

# みんなくりポジトリ

国立民族学博物館 学術情報リポジトリ National Museum of Ethnology

## Concept and Implementation

メタデータ	言語: eng 出版者: 公開日: 2009-04-28 キーワード (Ja): キーワード (En): 作成者: 高橋, 順一, 洪, 政国 メールアドレス: 所属:
URL	<a href="https://doi.org/10.15021/00002041">https://doi.org/10.15021/00002041</a>

## 3.1 Concept and Implementation

*Junichi Takahashi*

*Jung-Kook Hong*

*IBM Research, Tokyo Research Laboratory*

### 3.1.1 Concept

GDM is a virtual museum on the Internet for museum education. The National Museum of Ethnology, Osaka, Japan, the British Museum, London, UK, Cornell University, Ithaca, NY, USA, and IBM Japan jointly proposed the concept and developed a prototype through international collaboration among the experts of museum, education, and computer science (1). The concept of the GDM is based on studies of state-of-the-art multimedia information services and museum education on the Internet with a special focus on ethnology. The study includes surveying conventional teaching methods (2, 3) and the use of teaching materials in classrooms in the United Kingdom and Japan.

Examples of the teaching methods include *Resource Packs*, which are published by the British Museum and used as teach-the-teacher materials for schoolteachers (4-6). The materials correspond to the national museum education curriculum in the United Kingdom (7), and are intended to facilitate the activities of schoolteachers and students in the real classroom. Another example includes *Hypermedia Ethnographica*, which was published by ethnology researchers in the National Museum of Ethnology as an experimental ethnographica publication using hypermedia (8). The materials are not only intended for browsing content along with a narrative text (similar to a slide show), but also for picking up only the multimedia materials of interest by searching for key terms.

Although there are many museums that provide museum information services on the Internet, the problem is that interactive multimedia technologies as well as network technologies are not well matched with conventional museum education methods. In the ethnology domain, for example, cross-cultural references for a museum collection facilitate understanding of the underlying concepts of the collection by comparing it with other collections around the world. Moreover, collaboration between teacher and students is essential in education activities in museums. In a typical classroom, the teacher creates the teaching materials to prepare students by suggesting what to learn, how to learn, where to learn, and so on. With the materials, the students will learn by

themselves, or by collaborating with each other. The teacher sometimes makes a suggestion to the students for directing their studies more effectively. After working with the materials, the students will present what they learned in the classroom, and the teacher will evaluate the study.

### ***Requirement of GDM Prototype***

These studies imply the following requirements for combination of the interactive multimedia technologies with traditional education methods on the Internet:

#### ***1) Cross-cultural information service***

There is a growing need for interactive access to a large volume of multimedia collections of museums on the Internet. Especially, the information should be provided as a cross-cultural reference; the information of one museum is expected to complement the information of other museums.

#### ***2) Novel use of conventional education materials***

Museum experts, such as curators and museum teachers, expect Internet use to grow and eventually replace conventional paper-based teaching materials, by facilitating interactive hypermedia technologies on the Internet.

#### ***3) Framework and tools for making teaching materials***

Museum curators and schoolteachers need a framework and tools for creating theme-oriented teaching materials using multimedia.

#### ***4) Multimedia publishing***

Researchers in the ethnology field wish to publish their research activities through the Internet as hypermedia documents by using interactive multimedia materials.

### ***Technical Issues***

There are the following technical issues to be solved for creating multimedia information services and education services on the Internet.

#### ***1) Single interface to global museum content***

As the number of museum sites increases on the Internet, users have to consider where the information exists, and how to access the information. It is difficult for a user to access the larger number of museum sites to get museum information of interest because the method for accessing museum content differs among museum sites. A standard access modality is required so that the user can access museum information without considering the differences such as user interface and indexing method among museums.

### *2) Search quality and quantity*

There are many Internet search tools and directory services available, such as Lycos and Yahoo. It is often the case that Internet search results become too large for users to browse. The user will spend time only to find that many of the results are of no interest. Moreover, content is created not only by museum experts, but also by museum educators and school-children in museum education. Quality levels differ depending on the background of content creators. Users need to distinguish these differences in quality level to make use of the content effectively in museum education.

### *3) On-line content editing*

Museum content is not just read-only data, but is also updateable by museum users. Schoolteachers will search and browse the teach-the-teacher materials, and create individual teaching materials for their students. The students will create and update their own collections during their learning activities. For communicating and collaborating with other students and teachers, such personal collections should also be shared among other museum users. An access modality is required for integrated data management for both read and write access to the content (9, 10).

### *4) Interactive user interface*

An interactive user interface is one of the most important factors for implementing a digital museum (2). In such an environment, users will promptly receive feedback for their actions, and the client state will change according to the information. Because the current WWW makes use of stateless HTTP protocol for data transfer, it is difficult to keep the current client status. Users can-not receive feedback to their form submission until the server returns the result of processing the form changing Web page (11). Although the interactivity of interfaces is increasing in many WWW sites that use Java technology (12), the data transfer between server and client is ad hoc, and a standard protocol for bi-directional data communication is required for building an interactive user interface on WWW (13).

The *GDM* is a virtual museum that provides users a single view of multimedia content of multiple museums, and access modality to the content on the Internet. The concept of the *GDM* is characterized by global access to museum content and interactive creation of user collections (14). Figure 1 shows the cyberspace of the *GDM*. The collections of museums are digitized and archived into high quality multimedia materials at each museum around the world. Museum experts, such as museum teachers and curators, create a 'story' by selecting the materials from the archives and inserting them into a narrative about the collections. The story conforms to authorized educational materials, which help users' learning activities by using the entire collections. Users can globally access the stories as well as individual materials according to theme of interest, such as topics, culture, area and era, rather than considering which museum holds the collection and where the collection is located. The stories and the

individual materials are indexed with a common indexing system in order to facilitate global access. The GDM provides individual users a workspace for creating a new user collection. The GDM comprises the user collections, whether the collection is authorized by museum experts or not.

### ***Public museum and personal museum***

There are two kinds of museums in the virtual museum: public museums and personal museums. The difference is related to the level of quality of content. A *public museum* comprises content which is created by museum experts, or which are authorized by the experts. They can-not be updated by the users once they are created. *Personal museums*, on the other hand, are museums with content created by museum users; the users can create their own personal museum by authoring content with resources of the public museum and other private museums. The private museum can be updated by museum experts and other authorized museum users as well as the creator of the private museum.

### ***Museum, exhibit and scenario***

The virtual museum is hierarchically composed of component objects from a structural point of view. The GDM is a single *museum* composed of a set of virtual museums. A virtual museum consists of a set of *exhibits*. The exhibits are arranged in cyberspace according to the context of what the creator of the museum intends to show to the user. In some virtual museums, exhibits are arranged according to geographical property of the exhibits; in others, they are arranged according to chronological property. The user can walk through the exhibits in the order the creator of the museum intends to provide the museum information to the user. The user can also view the exhibits of interest to them. An exhibit consists of a set of *scenarios*. The scenario is a minimum unit of information which has meaningful information to users. The scenario is a composed sequence of *pages*, each of which users can browse one page at a time. The page is composed of multimedia *resources*, including text, image, sound and video resources. The resources may be shared among different pages, and the page may be shared, too, among different scenarios.

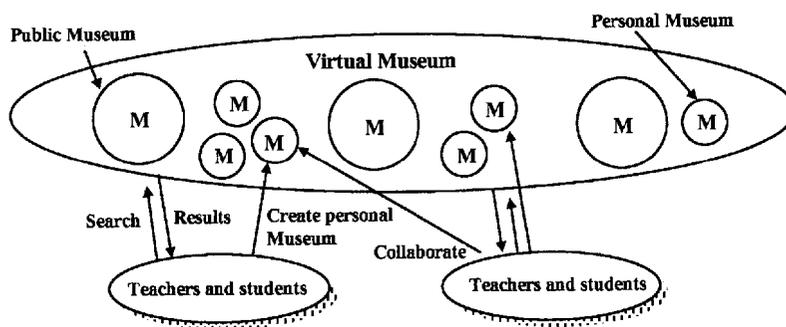
### ***Book and annotations***

A book is a personal collection of museum materials for individual users. The users can create a book by copying a scenario, or some particular part of the scenario, and can add *annotations* to the scenario. The book is distinguished from the original scenario in that the object is created and updated by the user. Individual books are identified even if they are based on the same scenario. The user can create multiple books which have different annotations to the same scenario.

### ***Textbook and notebook***

According to the usage of the collection, the museum is categorized into textbooks and notebooks. A *textbook* is intended for use as a teaching material. According to the

target reader, the textbook is categorized into textbooks for teachers and textbooks for students. *Textbooks for teachers* are created by museum experts, and show what information the museum can provide in the context of education. *Textbooks for students* are created by school-teachers who want to teach students by using materials. These textbooks are searchable by the students and other teachers. *Notebooks* are a working space for school-children. The notebook is a tool of communication between teacher and students and among students.



**Fig.1** Concept of Global Digital Museum

### 3.1.2 Characteristics of Prototype

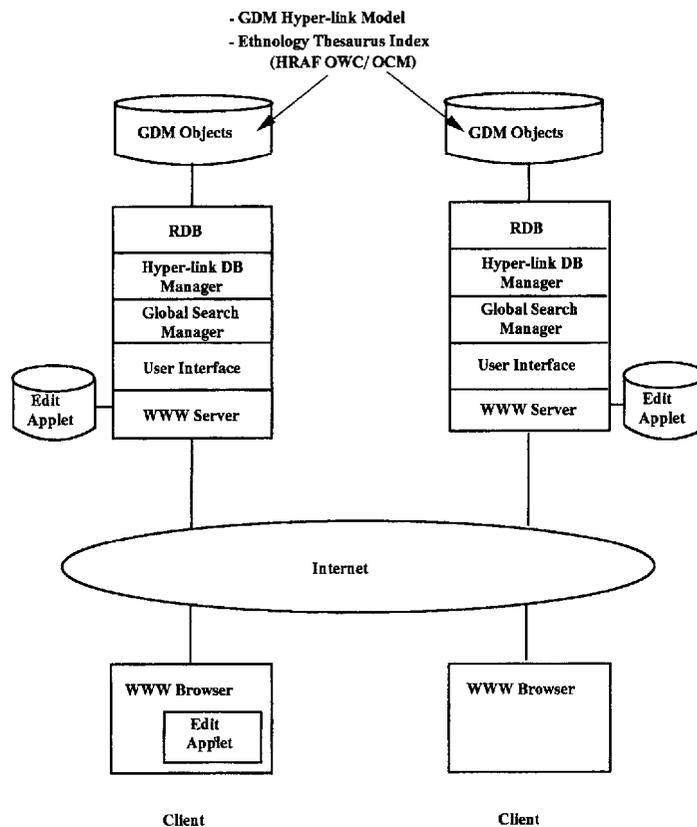
A prototype of the GDM was implemented based on client-server model and distributed server architecture (15-20). A global search function is one of the characteristics of the GDM prototype. Users can search high-quality multimedia museum information around the world just by visiting a single virtual museum. By using an international common indexing scheme in the ethnology domain, with fields such as topics, area and culture, search results can be classified in accordance with the type of object retrieved and quality level of the objects. Comparing these to the conventional search engines on the Internet, which facilitate a centralized index server for homepage searches, there may be URLs that are not accessible in query time. The units of a search are semantic objects, story and resources, etc., rather than an URL. Thus the prototype enables users to search the collections of different museums and compare the differences of the information.

Another characteristic of the prototype is collaborative learning environments. Users can also create and publish their own collections by combining different museum collections. Interactive tools for creating annotations to the user collection facilitate communication among users. The prototype provides the function to create and edit annotations to user-defined areas of images. Such annotations are shared among different users. Users can ask questions of their schoolteachers, museum experts, and

other users, for example. The user collections are identified by user ID and group assigned to individual users to distinguish the information source.

### 3.1.3 System Configuration

The prototype system of the GDM has been implemented on the Internet. Figure 2 shows the system configuration of the GDM. The GDM server consists of a global search manager, hyper-link database manager, user interface, WWW server and the backend DBMS. The museum content is stored in the hyper-link database at each of the museum sites. Users can access the museum content through one of the GDM servers on the Internet by using a WWW browser.



**Fig. 2** System configuration of the GDM

The global search manager manages the request from a client by communicating with other GDM servers. There are multiple GDM servers, as many as the number of

museums participating on a global network. One of the GDM servers takes the role of the meta-server, which has a list of GDM servers on the Internet. All the GDM servers can query the meta-server in client mode to get the list of GDM servers. When a GDM server receives a query from the client, it distributes the query to all the GDM servers, including the server itself, according to the GDM server list in server mode. The global search manager receives the query in server mode, and sends the query to the hyperlink database manager at the local site.

The hyper-link database processes requests from the global search manager for manipulating GDM objects in the site. All the GDM objects are stored in the relational database based on the GDM hyper-link data model (18). When the hyper-link database manager receives the search request from the global search manager, the request passes to the local database and returns the results to the global search manager. For the updating request, it receives the content of the objects from the client, and stores the content to the local database.

The user interface provides users with search, browse, and edit GDM object functions. The user interface is written in Perl CGI programs with JavaScript, and Java applets. Java applets are used for on-line editing for GDM objects, which is executed on the WWW browser at client side. The applet communicates with the global search manager for read and write access of GDM objects via GDM protocol. The user interface formats GDM objects in an HTML document. The WWW server provides the HTML document to a WWW browser.

### 3.1.4 Hyper-Link Database

The virtual museum of the GDM is composed of objects that are physically distributed on the Internet. Each object of the GDM represents information such as multimedia resources and structured information such as a story using the multimedia resources. The GDM provides the function of global search and interactive editing for the objects. We defined a novel hyper-link data model for the manipulation on the objects in the framework of the WWW (18).

All the objects in the GDM are composed based on two constructs: node and link among the nodes. There are two types of object: simple object and compound object. *Simple object* is a primitive object that holds the property of the object. The simple object has generic attributes to describe the application independent properties of an object, such as creator of the object, time stamp when the object is created, and URL of multimedia resource. Simple object also has user attributes that can be used for applications to do with the specific property of the object, such as keywords and other index information used for searches. *Compound object* specifies a link among simple objects. The compound object is associated to a simple object, and holds link

information from the simple object to another simple object. For example, let us consider a parent-child relationship between two simple objects: parent *p* and child *c*. Compound object *x* and *y* are associated to object *p* and *c*, respectively. Object *x* holds a link to simple object *p*, and a link to compound object *y*. The compound object *y* then holds the link to simple object *c*.

The properties of GDM objects are specified by using field values of the object. The fields of the simple object are divided into two categories: system field and application field. *System field* is an application-independent field for the object. *Application field* is an application-dependent field of which semantics is defined according to the object model in a particular application domain.

### ***Fields of simple object***

#### ***OBJECT\_ID:***

All the GDM objects are uniquely identified with an object ID on the GDM system. The object ID is specified with a host name that is qualified with domain name on the Internet and the local object identifier that is a serial number for objects in the domain. For example, the object ID 'xxx.yyy.zzz:0001' means that the object is stored in host 'xxx.yyy.zzz' with local identifier with '0001.'

#### ***RESOURCE\_TYPE:***

This field specifies the type of object that determines the semantics of the simple object. In the museum application model, the following set of values are defined:

*{GDM\_SCENARIO, GDM\_PAGE, GDM\_TEXT,  
GDM\_IMAGE, GDM\_SOUND, GDM\_VIDEO,  
GDM\_BOOK, GDM\_ANNOTATION}*

#### ***URL:***

This field denotes the location of the resource for the object. The resource is digitized multimedia data including text, image, sound and video. The resource may be located at a different site from the location where the simple object is resident.

#### ***CREATOR, CREATED\_DATE, ANALYST, TITLE\_ENGLISH, TITLE\_JAPANESE, DESC\_ENGLISH, DESC\_JAPANESE:***

These fields denote bibliographical data of the object.

#### ***CREATED and LAST\_UPDATE:***

These fields denote timestamp when the object is created and updated, respectively. The format of the timestamp is in the form of —'yyyy-mm-dd-hh.mm.ss.xxxxxx.'

#### ***READ\_USER and WRITE\_USER:***

These fields denote a list of user ID with read and write access to the object,

respectively.

***TYPE\_OF\_ACTION:***

Client interface interprets these values in accessing GDM objects and in format of the object into an HTML document. Client interface also refers to the value of attribute *TYPE\_OF\_ACTION* for additional information to the object. For example, *GDM\_IMAGE\_PORTRAIT* and *GDM\_IMAGE\_LANDSCAPE* are used for representation information of the image to the display.

***OBJECT\_CODE1, OBJECT\_CODE2, ..., OBJECT\_CODE9:***

Index information of the simple object is stored in the application attributes of the object. In the current implementation, up to nine fields, *OBJECT\_CODE1* to *OBJECT\_CODE9*, are available for a simple object. They are used for the set of OCM codes, OWC code, time, list of keywords, place, annotation text and axis of annotation area in the image.

***Fields of compound object***

***GO\_ID\_for\_Compound:***

Object ID of the compound object. The ID is assigned with the same rule as the ID of the simple object.

***GO\_ID\_for\_This\_Node:***

Object ID of the simple object associated to the compound object.

***TYPE\_OF\_LINK:***

This field contains a type of link to another object.

There are four types of links defined:

<i>CHILD</i>	Link to child object
<i>SUCCESSOR</i>	Link to succeeding object
<i>BOOK_ORIGIN</i>	Link to scenario of book
<i>RESOURCE_ORIGIN</i>	Link to resource of annotation

The hierarchical structure of the scenario object is specified by using *CHILD* and *SUCCESSOR* links. The scenario object has a link to the first page object, the first page object has a link to the second page object, the second page object has a link to the third page object and so on. Page links are designated with serial links. Books and annotations are specified by using reference information to the original scenario and resource, respectively. A book object has a *BOOK\_ORIGIN* link to the original scenario, and the *CHILD* link to the first annotation object. The annotation object has the *SUCCESSOR* link to the second annotation object.

***NEXT\_GOID:***

This field contains a link to another object of which type is specified in the field *TYPE\_OF\_LINK*.

***Application model***

The simple object and compound object are generic constructs of our hyper-link data model, and are the modules for constructing higher level objects in an application domain. Application models for a particular domain are made of a set of resource type for simple object, a set of link type for compound object, and the interpretation of these types. We defined one of the museum application models on top of the hyper-link model. In the application model, we have introduced new grouping units called scenario, page, resource, book and annotation objects. These objects are constructed based on the simple object and the compound object on the hyper-link database model, and are represented independently from specific document formats such as HTML.

***Scenario, page and resources***

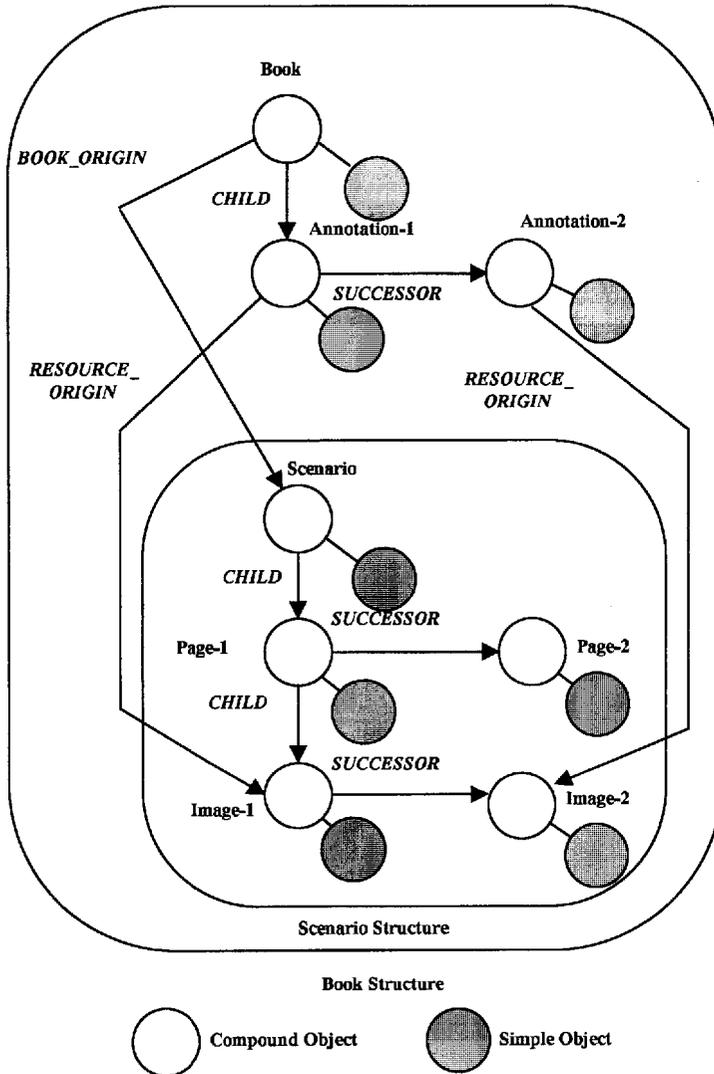
*Scenario* is the minimum unit of information that has meaning to the user. The scenario is a sequence of *pages*, each of which is displayed to users in one page. The page can be composed of *resources*, including text, image, sound and video. This hierarchical structure is represented by using the link field of the compound object.

***Book and annotations***

A book is a personal information area for individual users. A book object is distinguished from a scenario object in that the object can be updated by the user. Initially, a book is created by referring to the original scenario. Annotation is a text object that is associated with a resource object contained in the scenario. The book may have multiple annotations to different areas of an image, as well as annotations to different resources.

Figure 3 shows the example of the structure of the GDM objects. The scenario is defined as a compound object with a *CHILD* link to the first page. The page object is defined as a compound object with a *CHILD* link to the first resource, and a *SUCCESSOR* link to the next page object. The book object has a *BOOK\_ORIGIN* link to the original scenario, and a *CHILD* link to the first annotation. The annotation object has a *RESOURCE\_ORIGIN* link to the resource object to which the annotation is assigned, and a *SUCCESSOR* link to the next annotation object. Annotation text and annotated areas of image resource are stored in the user attribute of the simple object for the annotation.

The GDM object is stored as a record in a relational database. There are two types of tables used for storing the simple object and the compound object, respectively. On manipulating GDM objects, the hyper-link database manager has a function to keep



**Fig. 3** Example of structure of GDM objects

the link consistency between the simple object and the compound object. The compound object does not exist if the simple object is not created, and the simple object is not purged if the compound object is associated to the simple object.

### 3.1.5 Protocol

The GDM object is manipulated in a session by the combination of requests from

client including the commands: *OPEN*, *CLOSE*, *CREATE*, *PURGE*, *GET*, *PUT* and *SEARCH*. GDM protocol specifies the conformation of data transfer between server and client with the GDM commands (15).

The following BNF defined the syntax of the GDM command.

```

<GDM command> ::=
    <OPEN_command> | <CLOSE_command> |
    <CREATE_command> | <PURGE_command> |
    <SEARCH_command> |
    <GET_command> | <PUT_command>

<OPEN_command> ::= OPEN <mode>
<CLOSE_command> ::= CLOSE

<CREATE_command> ::= CREATE <object_type> <host>
<PURGE_command> ::= PURGE <object_type> <object_id>

<SEARCH_command> ::= SEARCH <search_type> <search_conditions>
<GET_command> ::= GET <object_type> <object_id>
<PUT_command> ::= PUT <object_type> <object_id>

<mode> ::= CLIENT | SERVER | META
<object_type> ::= S_OBJECT | C_OBJECT
<host> ::= string
<object_id> ::= string
<search_type> ::= BY_S_OBJECT|BY_C_OBJECT|SIMILAR_Y_S_OBJECT
<conditions> ::= <condition> | <conditions> <condition>
<condition> ::= <field> <value>
<field> ::= string
<value> ::= string

```

### *OPEN and CLOSE*

The session is opened with the OPEN CLIENT command between the client and the server, the session is opened with the OPEN SERVER command between the server, and the server and the session is opened with the OPEN META command between the server and the meta-server. These sessions are closed by only one CLOSE command.

### *CREATE and PURGE*

The new GDM object is created by the CREATE command. The CREATE command takes arguments of object type and host name where the object is stored. The object type is denoted by *S\_OBJECT* and *C\_OBJECT* for the simple object and the compound object, respectively. When the CREATE command is successfully

completed, an object ID for the new object is returned. The PURGE command removes the object that is identified with the object ID.

### *GET and PUT*

The GET command retrieves the object specified by the object ID. For example, GET S\_OBJECT host1:id1 returns field values of the simple object identified by 'host1:id1.' The PUT command transfers field values for the object to the server from the client.

### *SEARCH*

The search command looks for GDM objects with the conditions on the attributes of simple objects. Search conditions are specified by the list of attribute names and value pair of the simple object. Results of the search command are the number of results with a list of object IDs of the compound object that is associated to the matched simple object. There are three types of search: and the type is identified by the keyword: *BY\_S\_OBJECT*, *BY\_C\_OBJECT* and *SIMILAR\_BY\_S\_OBJECT*.

*BY\_S\_OBJECT*: Search compound object with the attributes of the simple object associated to the compound object  
*BY\_C\_OBJECT*: Search compound object with the attributes of the compound object  
*SIMILAR\_BY\_S\_OBJECT*: Search similar compound object according to the attributes of the simple object associated to the compound object

Search conditions are specified by the list of attribute name and value pair. For example, the following command search compound object associated to the simple object with conditions of *OBJECT\_CODE1* is equal to AH1 and *OBJECT\_CODE2* is equal to 123.

```
SEARCH BY_S_OBJECT
RESOURCE_TYPE GDM_SCENARIO
OBJECT_CODE1 AH1
OBJECT_CODE2 123
```

Results of the search command are a set of compound objects each of which has a link to an associated simple object. Search command returns the number of results with a list of the object ID that matched with search conditions. When the keyword *SIMILAR\_BY\_S\_OBJECT* is used, search results are ranked according to the similarity of the attribute specified in the conditions. The score ranges from 0 to 100, and the score is concatenated to the object ID in the list of results.

### 3.1.6 Global Search

Searching for the virtual museum of the GDM is based on distributed index management on the global network (19). In order to provide the user transparent access to the multiple servers, the search mechanism employs server-client interaction to the hyper-link databases at each of the local servers. When a user submits a query to the GDM, the query is distributed to all the GDM servers. Each server processes the query locally, and returns the results to the original server in which the results are merged into a results list.

The server consists of a hyperlink database manager, the global search manager and the server loop. The server loop accepts requests from users, parses the command line, dispatches to the hyperlink database or the global search manager, and gets a reply from them. The hyperlink database is responsible for maintaining the hyperlink data structure in the relational database, and it has a function to search for the hyperlink database. The global search manager has the responsibility for the remote search access, as it has the list of remote servers from the meta-server. The server is the forked process where the client request is received to start. Each client for the session has the process. If the session is completed or abnormally finished, the process is killed by the system.

In the global search, using a common indexing system is one of the key aspects with respect to the relevance of the search. We chose a code system called HRAF (Human Relation Area Files) for the global index scheme for the content of the GDM. HRAF is an authorized standard thesaurus used in the field of ethnology (13). The code system consists of two types of codes: OWC (Outline of World Culture) and OCM (Outline of Cultural Materials) (21, 22). OWC code classifies geographical areas of human culture. OCM code classifies all of the human activities in the real world. The OCM code set is defined as hierarchically in three levels of categories, and denoted in five-digit, three-digit, and two-digit numbers. For example, code 77 denotes "religious beliefs," code 776 for "spirits and gods," and code 776.01 for "conception of supernatural beings of a higher order than disembodied souls."

Search results are ordered according to the relevance of the matching object to the query in terms of hierarchy of OCM category based on how much the domain of the concept is overlapped in terms of the hierarchy of OCM categories. When query  $Q$  includes a set of OCM codes  $\{q_1, q_2, \dots, q_n\}$  in the query conditions, and the GDM object  $D$  includes a set of OCM codes  $\{d_1, d_2, \dots, d_m\}$ , the similarity is calculated for each of the matching objects by the following scoring function:

$$score(Q,D) = \sum_{i=1}^n \sum_{j=1}^m \frac{f(q_i, d_j)}{nm}$$

where

$$f(x,y)= \begin{cases} 100 & \text{if } x \text{ and } y \text{ match} \\ a & \text{if first three digits of } x \text{ and } y \text{ match} \\ b & \text{if two digits of } x \text{ and } y \text{ match} \\ 0 & \text{otherwise} \end{cases}$$

The HRAF code can also be used for creating a scenario structure without specifying particular resources used in the scenario. For example, suppose a creator of a scenario needs some image of a ball object to explain a story of a ball game, but there is no collection of balls in the museum. The creator can just specify the OCM codes as a placeholder, which corresponds to the concept of the ball game, instead of specifying the individual ball object. When the scenario object is searched by the user, the placeholder is automatically filled with the most relevant resources selected from the materials according to the OCM matching. The creator does not need to know where the resources are located on the Internet.

### 3.1.7 On-line Annotation Editing

There are four applets for the operation of the annotation for creating, editing, browsing, and erasing annotations.

*Append Annotation Applet (AnnoAdd class):*

The applet shows the image so that the user can define the rectangle areas of the image resource for the annotation and input text that explain the content of the image. The applet holds the annotation data for the relative axis of the defined rectangle and the annotation text. At the next step the applet requests the server to create the new compound object and the new simple object for the new annotation. The annotation data is filled into the field of the simple object. Finally, it looks for the last child compound object of the book, and updates the child link to the new compound object.

*Edit Annotation Applet (AnnoEdt class):*

The applet receives the list of annotation data that are axis and text from the server and displays the areas on the image resource. The user can choose one of the annotations by selecting one of the areas shown on the image, and then edit the text data associated to the area. The applet loads the simple object for the annotation and updates the value with the new text data from the user.

***Browse Annotation Applet (AnnoBrw class):***

The applet receives the list of annotation data that are axis and text from the server displays them on the image resource. The applet shows the annotation text associated to the image when the mouse cursor is focused on the annotation area on the image.

***Delete Annotation Applet (AnnoClr class):***

The applet receives the list of annotation data that are axis and text from the server and displays them on the image resource. The user can choose one of the annotations and deletes the compound object and the simple object for it. The applet searches the predecessor compound object for the deleted annotation and changes the successor links of the predecessor from the compound object of the deleted annotation to the successor compound object of the deleted annotations.

All of the applets are executed with parameters for the access to the GDM server, information of the resource to which the annotation is associated and scenario in which the resource is contained. Because of the restriction of JAVA virtual machine that the applet can only access the resources on the local server, the applet is downloaded from the server where the resource is located. The following procedure shows how the applets are invoked.

- 1) The client CGI program generates an HTML document that includes the applet for annotation.
- 2) When the HTML document is displayed, the applet is downloaded from the host where the resource is resident through HTTP.
- 3) The applet accesses an image file specified in the resource parameter from the host.
- 4) The applet accesses the GDM object through the GDM protocol.

### **3.1.8 User Interface**

The GDM client provides user interface for global search, browsing, and editing the content including scenarios and books on a Web browser. These pages are displayed as HTML documents that are served by a Web server through HTTP protocol.

***Global Search***

Figure 4 shows the interface of global search in the GDM. The search function allows a user to specify a number of search criteria, such as topics, time, area, keywords, and a list of exhibits is presented that meet those criteria. When a user chooses an exhibit in the list, a window is opened to display the first page of the exhibit, and the user can go through the exhibit by forwarding the page. In browsing the exhibit, the user can place a piece of the resource of interest into a 'basket' for private collection.

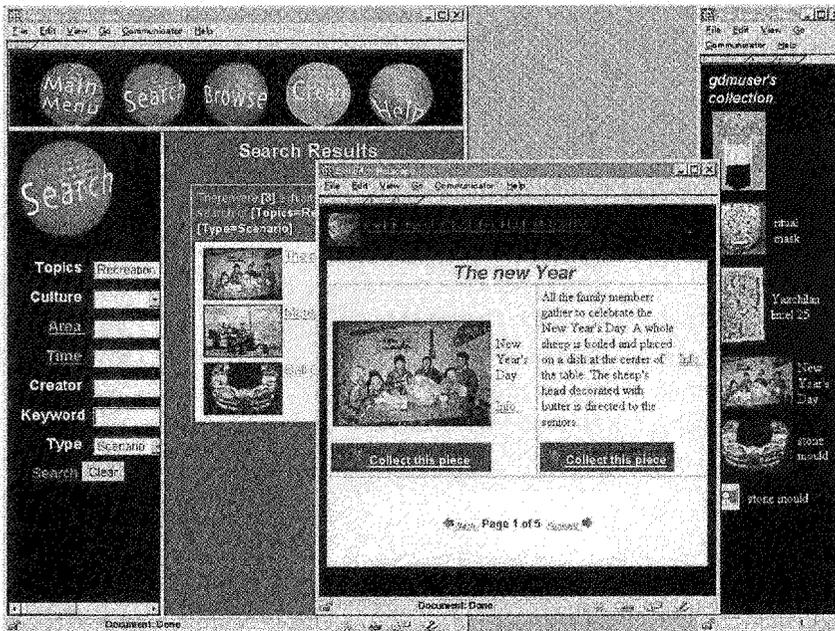


Fig. 4 Example of global search

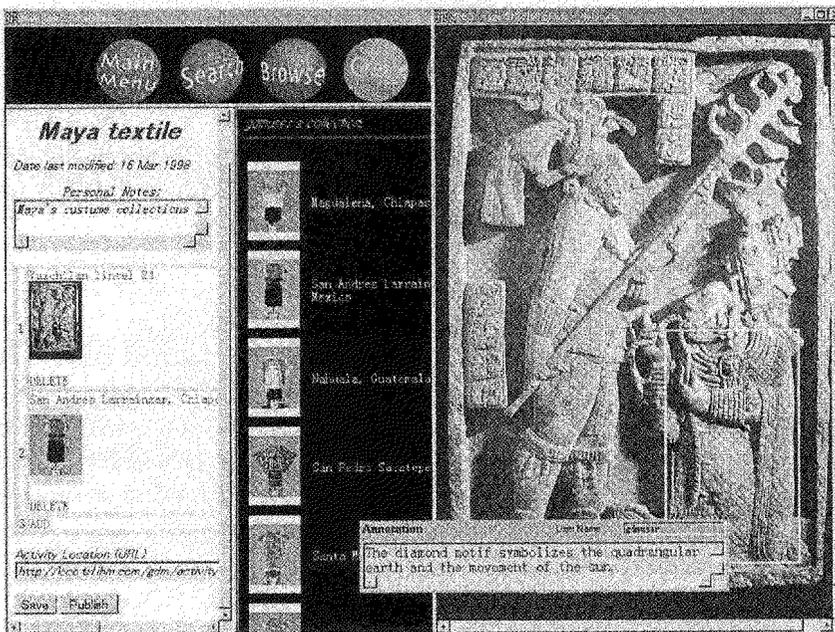


Fig. 5 Example of creation of an exhibit and annotations

### ***Book Creation and Editing***

Figure 5 shows how the user can create a new exhibit by combining the pieces in the basket. By clicking on an item in the basket, the user can continue to add items from the basket to the newly created exhibit. When the user is viewing an image of the exhibit in a separate window, the user can select an area of the image by dragging the mouse and drawing a box around it. Corresponding to the area, there is a text field for associating an annotation to the area, and for editing the annotation text. Java applets are used for adding, editing, browsing and deleting annotations. The applet communicates with the GDM server to create the new simple object for the annotation, and to update the link to the new object. During the user operation, the applet holds the data of the book object and the status of the edit operation on a Web browser, and users can perform the operation without accessing the server for each of the user operations. Publishing the exhibit puts it into the GDM database, making it available to all users by global search of the exhibits.

### **3.1.9 Content Input**

There are two types of tools provided for inputting the content to the GDM database: a content upload tool and database maintenance tool.

#### ***Content upload tool***

A set of GDM objects is loaded to the database from the files of simple objects and compound objects in CSV format. The tool executes DB2 upload utility (IBM95) to import the records of simple objects and compound objects into the tables, respectively. Among the attributes of GDM objects, Object ID and resource URLs depend on hostnames where the object is resident. Before executing the import, the upload tool automatically adds the host name to these attributes so the upload file is created independently from the host where the content is uploaded.

#### ***Database maintenance tool***

GDM objects can also be maintained by using a database maintenance tool. The tool allows database administration users to search and browse inputted content, and to verify the content interactively on a Web browser. The tool is implemented by a Perl CGI program, which issues a series of GDM commands for updating GDM objects. A GDM object to be maintained is retrieved by specifying either object ID or SEARCH command. The field values of the object are returned from a GDM server, and are displayed in an HTML form.

Inputting the content into the GDM database is proceeded in the following steps:

#### 1) Filling in content input form

The content creator who wants to create a new scenario prepares the structure of the

scenario and multimedia resources referred to in the scenario. We defined a standardized form for content input that is commonly used among museums. The form includes the specification of the GDM objects, which composed a scenario, properties of these objects, and resources that are referred to in the scenario.

#### 2) Preparing multimedia resources

Multimedia resources are digitized into separate files from the content input form. Each of the resources is associated to a URL, and is located to the file directory on the GDM server.

#### 3) Converting content forms to import file format

The text in content form is converted into the field data that correspond to the file format of the GDM database. The GDM supports importing GDM objects from a CSV file that is comma-delimited ASCII text file. The content form contains only the application attributes of the GDM object. In this step, the system attributes of the GDM object are added to the data in the content form.

#### 4) Importing data to the GDM database

GDM objects are loaded by using import utility in batch processing.

#### 5) Verify the content

Verify the imported data by browsing the content on the Web browser.

### 3.1.10 Discussion

We have built a prototype of the GDM based on client-server and distributed server architecture. The architecture is a feasible model for managing distributed databases on the Internet. One of such systems based on the distributed server architecture is DIENST (Distributed Interactive Extensible Network Server for Techreports): an operational protocol and server that employs the architecture for a distributed search of digitized technical reports on the WWW (23-25). The system architecture and design are intended for well-organized document searches, and not applicable to the museum applications in the perspective of heterogeneous multimedia data modeling, protocol, interactivity and performance. We studied the architecture and design of DIENST with respect to applying it to museum application. Although the basic idea of distributed search architecture of DIENST is also applicable to museum application, we found that the total system architecture and design are intended for well-organized document searches, and concluded that it is not applicable to our application in the perspective of document model, protocol, interaction and performance.

#### *Data model*

DIENST is based on the document model RFC-1807 (26). Although the format is

standard for well-structured documents such as technical reports, it is difficult to present inter-document structure, especially for content with multimedia collections. DIENST indeed supports intra-document structure; logical document structure such as pages of a document, and physical structure, such as file format of archived documents (25). The documents may have multiple representations (file formats such as plain text and Postscript format) and can be decomposed physically (by page) and logically (by structural aspects). Although the underlying bibliographical format is general for document databases such as technical reports, it lacks flexibility for structuring museum content with multimedia collections. Moreover, the document structure is explicitly presented as the directory structure of the file system. For example, the following protocol requests a GIF image of the first page of the document publisher:doc\_no.:

```
/DIENST/Repository/2.0/Page/publisher%2fdoc_no/inline/1
```

As the example shows, the semantics of the structure is encoded into the server program, and the server can not access the repository if the structure is changed. In the GDM, the location of the multimedia resource is not encoded in the model itself. Instead, the URL of the multimedia resources is used, of which encoding is independent from the document model.

The document ID of DIENST requires the document handle (27), which includes identification of the publisher of the document. Assuming the content may possibly be created by all the Internet users, it is not realistic to assign a publisher name for all the users of the Internet. In the GDM, the definition of an object ID is based on host name and local ID, uniquely generated by the system, and there is no difference in the generation of IDs among the creators of the object. Our object ID scheme also makes it easy to refer to remote site resources from an object of one site, thereby globalizing museum resources.

### ***Protocol***

The protocol of DIENST is by-service protocol (28), and it differs depending on what service the system is providing. In the DIENST architecture, there are four classes of services: index service, repository service, meta-server service and user interface service (28). The index service of DIENST provides a parametric search of a document collection based on the bibliographical data of documents. Meta-service provides a directory of all servers in a network domain. Repository service provides the content of digital documents based on the document model. The GDM uses a single protocol for the communications between client-server, server-server, and server-meta-server. All the other services of DIENST are intended for interaction between servers, except the user interface service, which is intended for communication with browsers used by humans, such as the query form. In the GDM, such an application level service is not provided with the protocol. User interface with

the GDM mediates this layer, in which GDM protocol is used for communication with the GDM server.

### ***Interactivity***

The user interface server of DIENST is CGI-based, and the user must wait for the reply from the server which is shown in another Web page. A Java applet in the GDM makes the user interface interactive because the applet can handle all the user actions on a Web browser client once the applet is downloaded to the client. We introduced the annotation object so a user can make a comment on any area in an image resource. Our choice of this model of annotation came from one of the museum requirements on making annotations on an image. Because an image has various meanings depending on what object in the image the user is interested in, one person may be interested in people in a photograph, while others may be interested in the background perspective of the photo.

Another interactivity provided by the GDM is an online database update. Loading of the DIENST database is batch oriented, and the index is rebuilt for the new entry. On the contrary, interactive maintenance of the GDM object is made available in the GDM by manipulating the object with GDM protocol. Such an interactive editing of a GDM object is essential for supporting book and annotation functions.

### ***Performance***

On the Internet, packet delivery is unreliable and there are no guarantees with transfer time like QoS (Quality of Service). With the distributed search architecture of DIENST, time-out occurs when a server can-not get a reply from other servers on the Internet. Current implementation of DIENST includes both the functions of a fault tolerance server, and a regional index server in which each DIENST server in a region accommodates all the index information of the servers in a region (24). Similar to the idea of a region index server, Chankhunthod, et al., proposed hierarchical arrangement of caches for mirroring the topology of a wide-area internetwork to distribute load away from a server and to protect the network from erroneous clients (29). The GDM employs object level caching for performance improvement on the Internet. The caching method is more effective when an object is accessed frequently in some short period. One such example is the use of teaching materials in museum education. Once a schoolteacher has accessed an object, materials for example, the students can follow the object on the cache rather than accessing the server individually.

### **3.1.11 Conclusion**

The Global Digital Museum offers a promising approach for developing the new paradigm of museum education. The prototype of GDM provides a virtual classroom on the Internet in which users can search and access museum information as the

content of a single federated museum. The users do not need to know or specify names of interesting files and sites of servers, and it is not necessary for them to search each file or site in turn. Instead, they indicate their subject interests only to get directly the museum information in which they are interested. In the virtually federated digital museums, schoolchildren and teachers, and the general public can collaborate in creating unique content with known and unknown users beyond the Internet. For developing the concept of the Global Digital Museum, there remain several technical issues, as well as intellectual property issues. We need to standardize global indexing systems and searching mechanisms, improve tools of collaborative creation works, especially annotation tools, improve user interfaces, and standardize the global digital museum education curriculum.

### Acknowledgements

The authors wish to thank to T. Nakagawa, T. Sasaki, A. Suzumura, and M. Ujitani, of the National Museum of Ethnology, Osaka, Japan, for discussions on the concept of the GDM and technical support for developing the GDM prototype. The author is grateful to Dr. H. C. Koh, East Rock Institute, New Haven, CT, who advised on using HRAF in indexing the content of the GDM.

### Notes

- 1) J. Takahashi, T. Kushida, J-K. Hong, R. Rieger, W. Martin, G. Gay, S. Sugita, Y. Kurita, J. Reeve, R. Loverance, "Global Digital Museum: Multimedia Information Access and Creation on the Internet," *Proceedings of the ACM Digital Libraries '98*, Jun.24-24, Pittsburgh PA, 1998, pp.253-257.
- 2) Hooper-Greenbil, E., ed., *The Educational Role of the Museum*. Routledge, London, 1994.
- 3) Hooper-Greenbil, E., *Museum and Gallery Education*. Leicester University Press, London, 1994.
- 4) Bateman, P., *Aztec Resource Pack*, British Museum Education Service, London, 1994.
- 5) Bateman, P., *Mexican Gallery Resource Pack*, British Museum Education Service, London, 1994.
- 6) McEwan, C., *Ancient Mexico in the British Museum*, British Museum Education Service, London, 1994.
- 7) *National Curriculum History: How the British Museum can help you*, British Museum Press, London, 1994.
- 8) Yuki and Friends, eds., *MONGOLICA ETHNOGRAPHICA*, National Museum of Ethnology, Osaka, 1996. <http://tenger.minpaku.ac.jp> (in japanese)
- 9) Streitz, N. et al., "SEPIA: A Cooperative Hypermedia Authoring Environment,"

- Proceedings of the ACM Conference on Hypertext, Milan, Italy, 1992, pp. 11-22.
- 10) Phelps, T. A. and R. Wilensky, "Toward Active, Extensible, Networked Documents: Multivalent Architecture and Applications," *Proceedings of the 1st ACM international Conference on Digital Libraries*, March 20-23, 1996, Bethesda, MD, pp. 100-108.
  - 11) Maly, K., et al., "Interactive Distance Learning over Intranets," *IEEE Internet Computing*, Vol. 1, No. 1, 1997, pp.60-71.
  - 12) Yourdon, E., "Java, the Web, and Software Development," *IEEE Computer*, Vol.29, No. 8, 1996, pp.25-30.
  - 13) Duan, N. N., "Distributed Database Access in a Corporate Environment Using Java," *Proceedings of the Fifth International World Wide Web Conference*, May 6-10, 1996, Paris, France.
  - 14) J. Takahashi et al., "Global Digital Museum (1) Concept," Information Processing Society of Japan Annual Convention, Osaka, Japan, Sep. 1996.
  - 15) J-K. Hong, J. Takahashi, T. Kushida, M. Kusaba, S. Sugita, Y. Kurita, T. Fujii, T. Morita, Y. Konagaya, Y. Yasugi, T. Asakura, M. Kubo, S. Fukuoka, K. Fukukawa, J. Reeve, R. Loverance, G. Gay, R. Rieger,, "Global Digital Museum (GDM): Prototyping a Virtual Museum on the Internet for Museum Education," *IBM Research Report*, RT0214, July 1997, IBM Corporation.
  - 16) T. Kushida, J. Takahashi, M. Kusaba, J-K. Hong, S. Sugita, "Global Digital Museum (2) Network Architecture," Information Processing Society of Japan Annual Convention, Osaka, Japan, Sep. 1996.
  - 17) M. Kusaba, J. Takahashi, T. Kushida, J-K. Hong, S. Sugita, "Global Digital Museum (3) Application to Museum Education," Information Processing Society of Japan Annual Convention, Osaka, Japan, Sep. 1996.
  - 18) J. Takahashi, T. Kushida, J-K. Hong, S. Sugita, IPSJ Annual Convention, "Global Digital Museum (4) Hyper-link Data Model," Information Processing Society of Japan , Fukuoka, Japan, Sep. 1997.
  - 19) T. Kushida, J. Takahashi, J-K. Hong, S. Sugita, "Global Digital Museum (5) Global Search," Information Processing Society of Japan Annual Convention, Fukuoka, Japan, Sep. 1997.
  - 20) J. Takahashi, J-K. Hong, S. Sugita, "Global Digital Museum (6) Prototype Evaluation," Information Processing Society of Japan, Annual Convention, Nagoya, Japan, Oct. 1998.
  - 21) *Outline of Cultural Materials*, Human Relations Area Files, Inc., 1982.
  - 22) *Outline of World Cultures*, Human Relations Area Files, Inc., 1983.
  - 23) Davis, J. R. and C. Lagoze, "A protocol and server for a distributed digital technical report library," *Cornell University Technical Report*, TR94-1418, 1994.
  - 24) Lagoze, C., E. Shaw, J. R. Davis and D. B. Krafft, "DIENST: Implementation Reference Manual," *Cornell University Technical Report*, TR95-1514, 1995.
  - 25) Davis, J. R. and C. Lagoze, "The Networked Computer Science Technical Report Library," *Cornell University Technical Report*, TR96-1595, 1996.
  - 26) Lasher, R. and D. Cohen, "A format for Bibliographic Records," 1995.

- <http://andrew2.andrew.cmu.edu/rfc/rfc1807.html>
- 27) "The Handle System Version 3.0 An Overview," Corporation for National Research Initiatives, 1996. <http://www.handle.net/docs/overview.html>
  - 28) Lagoze, C. and D. Ely, "Implementation Issues in an Open Architectural Framework for Digital Object Services," *Cornell University Technical Report*, TR95-1540, 1995.
  - 29) Chankhunthod, A., et al., "Hierarchical Internet Object Cache," *Technical Report* 95-611, Computer Science Department, University of Southern California, 1995.