

## A Continuous Quantity Factor in the Condition Formula Search

Toshio Kodama<sup>1</sup>, Tosiyasu L. Kunii<sup>2</sup>, Yoichi Seki<sup>3</sup>

<sup>1</sup>*CDS Business Dept., Advanced Computer Systems, Inc. and Maeda Corporation, Tokyo, Japan  
kodama@lab.acs-jp.com, <https://www.cellulardatasystem.com/e/index.html>*

<sup>2</sup>*Morpho, Inc., The University of Tokyo, Tokyo, Japan  
kunii@ieee.org, kunii@acm.org, <http://www.kunii.net/>*

<sup>3</sup>*Software Consultant, Tokyo, Japan  
gamataki61@mail.hinocatv.ne.jp*

### Abstract

*Cyberworlds in the era of 'cloud' computing are being created on the Web where data and its dependencies are constantly changing and evolving. The problem of combinatorial explosion in system development inevitably arises when dealing with cyberworlds. To solve the problem, we have developed a data processing system called the Cellular Data System (CDS), based on the Incrementally Modular Abstraction Hierarchy (IMAH) in the cellular model, which offers powerful mathematical reason. In this paper, we design and implement a continuous quantity factor to deal with objects that express continuous quantity and integrate it into the condition formula search, which is the effective data search function of CDS, used to analyze data in cyberworlds without limitation by defining schema. If you take advantage of continuous quantity factors with the condition formula search in business application development, application logic development is made much simpler. In addition, we verify the effectiveness of a continuous quantity factor by taking up examples of core logic development of business applications.*

**Keywords:** *cyberworlds, cellular model, formula expression, continuous quantity factor, condition formula*

### 1. Introduction

The system of cyberworlds is a distributed system. One of the features of cyberworlds is that data dependency is constantly changing in them. Cyberworlds are more complicated and fluid than any other previous worlds in human history and are constantly evolving. For example, millions of users manage their own blog information every day through Web services on mobile phones, like SNS in Japan, which is considered one of main elements of cyberworlds. At the same time, user requirements for cyberworlds also change and get more complicated as cyberworlds change. If you analyze data using the existing technology in business application

development, you have to modify the schema design and application programs whenever schemas or user requirements for output change. That leads to combinatorial explosion, because user requirements, and their combinations and schemas must be specified clearly at the design stage in general business application development. That is a fundamental problem, so we have to reconsider development from the data model level.

Is there a data model that can reflect the changes in schemas and user requirements for output to analyze data in cyberworlds? We consider that Incrementally Modular Abstraction Hierarchy (IMAH) of the cellular model proposed by one of authors (T. L. Kunii) is the most suitable model. The IMAH can model the architecture and the changes of cyberworlds and real worlds from a general level to a specific one, preserving invariants while preventing combinatorial explosion [1]. From the viewpoint of IMAH, existing data models are positioned as special cases. For example, UML can model objects at levels below the presentation level, and in the relational data model, a relation is an object at the presentation level which extends a cellular space because it has necessary attributes in which a type is defined, while the processing between relations is based on the set theoretical level. In the object-oriented model, an object is also the object in the presentation level, which extends a cellular space, while the relation between Class is the tree structure, which is a special case of a topological space. An Object in XML is considered a special case of a cellular space which extends a topological space, because an attribute and a value of it are expressed in the same tag format.

In our research, one of the authors (Y. Seki) proposed an algebraic system called Formula Expression as a development tool to realize the cellular model. Another (T. Kodama) has actually implemented CDS using Formula Expression. [12] In this paper, we have introduced the concept of a continuous quantity factor into CDS. A continuous quantity formula is effective when a continuous quantity is dealt with in business application development. In addition, we have placed emphasis on practical use by taking up some examples. Firstly, we explain the CDS and its main data search function briefly

in Section 2, 3. Secondly, we design the properties of a continuous quantity by Formula Expression and integrate them into the condition formula search function in Section 4. Next, we implement them in Section 5. We demonstrate the effectiveness of the continuous quantity factor by developing a business application system, thereby abbreviating the process of designing and implementing most application programs in Section 6. The business application system is the core logic of a meeting reservation system, where meetings are arranged in accordance with staff work schedules and room reservation schedules. Related works are mentioned in Section 7. Lastly, we conclude in Section 8.

## 2. The Cellular Data System (CDS)

### 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 to 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.

### 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  $\epsilon$  is Formula Expression
3. zero element  $\emptyset$  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  $\sqcup$  as 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, five maps (the expansion map, the bind map, the division map, the attachment map, the homotopy preservation map) are defined [9].

## 3. The Condition Formula Search of CDS

### 3.1. A condition formula

If users can specify search conditions, data search will become more functional when searching data from data storage. Here, we introduce the function for specifying conditions defining a condition formula by Formula Expression into CDS. Let propositions  $P, Q$  be sets which include characters  $p, q$  respectively. The conjunction, disjunction and negation of them in logical operation are defined by Formula Expression as follows:

- |                |                           |
|----------------|---------------------------|
| 1) Conjunction | $P \wedge Q = p \times q$ |
| 2) Disjunction | $P \vee Q = p + q$        |
| 3) Negation    | $\neg P = !p$             |

A formula created from these is called a condition formula. Here "!" 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.

### 3.2. A condition formula processing map

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.

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:

$$\begin{aligned} f(x, p) &= g(x, !p) \\ f(x, !p) &= g(x, p) \\ f(x, p+q) &= f(x, p) + f(x, q) \\ f(x, p \times q) &= f(x, p) \times f(x, q) \\ f(x, !(p+q)) &= g(g(x, p), q) \\ f(x, !(p \times q)) &= g(f(x, p), q) \end{aligned}$$

Here,  $f$  is a quotient acquisition map and  $g$  is a remainder acquisition map. Fig 2.3 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 four correspondences.

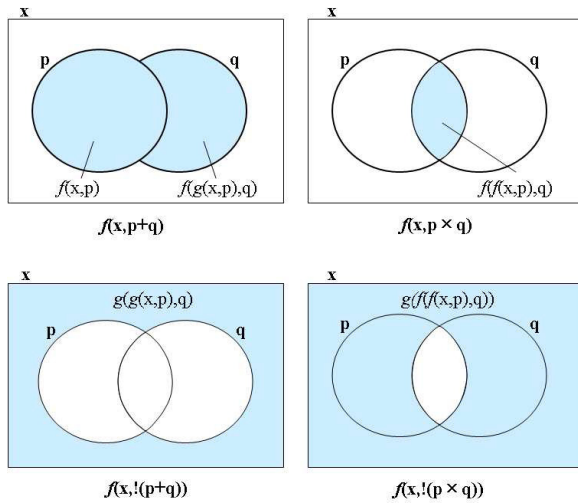


Fig 2.3 Images by the condition formula processing map  $f$

## 4. A Continuous Quantity Factor and Its Application to the Condition Formula Search

### 4.1. Definition

If it is assumed that  $r$  and  $s$  ( $r < s$ ) are arbitrary numerical identifiers, a continuous quantity factor to express continuous quantity from  $r$  to  $s$  is defined  $[r+s]$ . If you assume  $t, u$  ( $t < u$ ) and  $v, w$  ( $v < w$ ) are also arbitrary numerical identifiers and  $a$  is an arbitrary letter factor, the continuous quantity factor has the following properties:

1.  $[s+r] = \varnothing$  (if  $r < s$ )
2.  $[r+r] = \varepsilon$
3.  $[r+s] + [t+u] = [r+u]$  (if  $r \leq t < s \leq u$ )  
 $= [t+s]$  (if  $t \leq r < u \leq s$ )  
 $= [r+s]$  (if  $r \leq t < u \leq s$ )  
 $= [t+u]$  (if  $t \leq r < s \leq u$ )  
 $= [r+s] + [t+u]$  (if  $t \leq u \leq r < s$  or  $r < s \leq t \leq u$ )
4.  $[r+s] - [t+u] = [r+t]$  (if  $r \leq t < s \leq u$ )  
 $= [u+s]$  (if  $t \leq r < u \leq s$ )  
 $= [r+t] + [u+s]$  (if  $r \leq t < u \leq s$ )  
 $= \varnothing$  (if  $t \leq r < s \leq u$ )  
 $= [r+s]$  (if  $t \leq u \leq r < s$  or  $r < s \leq t \leq u$ )
5.  $[r+s] \times [t+u] = [t+s]$  (if  $r \leq t < s \leq u$ )  
 $= [r+u]$  (if  $t \leq r < u \leq s$ )  
 $= [t+u]$  (if  $r \leq t < u \leq s$ )  
 $= [r+s]$  (if  $t \leq r < s \leq u$ )  
 $= \varnothing$  (if  $t \leq u \leq r < s$  or  $r < s \leq t \leq u$ )
6.  $[r+s]([t+u] + [v+w]) = [r+s] \times [t+u] + [r+s] \times [v+w]$
7.  $([r+s] + [t+u])[v+w] = [r+s] \times [v+w] + [t+u] \times [v+w]$
8.  $a \times [r+s] = [r+s] \times a$

### 4.2. The image of a continuous quantity factor of the condition formula processing map

Here, the image of a continuous quantity factor by the condition formula processing map  $f$  is defined as follows:

$$\begin{aligned} f([r+s], [t+u]) &= [t+u] \text{ (if } r \leq t < u \leq s) \\ &= \varnothing \text{ (except if } r \leq t < u \leq s) \\ f([r+s], ![t+u]) &= [r+t] + [u+s] \text{ (if } r \leq t < u \leq s) \\ &= [r+s] \text{ (except if } r \leq t < u \leq s) \end{aligned}$$

A simple example is shown.

$$\begin{aligned} f((a+b \times c)[1+10], b \times [5+8]) \\ &= f(b \times c \times [1+10], ![5+8]) \\ &= b \times c \times ([1+5] + [8+10]) \end{aligned}$$

## 5. Implementation

This system is a web application developed using JSP and Tomcat 5.0 as a Web server. The client and the server

are the same machine. (OS: Windows XP; CPU: Intel Core2 Duo, 3.00GHz; RAM: 3.23Gbyte; HD: 240GB)

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

## 6. Case Study: A meeting reservation system

### 6.1. Outline

We have developed core logic of a meeting reservation system. In this case study, we assume that there are five staff; staff A ~ staff E and three meeting rooms; room1 ~ room3 and that each meeting is to be arranged by adjusting staff work schedules and room reservation schedules within a given period from 0 to 20.

Firstly, the formulas for staff scheduled time and room reserved time are designed using a continuous quantity factor and an operation for getting the formula for each's available time from them (6.2). Secondly, sample data of staff and rooms are inputted and the operation is carried out according to the design. Thirdly required data is outputted by the condition formula processing map (6.3).

### 6.2. Space design

We design a formula for the space and the operation as follows:

1. A formula for staff and their scheduled time as a topological space.

$$\text{Staff}(\Sigma \text{staff } id_i \times \Sigma [r_i + s_i])$$

*staff id<sub>i</sub>*: a factor which identifies a staff.

$[r_i + s_i]$ : a continuous quantity factor from  $r_i$  to  $s_i$ .

2. A formula for rooms and their reserved time as a topological space.

$$\text{Room}(\Sigma \text{room } id_i \times \Sigma [p_i + q_i])$$

*room id<sub>i</sub>*: a factor which identifies a room.

$[p_i + q_i]$ : a continuous quantity factor from  $p_i$  to  $q_i$ .

3. An operation for getting the formula for available times of staff and rooms

$$(\text{Staff}(\Sigma \text{staff } id_i \times [0+20]) + \text{Room}(\Sigma \text{room } id_i \times [0+20]))$$

$$-(\text{Staff}(\Sigma \text{staff } id_i \times \Sigma [r_i + s_i]) + \text{Room}(\Sigma \text{room } id_i \times \Sigma [p_i + q_i]))$$

[0+20]: a continuous quantity factor of the entire period (0 to 20).

We define the simple map  $h$  of CDS to multiply terms in a formula.

$$h: a+b+c+\dots \rightarrow a \times b \times c \times \dots$$

### 6.3. Data input/output

A formula for staff and rooms of the entire period is created as below:

$$\text{Staff}(\text{staff A} + \text{staff B} + \text{staff C} + \text{staff D} + \dots) \times [0+20] + \text{Room}(\text{room A} + \text{room B} + \text{room C} + \text{room D} + \dots) \times [0+20]$$

(formula 6.3-1)

If staff A schedules his/her work at time 1 to 3 and time 16 to 18, a term for staff A is created according to the space design 1 as below:

$$\text{Staff}(\text{staff A} \times ([1+3] + [16+18]))$$

And if staff B schedules his/her work at time 0 to 2 and time 5 to 8, a term for a staff B is created and added to the previous term as below:

$$\text{Staff}(\text{staff A} \times [1+3] + \text{staff B} \times ([0+2] + [5+8]))$$

Terms for staff C and staff D are also created as below:

$$\text{Staff}(\text{staff A} \times ([1+3] + [16+18]) + \text{staff B} \times ([0+2] + [5+8]) + \text{staff C} \times ([4+7] + [11+13]) + \text{staff D} \times ([10+14] + [17+19]) + \text{staff E} \times ([2+6] + [12+15] + [17+20]))$$

In the same way, a term for room reservations is created and added to the previous formula according to the space design 2 as below:

$$\dots + \text{Room}(\text{room1} ([0+4] + [10+14]) + \text{room2} ([4+7] + [15+17]) + \text{room2} ([7+10] + [12+14] + [17+20]))$$

(formula 6.3-2)

The following operation is done to get the formula for times staff and rooms are available from formula 6.3-1 and formula 6.3-2 according to the space design 3.

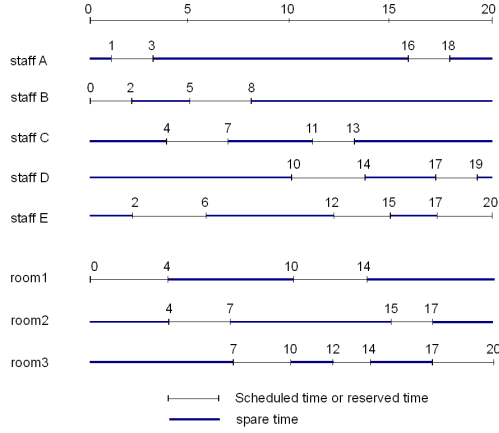
$$\text{formula 6.3-1} - \text{formula 6.3-2}$$

$$= \text{Staff}(\text{staff A} \times ([0+1] + [3+16] + [18+20]) + \text{staff B} \times ([2+5] + [8+20]) + \text{staff C} \times ([0+4] + [7+11] + [13+20]) + \text{staff D} \times ([0+10] + [14+17] + [19+20]) + \text{staff E} \times ([0+2] + [6+12] + [15+17])) + \text{Room}(\text{room1} ([4+10] + [14+20]) + \text{room2} ([0$$

$$+4)+[7+15]+[17+20])+room3([0+7]+[10+12]+[14+17]))$$

(formula6.3-3)

The simple figure is shown in Fig 6.3.



**Fig 6.3 The formula for times that staff and rooms are available**

If a user wants to answer the question “If a meeting is held from 10 to 12, who can participate in it and which rooms are free?”, he/she gets the image of formula 6.2-3 and the formula “[10+12]” by the map  $f$  as below:

$$f(\text{formula6.2-3}, '[10+12]')$$

$$=(\text{staff A}+\text{staff B}+\text{staff E})[10+12]+(\text{room2}+\text{room3})[10+12]$$

From the results, you can know that staff A, B, E can participate and room 2, 3 are free at the time of the meeting.

Next, if a user wants to answer the question “When are staff C, D, E free except from 6 to 16, and which rooms are free then?”, he/she gets the image of formula 6.2-3 and the condition formula “(staff D+staff E+staff F)×![6+16]” by the map  $f$  and the map  $h$  as below:

$$f(\text{formula6.2-3}, (\text{staff D}+\text{staff E}+\text{staff F})\times![6+16]')$$

$$=(\text{staff C}+\text{staff D}+\text{staff E})\times([0+2]+[16+17])$$

$$f(\text{formula6.2-3}, ([0+2]+[16+17])\times\text{room}')$$

$$=[0+2](\text{room2}+\text{room3})+[16+17](\text{room1}+\text{room3})$$

From the results, you can know that staff C, D, E can have a meeting between 0 and 2 in room2 or room3, or between 16 and 17 in room1 or room3.

Next, if a user wants to answer the question “When is it possible for all of staff to have a meeting together and which rooms are free then?”, he/she gets the image of

formula 6.2-3 and the formula “staff” by the map  $f$  and the map  $h$  as below:

$$h(f(\text{formula6.2-3}, \text{staff}'))$$

$$=\text{Staff}\times\text{staff A}\times\text{staff B}\times\text{staff C}\times\text{staff D}\times\text{staff E}\times([8+10]+[15+16])$$

$$f(\text{formula6.2-3}, ([8+10]+[15+16])\times\text{room}')$$

$$=[8+10](\text{room1}+\text{room2})+[15+16](\text{room1}+\text{room3})$$

From the results, you can know that all staff are able to meet between 8 and 10 in room1 or room2, or between 15 and 16 in room1 or room3.

## 6.4. Considerations

The condition formula search has become more effective by integrating a continuous quantity factor when a continuous quantity such as time, distance or temperature, etc. is dealt with.

If the existing method of business application development is used in this case study instead of the continuous quantity factor function of CDS, complicated input/output programs have to be developed according to needs, and their maintenance costs required to meet various and unexpected user requirements are considerable.

## 7. Related works

The distinctive features of our research are the application of the concept of topological process, which deals with a subset as an element, and that the cellular space extends the topological space, as seen in Section 2. Relational OWL as a method of data and schema representation is useful when representing the schema and data of a database, [3] but it is limited to representation of an object that has attributes. Our method can represent both objects: one that has attributes as a cellular space and one that doesn't have them as a set or a topological space.

Many works applying other models to XML schema have been done. The motives of most of them are similar to ours. The approach in [8] aims at minimizing document revalidation in an XML schema evolution, based on a part of the graph theory. The X-Entity model [9] is an extension of the Entity Relationship (ER) model and converts XML schema to a schema of the ER model. In the approach of [6], the conceptual and logical levels are represented using a standard UML class and the XML represents the physical level. XUML [10] is a conceptual model for XML schema, based on the UML2 standard. This application research concerning XML schema is needed because there are differences in the expression capability of the data model between XML and other models. On the other hand, objects and their relations in

XML schema and the above models can be expressed consistently by CDS, which is based on the cellular model. That is because the tree structure, on which the XML model is based, and the graph structure, on which the UML and ER models are based, is special cases of a topological structure mathematically. Entity in the models can be expressed as the formula for a cellular space in CDS. Moreover, the relation between subsets, as we showed in 3.2, cannot in general be expressed by XML.

Although CDS and the existing deductive database look alike apparently, the two are completely different. The deductive database [11] raises the expression capability of the relational database (RDB) by defining some rules. On the other hand, CDS is a proposal for a new tool for data management and has nothing to do with the RDB.

## 8. Conclusions and future work

In this paper, we have designed and implemented a continuous quantity factor and insert it into the condition formula search, which is the main data search function of CDS. A continuous quantity such as time, distance, etc. can be expressed as a factor in Formula Expression and integrated into business logic modeling in business application development. As a result, the cost of application program development can be dramatically cut.

Application logic development of CDS will proceed with the implementation of a process graph. [13] In the next step, a situation as a node is transferred to the next situation selecting a path as an edge. If we implement the process graph by developing CDS in future work, the automation of business application development will be completed. We believe that CDS will bring great social impact, changing existing development fundamentally. Our research is still in its infancy, but it is progressing every day and some business applications are in commercial use at several companies.

## Appreciation

We really appreciate for receiving some advice about the case study (Section 6) from Mr. Yuji Arai of ACS, Inc.

## References

- [1] T. L. Kunii and H. S. Kunii, "A Cellular Model for Information Systems on the Web - Integrating Local and Global Information", *In Proc. of DANTE'99*, IEEE Computer Society Press, pp.19-24, 1999.
- [2] Bernadette Farias Lósio, Ana Carolina Salgado, Luciano do Rêgo Galvão, "Conceptual modeling of XML schemas", *In Proc. of WIDM'03*, ACM Press, pp.102-105, 2003.

- [3] Takashi Washio, Hiroshi Motoda, "State of the art of graph-based data mining", *In ACM SIGKDD Explorations Newsletter*, ACM Press, pp.59-68, 2003.
- [4] David W. Embley, "Toward semantic understanding: an approach based on information extraction ontologies", *In Proc. of ADC'04*, Australian Computer Society, Inc., pp.3-12, 2004.
- [5] Cristian Pérez de Laborda, Stefan Conrad, "Relational.OWL: a data and schema representation format based on OWL", *In Proc. of APCCM '05*, Australian Computer Society, Inc., pp.89-96, 2005.
- [6] Nicholas Routledge, Linda Bird, Andrew Goodchild, "UML and XML schema", *In Proc. of ADC'02*, Australian Computer Society, Inc., pp.157-166, 2002.
- [7] Sean K. Bechhofer, Jeremy J. Carroll, "Parsing owl dl: trees or triples?", *In Proc. of WWW'04*, ACM Press, Inc., pp.266-275, 2004.
- [8] Giovanna Guerrini, Marco Mesiti, Daniele Rossi, "Impact of XML schema evolution on valid documents", *In Proc. of WIDM'05*, ACM Press, pp.39-44, 2005.
- [9] Bernadette Farias Lósio, Ana Carolina Salgado, Luciano do Rêgo Galvão, "Conceptual Modeling of XML Schemas", *In Proc. of WIDM'03*, ACM Press, pp.102-105, 2003.
- [10] HongXing Liu, YanSheng Lu, Qing Yang, "XML conceptual modeling with XUML", *In Proc. of ICSE'06*, ACM Press, pp.973-976, 2006.
- [11] Q. Kong, G. Chen, "On deductive databases with incomplete information", *In ACM TOIS*, pp.354-370, 1995.
- [12] Toshio Kodama, Tosiya L. Kunii, Yoichi Seki, "A New Method for Developing Business Applications: The Cellular Data System", *In Proc of CW'06*, pp. 65-74, IEEE Computer Society Press.
- [13] Tosiya L. Kunii, "What's Wrong with Wrapper Approaches in Modeling Information System Integration and Interoperability? ", *In Proc of DNIS2003*, pp. 86-96, Lecture Notes in Computer Science, Springer-Verlag.