A DATA MODEL FOR ADAPTIVE MULTI-RESOLUTION SCIENTIFIC DATA

Philip J. Rhodes, R. Daniel Bergeron, Ted M. Sparr
Computer Science Department
University of New Hampshire
{rhodes,rdb,tms}@cs.unh.edu

Abstract Representing data using multiresolution is a valuable tool for the interactive exploration of very large datasets. Current multiresolution tools are written specifically for a single kind of multiresolution data. As a step toward developing *general purpose* multiresolution tools, we present here a model that represents a wide range of multiresolution data within a single paradigm. In addition, our model provides support for working with multiresolution data in a distributed environment.

Keywords: Multiresolution, Scientific Data Model, Adaptive Resolution

Introduction

New data gathering and data generation tools have created an explosion in the amount of data available to scientists. The existence of such large amounts of data provides opportunities that have not previously been possible, but the dataset sizes present major obstacles to understanding and interpreting the significant underlying phenomena represented in the data. This problem is particularly apparent when we try to develop tools for allowing the scientist to visualize the data in a form that may provide insight. Even relatively modest size data sets can have orders of magnitude more information than can possibly be displayed at one time.

Fortunately, it is often the case that most of the data can be visualized at a relatively coarse resolution as long as that presentation uses an *authentic* representation of the original data. The scientist can then identify "interesting" subsets of the data which can be displayed

This work is supported by the National Science Foundation under grants IIS-0082577 and IIS-9871859

at increasingly higher levels of resolution. Our research involves the development and evaluation of a distributed adaptive data representation scheme that allows interactive analysis of vast amounts of data in a progressive refinement environment.

Support for *scientific data exploration* of this sort requires both good visualization tools and good data management tools that incorporate *error representation* as an integral component. To provide such support for a wide class of applications, it is essential to develop a formal model for describing multiresolution data representations *and* how they can be accessed efficiently in a distributed environment. This paper presents such a formal model that will provide a framework for building a system to manage a variety of distributed multiresolution data representations. Our goal is to provide facilities that allow application programs and users to focus on the *semantics* associated with a given data request, not the details of data storage or the particular nature of the multiresolution representation.

In the remainder of the paper, we establish the basic data exploration paradigm we strive to support, define a formal model for representing scientific data, and then define a formal model for describing multiresolution scientific data. Finally, we conclude with a brief summary of the features we feel should be incorporated into a data management system intended to support interactive exploration of very large distributed scientific data.

1. Scientific Data Exploration Paradigm

We are interested in developing tools that support scientists in the development and evaluation of hypotheses about a phenomenon, particularly when such efforts are based on data observations from the phenomenon. The scientific community is engaged in numerous massive data collection and archiving activities. In addition to supporting many current research investigations, both narrow and broad-based, these archives form a baseline for a diverse set of future research projects. The size of data archives is growing rapidly. Until recently, units of terabytes characterized large datasets. Lately, the term petabytes occurs increasingly often. This escalation of dataset sizes will continue and undoubtedly accelerate.

1.1 Focused Exploration

One initial task of a typical research effort can be characterized as selecting and downloading a potentially relevant subset of an archive (or archives) into a scientific database system for further evaluation. Often, data are transferred over a network. Potentially relevant subsets of huge archives can themselves be quite large, often too large to fully analyze directly. Rather, the scientist must rapidly and efficiently explore the data for interesting phenomena. Focus then usually moves to these interesting zones, ignoring and perhaps discarding the rest.

1.2 Data Models

Identifying a suitable data model is a critical step toward managing and processing scientific data. It forms the basis of the interaction with exploration and analysis software. An appropriate data model can also be a vital tool for accessing and importing archive data. Existing scientific data models include the fiber bundle model [6], the lattice model [1, 19], and various grid models [13, 20].

One desirable characteristic of a data model is the ability to express an effective compromise between the necessity to restrict working dataset sizes to levels manageable by a workstation or LAN, yet include the detail the science requires. Some grid models, for example, permit variable size grids, coarse in dull or quiet data zones and fine in critical zones [20]. Such models work well in single purpose and limited scope studies where all investigators share a similar view of the data and agree on what constitutes interesting and uninteresting zones.

Unfortunately, the determination of important data zones is application and task dependent. Archive data must support a wide variety of uses from different, perhaps unanticipated, applications. Instead of static grids, scientific database systems must support dynamically formed multiple-resolution grids based on the specific purpose and on the data itself. To do that, the database must include another special type of metadata, termed *semantic metadata*, which describes or characterizes database content so that visualization and analysis tools work effectively [8, 9, 14].

There are many analytical and statistical tools for characterizing data. In fact, one of the goals of experimental research is the characterization of data in a way that distills new meaning from it, illuminating the phenomenon under investigation. Such metadata can accumulate as the data is studied and it can provide important guidelines for subsequent research. In particular, semantic metadata helps scientists identify those portions of data that merit further exploration. If a user/scientist can specify in semantic metadata terms what constitutes interesting data (or uninteresting data), the archive can select the most promising subsets. Having the archive dynamically form the multiresolution representation of these subsets can save considerable network and visualization bandwidth.

1.3 Multiresolution Hierarchy

We envision the creation of a standard multilevel data hierarchy for large scientific data sets. The original dataset would be stored permanently at some repository site for that data. A preprocessing operation would generate a complete multiresolution representation with increasingly coarse representations and the associated localized error representations. All components of this data representation would be available for downloading arbitrary subsets at arbitrary resolutions. A scientist is likely to extract some small coarse component of the hierarchy to store at his or her workstation, may access the next several levels on a data server on the local network, and perhaps access the finest resolution representations over the Internet.

This model works because only very small subsets of the total dataset will need to be accessed at the higher resolutions. Note that we can assume that the original data is essentially *read only*, although we certainly need to be able to dynamically update both the lower resolution representations and the *metadata* associated with the dataset.

In order for this multilevel storage representation to be effective, it is essential that every representation include both the data at that representation level and the cumulative local error measure associated with that data. The error representation must be an integral component of the data exploration process providing the scientist with critical feedback concerning *where* the current representation is likely to be accurate and where it isnt. Without such information, the scientist could not trust anything but the finest resolution.

1.4 Multiresolution Examples

We are not attempting to develop new methods of generating multiresolution data. Instead, we are developing a model for describing multiresolution data that addresses the features that all MR methods have in common. In this section we describe two different application domains for which MR data is particularly appropriate. Although these examples use very different kinds of data our model must be able to represent them both effectively.

1.4.1 MR and Regular Data. Much scientific data is sampled on a regular grid. Such data is particularly suitable for the application of wavelet transformations to produce an MR hierarchy. A wavelet representation of data includes two parts: the *summary* and the *detail*. As the names imply, the summary is an approximation of the original data, while the detail can provide a more refined representation when combined with the summary. For orthogonal wavelets, the detail component can also be interpreted as a representation of the error in the summary component [15].

Since summaries are a coarser representation of the data, we can produce a hierarchy of summaries of decreasing resolution by recursive application of the wavelet operation. In this way we can construct an MR hierarchy in which top levels are extremely coarse but take up little space, while lower levels are physically larger, yet provide much finer resolution.

Each level of the hierarchy has uniform resolution. If some regions of the data are much more interesting than others, it might be desirable to produce a single representation of the data in which uninteresting areas are represented coarsely in order to save space, while more important areas are reproduced more faithfully with finer resolution. This is an Adaptive Resolution (AR) representation. For instance, an AR representation might be constructed so that every region of the data meets a specified error bound. A localized error measure makes this a much simpler task by indicating features that require more resolution in order to meet the error bound. Since each summary coefficient has an associated detail coefficient, the wavelet transform generates localized error which can be used to represent the authenticity of the summary [22].

A researcher might want access to a hierarchy of representations with varying error bounds, yet also not want to waste space on uninteresting data or by representing simple features with excessive resolution. We can produce an AR Hierarchy by manufacturing a series of AR representations using different error bounds. The levels of the hierarchy are still arranged according to increasing resolution, but none of the levels waste space with unnecessarily fine resolution.

1.4.2 Multiresolution And Irregular Data. Our approach can also deal with irregular data such as is used for surface representation. For example, De Berg and Dobrindt [4] have developed a multiresolution method for terrains that is specifically geared towards computer graphics. Their motivation is to eliminate the unnecessary rendering of detailed polygons in the far distance of a rendered scene. The method allows resolution to slowly decrease as the terrain recedes into the distance. An important feature of the algorithm is that it seamlessly blends these different resolutions into a single mesh. The De Berg and Dobrindt method could therefore be classified as an adaptive resolution representation.

The authors build their MR representation in a bottom-up fashion. The original data is first triangulated using the Delaunay method. In forming each level, they remove a set of points from the previous level and then retriangulate in the area of each removed point using Delaunay. The removed points are chosen so that the retriangulation can be done locally without interacting with the neighborhood of other removed points. Presumably, they could also be chosen so that error is minimized in the resulting level. The algorithm continues forming new levels in bottom-up fashion until the number of points in the topmost level is equal to or less than a specified threshold.

The De Berg and Dobrindt algorithm fits very nicely into our model because it has distinct levels formed by the removal of a set of points. Not all algorithms behave this way. For example, Cignoni et al. describe an algorithm that removes one point at a time, choosing a point that generates the least error when removed [2]. Although we could say that each level in this hierarchy differs from its finer neighbor by the removal of a single point, it might be more useful to organize this fine-grained hierarchy into sets of points that satisfy a series of error bounds.

1.4.3 Model Motivations. Currently, an application programmer must write code that is specific to the kind of MR data he or she is manipulating. Modelling the commonalities of MR methods will facilitate the development of interfaces for handling MR data. Such interfaces will allow application programmers to write code that works with a variety of multiresolution methods, perhaps within the same application.

The key properties of these examples include the ability to define what is meant by a level and the ability to measure the accuracy of any approximation. We should also be able to model the process through which a level is produced from an existing level.

2. Data Model Foundations

Pfaltz et al. [16] identify the major features of scientific data as large size, complex entities and relationships, and volumetric retrieval. Although such characterization is correct, we need a more rigorous definition if we hope to provide effective database support for scientific data. For our purposes, scientific data is a collection of values that represents some natural phenomenon [7] that is a function over a domain [11] which might be time, space, radio frequency, etc. or some multidimensional combination. The notion of a dataset *domain* is central to our model of scientific data. The *value space* of the function defined over the domain usually consists of the Cartesian product of the value ranges of several data attributes. This is equivalent to saying that any point in the domain has a number of attributes–the value of the data function at that point.

2.1 Dimensional Data

Much scientific data can be meaningfully represented in a continuous n-dimensional data space [2, 8]. If a dataset consists of *some* attributes that are ordinal, independent, and defined on a continuous value range, the dataset contains *dimensional data*, and those attributes are *dimensional attributes*. Each possible combination of dimensions defines a *view* of the data, a notion similar to the view capability found in traditional databases. *Spatial data* is dimensional data that represents an actual physical space. A dataset can be dimensional without being spatial but even non-spatial dimensional data can often be visualized as if it were spatial, since humans find this representation familiar and easy to grasp. It may also be convenient to treat a set of attributes as if they are dimensional attributes even though they may not satisfy all the conditions for dimensional data. For example, we might want to treat a set of attributes as independent for exploration purposes with the goal of either validating or disproving that assumption.

Spatial (dimensional) data is often represented as points defined on a wide variety of regular and irregular grid types [2, 4, 6, 20]. The choice of the grid, usually based on a natural organization of the data, impacts the nature of the representation chosen for the data and the specification of algorithms for analyzing it.

2.2 Geometry, Topology, and Neighborhoods

The terminology used in the literature to describe various systems of grids is not standardized. We propose to develop a more comprehensive and consistent framework for describing and defining grids that encompasses most reported grid structures [9, 11], including both point and cell data organizations. We separately represent the underlying space in which the grid is defined, which we call the *geometry*, and the point and cell relationships implied by the grid, which we call the *topology*. Thus, the geometry of a dataset refers to the space defined by the dimensions; the topology of a dataset defines how the points of the grid are connected to each other. A dataset's topology is a graph with data points as nodes and arcs between nodes representing a *neighbor* or *adjacency* relationship.

This approach enables database support for application algorithms to process data either geometrically or topologically. In many cases, the topology and/or geometry do not have an explicit representation within the dataset because they derive easily from the indexes of an array that stores the data points. The array and its index structure compose the *computational* space of a dataset. Other more complex geometries and topologies may have a separate representation from their computational spaces.

In other applications the data has no inherent geometry or topology. For example, categorical data is normally not defined in a geometric space and scatter data has no predefined topology. Our data model allows a user to represent and manipulate both kinds of data. However, it may be useful to impose additional structure on the data. For example, we could impose a topology on scatter data either for efficiency of access or to support an alternative conceptual model for the data. Similarly we have shown that topology can be an effective vehicle for imposing a metric space upon categorical data [11, 12].

Many scientific applications require selecting the *neighborhood* of a point [5, 12, 17]. The neighborhood of a point p consists of points near p. Nearness may be defined *geometrically* (e.g., as the set of points within distance d of p in the geometric space) or *topologically* (e.g., as the set of points within n arcs of p in a topological space).

2.3 Error

Most scientific data contains some inherent error. This includes measurement error from sampling or computational error from simulation. Furthermore, operations and analyses may introduce additional error. Our model of scientific data includes *localized error* (i.e., it is estimated at every point within the domain [3, 22]).

2.4 Data Representation

Effective exploration tools for very large data sets are best developed on a rigorous conceptual model of the data. Such a model must be accessible to both the programmer and the user and must be able to adapt to the actual data in a natural and efficient way. We now present a data model that forms a promising foundation for describing scientific data that can be organized into a multiresolution hierarchy. The basic motivation for this model is to support distributed *interactive* data exploration.

A rigorous definition of a *data representation* is the formal basis of our model of the scientist's dataset. Although the dataset represents a phenomenon defined over a continuous domain, D (a *geometric* space with an infinite number of points), the dataset is a finite sampling of this space. Consequently, our *data representation* is defined over a finite set of points $\Delta \subset D$, known as *sample points*, within the domain D [8]. A sampling function f_Δ maps Δ to a subset Ω of a value space V [10], denoted by

$$f_\Delta : \Delta \rightarrow \Omega \ , \tag{1}$$

with *sampling error* described by a localized error function E_Δ mapping each sample point to an error space E, denoted by $E_\Delta : \Delta \rightarrow E$. Formally, we define a data representation R to be a quadruple:

$$R = \langle \Delta, \Omega, f_\Delta, E_\Delta \rangle \tag{2}$$

where Δ is a set of sample points in D that are sampled using the sampling function with an error function E_Δ, and Ω is the range of f_Δ. (By convention, we use dot notation to refer to the components of a tuple. Thus, $R.\Delta$ identifies the sample points of the representation R.)

2.5 The Lattice Representation

Although the data representation definition is comprehensive enough to encompass most kinds of scientific data, it only represents the actual data and does not incorporate any notion of how the different data elements might be related in a grid structure. We incorporate the grid definitions into our data model by adopting and extending the *lattice* model [1]. A lattice includes a topology, τ, as well as a geometry [11]. A lattice L_k^n has n topological dimensions that define a topological space, and k attributes for a point located in that space. The dimensionality of the lattice, n, is also the dimensionality of the lattice topology. Thus, a 0-dimensional lattice is simply an unordered set, a 1-dimensional lattice is an ordered list, a 2-dimensional lattice lies in a plane, and so on. The lattice geometry need not have the same dimensionality as the lattice topology. For example, a 2D lattice can be mapped to a curvilinear surface that exists in three-dimensional space. Formally, a lattice L consists of a data representation R and a topology τ; that is, $L = \langle R, \tau \rangle$. Operations on lattices include *value transformations* (e.g., normalization), *geometric domain transformations* (including affine transformations like scaling, translation, and rotation), and *topological transformations* such as mesh simplification. Operations like extension and projection can be either geometric or topological transformations (or both) depending on which components are altered.

The separation of geometry and topology in our lattice model allows us to represent both cell-based and point-based datasets in a unified fashion. Either a cell-based or point-based topology can be imposed on the same dataset, allowing the user to easily switch between these two views. Similarly, data read into the system as a collection of cells can be converted into a point-based representation, and *vice versa*.

2.6 Simple Data Model

Our notions of *data representation* and *lattice* are sufficient to represent a gridded scientific dataset, but they do not provide a representation for the phenomenon that the dataset is intended to model. We now define a simple *data model* which uses the lattice to approximate the phenomenon in the domain, as well as the error. Formally, a *data model* M consists of a lattice, and functions f_D and E_C to approximate the data value and its associated error at every point in the domain; i.e.,

$$M = \langle L, f_D, E_C \rangle \ . \tag{3}$$

The *approximating function*, f_D is normally based on the sampling function and returns a value that approximates the phenomenon in the domain; i.e., $f_D : D \rightarrow V$. *Interpolating functions* are approximating functions that satisfy the condition: $\forall d \in \Delta, f_D(d) = f_\Delta(d)$. That is, the interpolating function and the sampling function agree at each point in the sample domain Δ.

A possible definition of the initial E_C, representing error over the entire domain, could be f_{E_Δ}, which uses the approximating function to find values E_D from the values of E_Δ, the original sampling error function. For data models derived from other data models, E_C is the *cumulative error* including both sampling error and the error introduced by the derivation process.

3. Multiresolution Data Model

Although the basic data model described above represents a very wide range of data sets, it is not a model for multiresolution data. A multiresolution (MR) model allows a researcher to view data using resolutions ranging from very coarse to very fine (the original data). Using a coarse resolution can vastly reduce the size of the data that needs to be stored, manipulated, and displayed. It also serves as an overview of the entire dataset, allowing the researcher to pick out regions of interest without examining the original data directly. The researcher may examine interesting regions at finer resolutions, perhaps even accessing the original data. Note that finer resolutions are thought of as lower levels in a hierarchy, with the original data on the very bottom. “Drilling down” through this hierarchy allows the researcher to examine only data of interest at fine resolution, minimizing processing and display costs.

3.1 Multiresolution Data Representation

The MR representation offers scientists a tradeoff between detail and efficiency. Incorporating multiple resolution capability into the data model allows the database system to provide direct support for managing and using data at the resolution most appropriate to the immediate task. We first develop a definition of resolution based on the process by which the data is generated.

A *reducing operator* ρ transforms one data model, M, into another data model, M', where the new representation is smaller than the old [2]. In other words, $M'.L.\Delta$ contains fewer sample points than $M.L.\Delta$. This change in Δ causes $M'.f_D$ to differ from $M.f_D$ since the reduction in data points may change the approximating function. This reduction introduces additional associated localized error E_ρ. The domains of M and M' are both D, the domain of the natural phenomenon. $M'.E_C$ is a composition of $M.E_C$ and E_ρ, to reflect the error that ρ introduces. An *MR hierarchy* $\mathcal{M}$ is a sequence of *levels* $\{\lambda_0 \ldots \lambda_n\}$. Each λ_i in $\mathcal{M}$ is a pair consisting of a data model and associated localized reduction error: $\lambda_i = \langle M^i, E_i \rangle$. In summary, we formally define the reducing operator with the notation:

$$\rho : M \rightarrow (M', E_\rho) \; . \tag{4}$$

If we wish, we can define a sequence of reducing operators $\mathcal{R} = \{\rho_1 \ldots \rho_{n-1}\}$ where $\rho_i : \lambda_i \rightarrow \lambda_{i+1}$. That is, the reducing function ρ_i maps a level λ_i to a coarser level λ_{i+1}. The MR hierarchy is formed by repeated applications of reducing operations. First, the original data is stored in M^0 which corresponds to level λ_0. E_0 is the error associated with the original data, if it is known. A reducing operator is applied to M^0 to form M^1 and an error E_ρ which is E_1. M_1 and E_1 make up level λ_1. The process is repeated a number of times until the size of the data has been reduced sufficiently or until further reductions would introduce too much error. Certain classes of wavelet functions form an ideal basis for reducing functions because of their localized error characteristics [18, 21, 22], but our model is also appropriate for very different kinds of data reduction techniques such as triangle mesh simplification [2, 4].

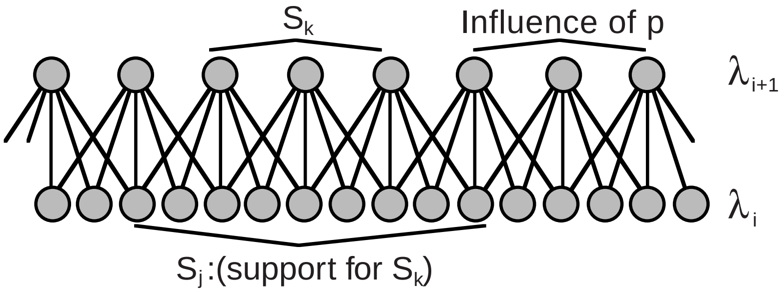

Figure 1. Support and Influence.

3.2 Adaptive Resolution and Adaptive Multiresolution

An adaptive resolution (AR) representation allows resolution to vary within a single level of a data model. The resolution near a point may depend on the behavior of the sampling function, on the behavior of the error function, or on the nature of the domain in the neighborhood of the point. A reducing operator that behaves differently over parts of D can define an adaptive resolution hierarchy, which is an MR hierarchy in which each layer is an AR representation. The reducing operator still reduces the size of Δ, but is more sophisticated in how it chooses to do so. For example, it can reduce resolution in areas with lowest error when forming the next level. It might also try to preserve resolution in areas of rapid value change and reduce resolution in less volatile areas. Because an AR hierarchy contains multiple resolutions within each level, it has the potential to achieve a representation with the same accuracy as MR using less storage. Alternatively, for a given amount of memory, it can retain increased detail and accuracy in important regions of the domain.

3.3 Support and Influence

Our model for MR is very general. In practice, most MR hierarchies are defined entirely by operations on the sampling set, Δ, and they often place further restrictions on a reducing function such as requiring spatial coherence. Typically, any neighboring set of sample points in λ_i should map to a neighboring set of sample points in λ_{i+1}. More formally, such a reducing function ρ can be defined as a collection of functions $\{r_0 \ldots r_n\}$ such that each r_j maps a neighboring set of points $S_i \subset \Delta_i$ in level λ_i to a neighboring set of points $S_i \subset \Delta_{i+1}$ in level λ_{i+1}. The set S_i

forms the *support* for S_k as shown in fig. 1. The union of all S_j for any level should equal Δ_i. Notice that this allows the domains of each r_i to overlap, meaning that a point p in λ_i might belong to the support for several different points in λ_{i+1}. These points form the *influence* of p (see fig. 1). By building the notions of influence and support explicitly into the data model (and into the database support system), we can provide a framework for better implicit support for efficient data distribution and distributed computation.

4. Multiresolution Data Support Model

Given the existence of both multiresolution and adaptive resolution representations of very large data sets, we need to develop a framework for a system to support such data sets in a distributed environment.

4.1 Goals

Applications should have access to a distributed multiresolution dataset without requiring explicit knowledge of either the distributed nature of the data, or its multiresolution structure. Ideally, the application should only be responsible for providing *error tolerance* measures along with a high-level *semantic* description of each application process that is applied to the data. We would like the distributed data support system to be able to make both resolution and location decisions based on these error measures and process descriptions. The system should be able to determine an appropriate resolution level to extract and to decide how and where the requested processing of that data should be carried out. In particular, depending on the size of the input data, the size of the output data, and the nature of the processing to be performed, the system ought to be able to decide whether to migrate code to the location of the data, or the data to the location of the process, or that both should be migrated to a third location.

4.2 Distributed Processing Support

We would like our data management system to provide support for distributing the processing associated with a distributed data representation. In order to balance the workload in a distributed processing environment, we need to have knowledge of the processing to be done and how that processing is affected by different data partitioning, including both spatial partitioning and partitioning introduced by the resolution hierarchy. This is particularly difficult since most of the processes applied to the data are both application- and task-dependent. On the other hand, we do not want the application programmer to have to be aware of the nature of the data distribution in order to benefit from the opportunities for distributed processing. In particular, we want to support both the migration of a user process to improve performance and the replication of the process to allow it to be distributed to separate data partitions defined over different regions and different resolutions.

The replication and partitioning is particularly difficult since boundaries between partitions usually have to be handled in a very process-dependent manner. Partition boundaries within the same resolution level can be difficult to handle; boundaries between neighboring regions that are represented at different resolutions are even more problematic because the semantics of merging the separate results becomes even less clear. For the system to provide meaningful automatic distributed processing support, the user process must be restricted to a class of operations whose input and output semantics can be represented by our data model. Within this context we plan to develop a semantic definition of user processes that will allow the system to determine how to distribute processing and combine results with minimal intervention from the process itself.

4.3 Application View

From the point of view of the application, the major role of the distributed data support environment is to find and extract any subset of the data. The application program ought to be able to:

- specify a range of data to extract, but allow the system to decide resolution level based on error measures;
- be able to override system resolution processing and request specific range/resolution;
- extend a set of predefined error measure functions with its own.

Once the appropriate subset of the dataset has been determined, the processing of that data should also be distributed. This is possible if the application provides a module that is migratable and defines the semantic effect of the operations to be performed so that the system can make some intelligent decisions about *where* to do the processing. This information could range from very precise parameter specifications to rather approximate heuristics. In particular, it is especially helpful if the system can predict the size of any output data sets and their eventual desired locations.

5. Conclusion

We have developed a preliminary formal data model for describing distributed multiresolution scientific data sets. A vital component of the data model is the incorporation of an error representation as an integral component of the model. The model is intended to be the basis for a scientific data management support environment that can provide nearly transparent access to a multiresolution dataset.

The present specification of the data model is just a beginning. In particular, we need to specify more fully the semantic metadata that defines processes and how to incorporate those definitions into the data model. We need to understand better how to characterize processes in terms of how they transform data model objects and how we can use that knowledge to more effectively access appropriate components of a distributed multiresolution dataset.

We have begun the development of our multiresolution distributed data support system. Our first goals are to provide the hierarchy generation support tools and straightforward access tools that require application knowledge of range and resolution. These tools will provide the basis for more intelligent interfaces.

References

[1] R. Daniel Bergeron, Georges G. Grinstein, “A Reference Model for the Visualization of Multi-Dimensional Data”, Eurographics ’89, Elsevier Science Publishers, North Holland, 1989

[2] Paolo Cignoni, Claudio Montani, Enrico Puppo, Roberto Scopigno, “Multiresolution Representation and Visualization of Volume Data”, IEEE Trans. on Visualization and Computer Graphics, Volume 3, No. 4, IEEE, Los Alamitos, CA, 1997

[3] P. Cignoni, C. Rocchini and R. Scopigno, “Metro: Measuring Error on Simplified Surfaces”, Computer Graphics Forum, Vol. 17, No. 2, Blackwell Publishers, Oxford, UK, 1998

[4] Mark de Berg, Katrin T.G. Dobrindt, “On Levels of Detail in Terrains”, Graphical Models and Image Processing 60:1–12, Academic Press, 1998

[5] Volker Gaede, Oliver Günther, “Multidimensional Access Methods”, ACM Computing Surveys, Vol. 30, No. 2, ACM, New York, 1998

[6] R.B. Haber, B. Lucas, N. Collins, “A Data Model for Scientific Visualization with Provisions for Regular and Irregular Grids”, Proceedings of IEEE Visualization 91, San Diego, CA, 1991

[7] W.L. Hibbard, C.R. Dyer, and B.E. Paul, “A Lattice Model for Data Display”, Proceedings of IEEE Visualization ’94, IEEE, Washington, DC, 1994

[8] W.L. Hibbard, D.T. Kao, and Andreas Wierse, “Database Issues for Data Visualization: Scientific Data Modeling”, Database Issues for Data Visualization, Proc. IEEE Visualization ’95 Workshop, LNCS 1183, Springer, 1995

[9] D.T. Kao, R. Daniel Bergeron, Ted M. Sparr, “An Extended Schema Model for Scientific Data”, Database Issues for Data Visualization, Proceedings of the IEEE Visualization ’93 Workshop (LNCS 871), Springer, Berlin, 1993

[10] D.T. Kao, M.J. Cullinane, R.D. Bergeron, T.M. Sparr, “Semantics and Mathematics of Scientific Data Sampling”,in Wierse, Grinstein and Lang (Eds.), Database Issues for Data Visualization, LNCS 1183, Springer-Verlag, Berlin, 1996

[11] D.T. Kao, *A Metric-Based Scientific Data Model for Knowledge Discovery*, Ph.D. Thesis, University of New Hampshire, Durham, 1997

[12] D.T. Kao, R.D. Bergeron, T.M. Sparr, “Efficient Proximity Search in Multivariate Data”, 10th International Conference on Scientific and Statistical Database Management, Capri, Italy, 1999

[13] P. Knupp, S. Steinberg, *Fundamentals of Grid Generation*, CRC Press, Boca Raton, FL, 1994

[14] Ulrich Lang, Georges Grinstein, and R. Daniel Bergeron, “Visualization Related Metadata”, Database Issues for Data Visualization, Proceedings of the IEEE Visualization ’95 Workshop, LNCS 1183, Springer, Berlin, 1995

[15] Stephane G. Mallat, “A Theory for Multiresolution Signal Decomposition: The Wavelet Representation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 7, pp. 674–693, July 1989

[16] John L. Pfaltz, Russell F. Haddleton, James C. French, “Scalable, Parallel, Scientific Databases”, Proceedings 10th International Conference on Scientific and Statistical Database Management, IEEE, Los Alamitos, CA, 1998

[17] Richard J. Resnick, Matthew O. Ward, and Elke A. Rundensteiner, “FED–A Framework for Iterative Data Selection in Exploratory Visualization”, Proceedings of Tenth International Conference on Scientific and Statistical Databases, IEEE Computer Society Press, Los Alamitos, CA, 1998

[18] Kiran K. Simhadri, S.S. Iyengar, Ronald J. Holyer, Matthew Lybanon, John M. Zachary, “Wavelet-Based Feature Extraction from Oceanographic Images”, IEEE Transactions on Geoscience and Remote Sensing, Voume 36, No. 3, IEEE Computer Society Press, Los Alamitos, CA, 1998

[19] T.M. Sparr, R. Daniel Bergeron, L. D. Meeker, “A Visualization-Based Model for a Scientific Database System”, Focus on Scientific Visualization, Hagen, Muller and Nielson (eds.), Springer-Verlag, Berlin, 1994

[20] D. Speray, S. Kennon, “Volume Probes: Interactive Data Exploration on Arbitrary Grids”, Computer Graphics, Vol. 24, No. 5, ACM, 1990

[21] Eric J. Stollnitz, Tony D. DeRose, David H. Salesin, *Wavelets for Computer Graphics: Theory and Applications*, Morgan Kaufmann Publishers, Inc. , San Francisco, CA, 1996

[22] Pak Chung Wong, R. Daniel Bergeron, “Authenticity Analysis of Wavelet Approximations in Visualization”, Proceedings of IEEE Visualization ’95, IEEE Computer Society Press, Los Alamitos, CA, 1995