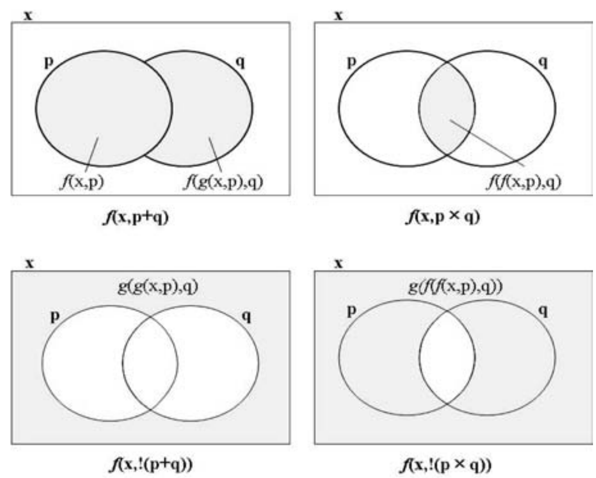

Figure 2.3 Images by the condition formula processing map *f*

Figure 2 is each image by the condition formula processing map *f*. It is obvious that any complicated condition formula can be processed by the combinations of the above six correspondences.

2.4 Implementation

A quotient acquisition map is the main function of a condition formula processing 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.

3. A Staff Assignment Information Management System

3.1 Outline

We propose the core logic of staff assignment information management using CDS. In this section, staff assignment information of a company is managed while a staff member is assigned to multiple departments (or projects) or the names of departments (or projects) change. In development of the application, firstly a formula for staff assignment record information is designed, along with the output for history information of staff assignments and the output for staff information of assignments by the maps of CDS. Next, a term is created according to the space design and added to the formula when a staff assignment changes. Then, required data can be outputted according to the output design.

3.2 The design of assignment space and data input/output

We design a formula for staff assignment record information and processing as follows:

$$\Sigma staff_i(time_j \times assignment_j) \qquad \text{(formula 3.2)}$$

$staff_i$: a factor which identifies a staff

$assignment_j$: a factor which expresses a staff assignment name at $time_j$

This is a disjoint union of terms for staff assignment record information as a topological space. Here, we assume that an initial value of each assignment is ε ($assignment_0=\varepsilon$) and also that if there is no assignment for a staff member, the value of the assignment is ε.

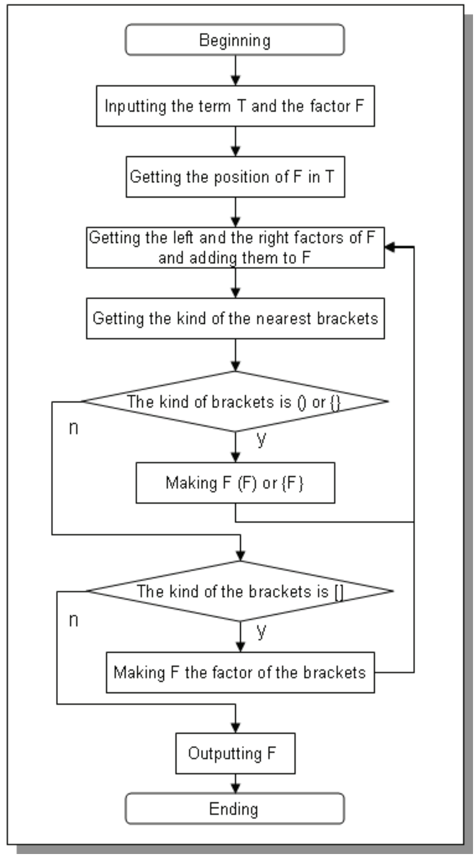

Figure 2.4 The flowchart of the algorithm for a quotient acquisition map

When the assignment of a staff member changes, a term for a staff assignment record is created. If a user wants to know the assignment history of $staff_n$, he/she gets the image of formula3.2 by the map f (2.3) and $staff_n$.

$$f(formula3.2, \text{'}staff_n\text{'}) = staff_n(time_0 \times \varepsilon + time_1 \times assignment_1 + time_2 \times assignment_2 + \ldots + time_j \times assignment_j)$$

You can see the staff assignment history of '$staff_n$'.

Next, if a user wants to get staff information of the $assignment_n$ at $time_m$, he/she gets the image of formula3.2 by the map f, the attaching map h [6] and '$time_m \times assignment_m$'.

$$h(f(formula3.2, \text{'}time_m \times assignment_m\text{'})) = (staff_o + staff_p + staff_q + \ldots) \times time_m \times assignment_m$$

From the result, you can see the staff information of the $assignment_n$. Here, we show simple examples of data input/output. If the assignment of $staff_1$, $staff_2$ in $time_0$ ~ $time_4$ are as in Table 3.2, the terms are created according to the design below (formula 3.2-2) and added to the formula.

time	**$staff_1$**	**$staff_2$**
$time_0$	(no assignment)	(no assignment)
$time_1$	marketing department in Osaka	technical department
$time_2$	research center, project A	research center
$time_3$	research center, business department	planning department, marketing department in Tokyo
$time_4$	(no assignment)	managing department

Table 3.2-1 Assignment data of $staff_1$ and $staff_2$

$staff_1 \times time_0 \times \varepsilon + staff_2 \times time_0 \times \varepsilon + staff_1 \times time_1 \times$marketing department$\times$Osaka$+staff_2 \times time_1 \times$technical department$+staff_1 \times time_2 \times$(research center+project A)$+staff_2 \times time_2 \times$research center$+staff_1 \times time_3$(business department+research center)$+staff_2 \times time_3$(planning department+marketing department$\times$Tokyo)$+staff_1 \times time_4 \times \varepsilon + staff_2 \times time_4 \times$managing department

(formula 3.2-2)

In Table 3.2-1, it is assumed that *project A* in $time_2$ has developed into *business department* in $time_3$.

Next, if $staff_3$ is added to staff members and assigned to the business department located in Tokyo at $time_4$ as seen in Table 3.2-2, the terms are created and added to formula 3.2-2 as below.

Time	$staff_3$
$time_3$	(no assignment)
$time_4$	research center in Tokyo

Table 3.2-2 Assignment data of $staff_3$

(formula3.2-2)$+staff_3 \times time_3 \times \varepsilon + staff_3 \times time_4 \times$Tokyo$\times$research center

(formula 3.2-3)

Here, if a user wants to know the assignment history information of $staff_1$, he/she uses the map *f* as below.

f(formula 3.2-3, '$staff_1$') = $staff_1(\varepsilon + \ldots + time_1 \times$marketing department$\times$Osaka$+time_2 \times$(research center+project A)$+time_3 \times$(research center+business department))

From the result, you can see the assignment history of $staff_1$ (marketing department at $time_1 \rightarrow$ research center and project A at $time_2 \rightarrow$ research center and business department at $time_3$).

Next, if a user wants to get staff information of the *research center*, he/she uses the map *f*.

h(formula 3.2-3, 'research center') = ($staff_1(time_2+time_3)+staff_2 \times time_2+staff_3 \times time_4 \times$Tokyo)$\times$research center

From the result, you can see that the staff members of the *research center* are $staff_1$ *and* $staff_2$ at $time_2$, $staff_1$ *at* $time_3$ *and* $staff_3$ at $time_4$.

If a user wants to answer the question "Who is engaged in both project A and the business department?", he/she uses the map *f*.

h(formula3.2-3,'projectA+business department') = $staff_1(time_2 \times$projectA$+ time_3 \times$business department)

You can see that the staff is $staff_1$.

3.3 Considerations

In developing a business application for staff assignment information management using CDS, it is quite capable of dealing with unexpected situations, such as the assignment of a staff member to multiple departments/projects, changes in the organization, or the addition of new staff, because of its topological modeling and the mechanism of disjoint union. In short, if you record changes in staff assignments by creating a term for the staff assignment and adding it to the formula one after another, you can output the data you want using the maps of CDS.

4. 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, 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 also satisfies requirements of an inductive system. 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. If you use the methods in [7], 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 design can be created so that the attributes of objects don't need to be designed in advance.

5. Conclusions

In this paper, we have applied CDS to the development of the core logic of staff assignment information management in a company. If you use CDS during development, application programs can become simpler because of CDS' flexibility at the time of data modeling and the useful maps, and a user can confirm the output result by the maps just after designing business objects. It will dramatically raise productivity and maintainability in business application development. Here, the point we should emphasize is that the quality of the business application developed 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 the system developer designing the formulas becomes more important when employing CDS.

References

[1] Kunii, T. L., Kunii, H. S. (1999). 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, p.19-24.

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

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

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

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

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

[7] Thakkar, Hetal., Mozafari, Barzan., Zaniolo, Carlo (2008). Designing an inductive data stream management system: the stream mill experience, *In*: Proceedings of the 2nd international workshop on Scalable stream processing system, Nantes, France, ACM Press, p.79-88.