In Multimedia Information Storage and Management, S.M. Chung ed., Kluwer Academic Publishers, 1996.

# 1

# THE OBJECT-ORIENTED DEVELOPMENT OF MULTIMEDIA INFORMATION SYSTEMS

**Gerald C. Gannod and Betty H. C. Cheng**

*Department of Computer Science*
*Michigan State University*
*A714 Wells Hall*
*East Lansing, Michigan 48824-1027*

## ABSTRACT

As object-oriented analysis and design techniques become more mature, the promises of object-orientation become more realizable. The advent of multimedia hardware and software technology give rise to a need for generalized multimedia data access methods. This chapter discusses the advantages of using object-oriented analysis and design techniques to design and develop a general client-server architecture for browsing, retrieving, and analyzing distributed multimedia data. The object-oriented development techniques are illustrated in the description of ENFORMS, an object-oriented distributed multimedia information system developed at Michigan State University.

## 1 INTRODUCTION

With the widespread use of web browsers, internet tools, and multimedia-equipped computers, there is an increasing demand for quick access to a wide variety of information. In almost every application area and domain, the need for electronic information is present. From industrial organizations to the K-12 arenas, there is an appreciation for all that is offered by the "information superhighway."

Correspondingly, there is clearly an abundant supply of information to be perused by users. In fact, many areas, such as scientific research addressing global change, continue to generate large quantities of information for analysis and understanding. However, the volume, distributed nature, and diversity of this information prohibits convenient access by many potential users. The users are forced to adapt to new browsing and presentation mediums with every new data set (collection of related data) or information domain.

For those information systems that have a specific objective, that is, a specific problem to be studied, a unified framework that is adaptable to new data sets and new analysis utilities will best serve potential users. Because the users may have varied backgrounds with different levels of expertise and the data as well as the tools for accessing and integrating the data may change over time, it is advantageous to build systems that support the separate development of the browsing and data management components, while providing an integrated environment to the users. Furthermore, for maintenance purposes, both components should be easily modifiable according to the data and user needs.

One technique that is particularly amenable to such demands placed on large scale information systems is object-orientation. Object-oriented analysis and design techniques are particularly suitable for modeling information-intensive systems, given their heavy emphasis on entity-relationships. Object-orientation also promotes reuse and facilitates maintenance. The support for sophisticated abstraction techniques provided by object-oriented analysis and design can be used to model systems in terms of specific domains, rather than be constrained by implementation details imposed by the constructs of a specific programming language. Furthermore, because multimedia information systems have the additional property of supporting the integration of different types of data access and manipulation capabilities, object-orientation enables the developer to simplify the modeling, and thereby the implementation of the multimedia capabilities.

This chapter describes how object-oriented analysis and design techniques can be applied to multimedia information systems. The object-oriented development approach has been applied to the development of ENFORMS [1, 2, 3, 4], a prototype system consisting of an integrated collection of software tools that allows a user to manipulate disparate data sets through a graphical user interface (GUI). Figure 1 shows a high-level view of the organization, where the rectangles represent potential participating sites, and the filled circles represent users.

The system, ENFORMS, currently has three different instantiations of data and tools for three separate projects involving environmental information. The same basic architecture has been used for all three projects. In fact, for two of the projects, a single system was used, only the run-time configuration of the

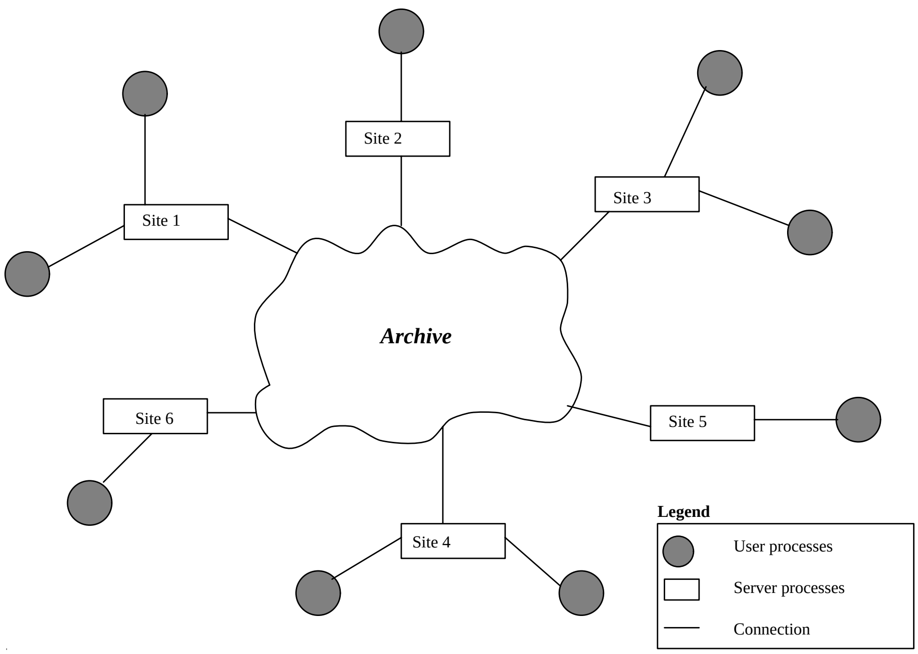



**Figure 1** A high-level view of the architecture.

system changed. ENFORMS has been developed using object-oriented techniques in order to make it extensible, flexible, and maintainable. Using ENFORMS enables decision-makers (who do not have to be computer experts) to better understand and, perhaps, learn about new environmental issues relevant to numerous regions, which potentially may have a global impact.

The remainder of this chapter is organized as follows. Section 2 gives background information in the area of object-oriented development techniques, digital libraries, and multimedia systems. General issues relevant to the architecture of multimedia systems are discussed in Section 3; this section also describes the impact of these issues on the development of ENFORMS. Descriptions of specific ENFORMS applications are briefly overviewed in Section 4. Finally, concluding remarks and future directions for object-oriented frameworks for multimedia systems are discussed in Section 5.

# 2 BACKGROUND

This section gives a brief overview of object-oriented analysis and design techniques. Multimedia systems and digital libraries are defined and important characteristics are reviewed.

## 2.1 Object-Oriented Analysis and Design

Object-oriented development techniques encompass analysis, design, and implementation. As with traditional development strategies, such as those offered by structured analysis and design [5], analysis addresses the identification of the functional and non-functional requirements of the system, design typically produces a family of solutions for the requirements, and the implementation defines a specific solution obtained by refinement and adding details to the design. The analysis and design stages are often facilitated by the use of graphical modeling techniques, such as data flow diagrams, state transition diagrams, and entity-relationship diagrams.

As systems continue to grow in size and complexity, alternative development techniques that offer a means for addressing the challenges imposed by the new systems are gaining increasing attention. One such alternative is *object-orientation* [6, 7, 8]. The most significant difference between object-oriented development (OOD) and structured analysis and design (SAD) is the point of focus of the respective development techniques. In SAD, the focal point is the procedure or function. The analysis stage centers around high-level descriptions of the functionality of the system. During the design phase, the refinement and decomposition of the high-level descriptions of functions yields more detailed descriptions of functions and procedures that incorporate implementation details. Finally, during the implementation phase, functions and procedures identified during design are decomposed into more specific functions. In contrast, OOD focuses on objects of the system, where objects refer to real-world entities (e.g., machines, person) as identified and defined by the developer. For each object, a set of specific operations for manipulating or accessing the object are defined. An object can only be manipulated by these operations. The implementation of these objects and their operations are "hidden" from users of the objects, thus facilitating the development and maintenance processes. Furthermore, concepts such as inheritance promote reuse in object-oriented systems.

Systems that involve a great deal of data with well-defined operations are particularly suitable for object-orientation [6, 7, 8]. Examples include database applications and information systems. With information-intensive systems, attention is focused on the data and its structure. Accordingly, the operations and the complexity of the logic of these systems is typically straightforward, thus making these systems particularly suitable for object-oriented development.

**Object-Oriented Modeling Technique.** Currently, there exists several approaches for performing object-oriented analysis and design [6, 7, 8]. Most of these approaches involve some type of graphical modeling that facilitates communication between the customer and the developer by offering a visual (simplified) representation of the system prior to its implementation. The *Object Modeling Technique* (OMT) [6] is commonly used in industry and in academia. The advantages to OMT include its support for analysis and design, and the simplicity in notation. OMT comprises three complementary models. The *object model* describes the static, structural data aspects of the system. The object model captures the objects of the system and the relationships between the objects. Succinctly, this model describes "what" the system is. The *dynamic model* depicts the temporal and behavioral aspects of the system. Simply, this model describes "when" system activities occur. Finally, the functional model gives the transformations of the data within the system. In essence, the functional model describes the services provided by the system. Respectively, entity-relationship diagrams, state transition diagrams, and data flow diagrams are used to represent the object, dynamic, and functional models. Clearly, the notation for each model is simple to construct and understand. By having complementary models, each model is only used to capture a specific perspective of the system. With recent work [9], rigorous analysis of each of the models is possible, thus enabling consistency and completeness checks at the model level prior to the implementation phase.

## 2.2 Multimedia and Digital Libraries

Many terms are specific to the areas of multimedia and digital libraries. For clarification purposes, we define terms relevant to both as well as terminology used in the description of the ENFORMS system.

**Multimedia.** A multimedia system is a computer-based system that integrates independent information and facilitates the control of that information via production, manipulation, storage, and communication services [10]. Multimedia systems can be characterized using the terms *strict* and *broad* multimedia, where the differentiating characteristic is the existence of continuous media (strict) [10]. The processing of individual images and text and the absence of continuous a medium characterize the *broad* definition of multimedia [10]. For the sake of genericity, broad definition of multimedia is assumed throughout this chapter although this presumption does not preclude the existence of continuous media in the development of information systems.

**Digital Libraries.** Gladney et al. define a digital library as [11]:

> *...an assemblage of digital computing, storage, and communications machinery together with the content and software needed to reproduce, emulate, and extend the services provided by conventional libraries based on paper and other material means of collecting, cataloging, finding, and disseminating information.*

For much of the discussion in this chapter, the definition of digital libraries and the requirements on the functionality of digital libraries provide the context from which the goals and motivations used in the approach advocated in this chapter for developing information systems are based. The concepts related to the implementation of multimedia, especially delivery of media, are however, the technologies that allow the goals to be satisfied.

## 3 ARCHITECTURE

The use of distributed information systems has seen an increase due to an improvement in hardware and software technology. Many issues impact the longevity of information systems, including adaptability to changes in the computing environment and configurability to changing data formats. Another important issue to consider in an information system is integration. When addressing integration, one must consider the degree to which integration is supported, and the means that a system uses to implement integration.

This section examines the issues related to the development of information systems, and the impacts those issues have had on an information system developed at Michigan State University called ENFORMS. For the remainder of this chapter we use the convention that the term "ENFORMS I" refers to features and concepts related to the early version of ENFORMS, the term "ENFORMS II" refers to the features and concepts related to the subsequent version of ENFORMS, and the term "ENFORMS" refers to features and concepts related to both versions.

## 3.1 Goals and Motivation

As described in Section 2, object-oriented analysis and design has become a widely used technology that facilitates reuse, extensibility, and maintainability. The ability to cope with change is often a defining characteristic of whether or not a software system survives changing computing technology and environments. Information systems are not immune to dynamic computing environments; indeed, these systems have flourished due to the improvements in network technology.

To assume that a software system will always be a viable product that requires no modification is clearly not realistic. With the existence of changing requirements and environments, the ability to satisfy the modifications due to those changes will often indicate whether or not a software system will have a long lifetime. Three main properties that facilitate coping with changes are *adaptation*, *integration*, and *configuration*.

*Adaptation* is the degree to which the architecture and design of a software system can facilitate modifications. In addition to the *maintainability* of software code, the adaptability of a system is also impacted by the products of good software engineering, including the existence of a set of requirements specifications, design specifications, and testing specifications that have been appropriately modified to reflect modifications in software code.

*Integration* refers to the degree to which an information system can incorporate new sets of data that may be different in format and nature to any data currently residing in the system. In addition to being able to incorporate this data, integration is impacted by the ability to manage these new resources in such a way that the core data is hidden from end users, but where the access methods are well-defined. Media integration is often associated with the ability to construct new multimedia objects (i.e., authoring). To this end, an information system with a high-degree of integration should provide the facilities for associating objects and their access methods while seamlessly incorporating these new objects into the system.

Finally, *configuration* is the degree to which an information system can cope with different domains of data. Configuration can cover many areas related to information systems including browsing, access, and retrieval methods. That is, the ability to tailor a system at run-time so that the browsing, access, and retrieval methods facilitate activities natural to the domain of interest makes a system highly configurable.

A *data custodian* is a person that is responsible for activities such as populating a library (also referred to as an *archive*) and creating catalogs for the data contained in the library. A *data end user* is any person that is interested in accessing the library. In general, the concepts of a *resource manager* and an *application enabler* provide a basis from which data custodians and data end users can choose modules that meet their needs and preferences [11]. A *resource* is a set of persistent data and the programs (or manipulators) that allow access to the data. When the only access allowed to data is a well-defined set of programs we refer to that set of programs as a resource manager [11]. An *application enabler* is a set of programs that provide services for editing, filtering, and formatting data such that the act of configuring a system or document is made easy [11].

The remainder of this section describes a system, called ENFORMS, that was designed and developed with the intention of maximizing the three criteria of adaptation, integration, and configuration through the use of concepts consistent with the notions of resource managers and application enablers.

## 3.2 Analysis Guidelines

Digital libraries generally share the same underlying constraint; they are based on the search, selection, and delivery of information that has been aggregated into some form that can be distributed easily. The means by which that data is disseminated depends on the nature of the data as well as the methods that have been used to catalog (classify) the data.

An *item* can be a single data entity or a collection of data that has been aggregated into a semantically related group or *resource*. As a group, an item is known as an *aggregate item*. Single data entities are referred to as *atomic items*. The set of programs that define the semantics for the item are referred to as the *item manipulators*. An *Item classification* scheme, or *search model* is the method that is used to catalog or *classify* items. The following guidelines provide a systematic method for developing information systems that are *item* and *item classification* centered. The steps are:

1. Analyze characteristic data
2. Choose classification model
3. Analyze the storage and ownership constraints

## *Data Analysis*

The data that is collected for use by an information system can determine the methods that will be used to access the data. Before an information system can be built, representative data to be accessed must be identified and analyzed. Specifically, there are two types of analysis activities:

- Atomic item access
- Aggregate item access

*Atomic item access* refers to the methods that will be used to manipulate atomic data items. For instance, when given an image, specific tools must be defined that allow for the manipulation of that image. *Aggregate item access* refers to the set of methods required to manipulate aggregate items.

Simple modeling of the items located in an information system archive serves as the basis upon which the remainder of the system is built. Decisions about how to browse, search, and retrieve items are all based on the items and their access methods.

Many activities in the browse and search of items involve abstract representations of the actual items. Although network technology has improved in performance, the delivery of actual items can still be a source of delays in response time. An *item descriptor* is a meta-item that describes the characteristics of an item. Item descriptors are smaller in size and avoid unnecessary network traffic. By using item descriptors, delivery of actual items is only performed when a specific manipulation of an item is requested by the user.

## *Search Models*

Another important activity in the analysis of representative data for an information system is the creation of classification paradigms or *search models*. A search model determines how a collection of items (either atomic or aggregate) are classified so that a system can be browsed and items can be retrieved using a particular *search engine*. Examples of classification methods are *hierarchical*, *spatial*, and *temporal*. In addition to facilitating browse and retrieval, the search model determines how new items are added to an archive.

### *Storage and Ownership Constraints*

The constraints of data storage (where the data is stored) and data ownership (who has access to the data) can have an impact on the architecture and topology of an information system. These constraints determine whether or not a system uses a client-server architecture, or whether it is a protected single machine system. The impact of these decisions play major roles in the delivery and access of items.

### *Object-Oriented Analysis*

Analyzing characteristic data, choosing the appropriate classification model, and determining the storage and ownership constraints make a direct exploitation of the advantages of object-oriented analysis and design. The focus of an information system is an item. Just as an object in the implementation of a software system has both data attributes and methods, items have attributes (the data itself) and methods (the manipulators).

By separating the data from a specific search model, support for cross-referencing across search models is facilitated. This ability is made possible by the treatment of items as autonomous data objects. In addition, the use of item descriptors facilitates the maintenance of the partition between the search model and the data items. As such, flow of data during browse and search is limited to small descriptions of items rather than large collections of items.

## 3.3 ENFORMS I

The architecture of the ENFORMS system is based on the understanding that an *archive* can be composed of a number of *items*. Each of the items found in an archive has characteristics that determine the types of operations that can be performed on that item. For instance, an archive may contain a document that can be manipulated by a user via tools that allow editing, viewing, keyword searching, and so on. From a set of basic requirements, we constructed an object-oriented model of the system [3]. The graphical notations of the *Object Modeling Technique* (OMT) [6] are used to represent all models.

## *High Level Analysis*

From the most abstract point of view, the ENFORMS I system allows a user to gain access to an archive of data, where the data of interest is identified by browsing related indices (i.e., a classification scheme) about the data. This method of access is depicted in Figure 2, where the circles represent executable

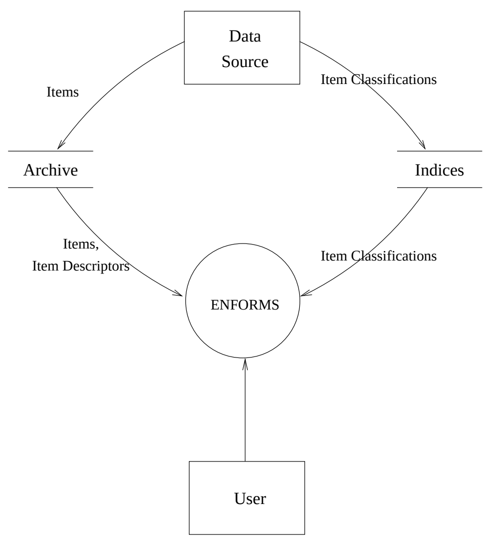
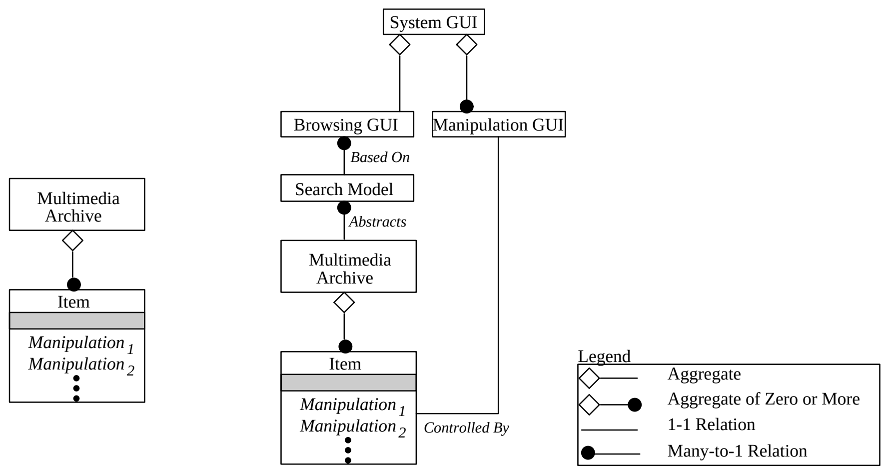



**Figure 2** Level 0 Data Flow Diagram

processes, the parallel lines represent data stores, the rectangles represent external entities, and the arrows represent flow of data. In the diagram, a *Data Source* represents originating sources of data. Once conversion of the data into suitable formats has been completed, the ENFORMS I system can manipulate that data in such a way that supports the data analysis needs of a *User*.

Figure 3(a) shows the basic object model for the archive. Two classes of objects are shown in this figure: a **Multimedia Archive** and an **Item**, where the roman font denotes classes. Using the OMT notation, the line connecting these two classes asserts the existence of a relationship; the diamond denotes the *aggregation* relationship, where the class touching the diamond is the aggregate. The filled circle at the opposite end of the line denotes "*many*", where *many* means zero or more; the absence of a filled circle at the endpoint of a line indicates "*one*". Given this notation convention, the diagram can be interpreted as "an object of type **Multimedia Archive** is an aggregate of many objects of type **Item**."




(a) Basic object model (b) Object model with access mechanisms

**Figure 3** Object models of the archive

Rectangular boxes, representing classes, can be partitioned into three layers: the top layer provides the name of the class, the middle layer lists attributes of the class, and the bottom layer enumerates operations associated with the class. The aggregation notation also identifies attributes of a class, thus introducing redundancy into the notation. In the case when either the attribute or operation part of the class notation is used, both layers are shown in order to clarify whether an enumeration lists attributes or operations. Note that shading represents unused layers.

In Figure 3(a), the notation for the class **Item** lists several *Manipulation* operations and indicates that **Item** objects are encapsulated units of information that can only be accessed through these services. This model for **Items** adds to the flexibility of the design by allowing nonatomic entities such as a group of related files, a set of maps, or even a software application to be treated as a single **Item**.

Figure 3(b) extends the basic object model by describing entities that provide access to the archive. Atop the **Multimedia Archive** is the **Search Model** that provides a browsing paradigm for the archive. The *Abstracts* relation between the **Search Model** and the **Multimedia Archive** is a *many-to-one* relation, and can be interpreted as "many different **Search Models** can provide abstractions of a single **Multimedia Archive**."

Figure 3(b) also introduces three interface entities: a **Browsing GUI**, a **Manipulation GUI**, and a **System GUI**. This figure shows that a given **Search Model** can have many **Browsing GUIs** based upon it. **Items** are *Controlled By* multiple **Manipulation GUIs**; the intended interpretation here is that each *Manipulation* operation of an **Item** is allowed to be coupled with its own GUI. Finally, the interface for the entire system, the **System GUI**, is modeled as an aggregation of a single **Browsing GUI** and zero or more **Manipulation GUIs**.

Figure 3(b) gives the general models for the archive, while the actual implementation uses specific choices relevant to the search, interface, and archive models, respectively. Figure 4 shows an extension to the general models, where **Multimedia Archive** is modified to include the notion of a **Distributed Multimedia Archive**. The remainder of this section gives details about the **Search Model** and the **Multimedia Archive**.

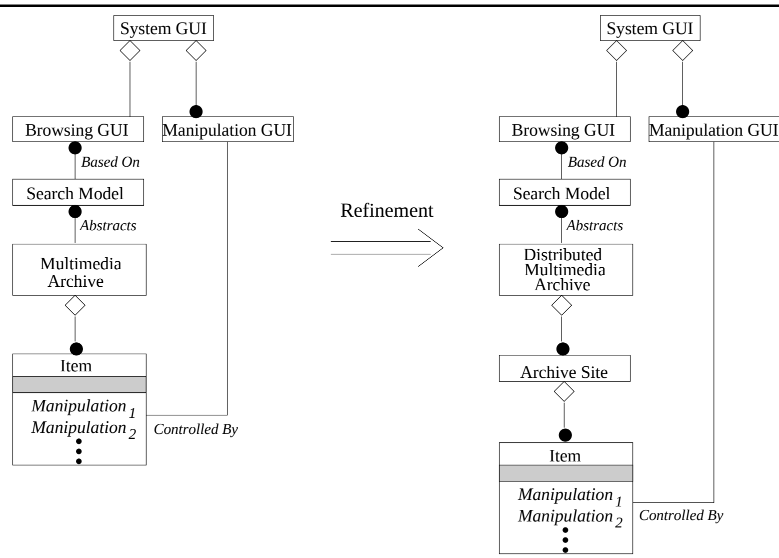



**Figure 4** Object model for ENFORMS I

At a high level, ENFORMS I has three states of operation, *System Initialization*, *System Operation*, and *System Termination*. Figure 5 depicts the dynamic model of ENFORMS, where the ovals represent states, the arrays indicate transitions (determined by the label), the solid dot is the start transition, and the outlined solid dot is the termination state. In this figure, System Initialization

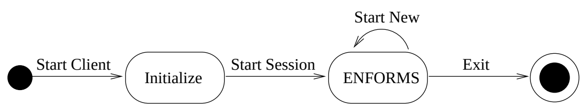



**Figure 5** Level 0 Dynamic Model

is represented by the **Initialize** state and System Operation is represented by the **ENFORMS** state.

ENFORMS uses a *browse-analysis* approach for decision support. Browsing is the method used to facilitate the construction of queries about topics of interest and the subsequent retrieval of data related to those interests. Analysis services utilize data that is retrieved from the archive after browsing. Figure 6 depicts a state machine that characterizes the browse-analysis method from a system perspective. In the browse-analysis framework, each browse-analysis

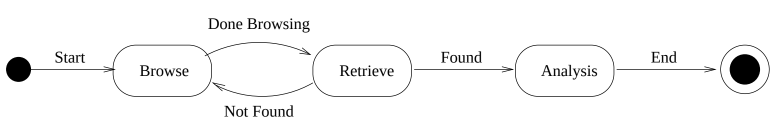



**Figure 6** Level 1 Dynamic Model for Browse and Analysis

*task* begins in the **Browse** state, where a user browses the archive based on the type of *search model* being used. Once the search criteria have been defined, data is retrieved, placing the browse-analysis task in the **Retrieve** state. If a search is successful (i.e., desired data has been found), the browse-analysis task transitions into the **Analysis** state, where a user can analyze the data using tools that can manipulate the data.

## *Search Models*

A **Search Model** defines how items are managed, classified, and retrieved. Examples of different search models include spatial, temporal, and hierarchical. Figure 7 shows the object model for a generic search model. A **Search Model** is an aggregation of **Classification** and **Item Classification**. An **Item Classification** is a construct for describing how a given item fits within a given **Classification** scheme. For instance, in the domain of zoology, the Bobcat "item" falls under the feline "classification". In the geography domain, the North America "item" could fall under the western hemisphere "classification". A search model provides the means by which a classification scheme can be browsed for subjects of interest. These subjects can then be used to create a **Query Formula** that enable the system to match the **Item Classification** to the specific **Item Descriptors** via the **Local Control Broker**.

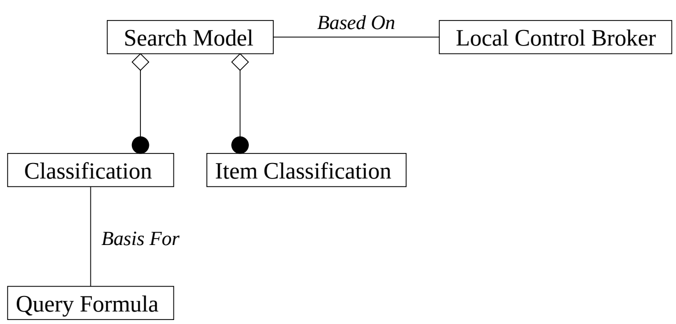



**Figure 7** Search Model

An interesting characteristic of the **Search Model** object model is the lack of an association between **Classification** and **Item Classification**. This feature allows items to have multiple **Item Classifications**. For instance, in the environmental science domain, the "ozone" item can fall under both the "atmospheric component" and as a "harmful gas" classifications.

The ENFORMS I system uses a **Search Model** called the **IPAR Search Model**. IPAR is a simple, hierarchical classification scheme where the height of the hierarchy is limited to four tiers, namely the *Issue*, *Problem*, *Aspect*, and *Refinement* tiers. An instantiation of the classification scheme consists of a specific hierarchy, such as that shown in Figure 8. The **IPAR Search Model** is a generic architecture that is completely independent of the specific instantiations of the classification scheme. IPAR partitions the classification domain into specific issues of concern. Each issue has an associated set of problems, and any given problem has several aspects. Figure 8 contains a simple example hierarchy for the data of the system.

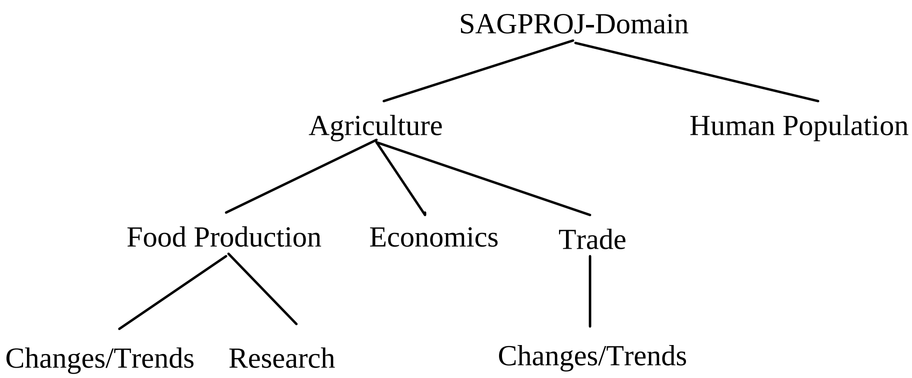



**Figure 8** An example IPAR classification hierarchy

Item classifications relate the set of items stored in the archive to the interests of the ENFORMS I user. When an item is added to the archive, it is described using the IPAR classification scheme in terms of those topics in the hierarchy to which the item relates. This approach to classification enables the user to locate items by constructing a query that describes the user's interests and requesting the system to find matches.

With respect to the analysis of the overall system, a generic IPAR model provides four basic categories of services to an IPAR-based GUI: elicitation of the current instantiation of the hierarchy, construction of classification-based queries, processing queries with respect to item classifications, and manipulation of query results. Conceptually, the IPAR model is composed of a classification hierarchy and a set of item classifications that are based on the hierarchy, as shown in Figure 9. The **Local Control Broker** provides the mechanism by which items are accessed in ENFORMS I. That is, it is the interface into the **Multimedia Archive**.

## *Multimedia Archive Analysis*

In general, a multimedia archive manages access to items. This relationship is captured by modeling the **Multimedia Archive** class as a collection of **Items**, each of which has its own set of associated access methods (as shown

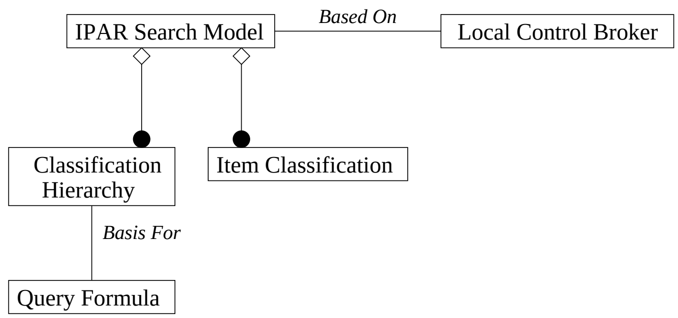



**Figure 9** A simple object model for IPAR

in Figure 3(a)). However, archive items are actually objects that exist external to the archive software, that is, in the domain of the operating system. Given this constraint, it is intuitive to model items abstractly using indirection. Figure 10 shows an object model for **Multimedia Archive**, where the **Items** are indirectly managed by (*Registered By*) **Item Descriptors**.

For each **Item** in the **Multimedia Archive**, there exists an **Item Descriptor** that contains all relevant information for the **Item**, which may only be examined by allowable manipulators. Manipulations for an aerial photograph item, for example, might be to *describe the photograph* and to *display the photo*, while those of a bibliography might be *search by keyword*, *sort by author*, and so on. In order to perform such manipulations, the appropriate software tools, or **Manipulators**, must be available for use.

The **Local Control Broker** realizes the conceptual aggregation of a **Multimedia Archive** and its collection of items by managing **Items** and **Manipulators**. Access is supported in two ways: through the retrieval of **Item Descriptors** and the activation of **Manipulators**. In ENFORMS I, six types of manipulators are supported: a GIS analysis tool, a text file display tool, an image viewing utility, an audio player, an MPEG animation player, and a generic application launching tool for applications that have their own GUI (e.g., data analysis models, pre-defined animations, etc).

## *Distributed Multimedia Archive*

As stated earlier, one of the primary goals of the ENFORMS project is to provide users with convenient access to large amounts of multimedia information that

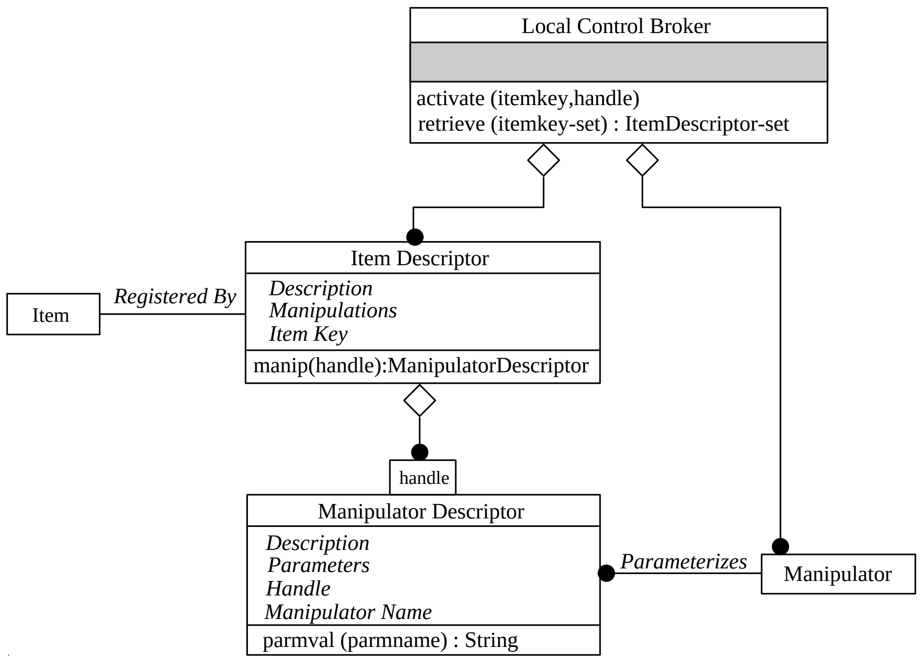



**Figure 10** Analytic model of **Multimedia Archive**

may be distributed across many sites. As such, we extended the concept of the **Multimedia Archive** from a single machine, standalone entity to a **Distributed Multimedia Archive** as shown in Figure 4. We refined this concept further by extracting the appropriate components of the **Search Model** and **Multimedia Archive** classes (shown in Figure 4). This refinement allows us to derive a new model that supports the access and manipulation of distributed data classes in a *client-server* framework. In this type of framework, *servers* accept requests over a network, perform the relevant services, and return the results. The *client* is a program that requests the services and receives the results.

Figure 11 depicts the results of the refinement of the ENFORMS system into a client-server model. One of the main features of the **Distributed Multimedia Archive** is the introduction of a **Distributed Control Broker** that facilitates the virtual communication between client and server **Search Model** objects. This communication is shown in Figure 11 by the *Talks To* relationship between the client and server **Distributed Control Broker** classes, and the *Talks*

*To* relationship between the client and server **Search Model** classes, which represent real and virtual communication, respectively.

Conceptually, this separation of client and server allows for the implementation of many different kinds of servers, each providing support for a different search model. These search models could include the **IPAR** model as well as search models that allow for spatial- and temporal-based searches.

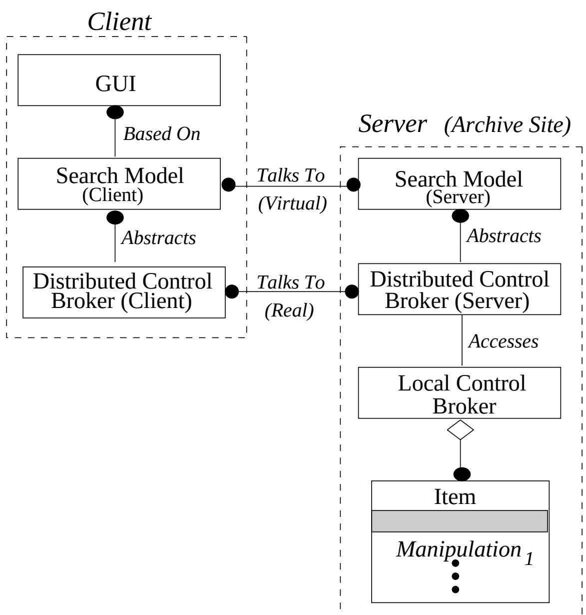



**Figure 11** The client-server architecture

## 3.4 ENFORMS II

A primary feature of the ENFORMS I system was the use of a generic client-server architecture that allows for the introduction of different **Search Models**. The analysis and design of the ENFORMS II system makes use of this feature by introducing new browsing semantics, including spatial and hypertext browsing. This section describes the design of the ENFORMS II, a system that integrates the use of spatially referenced data and hypertext documents as well as the class of data used in the ENFORMS I system.

## *Expanded Data Requirements*

*Items* in the ENFORMS I system were restricted to a single search model paradigm. That is, the items were accessible only through a hierarchical search mechanism. In contrast, the ENFORMS II supports the use of *spatial*, *temporal*, and *hypertext* search.

Figure 12 depicts an object model of the ENFORMS II archive. Within the

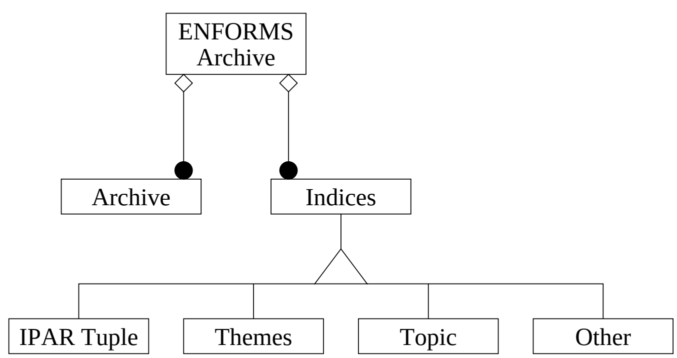



**Figure 12** High Level Object Model

ENFORMS II architecture, the **ENFORMS archive** is an aggregation of many **Archives** and many **Indices**[1]. More succinctly, the overall archive has many archives that are indexed by some classification method. Each of the archives can be considered as a server implementing a different search model. The **Indices** class demonstrates another kind of association used by the OMT notation. Namely, the inherits or "isa" association, indicated by a triangle. In the figure, there are four kinds of *Indices* modeled: **IPAR Tuple**, **Themes**, **Topic**, and **Other**. Each of these types of indices except the **IPAR Tuple** represents a new kind of search model. **Themes** classify items that fall under a *hypertext* search model. **Topic** is an item classification class for describing items that fall under a *spatial* search model, and **Other** is used as a generic item classification type.

[1] In the ENFORMS I context, an index (or its plural, indices) is the same as an item classification.

## *Refined Functionality*

Figure 13 depicts a data flow diagram for ENFORMS II. The figure shows how

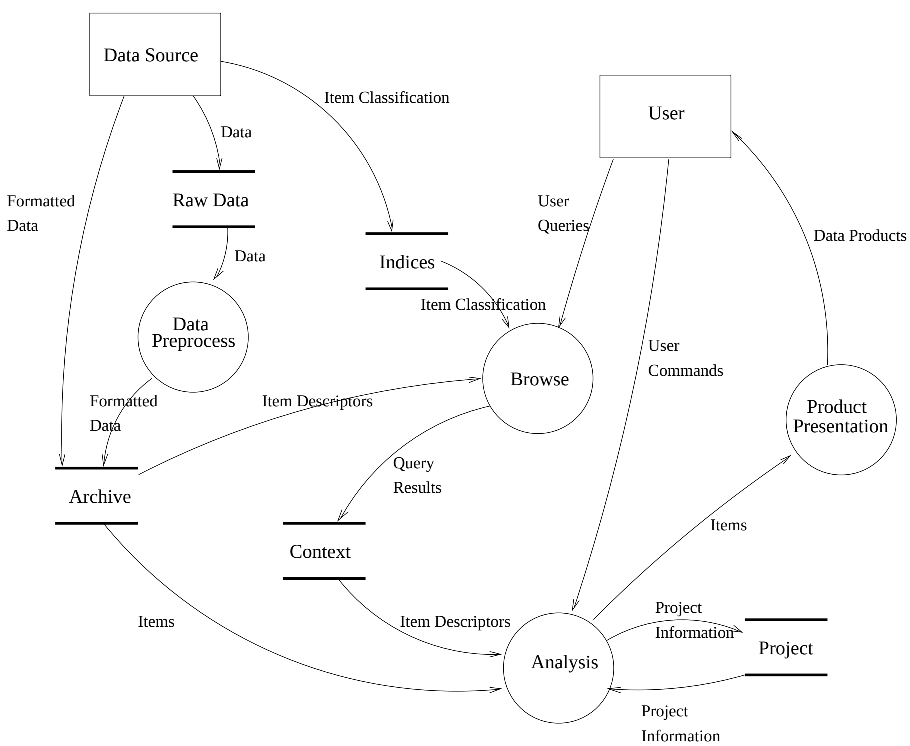



**Figure 13** Level 1 Diagram

the ENFORMS II system extends the idea of search models by introducing new browse semantics based on the use of spatially referenced and hypertext data.

A User begins by browsing and selecting areas of interest. The system uses those selections to form search parameters (the query) that are used to retrieve data items from an archive. The results of the query are used to create an *analysis context* for a user. An analysis context is a set of query formulas and associated data item descriptors that are used to describe the subset of data that has been identified by a user via the browsing activity. Using items found in the analysis context, a user can perform a number of analysis operations. These operations can then be used to generate output using some presentation package. After each session, a user has the option of saving their context and the results of analysis into a project file that can be recalled for later use.

## 3.5 Classification Software

A main feature of the ENFORMS system is the ability to easily add new items to the archives. To facilitate that activity, a software system called CLASSIFY was developed. The primary objective of this tool is to support two activities. First, it allows data custodians to create or modify item classifications. That is, item classifications can be modified based on the search model(s). Second, CLASSIFY enables data custodians to create new items by defining the data and the set of tools that provide access to the data.

Figure 14 depicts the architecture of the CLASSIFY tool. Based on the architecture of search models in ENFORMS, the CLASSIFY tool has a straightforward analysis, design, and implementation. Specifically, CLASSIFY incorporates the use of the concepts defined earlier for **Item Classification**, **Classification**, and **Multimedia Archive**.

### *Domain Classifications*

In both versions of the ENFORMS system, a **Classification** is a class that implements a cataloging feature. For instance, items in ENFORMS II can be classified (catalogued) hierarchically and/or spatially. The CLASSIFY tool allows a data custodian to construct a *classification scheme* for the domain of items contained in an archive, thus termed a *domain classification*. Using the products of the CLASSIFY tool at run-time, ENFORMS instantiates a graphical interface that is used by a user during the browse activity.

### *Item Creation*

There are two activities that must be performed when a data custodian needs to add a new item to an ENFORMS archive. An item must first be *described*, and then the item must be *classified*. Both of these activities are supported by the CLASSIFY tool.

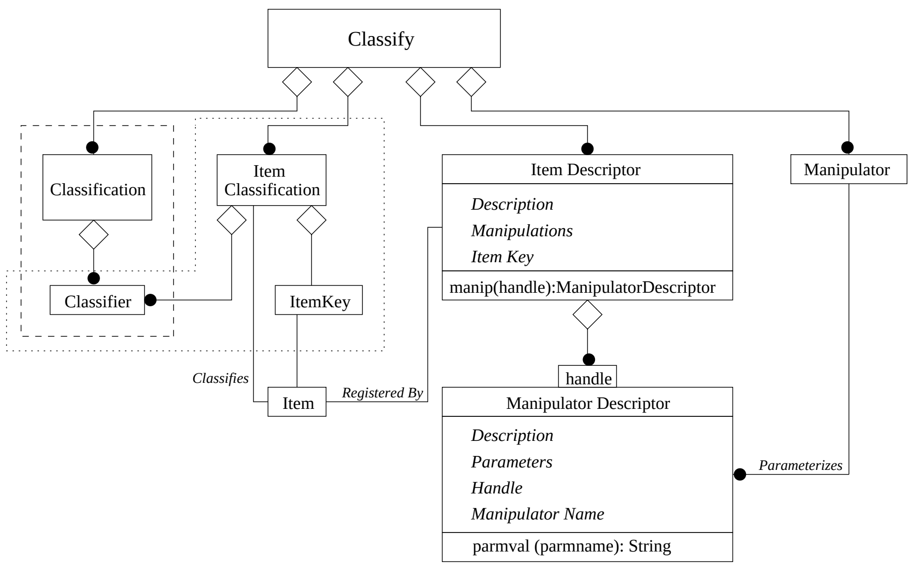



**Figure 14** Analytic object model of Classification Software

The act of describing an item using the CLASSIFY tool essentially allows a data custodian to create an **item descriptor**, a concept discussed in Sections 3.2 and 3.3. As such, a data custodian is responsible for defining the access methods for the new data item (i.e., the item manipulators) by either using a pre-existing manipulator, or by describing new manipulators. Once an item has been properly described, it must be placed into the archive. Placement of an item into the archive requires describing the item using an **item classification**, an activity also supported by the CLASSIFY tool. As was the case with the domain classifications, the products of the CLASSIFY tool are used by ENFORMS at run-time to define the contents of the archive.

# 4 APPLICATIONS

During this decade, NASA will launch many new platforms into earth orbit, including the satellites that will make up the Earth Observing System (EOS). The remotely sensed data obtained from EOS can be used to promote global and national security, extend international cooperation, and improve our ability to understand and manage global environmental, economic, and social problems. In the past, NASA and other agencies have focused on the acquisition of data rather than the integration or the dissemination of data. Many organizations addressing grand challenge problems, such as those defined by earth sciences, require the integration of both physical and human resource databases in an interactive manner. Such a capability allows reasonably informed policy analysts and related staff to query an "Environmental Science Workstation" so as to better understand how human uses impact our natural resource base.

The ENFORMS project is a multidisciplinary effort involving researchers from more than twenty-two different departments from Michigan State University and numerous scientists from CIESIN, EPA, USDA, and NASA. ENFORMS has been designed to provide access to data and data integration utilities for the purposes of facilitating decision making processes relevant to environmental policies. The user constraints included a user-friendly system, usable by non-computer scientists, focused search capabilities, and transparency of the distributed and heterogeneous nature of the data and analysis utilities. ENFORMS contains a wide variety of items, such as image files, research papers, executable environmental models, and research data sets. Using the graphical user interface, the user is able to examine these items interactively through the archiving software and display images and execute environmental models without requiring an extensive computer background.

Based on the original object-oriented architecture, there have been three separate instantiations of ENFORMS for use in three disjoint projects. For each project, four tasks were performed. First, a *user needs analysis* was performed. The objectives of this study were to identify the users of the system, what problems are of interest to users of the system, and what types of answers are sought for the questions. Second, based on the results of the user needs analysis, a group of application scientists assisted in identifying the sets of data to be used to address the problems and answers of the users. In most cases, the required data came from different sources, were stored in numerous formats, and required different access utilities. Therefore, numerous utilities have been developed to provide unified access to the relevant data, making the heterogeneous formats, and distributed nature of the data transparent to the user. The third task was to determine what types of analysis tools were needed by the users in order to manipulate and integrate the data as a means to find answers to the questions as well as facilitate the decision making processes. The final task was to determine how the information in the system should be presented to the user. This task included determining the search (browsing) model, the type of interface to the analysis tools, and the type of presentation for analysis results. This last task relied heavily on the use of the CLASSIFY utility as it provides a user-friendly mechanism to be used by the development team and the data custodians for determining the presentation of information, how data is accessed and manipulated, and a facility for classifying the data.

The objectives for each of the projects are described briefly below in chronological order. For each project, a description of the user needs analysis is given, followed by a discussion of the data types and analysis utilities needed by the respective projects. It is noted that with each instantiation of ENFORMS, the majority of the original architecture was preserved. The most significant difference between ENFORMS I and II is in the search/browse models. Different data manipulators are included for the three instantiations, but the need to include different manipulators does not require any changes to the overall architecture. For a given instantiation of ENFORMS, the CLASSIFY tool was used to define the classification hierarchy, individual item classification, and individual item descriptors (e.g., textual descriptions, relevant manipulators, associated items).

## 4.1 Saginaw Bay Watershed Regional Analysis

**User Needs Analysis.** The objective of the Saginaw Bay Watershed Regional Analysis project (under sponsorship from NASA) was to focus on the sustainability of food production systems in the Saginaw Bay Watershed in Michigan, with a particular emphasis on projections into the coming century. Ongoing work by numerous scientists at Michigan State University (MSU) was used to accelerate the user needs analysis. These researchers analyzed responses from user surveys in order to determine the most pressing questions involving human use of natural resources in food production. From the user survey, it was determined that the relevant areas of study included land and water features, agriculture, animal populations, human populations, pollution, climate, and environmental health.

**Data and Analysis.** The results of the assessment were analyzed by science members of the project team to identify data and the different types of data analysis that would help to address some of the users' concerns. A sampling of the types and sources of data that were identified and used to populate the watershed configuration of ENFORMS I are given in Figure 15. Once the data was identified and acquired, it was classified according to the perspective obtained from the survey study.

| TYPES OF DATA | DATA SOURCE |
|---|---|
| *PHYSICAL* | |
| Land use | Landsat TM (1984 and 1992) and CRIES |
| Land use change | Landsat TM (1992) versus MIRIS 1978 |
| Lakes, rivers, streams and drains | Michigan Resource Information System (MIRIS) of the Michigan Department of Natural Resources |
| Watershed boundaries | MIRIS and U.S. Geological Survey |
| Climate | NOAA's National Climatic Data Center (NCDC) |
| Soils (STATSGO) | USDA Soil Conservation Service |
| Topography (digital elevation model) | USDA Soil Conservation Service and the Defense Mapping Agency |
| Geology | MSU Center for Remote Sensing |
| Aquifer vulnerability | MSU Center for Remote Sensing |
| *BIOLOGICAL* | |
| Surface conditions | NOAA AVHRR |
| Endangered species | Michigan Natural Features Inventory Heritage Program and the Nature Conservancy |
| Non-indigenous species | MSU Department of Entomology |
| Forest maps | Comprehensive Resource Inventory Evaluation System |
| Watershed bibliography | Saginaw Valley State University |
| *SOCIO-ECONOMIC* | |
| Urban land use | MSU Imagery Archive Laboratory |
| Population and economics (TIGER) | U.S. Bureau of the Census |
| Employment and income | MSU Institute for Public Policy and Social Research |
| Human health, economics | MSU Institute for Public Policy and Social Research |
| County boundaries and transportation | MIRIS |
| Crop production | Michigan Agricultural Statistics Service and National Agricultural Statistics Service |
| Management practices | MSU Cooperative Extensions Service and USDA Soil Conservation Service |
| Government policies | Documents from Various Sources |
| Policy analyst directory | MSU Institute for Water Resources |

**Figure 15** Sampling of data types and sources available through the demonstration prototype.

## 4.2 US/Mexico Border Water Quality and Quantity

**User Needs Analysis.** ENFORMS I has also been used to facilitate data access and analysis for information specific to the El Paso/Ciudad Juarez area in west Texas, southern New Mexico, and Mexico. The specific objective of the USDA sponsored project was to provide a graphical-based environment for accessing the integration of socioeconomic and water quality/quantity data.

The area of focus is a critical region receiving a great deal of attention due to the North American Free Trade Agreement. In general, water quality and water quantity issues are of vital concern to planners in the region because of the arid climate, limited water supply, forecasted increases in population, and the need for water in agricultural production in the region.

**Data and Analysis.** The project specifically focused on water quality and quantity issues in the US/Mexico border region. The data relevant to the project included information for both surface and ground water. The system provided access to three major types of data for the project: text data, spreadsheet data, and image data.

A primary source of information was a set of Lotus spreadsheet files containing water pumpage data, water level data, and chemical and trace analysis data for wells in the Hueco Bolson aquifer, a major source of groundwater for the region.

The users emphasized the need to be able to perform analysis of the spreadsheet data related to water quantity and water quality in the wells supplying groundwater to the region. In addition to standard spreadsheet analysis, the users also requested the ability to dynamically generate graphs from the spreadsheet data that enabled them to determine correlation between different parameters of well data. The graphing utilities supported the integration of temporally referenced data and individual parameters, such as chemical levels, well pumpage rates, and water levels. All of the data analysis needs were handled by providing access to the appropriate manipulators, some of which were developed by the ENFORMS team. There was no need to change the core of the architecture of ENFORMS I to include the new manipulators.

## 4.3 Great Lakes Regional Decision Support System

**User Needs Analysis.** An *environmental quality assessment* is defined to be a process that delineates the extent of environmental impact within a given geographic area. It involves the identification of potential sources of contamination and an evaluation of the type and degree of ecological impairment within the given area. A *monitoring needs assessment* is defined as a process to evaluate the adequacy of monitoring efforts for a given set of parameters and/or geographic area and to identify further monitoring requirements. The objective of ENFORMS II (under sponsorship from the EPA) was to study the impacts of water monitoring at a regional level (Great Lakes) on environmental quality and monitoring needs assessments.

Due to the heavy emphasis on spatially and temporally-referenced data, the ENFORMS search model and analysis utilities were expanded significantly, thus warranting the name ENFORMS II.

**Data and Analysis.** Historical monitoring data is extremely important in the assessments of this project, providing insight into environmental change and, more importantly, trends. Several data sets were identified to be important to water quality assessment activities. Figure 16 contains a summary of the representative data sets available through the system.

The two main data types are *georeferenced point data* and *ancillary data.* Georeferenced point data (also known as *point data*) is a set of attribute values that is indexed using the latitude and longitude of the site where the data sampling took place. Ancillary data consists of images, charts, documents, MPEG movies, etc. Although ancillary data may be specifically related to a particular geographic area, the main characteristic that differentiates ancillary data from point data is that point data contains a *sampling* of some attribute, while ancillary data serves to describe, explain, show, or illuminate facts about the subject being examined. Point data more naturally lends itself to further analysis by Geographic Information Systems (GIS), spreadsheet programs, or statistical analysis packages.

There are four different kinds of analysis that are available in the ENFORMS II: Point Tool, Full Spatial, Guide, and IPAR. Point Tool is a tool that supports the manipulation of spatially referenced data that is not numeric in nature (i.e., it is ancillary). Full Spatial analysis supports the manipulation and analysis of spatially referenced tabular numeric data via the use of some Geographic

| Data Name | Description |
|---|---|
| EPA STORET | Chemical analysis of water samples |
| USGS Flow Gauge | Measurements of rate of water flow (river/stream) the vulnerability of a river or stream to sedimentation. |
| Permit Compliance | Permitted surface water discharge for sites |
| Bio-reports | Field reports representing observations made at a given location regarding plant, animal, and insect populations. |
| DNR Boundary Data | Information about different spatial regions (political and hydrological boundaries). |
| Physical Features | Physical features of a region (roads, highways, rivers) |
| Ancillary and Metadata | Information related to previously mentioned databases human populations, quality control information, various, reports bibliographies, photographs, data catalogs, and precomputed charts, graphs, and documents |

**Figure 16** Representative data for EPA Regional Analysis System

Information System (GIS) tool. Guide analysis provides facilities for viewing hypertext documents related to certain environmental areas of interest. Finally, IPAR analysis supports the manipulation of data items based on their access methods. One of the IPAR access methods is parameterized modeling. The system supports both predictive model-based analysis and historical data analysis, both of which required sophisticated visualization and animation capabilities. The analysis of the historic data have both spatial and temporal dimensions, which enables users to analyze data based on spatial location or time periods. Most of the spatial analysis requirements can be carried out with the help of a particular geographical information system (GIS), which is designed specifically to manipulate the spatial relationships.

## 5 CONCLUSIONS AND FUTURE DIRECTIONS

As multimedia technology continues to improve and expand into more applications and the volume of data increases, the demand for developing extensible and configurable multimedia information systems increases. Economic realities will not permit the development of new systems each time the multimedia technology changes or new data needs to be added to a system. This chapter has presented the basic architectural framework of object-oriented information systems. Sample analysis and design techniques for developing object-oriented systems has also been described. The context for much of the discussion was in terms of the object-oriented development of ENFORMS, an object-oriented, distributed multimedia information system. In addition to the object-oriented design of ENFORMS, the system is runtime configurable to the data and analysis tools according to metadata descriptions of the system classification, item classification, and item descriptors. Three separate instantiations of ENFORMS has also been described.

As multimedia systems gain more widespread use, more attention will be focused on process development models for multimedia systems. More CASE (computer-aided software engineering) tools will be needed to develop multimedia applications. Currently, multimedia application development requires knowledge in a number of areas within computing, including networking, large-scale databases, software engineering, media manipulation, interface design, and so on. The increasing number of multimedia application developers will motivate the development of reusable libraries for each of these areas that can be readily used to improve the productivity of multimedia system development. Object-oriented technology is likely to play a large role in the development of these reusable libraries. Finally, it is anticipated that the work in the area of standards for the different components within multimedia applications will become more mature in the near future. Examples include data formats, communication protocols, and data manipulators (e.g. animation tools, movie players).

# 6 ACKNOWLEDGEMENTS AND FURTHER INFORMATION

The projects described in this chapter have been sponsored in part by NASA, EPA, USDA, the Consortium for International and Earth Science Information Network (CIESIN), and Michigan State University. The success of the ENFORMS project has only been possible due to the devoted efforts of a number of people, including Robert Bourdeau, David Robinson, Steve Schafer, Joe Sharnowski, Yonghao Chen, Paul Fraley, Dan Judd, Heather Richter, Stephen Wagner, and Enoch Wang. The application scientists provided invaluable information in the organization of data, classification, and usability of the system: Bryan Pijanowski, Stuart Gage, Jon Bartholic, Lois Wolfson, T. Kang, and Todd Zahniser.

For further information regarding the ENFORMS project and other related materials, please see the ENFORMS World Wide Web homepage at

`URL:http://www.cps.msu.edu/groups/enforms/enforms.html`.

Specific questions can be addressed to B. Cheng (chengb@cps.msu.edu).

# REFERENCES

[1] R. H. Bourdeau, B. Pijanowski, and B. H. C. Cheng, "A decision support system for regional environmental analysis," in *Proc. of 25th International Symposium on Remote Sensing and Global Environmental Change: Tools for Sustainable Development (Vol. II)*, (Graz, Austria), pp. 223–233, 1993.

[2] R. H. Bourdeau, B. H. C. Cheng, and B. Pijanowski, "A regional information system for environmental data analysis," *accepted to appear in Photogrammetric Engineering & Remote Sensing.*

[3] B. H. C. Cheng, R. H. Bourdeau, and G. C. Gannod, "The object-oriented development of a distributed multimedia environmental information system," in *Proc. of 6th International Conference on Software Engineering and Knowledge Engineering*, pp. 70–77, June 1994.

[4] J. L. Sharnowski, G. C. Gannod, and B. H. Cheng, "A distributed multimedia environmental information system," in *Proc. of IEEE Int.*

*Conference on Multimedia and Computing Systems*, (Washington, DC), pp. 142–194, May 1995.

[5] E. N. Yourdon and L. L. Constantine, *Structured Design*. Yourdon Press, 1978.

[6] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *Object-Oriented Modeling and Design*. Englewood Cliffs, New Jersey: Prentice Hall, 1991.

[7] P. Coad and E. Yourdon, *Object-Oriented Analysis*. Englewood, New Jersey: Yourdon Press, Prentice Hall, 1990.

[8] R. Wirfs-Brock, B. Wilkerson, and L. Wiener, *Designing Object-Oriented Software*. Englewood, New Jersey: Prentice Hall, 1990.

[9] R. H. Bourdeau and B. H. C. Cheng, "A formal semantics of object models," *IEEE Trans. on Software Engineering*, vol. 21, pp. 799–821, October 1995.

[10] R. Steinmetz and K. Nahrstedt, *Multimedia: Computing, Communications, and Applications*. Prentice-Hall, 1995.

[11] H. M. Gladney, E. A. Fox, Z. Ahmed, R. Ashany, N. J. Belkin, and M. Zemankova, "Digital Library: Gross Structure and Requirements: Report from a March 1994 Workshop," in *Proceedings of the First Annual Conference on the Theory and Practice of Digital Libraries*, 1994.