Title	Infrastructure management information system framework requirements for disasters
Authors(s)	Pradhan, Anu, Laefer, Debra F., Rasdorf, W. J. (William J.)
Publication date	2007-03
Publication information	Pradhan, Anu, Debra F. Laefer, and W. J. (William J.) Rasdorf. “Infrastructure Management Information System Framework Requirements for Disasters” 21, no. 2 (March, 2007).
Publisher	American Society of Civil Engineering (ASCE)
Item record/more information	http://hdl.handle.net/10197/2377
Publisher's version (DOI)	10.1061/(ASCE)0887-3801(2007)21:2(90)

Infrastructure Management Information System Framework Requirements for Disasters

Anu R. Pradhan,[1] Debra F. Laefer,[2] M. ASCE, and William J. Rasdorf,[3] F. ASCE

ABSTRACT

A three-tiered, enterprise, GIS architecture offers a robust, efficient, and secure platform to potentially revolutionize disaster management by enabling support of all of the phases of governmental activity that must occur before, during, and after a disaster. Presently, both publicly and privately initiated, computer-based systems designed for disaster management cannot meet the real-time data access and analysis needs at crucial stages, especially those occurring during an actual disaster. Impediments are reflective of the proprietary, standalone, and segregated nature of current systems. This paper proposes an integrated, infrastructure management information system as a reliable and effective alternative. Issues related to sharing data, customizing applications, supporting multiple data formats, querying visually, facilitating ubiquitous computing, and upgrading are all addressed. Achieving maximum flexibility and capacity in a disaster management system relies upon recent advances in the following areas: (1) standardized data specifications, (2) middleware services, and (3) web-enabled, distributed computing. Key resources in designing and implementing such an arrangement are prototyped in a system that was initially designed for addressing disaster management of urban explosions. The critical details of that system are presented herein.

KEYWORDS

Disasters, Hazards, Database, Spatial Data, Emergency Services, Damage Assessment, Disaster Relief, Computer Networks, Computer Applications, Computer Models, Computer Software, Information Management, Information System, Geographic Information System

---

[1] Doctoral Candidate, Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA 15213.

[2] Corresponding Author, Lecturer, School of Architecture, Landscape and Civil Engineering, College of Engineering, University College Dublin, Earlsfort Terrace, Room 115, Dublin 2, Ireland 011-353-1-716-7276 (phone), 011-353-1-716-7399 (fax). Email: debra.laefer@ucd.ie.

[3] Professor, Department of Civil, Construction, and Environmental Engineering, North Carolina State University, Campus Box 7908, Raleigh, NC 27695.

INTRODUCTION

The September 11, 2001 attacks on the World Trade Center (WTC) acutely demonstrated the need for an integrated management information system (IMIS) to assist in simultaneously assigning governmental services and resources for regular programs and emergency response (Flax et al. 2002). Although the majority of American cities now have electronic base-maps to facilitate urban planning, few of these base-maps have disaster management capabilities. Even amongst those that do, there are significant functional limitations with respect to (1) accessing data collected by non-disaster management branches of government and (2) supporting data analysis capabilities. Impediments exist because of organizational, computing, and charter characteristics of these systems (e.g. Uddin and Engi 2002). The challenges of data access, spatial querying, and information sharing across multiple users, particularly during disaster response have been identified as major technical challenges (Goodchild 2003, Mansourian et al. 2005, Zerger and Smith, 2003). These obstacles exist in large part due to the proprietary, standalone, and segregated nature of the current DMSs, which prevents participation in efficient data gathering and sharing capabilities beyond the purview of individual departmental authority. Consequently, these systems are of circumscribed utility, when responding to major disasters (Greene 2002, Flax et al. 2002, Cutter 2003). A new approach to the design and implementation of IMISs for managing disasters is needed. This paper assesses various disaster related phases in terms of the emergency response community, outlines current system limitations, defines essential system features, and introduces a portion of DIsaster Operations, Risk, Abatement, and MAnagement (DIORAMA), a recently prototyped disaster management system (DMS) developed by the authors.

The initial motivation to develop DIORAMA was to create a system capable of supporting highly detailed, blast-damage analysis for urban areas that had been subjected to high explosive detonations. Patterns of energy dissipation and blast-damage cannot be understood thoroughly, without powerful three-dimensional (3D) querying capacities. Since inclusion of actual pre- and post-blast condition data of building components, as well as entire structures, was deemed crucial to comprehending vulnerabilities, extensive data collection of the built-environment became paramount. Such information was essential both in terms of post-incident performance assessment and in applying those assessment outcomes to other structures. As such, a continuous cycle of information was needed to analyze urban explosions and their attendant hazards throughout all phases of the disaster management process. In this paper, the requisite computer architecture for such a disaster management system is illustrated and multi-hazard, disaster analysis needs are used to illustrate how the concepts here presented are widely applicable.

BACKGROUND

A disaster is defined as a “sudden calamitous event bringing great damage, loss, or destruction” (Merriam Webster 2003). A sampling of disasters may be generalized into interrelated or similar groupings based on the cause, location, and type of community affected (Table 1). Although these broad categories include events that are highly distinct in their nature, environment, and impact, many have commonalties with respect to the data sets required to address these disasters in a multi-phase context (Figure 1). A simple example is in the planning and execution of large-scale evacuations. Such evacuations may be necessary during earthquakes, hurricanes, forest fires, and nuclear contamination. To find proper evacuation routes, the same road network data set is required, irrespective of the disaster. By focusing on common data sets, although not fully reflective of any particular disaster or specific phase of the disaster management cycle, an initial framework can be created from which to prototype a system to address a wide range of disasters. The premise of basing disaster management systems on common data sets was recently confirmed by a group of over 30 emergency management personnel in the state of North Carolina (Rasdorf et al. 2005). Such an approach holds promise for addressing the complexities of disaster management. Unfortunately, as will be fully described below, the emerging geographic information system (GIS) based solutions being developed by governmental agencies and private companies do not accommodate the multiple phases that occur as part of the disaster management process.

Importance of GIS

Increasingly, community vulnerability to natural and manmade disasters is being recognized (Godschalk 2003). The level of vulnerability is an indication of the extent of the community's disaster readiness. Although disasters occur in a wide variety of manifestations, all happen within a specified geographic area, and, thus, all disaster analysis must be viewed in relation to that location (Johnson 2003). Understanding geographically-based information is critical to protecting communities from disasters (Radke et al. 2000).

The general availability of computing as a routine part of community planning and management facilitates an information technology role in decision-making before, during, and after disasters. For instance, during a disaster, multiple emergency management responders need to share information, which requires access to databases and computer-generated maps. Simultaneously, they need to perform mission-critical, data analysis to assist search and rescue operations (ESRI 1999). GIS-based tools have proven to be important aids in disaster preparation and rapid response (Johnson 2003). Digital maps and spatial analysis tools (enabled by GIS) offer a common template on which a wide variety of data can be quickly placed, measured, and analyzed, which can result in better informed decisions (Greene 2002). Thus, to be truly prepared for a disaster, emergency managers need a GIS-based tool to organize, integrate, analyze, and distribute data associated with various phases of disaster management.

Disaster Management Phases

The authors propose that the success of any disaster management system is dependent upon capabilities in six disaster-related phases: (a) Identification, (b) Prediction, (c) Mitigation, (d) Preparation, (e) Response, and (f) Recovery (Figure 1). Within DIORAMA these terms have been defined as follows. Identification involves ascertaining and inventorying community assets, which are dominated by the physical infrastructure (e.g. roads, buildings, bridges) but also include transportable resources such as fire trucks and hospital beds. Prediction entails conceiving dangers and affiliated impact levels, which may include casualties, property losses, and utility service interruptions. Mitigation concerns active reductions of the disaster's impact, while Preparation includes needed actions to contend with the portion of the disaster's impact that cannot be mitigated (i.e. there will always be some property damage and some injuries). Response addresses real-time actions as the disaster evolves, while Recovery encompasses restoring services and rebuilding disaster-stricken areas. Figure 1 conveys a sense of the scope of the disaster management problem, although phase labeling is somewhat arbitrary.

To illustrate these six phases and to demonstrate the multi-disaster applicability of the proposed structure, a hurricane evacuation scenario is provided. To minimize hurricane-induced losses, each of the above-defined phases must be addressed within the disaster management system (DMS). The DMS must host the required data and support relevant analytical capabilities. During Identification, susceptible communities are ascertained, based on information from past hurricanes and coastal proximity. In Prediction, the identified communities are assessed for potential casualties, based on local population density, physical infrastructure, and storm intensity history. As part of Mitigation, particular evacuation routes for each designated community are identified and upgraded, based on current infrastructure capabilities versus projected casualty levels. Throughout Preparation, these communities are educated as to the periodic necessity to evacuate and as to evacuation route locations and procedures, based on specific geographic and demographic characteristics. Within Response, knowledge of rapidly changing conditions is collected and integrated, and preferred evacuation routes are communicated to the public, based on storm path, flooding levels, and traffic conditions. In the Recovery period as the hurricane wanes, community resources such as shelter, water, and food are allocated to survivors and debris is prioritized for removal, based upon temporary population distribution and evacuation route accessibility.

The four phases of Identification, Prediction, Mitigation, and Preparation can be considered temporally as pre-disaster. In contrast, the phase of Response occurs while the disaster is happening, and Recovery is strictly a post-disaster phase. These six phases illustrate the importance of the interrelationship of the various phases before, during, and after a disaster. Having all temporal aspects addressed is essential, as the information pertinent to one phase becomes vital to other phases. For instance, the analysis done during the Preparation phase informs the Response phase. Choices during Response impact Recovery strategies. Similarly, post-damage data collected during the Recovery aids in the Identification, Prediction, Mitigation and Preparation aspects for future disasters. A DMS must be able to address systematically all six phases, because of their intrinsic inter-dependency.

Existing Systems

Unfortunately, despite multi-billion dollar investments in preparedness and simulated training exercises (GAO 2003), the majority of American cities and municipalities are still inappropriately or minimally prepared for disasters as clearly demonstrated by emergency response to hurricanes Katrina and Rita. Reasons range from the fiscal to the political, but the computer architecture of many of the best known DMSs have inherent computing limitations that prevent their full applicability to disaster management. For instance, cities such as New York have baseline systems with digital maps of public utilities, road networks, and building footprints, but the system is standalone and of circumscribed functionality, because it lacks the capability to exploit the stored data for analysis (Flax et al. 2002). The establishment of a baseline system represents a necessary but insufficient condition for a viable, fully functional, multi-phase system. Of those cities with baseline systems, only a small percentage of them have disaster related capabilities, and of those, no city has yet designed and implemented a computer system capable of supporting the requisite Response and Recovery that are critical during a major natural or manmade disaster. Instead, emergency aspects of Response and Recovery are strictly supported by 911 calls and aerial surveillance, which do not support other aspects of disaster management.

Existing systems are typified by discrete, non-networked arrangements that are largely ineffectual in many aspects of disaster management. Two of the best-known DMSs are known by their acronyms HAZUS and CATS. Evaluations of these and a commercial system are presented below, with respect to multi-phase disaster needs.

HAZUS-MH

The Federal Emergency Management Agency (FEMA) developed a hazard prediction program under contract with the National Institute of Building Sciences: Hazards United States – Multi-hazard (HAZUS-MH) uses GIS software to map and graphically depict hazard data, economic losses, and buildings and infrastructure damage from hurricanes, floods, and earthquakes (HAZUS 2005). The GIS technology combines hazard layers with national databases, including datasets of demographics, building inventory, critical facilities, transportation systems, and utilities. The system applies both loss estimation and risk assessment methodologies for a limited set of disasters. Although HAZUS-MH is equipped to model the probability and level of pre-incident risk, the system can neither support real-time response and recovery and nor multiple device types (ubiquitous computing). Thus, the software addresses only the Identification, Prediction, and Mitigation phases (Table 2).

CATS

The Consequences Assessment Tool Set (CATS) is a product of the Defense Threat Reduction Agency in conjunction with FEMA (DTRA 2002). CATS is an ArcView application for predicting hazards from hurricanes, earthquakes, weapons of mass destruction, and conventional explosives and for assessing the ensuing consequences. CATS allows the user to simulate scenarios for training and planning, assess affected population and infrastructure based on real-time weather information, initiate resource deployment, evaluate further needs, and locate resources for a sustained response. As CATS is primarily a training tool, it cannot support real-time Response and Recovery activities (Table 2).

Commercial System

There are also commercially available programs for post-disaster data consideration, such as the one developed by Deaton and Frost (2001), which integrates a global positioning system, GIS software, digital photography, and handheld computing technology. The system allows both quantitative and qualitative data collection, but primarily focuses on baseline and post-disaster data collection. Thus, this system addresses only the Identification and Recovery phases (Table 2).

In summary, readily available public and private systems each support only some of the six disaster management phases. Thus, they cannot process disaster-related information within the context of an integrated system.

As a DMS's architecture o necessarily defines future expansion and integration options, hardware and software choices must be evaluated for multi-phase applicability.

SYSTEM REQUIREMENTS

An ideal DMS should be enabled for performance in all six disaster management phases within a single integrated system. To achieve this, a DMS must have three components: (a) a baseline system, (b) spatial querying, and (c) ubiquitous computing.

Baseline System

A baseline system establishes the basic information backbone required for a DMS (Laefer et al. 2005a). A DMS must be capable of three key functions: depiction, integration, and connection.

Depiction

Depiction is the physical representation in a digital format of the environment – including streets, utilities, buildings, and other major infrastructure components. Depiction includes data selection, collection, and digitization. The data may encompass various sets including photographs; digitized and paper drawings of architectural and structural elements; transportation network information; utility maps; periodic building or façade inspection reports; and GPS-based building and street locations (Laefer et. al 2005b).

Integration

Integration is the combination of various required data sets into a single, synthesized repository. An integrated database would facilitate rapid and detailed disaster response that would be unachievable with traditional tools, because the integrated database allows unprecedented use of multiple data types (attribute data, text, drawings, maps, photos, and videos, etc.) in combination with electronic, spatial data fusion, which links together otherwise inaccessible data. These processes result in new ways to share and distribute data.

Connection

Connection is the establishment of relationships between various attribute data and key infrastructure features (e.g. size, shape, age, and depth of embedment of a section of a utility line) and their linkage to their identifiable, visual representations (e.g. gas lines distinguishable from water lines). Attribute data constitutes the characteristic information about the spatial features. For instance, a road (represented by a line feature) has various attributes such as width, bedding composition, pavement condition, and speed limit. Likewise, a building (represented as a spatial feature) has attributes such as height, street address, and number of stories. By establishing connections between each spatial feature and its relevant attribute information, a simple digital map can be transformed into a substantially more powerful tool, because of the capability for spatial querying.

Spatial Querying

A DMS should support the complex spatial analysis required for all of the six disaster-related phases. As mentioned earlier, from identification to prediction, mitigation to preparation, response to recovery, each phase inevitably leads to the next, in a continual quest for a safer environment in which to live and work (Johnson 2003). Complex spatial queries offer unprecedented opportunities for advances in community safety. Examples of useful spatial queries include locating the shortest evacuation path and optimizing post-disaster resource distribution (e.g. shelters, food depots) for displaced people.

Ubiquitous Computing

Ubiquitous computing enables multiple computing devices (e.g. workstations, laptops, and handheld devices) to be simultaneously and seamlessly available as a composite system (Weiser 1993). Such architecture would maximize existing resources and provide the highest level of functionality for various end users, as different types of devices have various capabilities from simple data collection to resource intensive computing. For example, handheld devices can play an important role in post-disaster data collection, because of their size and portability (Deaton and Frost 2001). Similarly, resource intensive analysis (most pertinent to identification, prediction and response) needs to be done with powerful workstations. Thus, a system is required that facilitates the use of multiple computing devices of varying power levels and capabilities, because a DMS should be accessible to many different clients. The concept of ubiquitous computing should also be expanded to include peripheral hardware including data collection and measurement devices and gages. At present, such systems exist to a limited extent at specific academic institutions (e.g. Liu et al. 2003, Pena-Mora 2001), but are not part of standard disaster management systems such as HAZUS-MH and CATS.

ENTERPRISE GIS ARCHITECTURE

To begin to meet the above described needs, an Enterprise GIS architecture consisting of a database system (data tier), middleware (business logic tier), and customized user interface (client tier) is proposed. The main goal of an Enterprise GIS architecture is to provide a common data sharing infrastructure (Von Meyer and Oppmann 2000). As such, it promotes utilization of GIS capabilities by timely channeling spatial and non-spatial data from various sources within an organization (Landrum 2001). Currently available DMSs do not harness the power of an Enterprise GIS architecture, because these systems are designed either with stove-pipe or client-server architecture. In stove-pipe architecture, the three logical components of GIS (data, business logic, and client) are tightly coupled and distributed as a single entity. The data component contains the organization’s data, while the business logic component implements the functionality or business rules of an application, organization, or process, and the client component interacts with the business logic component to access the data component. The arrangement dictates a high level of interdependencies between tiers.

In a client-server architecture, the data and client components are separated, with the business logic component being integrated and, thus, tightly coupled with either data or client component. The tight coupling makes future changes or upgrades to these components difficult, which generally results in high maintenance costs and poses a risk of system failure. Additionally, the client conducts most or all of the processing, thereby leaving little, if any, to be done by the server. When multiple clients must be serviced, as in the case of a DMS, relying on a client to process data is highly resource intensive (in terms of computing power and time) and is, thus, generally unacceptable. As such, the concept of a customized client, in which different client applications are designed to meet various functional requirements, is preferable to a traditional client/server architecture but still poses impediments to flexibility and scalability. To optimize such an arrangement, a three-tiered architecture (with the capability of being an N-tiered design), is proposed, where there is a loose coupling and separation of at least three components. This arrangement is typical of an Enterprise architecture.

The three-tiered architecture is commonly referred to as N-tiered, which is an allusion to the unlimited number (N) of intermediary layers between the client and server. These layers reduce redundancies in both data entry and processing. In N-tiered architectures, the business logic components are explicitly restricted from being coupled to the client and data components (Morais 2000). The introduction of a middle component to capture business logic enhances performance, flexibility, and maintainability by centralizing process logic; Figure 2 shows a typical arrangement of this structure that has yet to be employed in commercially available DMSs. With other architectural designs, a change to business logic would need to be written into every application (Eckerson 1995). An N-tiered architecture circumvents this requirement, thereby promoting a substantially more efficient and flexible system.

Database System (Data Tier)

The data component in Figure 2 is represented by a database system, whose primary function is to store data. Depending upon the number of users who access the data and the complexity of the data, different database servers are used; Figure 2 shows three Oracle 9i database servers. Small volumes and simple data commonly used by a small number of people can be stored in standard data files, while large, complex data volumes simultaneously needed by many users require a special database management system to ensure integrity and longevity (Longley et al. 2001). As IMIS data is complex in nature and may be accessed by many users, spanning multiple organizations and geographic locations, it requires special database management system support. Specifically, there is a necessity for a standardized data format and an integrated data repository.

Importance of Standard Data Format

Along with public safety departments (i.e. police and fire), emergency managers must coordinate their efforts with city planners, building officials, public works officers, and other governmental agencies, in order to implement many of the disaster management phases. Off-the-shelf desktop GIS systems are inadequate, because spatial data and attribute information are stored in proprietary file formats. Such file formats hinder sharing and integration, because they are not based on open standards, such as those recently developed by both governmental and non-governmental organizations. As examples of standardization, industry consortiums such as the Open GIS Consortium (OGC) have formulated the “Simple Features Specification” (OGC 2003), while the United States Geological Survey (USGS) has established the “Spatial Data Transfer Standard” specification for standardized data exchange

(SDTS 1997). These common standards help to fill a key need of an Enterprise GIS system; that of non-proprietary data storage formatting.

Importance of an Integrated Data Repository

One of the major problems faced during the New York City (NYC) rescue operations after September 11, 2001 was the lack of an integrated data repository (Flax et al. 2002). Data sets from different organizations had to be collected, before any computer-based search and rescue operations could be undertaken. The enormity of the situation highlighted the acute and immediate need for accurate and compatible spatial and attribute data. During a disaster, decision makers, incident commanders, emergency responders, and city mangers need spatial data instantaneously. Since timing is critical during all Response operations, all potentially required data sets must be pre-stored in a centralized repository or pre-connected to an integrated repository. This ensures that mission-critical data can be retrieved and redistributed to different agencies or personnel in a timely manner.

The NYC problem was exacerbated by the fact that although there had been a significant investment in the establishment of an Emergency Mapping and Data Center (EMDC), the center was located at WTC and was entirely destroyed during the attack. There was no backup. The EMDC had to be completely reestablished off-site, to aid the search and rescue operations. What should be understood from this example is that an integrated repository must not be not confused with a repository having a single point of failure. The system's actual implementation should consist of multiple repository hardware components (different servers) that are physically distributed across multiple locales. From the user's perspective, however, all the required data sets are visible and available, as if there was a single physical repository, and any updates made on one data set are immediately reflected throughout the distributed system. An integrated repository that utilizes a distributed database precludes data loss or data access, but presents other challenges: (a) multi-version concurrency control (Bernstein and Goodman 1983), (b) recovery (Elmasri and Navathe 2000), (c) system security, and (d) support of multi-format data. The business logic to process the data stored in such repository system for meaningful analysis is contained in the middleware system.

Middleware (Business Logic Tier)

Middleware is a set of common services that enable applications and end users to exchange information across networks (Umar 1997). These services reside in the middle portion of the architecture: above the networking software and operating system and below the distributed applications (Bernstein 1996). Middleware technologies are generally categorized as either general or service-specific. General middleware technology (e.g. Java 2 Enterprise Edition and Microsoft .NET) provides a common framework to develop business logic, while circumventing the need to know the technical details of the intended use, underlying hardware, operating systems, and communication protocols. These frameworks are not tied to any service specific technology.

In contrast, service-specific middleware technology serves a specific domain and can be customized to facilitate a particular task. For example, contemporary database management systems cannot handle spatial data in an efficient and intelligent manner, because they do not contain the basic structures for topology and geometry, whereas middleware for a spatial database may be designed explicitly for that purpose. Many commercial spatial database middleware products are available, but the following criteria must be met to provide full functionality in a DMS:

- Provide the required infrastructure to manage multiple users
- Provide business logic software to support advanced GIS data types (e.g. images, networks, features with integrated topology, and shared geometry) and associate these various data types with rules, behaviors, and other object properties
- Allow GIS data to be directly maintained in the format of "spatial types" supported by the database management systems vendors
- Integrate spatial search capabilities provided by database management systems vendors within the GIS client software applications

In general, a DMS requires a number of middleware systems that are both general and service specific. The efficiency of the entire system is highly dependent on the efficacy and proper use of middleware. For example, performing spatial queries that employ huge network data sets is a resource-intensive task for the middleware. Such queries mandate that the DMS exploits processing power advancements such as parallel or distributed computing. Middleware helps to achieve increased flexibility and enhanced performance by: (1) distributing the GIS application between the client layer, the application server, and the database server; (2) improving data integrity; (3) reducing database and application development costs, and (4) facilitating ubiquitous computing. Middleware fosters and facilitates the development of customized client applications that comprise the client tier.

Customized User Interface (Client Tier)

When the middleware provides the business logic to manipulate the data, customized client applications can simply access pre-existing business logic to fulfill most needs. This architecture helps to develop client applications that contain minimal business logic, thus reducing development time and costs. For instance, the middleware may provide a service (business rule/logic for a DMS) to find the shortest route to the nearest hospital from a given location. To accomplish this, the system must identify hospital structures, locate the point of initiation, have access to a transportation model, and conduct a spatial analysis to optimize routing options. Then, a handheld GPS device with limited computing power (as carried by ambulance personnel) or a powerful personal computer (as employed by other rescue personnel) can use the same service, irrespective of computing power, because the logic to process the complex spatial query resides in the middleware and not in the handheld device or personal computer. Such middleware-intensive arrangements foster the development of client applications with minimal functionalities.

DIORAMA – A PROTOTYPE IMIS FOR DISASTER MANAGEMENT

The infrastructure management information system DIORAMA was designed as a multi-phase DMS. Prototype development occurred at the rate of approximately 10 hours per week over 18 months with the following objectives: (1) look at a single aspect of one phase of a disaster, specifically the data entry and retrieval necessary for the categorization) of blast damage from the 2001 WTC terrorist attacks, (2) develop a GIS-based map of the portion of Manhattan that could support the inclusion of the over 300 blast-damaged buildings, (3) build a database sufficiently detailed to collect and categorize the necessary data to conduct blast damage categorization, (4) spatially query the system. To these ends, the main objective technical was to build a system that separated the three critical tiers (i.e. data, business logic and client). Prototype development required knowledge of system design, GIS, database data modeling, and Java. DIORAMA's database system, middleware, customized client interface, and general functionality are described below.

Database System (Data Tier)

DIORAMA's system architecture utilizes Oracle 9i database server for its data tier, which was chosen for its capacities to both serve as a central repository for spatial data and support integration of spatial data with other core organizational data. The server can supports multiple users, exploits enhanced database management features (e.g. administration and maintenance utilities, replication, and fast backup and recovery), and employs Structured Query Language (SQL) – an open application, programming interface; SQL is a comprehensive database language that standardizes data definition, querying, and updating tasks, as established by the American National Institute (Elmasri and Navathe 2000).

Such a database has broad storage capacity (e.g. vector data of building footprints, street centerlines, and other linear features in an Environmental Systems Research Institute (ESRI) geodatabase formats in use with ESRI's ArcSDE application server). A geodatabase physically stores spatial data inside a database management system, thus enabling spatial data to be kept in standard relational databases (MacDonald 2000). ArcSDE uses the default binary schema for an Oracle database server, which is fully compliant with the OGC Simple Features Specification for SQL's binary geometry (ESRI 2003). At the time of DIORAMA's development there were three database systems with adequate capabilities in common usage (IBM's DB2, Oracle's and Microsoft's SQL). The widespread availability of Oracle's database was the main reason for its selection.

A relational data model [a conceptual representation of the data structures required by the database (Adams et al. 2002)] was designed and implemented to store building attribute information. The attribute information was that, which was required to conduct post-blast building damage identification. Blast analysis in urban areas ultimately requires highly sophisticated, 3D capabilities to predict energy propagation and dissipation (Figure 3). Supporting this type of application is critical in a DMS and was primary in the prototyping DIORAMA.

Figure 3 shows a simplified conceptual data scheme of DIORAMA's database represented using the Unified Modeling Language; an industry standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems (Rumbaugh et al. 1999). Within the scheme are various entities (e.g. BUILDING entity, STORY entity, FAÇADE entity) that model a building as an entity consisting of a sequence of façades, stories and building components. The entity BUILDING captures the attribute information of a building (e.g. building name, street address, structural system type, date of construction), while BUILDING_FOOTPRINT stores its plan as spatial data (i.e. geometry and topology). The entities STORY, FAÇADE, and WINDOW capture attribute data of the building's story, façade, and window, respectively using simple text and numerical data types. Computer aided design (CAD) drawings and images are stored as a binary large object (BLOB) data type (e.g. WINDOW_ARCHITECTUREBLOB, FAÇADE_PHOTOBLOB). Unfortunately, the BLOB data format cannot be read as meaningful spatial information without a spatial middleware system, as described below. Figure 3 illustrates a few attributes required to support one application – blast damage identification. As such, special emphasis is given to window and façade features. A complete enumeration of the functionality of this data model is provided elsewhere (Pradhan 2003).

Middleware (Business Logic Tier)

DIORAMA's middleware is comprised of ESRI ArcSDE server, an ESRI ArcIMS server, and a HTTP server from Apache Software Foundation for its business logic component (Figure 2). The ESRI ArcSDE server functions exclusively as a spatial database-specific middleware and enables a standard database management system to store and manage geographic data by adding a spatial data type to a relational data model. The server facilitates the storage of a geographic feature via a row of the database table and the coordinates that represent the geometry of the feature as a single binary object value in a column of the table (Miller and Shaw 2001). Additionally the server supports significant functions and capabilities, including the geometric data management required for complex topological analysis. The business logic to perform spatial analysis is also embedded in the ArcSDE server. Thus, the client applications need not provide the business logic, only invoke the required functionalities.

ArcIMS provides the foundation for distributing high-end GIS and mapping services via the Internet (ARCIMS 2003). ArcIMS, with the Apache HTTP server, enables users to integrate local data sources with Internet data sources for display, query, and analysis in an Internet browser. Thus, they provide an infrastructure to web-enable DIORAMA. Unlike most GIS vendors, ESRI provides a wide variety of GIS software based on either two-tier to three-tier architectures. Additionally, at the time of DIORAMA's development none of ESRI's competitors offered GIS software that could simultaneously support three tiers and an Enterprise system. As the main focus of the prototype was the development of a three-tiered, Enterprise DMS, ESRI products were generally selected: DIORAMA relies on concepts implemented in other Enterprise GIS, as the objective was not to create a new Enterprise GIS architecture, but to understand and prototype an existing GIS architecture to enhance DMSs.

Capturing Business Logic

"Use Case" modeling is the de facto technique for performing software requirements analysis, thereby capturing the business logic for a given system. Use Case modeling describes the system behavior of the target system from an external point of view (Fowler et al. 1999) by capturing the functional and behavioral requirements of the system that help the users perform their tasks. Use case diagrams contain actors (anyone or anything interacting with the business logic) and use cases [actions pertaining to business logic that benefit the initiating actor(s)]. Examining the actors and defining their capabilities with the system establishes how best Use Case deployment. Use Case diagrams are typically represented in Unified Modeling Language (Rumbaugh et al. 1999) to (1) capture system requirements and (2) communicate with end users and domain experts.

Since all system needs can rarely be covered in one Use Case, there is typically a collection of Use Cases. Figure 4 depicts the process for querying information from the database to the end user (database administrator or data entry personnel). The user logs into the system and verifies user credentials. As per pre-specified access privileges, the user queries information (spatial and attribute) stored in the database system. Results are displayed on the client application.

After Use Case modeling, realization is conducted, which describes how a specific Use Case is realized in terms of the action sequence the actor invokes by notifying the system (Figure 5). A complete enumeration of DIORAMA's various use cases is provided elsewhere (Pradhan 2003).

Customized User Interface (Client Tier)

For DIORAMA, a GIS client application was developed for a specialized solution to enter/update/delete data and perform spatial queries (Figure 6). This was the prototype's main emphasis to eventually assess comparative building performance from the WTC attacks. The application provides a graphical user interface (GUI), with mapping features enabling insertion, updating, and deletion of various data (e.g. photographs and CAD files). The GUI also supports the formulation of spatial and attribute-based queries. Query results are displayed graphically and tabularly. The application uses Java, a purely object-oriented language designed to enable the development of secure, high performance applications on multiple platforms in distributed networks (Gosling and McGilton 1996).

The GUI was developed using the Java Foundation Classes library, as it provided an extensive set of technologies to create an interactive format for client applications that run on multiple platforms (JFC 2003). In addition, ESRI's Java MapObjects 1.0 library was employed to perform geographically-based display, query, and data retrieval activities (MAPOBJECTS 2003). This application was based on a Model View Controller design pattern similar to the Java Application Framework (Sunkpho and Garrett 2003).

Features

The prototype customized client application was designed for usage ease and sophisticated spatial and attribute-based queries as illustrated below.

Usage Ease

Because real systems employ data entry personnel with minimal computer training, usage ease is critical. Apart from standard GUI components (e.g. menus, toolbars, textboxes, window interfaces), the DIORAMA's GUI provides visual maps displaying building footprints, street networks, and other infrastructure. While entering/updating data, personnel can visually associate data with a particular geographic entity – selecting a specific building from the map and displaying a list of available menu options to enter/update/delete building attribute data through multiple, interconnected data entry frames. The initial frame focuses primarily on building location, thereby establishing a spatial reference frame. Subsequent screens request increasingly detailed levels of information predominantly related to building materials, geometry, and configuration. A complete enumeration of DIORAMA's functionality, including flow diagrams, data structure definition, and user interface screen, is available elsewhere (Pradhan, 2003).

Support for Spatial and Attribute-based Queries

The client application enables complex spatial and attribute queries based on a single attribute table. Without customization, data must be distributed among multiple tables, which requires user comprehension of the underlying data structure to query, which may require programming skills. For instance, ESRI ArcMap users need to understand concepts of "Joins" and "Relates" to associate data stored in tables with geographic features. When the user "Joins" two tables, the attributes from one table are appended onto the other, based on a field common to both tables. "Relates" only defines a relationship between two tables (also based on a common field) but does not append the attributes of one to the other. DIORAMA, however, precludes the need of SQL language knowledge to formulate spatial and attribute queries.

In DIORAMA, the user simply selects the desired fields from each table and then specifies the conditions related to that field. A "Query Interface" screen provides further simplification by providing check boxes for query formulation and additional screens with buttons for query formulation. For instance, boxes adjacent to Building Query, Façade Query, Storey Query, and Window Query each launch subsidiary screens of further details. Thus, the user need know only selection criteria, as the interface constructs the query (Figure 6). Figure 7 depicts sample query results (tabular and graphical). Additionally, architectural drawings are available by selecting the image identified under an "architectural drawing" column.

As an example, DIORAMA is able to query damaged and undamaged structures based on extremely general or highly specific criteria. For instance, the system can search for every building that had at least 10 percent of its windows damaged, or it can search for every building built between 1914 and 1928 that lost between 20 and 30 percent of its windows at the height of at least 10 m above the ground on the façade that faces away from the World Trade Center complex. The system can display the results both graphically and tabularly, which facilitates establishing the perimeter of the impacted zone. Following large amounts of additional data entry, the system will be able to provide key pieces of fundamental information related to blast damage distribution. This will be done by coupling the results fields with algorithms that determine least distance from a single point of interest, such as where each hijacked plane was estimated to be when the fuel tank exploded. Such information will greatly aid understanding of both existing building vulnerability and blast energy dissipation in a non-free field environment.

CONTRIBUTION

Presently, DIORAMA supports data entry, update, retrieval, spatial querying, and two-dimensional spatial visualization. One component that distinguishes DIORAMA from off-the-shelf GIS software is DIORAMA's ability to directly query from multiple normalized tables, as in the case of normalized database schema. Off-the-shelf software accommodates only simple table structures. In DIORAMA, complex queries using multiple normalized tables can be performed directly with an intuitive GUI, which obviates the user having to understand complex table relations. Similarly, data entry/updating is conducted via a GUI – customized to a level not available in off-the-shelf products. To achieve these capabilities, as well as those outlined in the general criteria, the major area of customization implemented was in the system architecture, which was based on a distributed, three-tier architecture (data, business logic, and client). Each tier was designed and implemented individually and is installable on separate node. Node system heterogeneity is not a main issue as communication between the data tier and middle tier is done using SQL (supported by most existing database vendors) and there is support for Java – the selected (platform independent) programming language.

DIORAMA's strengths include the following: (a) reliable architecture through use of a relational database (data tier), spatial middleware (business-logic tier), and customized Java MapObject client (client-tier), in contrast to off-the-shelf GIS software, in which all components are coupled as one monolithic system; (b) non-resource intensive Java MapObject client applications, instead of resource intensive, commercial GIS software, (c) straightforward data sharing, since data is stored based on OGC open system standards (Simple Features Implementation Specification); and (d) simplified future upgrading, as business rules are transparent and coded for specific domains, as opposed to proprietary GIS applications, where business rules are coded to encompass many domains of expertise and are treated as black box applications. These features are fundamental to the ultimate use of DIORAMA as a multi-phase, disaster management system with ubiquitous computing capabilities.

DIORAMA's main, current functionalities can be summarized as follows: storage of spatial information in an Open GIS Consortium compliant data format, handling of complex spatial and attribute-based queries with an easy and intuitive GUI, allocation of resource intensive computation in real time, and support of multiple users for simultaneous accessing and updating within a common spatial repository. This combination of off-the-shelf and customized functionalities are enabled by the construction of a three-tier, Enterprise level GIS system. DIORAMA also has interoperability by usage of a multiple platform compatible language (Java) and data storage in an OGC compliant file format, although no special tools or methodologies supported by OGC were implemented. System validation was based upon four items: (1) level of user friendliness, (2) ease of data entry/storage, (3) intuitive querying, and (4) real time processing speed. The first three were verified by successful undergraduate usage of the system and the fourth by data entry and retrieval without perceived interruption in user interaction. Follow on work includes large quantities of data entry and high-level analysis development for categorization of blast-energy dissipation in a non-free field environment.

CONCLUSION

Current DMSs are presently focused on an overly restrictive definition of disaster management and, thus, do not adequately address the multi-phase nature that is fully reflective of a community's disaster management needs. As such, the architecture of existing systems heavily impedes full applicability and further expansion of these systems, as a direct outgrowth of their hardware and software limitations. By adopting an alternative architecture based on an Enterprise GIS framework, the six disaster management phases (*Identification*, *Prediction*, *Preparation*, *Mitigation*, *Response*, and *Recovery*) can be realized in terms of customization, computing resource distribution, data sharing, and upgrading. An integrated disaster management system based on an Enterprise system that supports GIS, a relational database, and multiple data formats offers substantial advantages over existing standalone systems in terms of data collection, administration, retrieval, distribution, and usage. The prototype architecture demonstrates the constructability of such a system through the employment of a data repository, a middleware system customized for disaster management, and customized client applications for querying blast damage in an urban area. The prototype exploits recent innovations in both hardware and software: enabling tabular and graphical queries for data analysis. From a software architecture perspective, the proposed prototype represents an advance over existing DMSs, because despite widespread industry support for three-tier software architecture, it has yet to be adopted by the disaster management community.

REFERENCES

Adams, T.M., Malaikrisanachalee, S., Blazquez, C., Lueck, S. and Vonderohe, A. (2002). "Enterprise-wide data integration and analysis for oversize/overweight permitting," *J. Comp. Civil Eng*, 16(1), 11-22.

ARCIMS (2003). "ArcIMS – GIS for the Internet," <http://www.esri.com/software/arcims/index.html> (Sept. 30, 2005).

Bernstein, P. A., and Goodman, N. (1983). "Multiversion concurrency control - theory and algorithms." *ACM Transactions on Database Systems*, 8(4), 465-483.

Bernstein, P. (1996). Middleware: a model for distributed systems services, *Comm. ACM*, 39(2), 86-98.

Cutter, S. L. (2003). "GI Science, Disasters, and Emergency Management." *Trans. GIS*, 7(4), 439-46.

Deaton, S. L. and Frost J. D. (2002). "Integrated Digital Earthquake Reconnaissance." *urban earthquake risk, 7th U.S. nat'l conf. earthquake eng.*, (CD-Rom) EERI.

DTRA (2002). "Consequences Assessment Tool Set - Joint Assessment of Catastrophic Events" <http://www.dtra.mil/press_resources/fact_sheets/display.cfm?fs=cats-jace>(Sept. 30, 2005).

Eckerson, W. W. (1995). "Three tier client/server architecture: achieving scalability, performance, and efficiency in client server applications." *Open Information Systems* 10, 1(3), 20.

Elmasri, R. and Navathe, S. B. (2000). *Fundamentals of database systems*, 3rd ed., Addison-Wesley, Lebanon, Indiana, Chpt 8, 243-244 and Chpt 19, 329-357.

ESRI (1999). "GIS for Emergency Management." ESRI White Paper. http://www.esri.com/library/whitepapers/pdfs/homeland_security_wp.pdf(Aug. 30, 2005).

ESRI (2003). "Standards and interoperability - newsletter", < http://www.esri.com/library/newsletters/standards-interoperability/sinews-summer03.pdf : 2005-08-05> (Sept. 30, 2005).

Flax, L. K., Jackson, R. W., and Stein, D. N. (2002). "Community vulnerability assessment tool methodology." *Natural Hazards Rev.* 3(4), 163–176.

Fowler, M. and Scott, K. (1999). *UML distilled: a brief guide to the standard object modeling language*, 2nd ed., Addison-Wesley, Essex, U.K., Chpt 3.

GAO (2003). "Major management challenges and program risks- federal emergency management agency." *Performance and accountability series*, U.S. General Accounting Office, Washington, D.C., January, 6.

Godschalk, D. R. (2003). "Urban hazard mitigation: creating resilient cities." *Natural Hazards Rev.*, 4(3), 101-167.

Goodchild, M. F. (2003). "Geospatial data in Emergencies." *The geographical dimensions of terrorism*. Ed.s Cutter, S. L., Richardson, D. B., and Wilbanks, T. J. Routledge, U.K., 99-104.

Gosling J. and McGilton H. (1996). "The java language environment," <http://java.sun.com/docs/white/langenv> (Sept. 30, 2005).

Greene, R. W. (2002). *Confronting catastrophe: A GIS handbook.* ESRI Press, Redlands, CA.

HAZUS (2003). "HAZUS- multi disasters, a loss estimation software." <http://www.fema.gov/hazus/hz_overview.shtm>(Sept. 30, 2005).

JFC (2003). "Java foundation classes (JFC) – cross platform GUIs & graphics." <http://java.sun.com/products/jfc/>(Sept. 30, 2005).

Johnson, G. (2003). "Solving disaster management problems using ArcGIS," <http://campus.esri.com> (Nov. 9, 2003).

Laefer, D.F., Koss, A.L. and Pradhan, A.R. (2005a). "The Need for Baseline Data Characteristics for GIS-based Disaster Management Systems," *J. Urban Planning*, (in press).

Laefer, D.F., Pradhan, A.R., and Koss, A.L. (2005b). "GIS-based Disaster Management Systems: a Cogent Data Framework." *Trans. Res. Board* (in press).

Landrum, T. B (2001). "Enterprise GIS – Building a Strong Foundation," <http://www.gisdevelopment.net/proceedings/gita/2001/lgp/lgp005.shtml>(September 30, 2005).

Liu, D., Cheng, J., Law, K. H., Wiederhold G., and Sriram, R. D. (2003). “Engineering information service infrastructure for ubiquitous computing.” *J. Comp. Civil Eng*., 17(4), 219-229.

Longley, P. A., Goodchild, M. F., Maguire, D. J., and Rhind, D. W. (2001). *Geographic Information Systems and Science*, 1st ed., John Wiley & Sons, Ltd, West Sussex, U.K. Chapter 11, 226.

MAPOBJECTS (2003). “MapObjects – Java edition”, <http://www.esri.com/software/mojava/>(Sept. 30, 2005).

MacDonald, A. (2000). *Building a Geodatabase*, ESRI, Incorporated, Redland, CA.

Mansourian, A., Rajabifard, A. Valadan Zoej, M. J., and Williamson, I. (2005). “Using SDI and web-based system to facilitate disaster management.” *Computers and Geoscience*. (In press).

Merriam-Webster (2003). *Merriam-Webster’s Eleventh Collegiate Dictionary*, Merriam-Webster, Springfield, MA

Miller. H. J. and Shaw, S. (2001). *Geographic Information Systems for Transportations*, 1st ed., Oxford University Press, Oxford, U.K., Chpt. 7, 219.

Morais, M., (2000). Realizing the benefits of an n-tiered enterprise GIS <http://www.gislounge.com/features/aa062100.shtml>(Sept. 30, 2005).

OGC (2003). “OpenGIS® Implementation Specifications” <http://www.opengis.org/techno/implementation.htm>(September 30, 2005).

Penã-Mora, F. and Dwivedi, G. H. (2002). “Multiple device collaborative and real time analysis system for project management in civil engineering.” *J. Comp. Civil Eng.*, 16(1), 23-28.

Pradhan, A. R. (2003). *Infrastructure management information system framework for disasters*, MS thesis, Dept. Civil, Construction, and Environmental Eng., North Carolina State Univ.

Radke, J. T., Cova, M. F., Sheridan, A. ,Troy, L. Mu, and R. Johnson (2000). “Challenges for GIS in emergency preparedness and response. ESRI white paper.” http://www.esri.com/library/whitepapers/pdfs/challenges.pdf(Aug. 30, 2005).

Rasdorf, W. J., Harris, E., and Laefer, D. F. (2005). Common data collection needs: A North Carolina emergency management community perspective, Nat’l Science Fndn Final Report.

Rumbaugh, J., Jacobson, I., and Booch, G. (2004). *The unified modeling language reference manual*, Essex:U.K. Prentice Hall, Pearson-Professional Education, Chpt. 1.

SDTS (1997). American national standard for information systems – spatial data transfer standard (sdts) – part 1. logical specifications, Draft for review, Am. Nat'l Standard Inst., New York.

Sunkpho J. and Garrett, J. H. (2003). "Java inspection framework: developing field inspection support systems for civil systems inspection," *J. Comp. Civil Eng.*, 17(4), 209-218.

Uddin, N., and Engi, D. (2002). "Disaster management system for southwestern Indiana." *Natural Hazards Rev.*, 3(1), 19-30.

Umar, A. (1997). Object-oriented client/server internet environments: Developing new applications and dealing with legacies, Prentice Hall, Inc. Essex,U.K.

USFDA (2003). <http://www.usaid.gov/our_work/humanitarian_assistance/disaster_assistance> (Sept. 30., 2005).

Von Meyer, N. R. and Oppmann, R. S. (2000). "Enterprise GIS." Urban and regional information systems association, Park Ridge, IL, Chapter 1.

Weiser, M. (1993). "Some Computer science problems in ubiquitous computing." *Comm. ACM*, July. (reprinted as "Ubiquitous computing", Nikkei Electronics, Dec. 6, 1993, 137-143).

Zerger, A. and Smith, D. I. (2003). "Impediments to using GIS for real-time disaster decision support," *Comp., Envir. Urban Sys.,* 27, 123-141.

Table 1. Sampling of Disaster Categories

Disaster	Cause		Location			Impacted Community	
	Natural	Manmade	Land	Water	Air	Urban	Rural
Tornado	X		X	X	X	X	X
Hurricane	X		X	X	X	X	X
Flood	X		X	X		X	X
Ice storm	X		X			X	X
Earthquake	X		X	X		X	X
Building collapse	X	X	X			X	
Explosion		X	X		X	X	
Fire	X	X	X			X	X
Hazardous liquid spill		X	X	X	X	X	X
Biological outbreak	X	X	X	X	X	X	X
Nuclear contamination		X	X	X		X	X
Oil spill		X	X	X		X	X

Table 2. Sampling of Existing Systems' Capabilities

Supported features / Existing systems	Support for baseline system	Analysis required for six disaster phases	Support for ubiquitous computing
HAZUS-MH	Available	a, b, d	Not available
CATS	Available	a-d	Not available
Deaton and Frost	Not available	a, f	Available

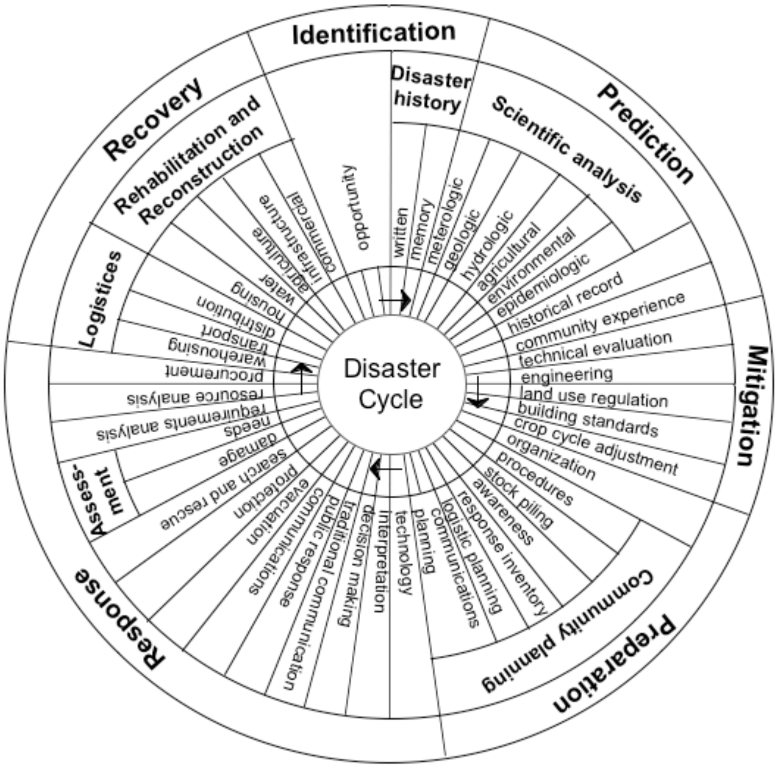


Figure 1. Disaster Preparedness Phases (adapted from Johnson 2003 and USDFA 2003)

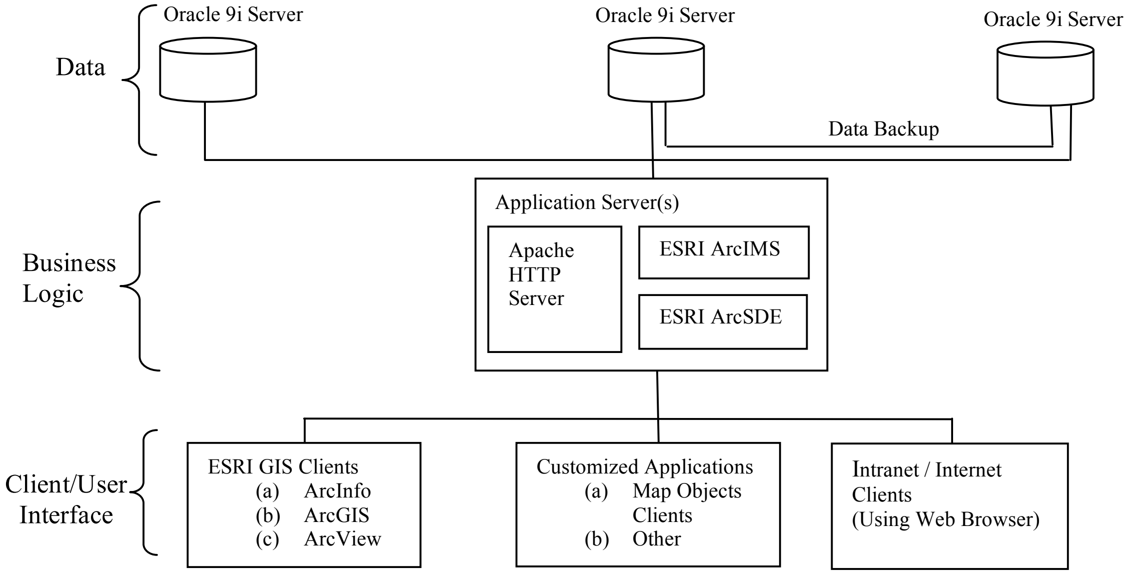


Figure 2. DIORAMA's 3-Tiered System Architecture Design

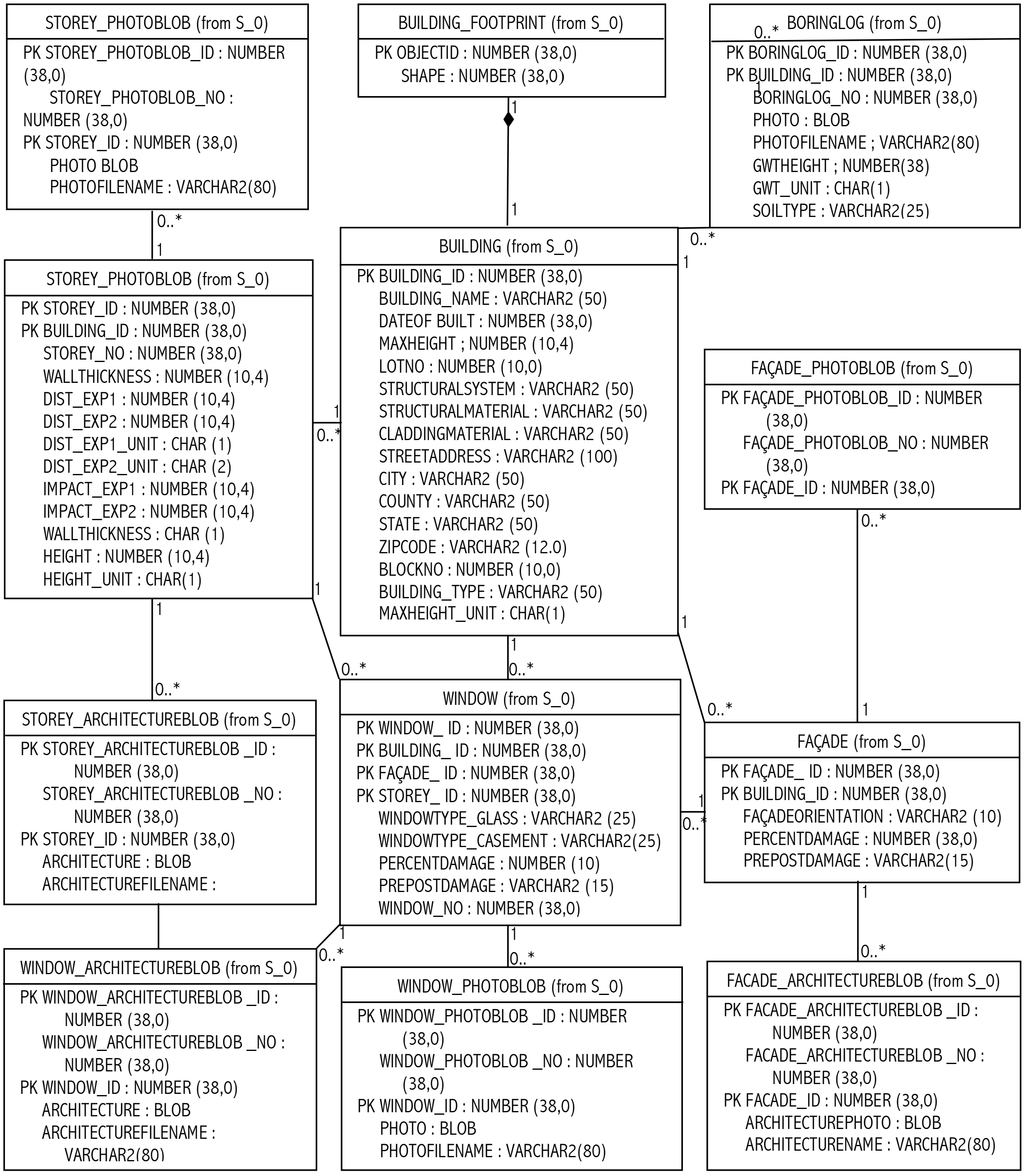


Figure 3. UML Conceptual Schema for DIORAMA's Database (Pradhan 2003)

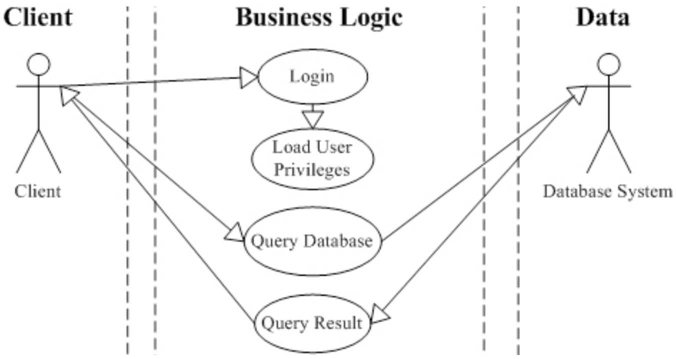


Figure 4. Use Case Modeling for Querying Database

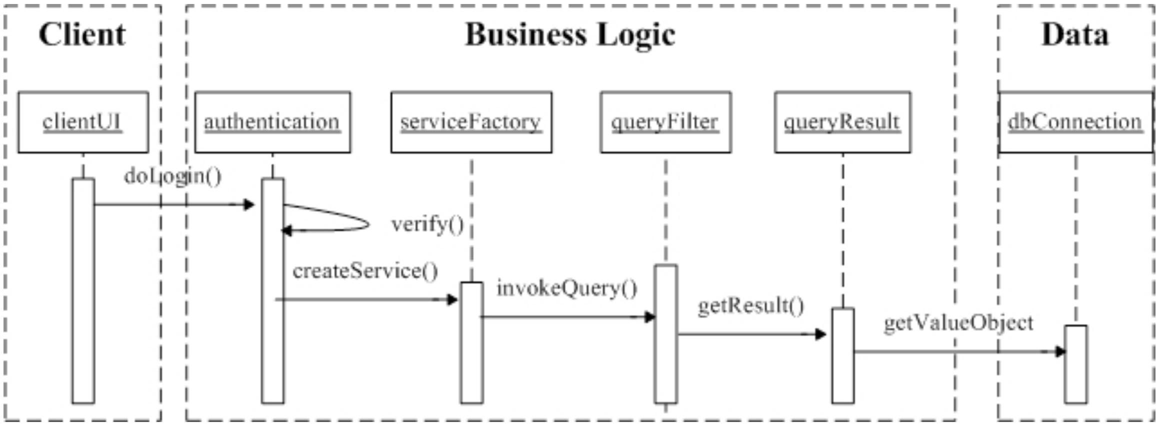
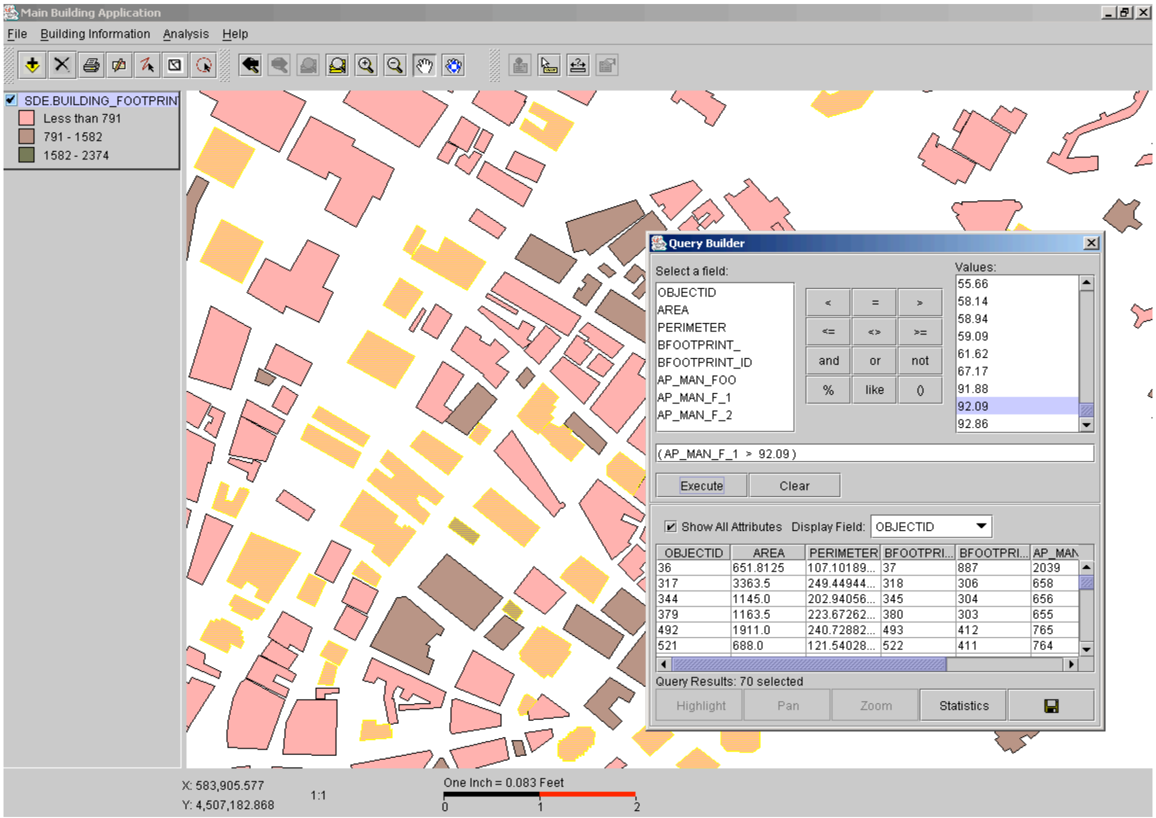


Figure 5. Sequence Diagram for Querying Database



☑ SDE.BUILDING_FOOTPRIN
- Less than 791
- 791 - 1582
- 1582 - 2374

(AP_MAN_F_1 > 92.09)

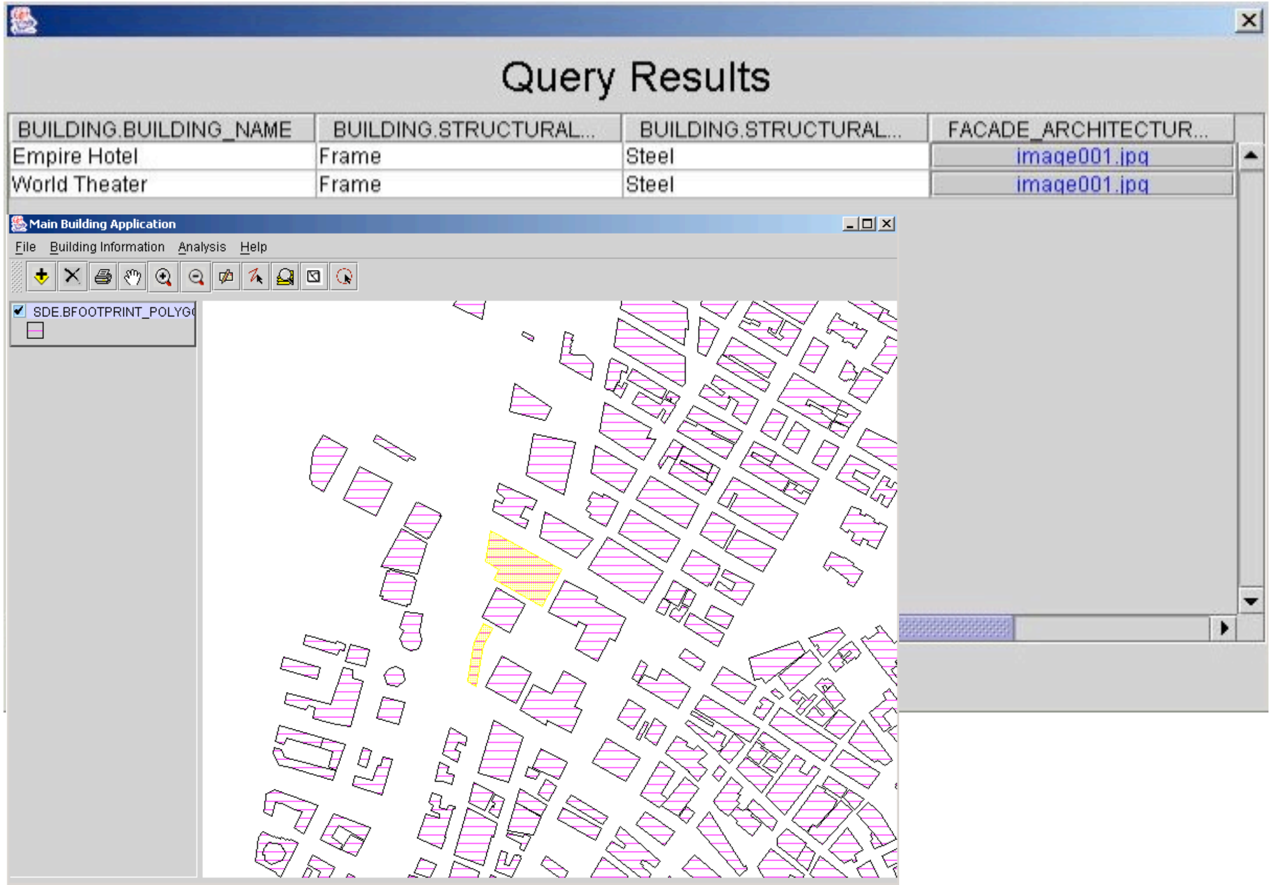
☑ Show All Attributes Display Field: OBJECTID

OBJECTID	AREA	PERIMETER	BFOOTPRI...	BFOOTPRI...	AP_MAN
36	651.8125	107.10189...	37	887	2039
317	3363.5	249.44944...	318	306	658
344	1145.0	202.94056...	345	304	656
379	1163.5	223.67262...	380	303	655
492	1911.0	240.72882...	493	412	765
521	688.0	121.54028...	522	411	764

Query Results: 70 selected

X: 583,905.577
Y: 4,507,182.868

Figure 6. Example of Querying Portion of Graphical User Interface



Query Results

BUILDING.BUILDING_NAME	BUILDING.STRUCTURAL...	BUILDING.STRUCTURAL...	FACADE_ARCHITECTUR...
Empire Hotel	Frame	Steel	image001.jpg
World Theater	Frame	Steel	image001.jpg

Figure 7. Graphical and Tabular Query Results

TABLE CAPTIONS:

TABLE 1. Sampling of Disaster Categories
TABLE 2. Sampling of Existing Systems' Capabilities

FIGURE CAPTIONS:

FIGURE 1. Disaster Preparedness Phases (adapted from Johnson 2003 and USDFA 2003)
FIGURE 2. DIORAMA's 3-Tiered System Architecture Design
FIGURE 3. UML Conceptual Schema for DIORAMA's Database (Pradhan 2003)
FIGURE 4. Use Case Modeling for Querying Database
FIGURE 5. Sequence Diagram for Querying Database
FIGURE 6. Example of Querying Portion of Graphical User Interface
FIGURE 7. Graphical and Tabular Query Results