An Example of a Charge Calculation System using the Numerical Value and Exponential Calculation of the Cellular Data System

Toshio Kodama
CDS Project
Maeda Corporation
Tokyo, Japan
kodama.ts@jcity.maeda.co.jp
kodama@lab.acs-jp.com

Tosiyasu L. Kunii
Morpho Inc
Tokyo, Japan
kunii@ieee.org

Yoichi Seki
Software Consultant
Tokyo, Japan
gamataki51@hotmail.com

Abstract— in the era of cloud computing, data is processed in a “Cloud”, and data and its dependencies between systems or functions progress and change constantly within the cloud, as a user’s requirements change. Such information worlds are called cyberworlds. We need a more powerful mathematical background which can model the cyberworlds in the "cloud" as they are. We consider the Incrementally Modular Abstraction Hierarchy (IMAH) to be appropriate for modeling the dynamically changing cyberworlds by descending from the most abstract homotopy level to the most specific view level, while preserving invariants. We have developed a data processing system called the Cellular Data System (CDS) based on IMAH. In this paper, we introduce numerical value calculation, exponential calculation and the processing maps on the presentation level of IMAH into CDS. This function is quite effective in business application development because numerical values and exponential identifiers can be put into formulas and be calculated. We show its effectiveness through examples of core processing of a cellular phone charge calculation system using numerical value and exponential calculations.

Keywords-component; incrementally modular abstraction hierarchy, formula expression, numerical value calculation, exponential calculation

I. INTRODUCTION

Cyberworlds are information worlds formed in a cloud either intentionally or spontaneously, with or without design. As information worlds, they are either virtual or real, and can be both. In terms of information modeling, the theoretical ground for the cyberworlds is far above the level of integrating spatial database models and temporal database models. They are more complicated and fluid than any other previous worlds in human history, and are constantly evolving. The number of companies that conduct business in cyberspace, such as Google and eBay is increasing and the market is growing remarkably. On the other hand, in general business application system, as the scale of systems becomes larger and the specifications of systems changes more frequently, development and maintenance becomes more difficult, leading to higher costs and delays. In some cases, a huge system as the mainstay system in a large company, where the number of program steps is the hundreds of millions, needs several years to develop. Increases in development and maintenance cost squeeze management. Such situations arise because of combinatorial explosions. The era of cloud computing requires a more powerful mathematical background to model the cyberworlds and to prevent combinatorial explosions. In the cloud, every business object and business logic should be expressed in a unified form to eliminate discontinuity between systems or functions and to meet changes in user requirements. The needed mathematical mechanisms are: 1. disjoint union of spaces by an equivalence relation; 2. change in spaces to preserve invariants; 3. attachment of different spaces by an equivalence relation; 4. space with dimensions as a special case. We consider the Incrementally Modular Abstraction Hierarchy (IMAH) that one of authors (T. L. Kunii) proposes able to satisfy the above requirements, as it models the architecture and the dynamic changes of cyberworlds from a general level (the homotopy level) to a specific one (the view level), preserving invariants while preventing combinatorial explosion [1]. It also benefits the reuse of information, guaranteeing modularity of information based on the mechanism of disjoint union. Unlike IMAH, other leading data models do not support the disjoint union or the attaching function by equivalence relation. In this research, one of the authors (Y. Seki) proposed a finite automaton called Formula Expression as a development tool to realize IMAH. One of authors (T. Kodama) has actually designed spaces and implemented the data processing system called the Cellular Data System (CDS) using Formula Expression. In this paper, we put emphasis on practical use by taking up some examples. First, we design numerical value and exponential calculations to put numerical values and exponential identifiers in Formula Expression as a function on the presentation level and implemented it. We demonstrate the effectiveness of CDS by developing a general business application system of a cellular phone charge calculation and abbreviate the process of implementing application programs.

II. IMAH AND FORMULA EXPRESSION

A. The Incrementally Modular Abstraction Hierarchy

The following list is 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 calculations. When we define a function in a cyberspace, we need domains that guarantee continuity such that the 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. An example of an adjunction space level is shown in Figure 1.

**B. 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 $\epsilon$ is Formula Expression
3. zero element $\phi$ 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, $\times, (, [\ ]$ can be abbreviated. $+$ means disjoint union and is expressed as $\Sigma$ specifically and $\times$ is also expressed as $\Pi$. 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, four maps (the expansion map, the bind map, the division map, the attachment map) are defined [9].

**C. A Conditional 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, and the map is called a condition formula processing map. 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. The condition formula processing map $f$ is a map that gets a disjoint union of terms that 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. Figure 2 shows each image by the condition formula processing map $f$.

**III. NUMERICAL VALUE CALCULATION AND EXPONENTIAL CALCULATION ON THE PRESENTATION LEVEL**

**A. The properties of numerical value calculation and exponential calculation**

If we assume that $p, q, r$ are arbitrary numerical factors, and that $s, t, u$ are arbitrary factors, the numerical value
calculation in Formula Expression has the following properties:

1. \( sxI = s \)
2. \( sx0 = 0 \)
3. \( ps = ps \)
4. \( sxp+sxq = sx(p+q) \)
5. \( sxpixq = sx(p^2q) \)
6. \( sxp = pxs \)
7. \( sxp(1xq+uxr) = sxx(p^2r)+uxx(q^2r) \)
8. \( sxp+1xq)uxr = sxux(p+q)r+txux(q^2r) \)

And the exponential calculation in Formula Expression has the following properties:

1. \( s^0 = 1 \)
2. \( s^1 = s \)
3. \( hs^q = s^q \)
4. \( s^0sx^q = s^{0+q} \)
5. \( (s^q)^r = s^{qr} \)

### B. The numerical value and exponential calculation map

The numerical value and exponential calculation map \( f \) is defined based on the above mentioned properties. If you assume the entire set of formulas, including the numerical factors, to be \( A, f: A \rightarrow A \) and \( f \) is the followings:

\[
\begin{align*}
&f: s \rightarrow sxI \\
&f: ps \rightarrow sxp \\
&f: sxp+sxq \rightarrow sx(p+q) \\
&f: sxpixq \rightarrow sx(p^2q) \\
&f: sxp(1xq+uxr) \rightarrow sx(p^2r)+uxx(q^2r) \\
&f: sxp+1xq)uxr \rightarrow sxux(p+q)r+txux(q^2r) \\
&f: s^0 \rightarrow 1 \\
&f: s^1 \rightarrow s \\
&f: hs^q \rightarrow s^q \\
&f: s^0sx^q \rightarrow s^{0+q} \\
&f: (s^q)^r \rightarrow s^{qr}
\end{align*}
\]

And if we assume that \( T \) is an arbitrary term, and that \( E \) is an arbitrary formula, \( f \) is:

\[
f(T+T) = f(T)f(T) \\
f(T+T) = f(T)+f(T) \\
f(E) = f(E)
\]

An example of the map \( f \) is shown below.

\[
f(\text{cat+dog+rabbit+dog+cat+rabbit+dog+rabbit+mouse})
\]

As a further example, when three boxes of cigarettes (which are $5 per box) are bought, the formula for calculation and processing by the map \( f \) is:

\[
f(\text{cigarette}\times5\times(1\times\text{box})\times3\times\text{box})
\]

The following is the coding for the calculation of numerical value and exponential calculation. The focus is the recursive process (line 7, in bold) that is done if a coming numerical calculation is of the type \((\)\). The explanation is abbreviated due to space limitations.

1. term = null; factor = null;
2. while(factor is not null){
3. term = getTerm(factor);
4. while(term is not null){
5. factor = getFactor(term);
6. if(factor is of the type \((\)\){
7. \( \text{factor} = \text{calculate}(\text{the contents}); \)
8. \}
9. \}
10. \}
11. \}
12. newTerm = newNF + newLF;
13. newFormula = newFormula + newTerm;
14. return newFormula;

### IV. DEVELOPMENT OF A CELLULAR PHONE CHARGE CALCULATION SYSTEM

#### A. Outline

We take up the simple example of a cellular phone charge calculation system to secure generality, because charge calculation systems are generally developed in most industries. The most important thing with a charge calculation system is to be able to deal with unexpected changes of users' charge plans or of user contract data. If a cellular phone company is going to introduce new charge plans for the cellular phone to further their strategy of gaining more cellular phone subscribers, the development of an application program to calculate charges will cost a lot and take a lot of time, because the program needs to be consistent with previous data/programs, and user contract data will change according to the new charge plans. To solve these difficulties, we apply CDS to the development of core processing of the cellular phone charge calculation system. Firstly, each formula for contract data and plan type data is designed as a topological space. Secondly, formulas from January to December of a year, where basic charges or discount rates are expressed using numerical values and exponential calculation, are created according to the design. Every time charge plan or user contract data is changed over time, another topological space is created and added, forming a disjoint union of topological spaces. Thirdly, each charge for use of the cellular phone can be calculated using the maps of CDS, such as the condition formula processing map (II.C) or the numerical value and exponential calculation map (II.A.III.B). In these designs, numerical value calculation, exponential calculation, and the calculation map are used to express unit price per month, duration of use, discount rates and the calculations among them. Here, actual data and functions are simplified to focus on verifying development of core processing without losing generality.
B. The design of topological spaces

We design a formula for topological spaces for (1) the user contract data, and (2) the charge plan data. Numerical value identifiers are used to express unit price per month, duration of use, and discount rates. The formulas for (1) and (2) are designed as follows:

(1) \text{CONTRACT}\{\Sigma \text{user id}_i\} \{\Sigma \text{charge plan type}_j\} \{\Sigma \text{month}_k\}

user id\_i; a factor which identifies a cellular phone user
charge plan type\_j; a factor which shows a charge plan
type of a cellular phone
month\_k; a factor which shows a month of use of a

(2) \text{PLAN}\{\Sigma \text{charge plan type}_i\} \{\Sigma \text{unit price per month}_j\} \{\Sigma \text{month}_k\}

unit price per month\_j; a factor which shows unit price per month of charge

C. Data input according to the design

-January to March of the year-

First, for the period January–March, assume that seven users (Tom, Mike, Joy, Alice, Jack, Nancy, John) sign contracts for the use of cellular phones and that there are two plan types: 1. “adult” whose basic fee is “3000yen” per month and 2. “child” whose basic fee is “2000yen” per month, as seen in Figure 3. You create the following formula for the topological space (formula C-1), according to the above design in B, and add it to data storage.

formula C-1:

\text{CONTRACT}\{\text{Tom+Mike+Joy+Alice+Jack+Nancy+John}\} \{\text{adult+adult+adult+child+child+student+adult}\} \{\text{Jan+Feb+March}\}

-Jan–Mar

<table>
<thead>
<tr>
<th>User Contract Data</th>
<th>PLAN TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tom</td>
<td>adult</td>
</tr>
<tr>
<td>Mike</td>
<td>adult</td>
</tr>
<tr>
<td>Joy</td>
<td>adult</td>
</tr>
<tr>
<td>Alice</td>
<td>child</td>
</tr>
<tr>
<td>Jack</td>
<td>child</td>
</tr>
<tr>
<td>Nancy</td>
<td>adult</td>
</tr>
<tr>
<td>John</td>
<td>adult</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Charge Plan Data</th>
<th>BASIC CHARGE PER MONTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>adult</td>
<td>3000</td>
</tr>
<tr>
<td>child</td>
<td>3000</td>
</tr>
<tr>
<td>student</td>
<td>3000*0.7</td>
</tr>
<tr>
<td>the earthquake victim</td>
<td>-1000</td>
</tr>
</tbody>
</table>

-Apr-Jun

Next, for the period April–June, assume that a cellular phone operator introduces a new charge plan type, “aged”, for elderly people, whose basic fee is 50% off the “adult” fee, that John qualifies for the plan, and that the fee for the “child” plan changes from “2000yen/m” to “1000yen/m”, as seen in Figure 5. You create the following formula for the topological space (formula C-3), according to the design, and add it to the previous formula.

formula C-3:

\text{CONTRACT}\{\text{Tom+Mike+Joy+Alice+Jack+Nancy+John}\} \{\text{adult+adult+child+child+student+aged}\} \{\text{July+August+September}\}

You create the following formula for the topological space (formula C-2), according to the above design, and add it to the previous formula.

formula C-2:

\text{(formula C-1)+CONTRACT}\{\text{Tom×0.5+Mike+Joy+Alice+Jack+Nancy+John×0.5}\} \{\text{student+adult+adult+child+child+student+adult}\} \{\text{April+May+June}\} \{\text{3000 yen/m+2000 yen/m+3000yen/m×0.7}\} \{\text{April+May+June}\}

-Apr-Jun

Figure 3. User contract data and charge plan data from January through March

-Apr-Jun

Next, for the period July–September, assume that cellular phone operator introduces a new charge plan type, “aged”, for elderly people, whose basic fee is 50% off the “adult” fee, that John qualifies for the plan, and that the fee for the “child” plan changes from “2000yen/m” to “1000yen/m”, as seen in Figure 5. You create the following formula for the topological space (formula C-3), according to the design, and add it to the previous formula.

formula C-3:

\text{(formula C-2)+CONTRACT}\{\text{Tom+Mike+Joy+Alice+Jack+Nancy+John}\} \{\text{student+adult+adult+child+child+student+aged}\} \{\text{July+August+September}\} \{\text{3000yen/m+1000yen/m+3000yen/m×0.7+3000yen/m×0.5}\} \{\text{July+August+September}\}

-From July through September-

Next, for the period July–September, assume that cellular phone operator introduces a new charge plan type, “aged”, for elderly people, whose basic fee is 50% off the “adult” fee, that John qualifies for the plan, and that the fee for the “child” plan changes from “2000yen/m” to “1000yen/m”, as seen in Figure 5. You create the following formula for the topological space (formula C-3), according to the design, and add it to the previous formula.

formula C-3:

\text{(formula C-2)+CONTRACT}\{\text{Tom+Mike+Joy+Alice+Jack+Nancy+John}\} \{\text{student+adult+adult+child+child+student+aged}\} \{\text{July+August+September}\} \{\text{3000yen/m+1000yen/m+3000yen/m×0.7+3000yen/m×0.5}\} \{\text{July+August+September}\}

-From April through June-

Next, for the period April–June, assume that a cellular phone operator introduces a new charge plan type, “student”, whose basic fee is 30% off the “adult” fee, and that Tom and Nancy qualify as students, as seen in Figure 4. Moreover, assume that a special discount (1000yen off per month) for earthquake victims is offered, and that Tom and John qualify.
and add it to the previous formula.

topological space (formula C-4), according to the design,
seen in Figure 6, and that Alice and John cancel their

From October through December-

Finally, for the period October-December, assume that a
cellular phone operator adds an extra charge for all users, as
seen in Figure 6, and that Alice and John cancel their contracts. You create the following formula for the
topological space (formula C-4), according to the design, and add it to the previous formula.

formula C-4:

\[ \text{formula C-3}) + \text{CONTRACT}\{\text{Tom} + \text{Mike} + \text{Joy} + \text{Jack} + \text{Nancy}\} + \text{student} \times (\text{Oct} + \text{Nov} + \text{Dec}) \]

Secondly, from the result, you make the following condition formula.

\[ \text{Plan} \times (\text{adult} \times (\text{Jan} + \text{Feb} + \text{Mar}) + (\text{student} + \text{earthquake victim}) \times (\text{Apr} + \text{May} + \text{Jun}) + \text{student} \times (\text{July} + \text{Aug} + \text{Sep}) + \text{student} \times (\text{Oct} + \text{Nov} + \text{Dec})) \]

And you get the image of formula C-4 by the condition formula through the map \( f \) again. The result is below.

formula D-2:

\[ \text{PLAN} \times (\text{adult} \times 3000 \text{yen/m} \times (\text{Jan} + \text{Feb} + \text{Mar}) + (\text{student} \times 3000 \text{yen/m} \times 0.7 + \text{earthquake victim} \times -1000 \text{yen/m}) \times (\text{Apr} + \text{May} + \text{Jun}) + \text{student} \times 3000 \text{yen/m} \times 0.7 + \text{student} \times (\text{July} + \text{Aug} + \text{Sep}) + \text{student} \times (\text{Oct} + \text{Nov} + \text{Dec})) \]

Thirdly, you replace “(Jan+Feb+Mar)”, “(Apr+May+Jun)”, “(July+Aug+Sep)” and “(Oct+Nov+Dec)” by “3m”, which means “three months”, in formula D-1, and calculate it by the

\[ \text{CONTRACT} \times \text{Tom} \times (\text{adult} \times (\text{Jan} + \text{Feb} + \text{Mar}) + (\text{student} + \text{earthquake victim}) \times (\text{Apr} + \text{May} + \text{Jun}) + \text{student} \times (\text{July} + \text{Aug} + \text{Sep}) + \text{student} \times (\text{Oct} + \text{Nov} + \text{Dec})) \]

From the result, you can know that the answer is 25,800yen.

Next, if a user wants to answer the question “How much do all users pay in fees in July?”, firstly you can get the
image of formula C-4 by “\( \text{CONTRACT} \times \text{Jul} \)” through the map \( f \). The result is below.

formula D-3:

\[ \text{PLAN} \times (\text{adult} \times 3000 \text{yen/m} \times 3 \text{m} + (\text{student} \times 3000 \text{yen/m} \times 0.7 + \text{earthquake victim} \times -1000 \text{yen/m}) \times 3 \text{m} + \text{student} \times (\text{July} + \text{Aug} + \text{Sep}) + \text{student} \times (\text{Oct} + \text{Nov} + \text{Dec})) \]

And you get the image of formula D-3 by the condition formula through the map \( f \) again. The result is below.

formula D-4:

\[ \text{PLAN} \times (\text{student} \times 3000 \text{yen/m} \times 0.7 + \text{adult} \times 3000 \text{yen/m} + \text{student} \times 3000 \text{yen/m} + \text{child} \times 1000 \text{yen/m} + \text{child} \times 1000 \text{yen/m} + \text{student} \times 3000 \text{yen/m} \times 0.7 + \text{student} \times 3000 \text{yen/m} \times 0.5 + \text{adult} \times 3000 \text{yen/m} \times 0.5 + \text{child} \times 2000 \text{yen/m} + \text{child} \times 2000 \text{yen/m} + \text{student} \times 3000 \text{yen/m} \times 0.5 + \text{student} \times 3000 \text{yen/m} \times 0.5 + \text{student} \times 3000 \text{yen/m} \times 0.5 + \text{student} \times 3000 \text{yen/m} \times 0.5) \]

D. Data output

If a user wants to answer the question “How much does Tom pay in fees during the year?”, firstly you can get the
image of formula C-4 by “\( \text{CONTRACT} \times \text{Tom} \times (\text{Jan} + \text{Feb} + \text{Mar} + \text{Apr} + \text{May} + \text{Jun} + \text{July} + \text{Aug} + \text{Sep} + \text{Oct} + \text{Nov} + \text{Dec}) \)” through the condition formula processing map \( f \). The result is below.

formula D-1:
2.

In the above example, from April through June, some users also qualifies for two plan types, "the earthquake victim" and "student", and John can belong to multiple plan types (Tom qualifies for two.

E. Considerations

In general, it costs a lot to develop and maintain application programs for the cellular phone charge calculation system because of the complexity and frequent changes in the business services. This example shows that every time the service changes, if you only have to create new formulas for the user contract data and the charge plan data according to the formula design, you can get the output using the processing map, thereby reducing the amount of application development and maintenance. This is mainly because of:

1. the mechanism of the topological processing function of CDS, which deals with subsets as an element

In the above example, from April through June, some users can belong to multiple plan types (Tom qualifies for two plan types, “the earthquake victim” and “student”, and John also qualifies for two plan types, “the earthquake victim” and “adult”).

2. numerical values and exponential identifiers that can be put into formulas and calculated according to the definition of numerical and exponential calculation

In the above example, unit price per month and duration of use are calculated using numerical and exponential calculations.

REFERENCES


VI. CONCLUSIONS

We have developed a data processing system called the Cellular Data System (CDS) based on IMAH. In this paper, we introduced numerical value and exponential calculation and the processing maps on the presentation level of CDS. Using the function in business application development, you can put numerical values and exponential identifiers into formulas as identifiers, and they can be calculated based on definition, so that system development become simpler, business processes more visible, and business applications more flexible to changes in business conditions, as shown in section 4. As a result, use of CDS can make developers more creative, preventing from frequent troubles between the customer side and the supplier side, while preventing combinatorial explosions.

The fruits of this research have already been put to practical use in many companies, and further strides are being made every day. We are sure that CDS has possibilities to bring great social impact in the era of cloud computing.


