The Gaea System: A Spatio-Temporal Database System for Global Change Studies *

Nabil I. Hachem, Michael A. Gennert, Matthew O. Ward
Department of Computer Science
Worcester Polytechnic Institute
Worcester, MA 01609, USA
e-mail: hachem,michaelg,matt@cs.wpi.edu

Abstract

The Gaea system is a spatio-temporal database management system under development at Worcester Polytechnic Institute. Gaea is intended to provide advanced data management and analysis to geographical information systems (GIS) for global change studies. We present the objectives and long-term vision of the Gaea project, describe the Gaea system architecture and discuss the current state of development.

1 Introduction

1.1 Significance

In an office of the not-too-distant future, a climatologist sits down at her computer workstation and types in a simple question, "Has the rainfall across Southern California been affected by the shrinking rain forests of Brazil?" In a few moments, the screen is filled with a patchwork of images, icons and menus.

Using the computer's mouse to link together certain pictures and icons, then typing simple answers to questions the computer poses, she sharpens her search and then goes beyond her initial query, drawing in data on wind patterns, atmospheric gas exchange and solar radiation—some of it drawn from computers located thousands of miles away—and integrating it with information coded in weather satellite images pulled from a data bank in Washington, D.C. Finally, she displays the results of her new computer model as a time-sequenced series of high-resolution maps.

It may sound like science fiction, but this is the ultimate goal of the Gaea system, a project being undertaken by a research team in WPI's Computer Science Department and in the George Perkins Marsh Institute at Clark University. The team hopes to take the first steps toward a radical new way of handling the huge amounts of information that confront scientists studying the phenomenon of global change.

For the most part, current scientific databases can't easily handle the staggering quantity of data often gathered in geographic or climatological research (weather satellites, for example, collect billions of bits of data every day). In addition, business-oriented databases generally lack the flexibility to adapt easily to new types of data or to new ways of analyzing it. This quality—called extensibility—is also required to accommodate meta-data, or data about the data, which can include information about how, when and where the data were created, and about how they have been processed or recalibrated in the meantime.

A goal of the global change database project is finding a way to integrate spatial and temporal data, making it easier for researchers to model and study how global systems change over time.

Another important goal of the global change database project is making the program as easy to use as possible. Current GIS force users to think in terms of bits and bytes, when users should be concentrating on the concepts represented by those bits and bytes. In order to make the system usable by non-computer scientists, the Gaea system lets a user describe an experiment in terms of concepts, automatically relating the concepts to their definitions. For example, a user may prefer to examine DESERTIC REGIONS without regard to their precise definition, which may vary from user to user. Another aspect of Gaea which addresses ease of use is its graphical user interface, which provides an intuitive front-end for browsing and querying the database as well as a visual language for the design and management of experiments.

*This work is supported by the National Science Foundation under Contract IRI-9116988.

The Gaea project started in October 1991, and the first phase will be completed by September 1993. The research team is developing a working prototype of the system that will be tested with a wide range of geographic, cartographic and remotely sensed data provided by Clark University. While the early prototypes of the new global change database software will be tested with data residing on computers at WPI and Clark, the team would ultimately like to give the system the ability to tap into international computer networks to gather information from diverse and potentially incompatible computer systems located around the world.

1.2 Project Goals

The long-term goal of the Gaea project is to develop an extensible, object-oriented data management and analysis system to be used by researchers in the field of global change [6].

The current goal is to develop a prototype, which can be used by geographers in a user-friendly manner, yet permits integration of diverse data types and interactive development of sophisticated methods for data analysis, prediction, and display. Focus is on the object manipulation and analysis aspects of the system as well as the management of "meta-data," that is, data about the data.

The general abilities of the system will include: 1) deriving information about spatio-temporal objects, 2) maintaining information on the evolution of data objects, 3) providing derivation semantics for data objects, 4) integrating data and analysis management of user-specified experiments, 5) providing a visual environment for browsing, querying, and analysis, and 6) providing user extensible data and operator types.

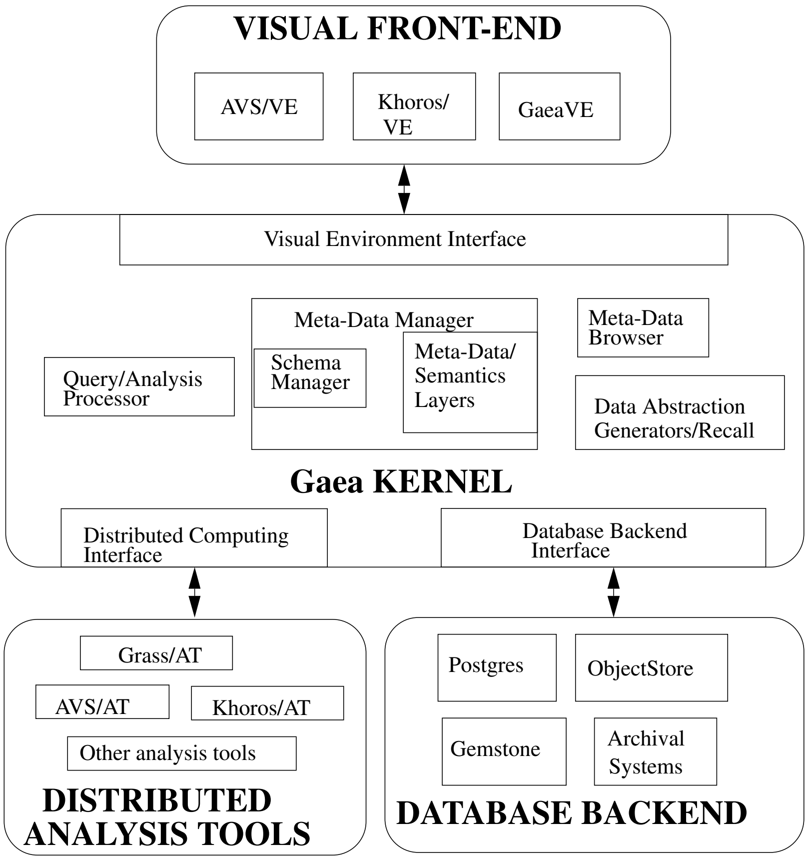

Figure 1: A Visionary Look into the Architecture of Gaea

2 Long Term Vision of the Gaea Architecture

Our long term vision is based on our view of a scientific data management and analysis environment which can be layered along three levels (Figure 1): 1) The visual frontend, which allows the user to pose visual queries, apply analysis operators to data, and visualize data, including analysis results; 2) the Gaea Kernel, which provides

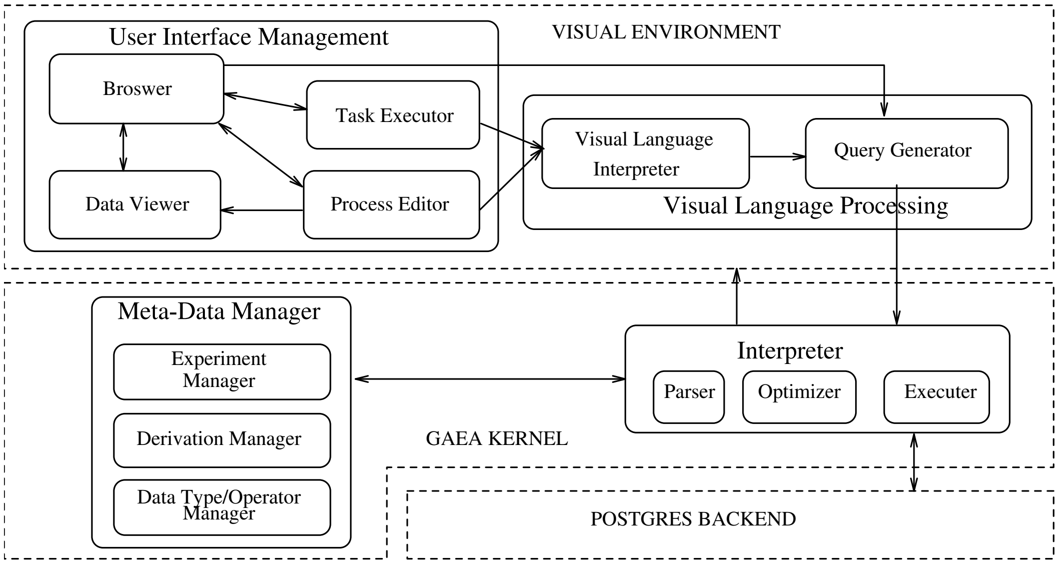

Figure 2: Architecture of the Gaea system prototype.

support for meta-data and converts simple queries from the visual frontend into a complex series of database accesses and operations; and 3) The Database Backend, which actually stores the data, providing network and archiving functions. We describe each subsystem in turn.

The **Visual Frontend** mediates all interaction with the user. Our objective is to provide sufficient flexibility so that a variety of popular visual environments can be interfaced to the Gaea Kernel. There exists many such packages, either commercial (e.g., AVS [2]) or publicly available (e.g., Khoros [7]). These visual environments come with complete analysis subsystems; we would like to make use of the frontends and analysis operators separately, as shown in Figure 1. In addition, we have written our own visual frontend tailored to the Gaea Kernel [9]. One challenge on which we are currently working is the definition of a query and analysis language in which any visual query can be expressed. When that language is defined and implemented, any visual environment may be incorporated into Gaea by converting commands into the common query and analysis language. At present, the query and analysis language is an extension of Postquel ([1], pp. 78–93), but more functional languages are also under consideration.

The most important function of the **Gaea Kernel** is the management of meta-data and the semantics of derived data. Users can query meta-data to obtain the meaning of derived data. Furthermore, capturing a data object's derivation process information enables the user to repeat that process and derive new data, given different input data. The kernel will include a schema manager which manages the meta-data and the associated derivation semantics and analysis operators (Figure 1). The Query/Analysis Processor (QAP) is responsible for processing queries, deriving new data whenever necessary, and using meta-data. The kernel includes a semantic and meta-data browser to allow a user to find relevant data without knowing specific file and path names. There is also a Data Abstraction Generation and Recall module which allows previously generated data to serve as a template for additional queries, i.e., queries can be abstracted. Finally, generic interfaces to the frontend visual environments and backend distributed computing and distributed databases and archives are provided.

The **Backend System** consists of distributed and archive databases such as Postgres, Object Store and Gemstone ([1], pp. 34–93). The distributed computing environment consists of scientific analysis operators which are available within commercial or public domain software systems. Examples are the analysis tools available within AVS, Khoros, and GRASS. These tools may be imported into Gaea because the meta-data manager will have registered information about analysis operators, their domains of application, data types and formats they apply to, among other meta-data. The Gaea kernel will be able to chose from these available tools and use them to provide a seamless integration between analysis and data management for scientific environments.

This visionary architecture of Gaea is detailed in [4]. We provide a description of the current Gaea prototype, discussing the core of the meta-data manager and overviewing the Gaea visual environment, in the next section.

3 Current View of the Gaea Architecture

The Gaea prototype architecture is divided along three levels (Figure 2): 1) The Gaea Kernel is the core of the prototype, providing the essential meta-data management capabilities; 2) The Visual Environment is designed to provide visual facilities for scientific experiment design, data definition and manipulation, including querying, and semantic browsing; and 3) The Postgres ([1], pp. 78–93) 3rd Generation DBMS serves as the backend.

3.1 The Gaea Kernel

The most important aspect of the Gaea Kernel is the meta-data manager. It provides a framework for capturing and managing scientific data derivation histories [5]. The actual meta-data are viewed by the system at three semantic levels (Figure 3): 1) The high-level semantics view which will provide the user with means to design and develop logical views of experiments; 2) The derivation semantics view which provides for the management of (scientific) derivations of data; and 3) The system-level semantics which are essentially the abstract data type (ADT) view of the system.

3.1.1 High-Level Semantics

This level records the information that is necessary for the understanding of a specific experiment. In global change research, it is difficult to agree on carefully designed experiments. The Gaea kernel supports experiments through the experiment manager module of the meta-data manager. This module is capable of manipulating conventional semantic modeling constructs. In addition, we introduce the notion of *concepts*. A **concept** is a representation of a spatio-temporal entity set, extended with an imprecise definition. Concepts are very common in scientific databases.

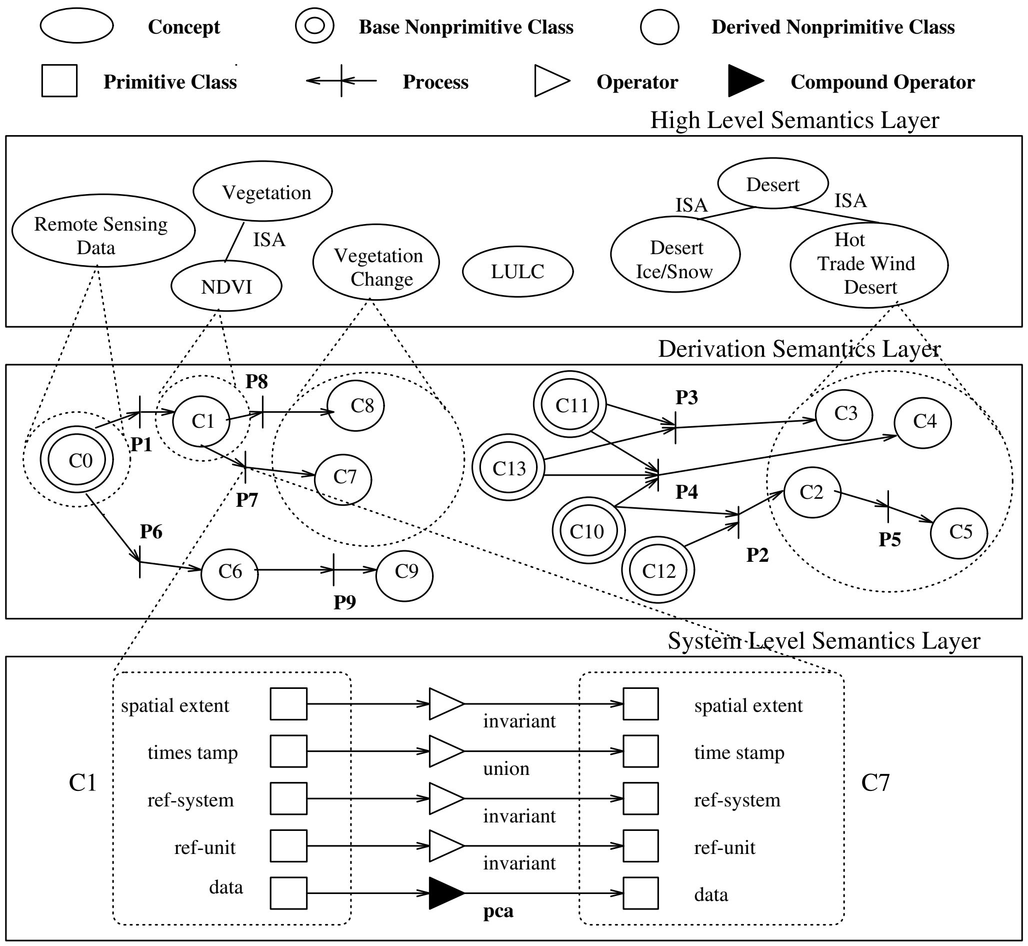

Figure 3: The three semantic layers in Gaea.

For excample, PERSON is an entity set with a well understood definition. It may be considered as a concept with a well defined and agreed upon meaning. In GIS, a DESERTIC REGION is an entity set whose definition may differ from one user to another. An acceptable definition of a desert must include consideration of the amount of precipitation received, the distribution of this precipitation over a calendar year, the amount of evaporation, the mean temperature during the designated period, and the amount and utilization of the radiation received. Furthermore, every one of those factors may have different metrics.

At this high level of abstraction, we model deserts with a specialization hierarchy (Figure 3). This hierarchy does not capture the relationships between other concepts involved in the definitions of deserts. While general relationships can be provided using the well proven semantic modeling technology, new semantics for data derivation are necessary.

3.1.2 Derivation Semantics Level

The leaves of the concept hierarchy in the high level semantic layer maps to a set of non-primitive classes in the derivation semantics layer (e.g. "hot trade-wind desert" map to the set of (non-primitive) classes {C2, C3, C4, C5} in Figure 3).

The derivation semantics layer records the derivation relationships among classes of data. Such relationships can also be used for the generation of new classes of data. Typically, when data are not stored in the database, we may generate the needed data with the help of such derivation relationships. The basic constructs used are: 1) A **Process** which captures the description of a scientific procedure used for the generation of new concepts from other concepts and 2) A **Task** which is the instantiation of a process with input data objects. Every task will generate a set of objects (most of the time just one) for the output class.

Formally, a process defines a mapping between a set of input object classes and an output object class. Essentially, the outcome of a process is a unique class which is a member of a concept. Thus, object classes which do not represent base data are solely defined by their derivation processes. In this way a process captures the semantics of data derivations.

3.1.3 System Level Semantics

The system level semantics of Gaea is responsible for the management of abstract data types (ADTs). Following the object-oriented paradigm, ADTs in Gaea are primitive classes encapsulated with the methods or functions applicable to them.

The mapping between the derivation semantics layer and the system layer consists of the mapping of a process as a transformation of a set of input classes to an output class using operators that are applied to primitive classes. For example, "vegetation change" can be derived as either class C7 or C8. Consider, for example, C7 to be derived using principle component analysis (PCA) which is part of process P7. The mapping between input and output attributes is shown in the lower portion of Figure 3. Observe that `pca` is a compound operator composed of a network of intercommunicating operators, whose structure is discussed in [5]. This network can be considered as a data flow network of functional operators that are applied on primitive classes, such as spatial coordinates, temporal attributes, and raster images.

3.2 Gaea Visual Environment

The Gaea Visual Environment provides the user interface to Gaea [9]. Its functionality can be decomposed into four interrelated activities (Figure 2).

The **Browser** permits perusal of the contents of the database, including concepts, data, operators, processes, experiments, or meta-data. Users specify actions in the browser by interacting with visual representations such as images, graphs, lists, and icons to indicate constraints on searches. Thus, to determine for which years there exists rainfall data for Australia, the user would draw a box around the section of the world map which contains Australia, browse all concepts available for Australia, choose rainfall from the result list, and specify the temporal resolution via a menu to be yearly. The time line then indicates the particular years for which the data exist.

The **Data Viewer** is a data visualization toolkit. Various types of data, such as tables, images, and vectors can be visualized using a set of visualization operators. Users can interactively specify the parameters for visualization operators. The results of a visualization, such as a map of vector data, can be used for specifying spatial context in the Browser.

The **Process Editor** permits the user to interactively specify a process via a visual language, whereby data and operators are linked together in a network. A user may choose to edit an existing process, which may then be linked via a concept, or create a process from scratch and associate it with a new or existing concept. The system performs consistency checking on the connections between nodes, and the Browser may be invoked to search the database for compatible operator or data classes. Processes can be tested in the Process Editor by specifying input data objects and results visualized by the Data Viewer. Parameters to be shared by operators may be linked via a point-and-click mechanism.

The **Task Executor** allows the user to associate specific data objects of specific concepts to a process or a series of processes and execute the result. As in the Process Editor, shared parameters may be indicated interactively. The Browser may also be used to locate and specify the desired objects. A fully specified task may be saved as an experiment, with meta-data extracted from the nodes and optionally supplemented by the user.

4 Summary

In the first phase of this project we identified the needs of the global change research community in the areas of data management, operators, and experiments. This led to an extensible design for the Gaea system architecture composed of the visual interface frontend, Gaea kernel, and database backend.

A prototype system is nearing completion in the second phase of the project. The prototype includes spatio-temporal data and operator models, data language, and the visual environment. Preliminary versions of the frontend, kernel, and backend have been written and serve as bases for additional work. Ongoing work includes the development of the visual browser and simplified object editors. A major accomplishment, the formalization of meta-data semantics, provides for the semantics of meta-data in a non-ad hoc manner.

The third phase will primarily address usability issues, including a history mechanism to cache the current spatio-temporal objects of interest and an entry processor to facilitate the importation of new data types into Gaea. We will also expand the operator set by including operators from other GIS, image processing, and scientific visualization packages.

Future phases will focus on query optimization, network access, and the application of Gaea technology to real-world scientific database problems such as NASA's Earth Observation System Data Information System and NSF's High Performance Computing and Communications Program. Based on our view of the scientific database landscape, the integration of very large databases with powerful analysis systems and intuitive interfaces will prove to be a development of immense importance, ultimately enabling us to answer the question, "Has the rainfall across Southern California been affected by the shrinking rain forests of Brazil?" and others.

Acknowledgments

We would like to thank Michael Dorsey for helping out with the writeup of the significance section. Also we extend our thanks and appreciation to the current members of the Gaea Project team including Robert Dugan, Ke Qiu, Yelena Yogneva-Himmelberger, and Yuhong Zhang.

References

[1] "Next-Generation Database Systems," Comm. of the ACM, Special Issue, Vol. 34, No. 10, Oct. 1991.

[2] Advanced Visual System Inc., 300 5th Avenue, Waltham, MA 02154. *AVS Technical Overview*, Oct. 1992.

[3] J. Westervelt, "Introduction to GRASS 4," U.S. Army CERL, July 1991.

[4] N. I. Hachem, M. A. Gennert and M. O. Ward, "Distributed Database Management for Scientific Data Analysis," Int. Workshop on Global GIS, ISPRS WG IV/6, Aug. 1993, Tokyo, Japan.

[5] N.I. Hachem, K. Qiu, M.A. Gennert, and M.O. Ward, "Managing Derived Data in the Gaea Scientific DBMS," Proc. Int. Conf. Very Large Databases, August 1993. (also WPI-CS-TR 92-08, Dec. 1992)

[6] N.I. Hachem, M.A. Gennert and M.O. Ward, "A DBMS Architecture for Global Change Research," Proc. ISY Conf. on Earth and Space Science Information Systems, Pasadena, CA, Feb. 1992.

[7] J. Rasure, D. Argiro, T. Sauer, and C. Williams, "A Visual Language and Software Development Environment for Image Processing," Int. J. Imaging Systems and Technology, Vol. 2, pp. 183–199, 1990.

[8] K. Qiu, N.I. Hachem, M.O. Ward and M.A. Gennert, "Providing Temporal Support in Data Base Management Systems for Global Change Research," Proc. 6th SSDM Working Conf., Switzerland, June 1992.

[9] M.O. Ward, Y. Zhang, N.I. Hachem, and M.A. Gennert, "A Visual Programming Environment for Supporting Scientific Data Analysis," to appear in Proc. Workshop Visual Languages, Aug. 1993.