A Scalable Mediator Approach to Process Large Biomedical 3-D Images

Konstantinos Liakos, Albert Burger, and Richard Baldock

***Abstract*—The Edinburgh Mouse Atlas is a spatial–temporal framework to store and analyze biological data including three-dimensional (3-D) images that relate to mouse embryo development. The purpose of the system is the analysis and querying of complex spatial patterns, in particular the patterns of gene activity during embryo development. The framework holds large 3-D gray level images and is implemented in part as an object-oriented database. In this paper, we propose a dynamic layered architecture, based on the mediator approach, for the design of a transparent and scalable distributed system which can process objects that can exceed 1 GB in size. The system's data are distributed and/or declustered across a number of image servers and are processed by specialized mediators.**

***Index Terms*—Bioinformatics, biomedical three-dimensional (3-D) image processing, mediators, network systems.**

I. INTRODUCTION

THE Edinburgh Mouse Atlas [1], [2] is a digital atlas of mouse development, created at the MRC Human Genetics Unit (HGU), Edinburgh, to store, analyze, and access mouse embryo development. The stored objects contain three-dimensional (3-D) images that correspond to conventional histological sections as viewed under the microscope and can be digitally resectioned to provide new views to match any arbitrary section of the experimental embryos [2]. The volume of the processed 3-D objects can exceed 1 GB and typical laboratory-based machines fail to efficiently browse such large biomedical image reconstructions, particularly since the users may wish to access more than one reconstruction at a time. In this paper, we propose a layered distributed architecture to provide a scalable solution to this problem. The system is transparent to the user allowing reconfiguration to improve efficiency in response to demand with no effect on the client application.

The proposed architecture has been influenced by the mediator approach. Mediators were first described by Wiederfold [3] to provide a coherent solution to the integration of heterogeneous and homogenous data by "*abstracting, reducing, merging and simplifying them*." In more detail mediators are "*modules occupying an explicit active layer between the user applications and the data resources. They are a software module that exploits encoded knowledge about certain sets or subsets of data to create information for a higher layer of application* [3]." They encapsulate semantic knowledge about a number of distributed sources providing to the user applications a unique representation of the distributed database context.

The middleware initially adopted for our purposes is the common object request broker architecture (CORBA)[1] [4]. The CORBA framework provides the means to develop open, flexible, and scalable distributed applications that can be easily maintained and updated. Data integration can be partially resolved via the use of the interface definition language (IDL) that separates the data model from its implementation. In the past, CORBA has been proposed as an efficient middleware solution for bioinformatics projects [5]–[8]. In addition, the European Bioinformatics Institute (EBI)[2] has adopted CORBA as a middleware for a number of its bioinformatics systems. A competing technology to CORBA, also adopted by EBI, is Web Services.[3] However, Web Services are less efficient than CORBA/IIOP over slow connected networks [9]. As a consequence, they currently cannot give optimum solutions to applications that require fast processing and transmission of large amounts of data. It should be noted though, that our proposed design is independent of the middleware platform. Web services are evolving rapidly and may become the preferred option in many bioinformatics applications.

A property of this application is that a query on the data requests a virtual object. These objects are analogous to a view in database terms that represent two-dimensional (2-D) section images that are computed at run-time from the original 3-D voxel models.

Our layered design is a modification of the mediator approach and is based on the requirement for dynamic configuration of the mediators due to performance reasons. Simple client–server designs such as that implemented in [2] for a genome-mapping prototype are inadequate to handle the requirements derived from large voxel image files. Although smaller size voxel images can be efficiently processed, larger images result in an unacceptably slow response, due to memory paging and central processing unit (CPU) processing time. Furthermore, integration of additional object resources becomes impractical. The volume of our current as well as our future objects, in addition to the requirement of providing very fast response times, introduce the necessity of distributing the cost of processing by declustering biological images in order to provide a scalable solution, minimize the cost of the overall query response time, and make an optimum usage of the available hardware resources. Our design adopts an n-tier solution by distributing the cost of processing to a number of image servers. Such a task is accomplished by declustering and distributing the image data across

Manuscript received May 5, 2003; revised March 9, 2004.

The authors are with MRC Human Genetics Unit, Western Hospital, Edinburgh EH4 2X0, U.K. (e-mail: konstantinas.liakos@hgu.mrc.ac.uk).

Digital Object Identifier 10.1109/TITB.2004.834374

---

[1] Object Management Group, OMG. [Online]. Available: www.omg.org.

[2] European Bioinformatics Institute. [Online]. Available: http://www.ebi.ac.uk/.

[3] World Wide World Consortium. [Online]. Available: www.w3.org.

different image servers, processing them in parallel. The image server processing is hidden by the use of one or more mediator layers, which are responsible for providing transparent access and to monitor image servers so that user requests are directed appropriately. In addition, mediators are designed to provide other services, such as query processing, decomposition and reassembly, query optimization, and user behavior prediction or "look-ahead." This latter service enables precomputation of predicted requests in order to accelerate the response time of the system. Finally, an important aspect of our design is its ability to vary the number mediators and allow dynamic configuration of the system to optimize the response.

The emphasis of this paper is on the scalability of our design to adjust the number of the mediators and to respond to the requirement to process large reconstructions by distributed processing. At this prototype stage, the layered approach enables open and simple extensibility to future requirements. A hierarchy of mediators is expected to improve overall performance when additional services will be introduced at the mediator side and will be investigated in a later paper.

The remainder of this paper is organized as follows. Section II provides a brief discussion of the current Mouse Atlas system. Section III provides a detailed description of the proposed distributed architecture, Section IV the proposed query processing design, and in Section V, some initial performance results are provided. Finally, Section VI discusses future related issues and concludes the presented work.

II. MOUSE ATLAS

The MRC HGU, Edinburgh, has developed the Edinburgh Mouse Atlas [1], [2] based on an object-oriented architecture in order to store and analyze biological data, including 3-D images that relate to mouse embryo development [2]. The embryo framework for the database is represented as a set of voxel models (3-D images), initially at each development stage defined by Theiler [10], but with the possibility of extension to finer time-steps especially at earlier stages. The voxel models correspond to conventional histological sections as viewed under the microscope and can be digitally resectioned to provide new views to match any arbitrary section of an experimental embryo.

The purpose of the system is the analysis and querying of complex spatial patterns, in particular the patterns of gene activity during embryo development. The underlying image processing and manipulation uses the Woolz image-processing library [1], [2], [11], [12] that was developed for automated microscope slide scanning and is very efficient for binary set and morphological operations. The users navigate in the 3-D space via the use of user interface components that correspond to particular viewing parameters, to define 2-D sections at any orientation. At any given time, only one such component can alter its value to generate a new query. The requested objects are *virtual* objects, which are not saved in the data sources, but are computed during run-time by Woolz library functions associated with the original 3-D voxel data. This is analogous to a view in database terms. The rotation within the original 3-D space that results in the generation of the section view is described below.

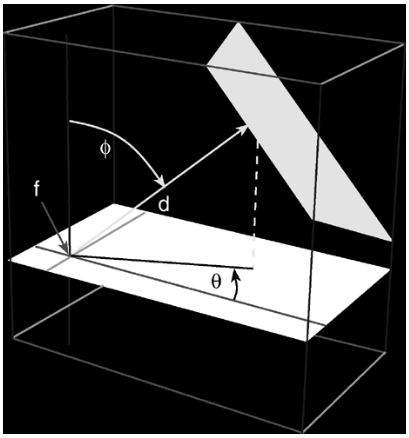

Fig. 1. Viewing plane is defined to be perpendicular to the viewing direction given by angles ϕ and θ. The actual plane is distance d from the fixed-point f. For the user interface, ϕ is termed *pitch* and θ is *yaw*.

Given an original coordinate $r = (x, y, z)^T$, the viewing plane (Fig. 1) is defined as a plane of constant z in the new coordinates $r' = (x', y', z')^T$, i.e., the new z axis is the *line-of-sight*. This axis is fully determined by defining a single fixed-point f that is the new coordinate origin, and a 3-D rotation. The actual view plane is then defined to be perpendicular to this axis and is determined by a scalar distance parameter d along the new axis. In this way, the transformation between the original r and viewing coordinates r' is determined by a 3-D rotation and translation with the viewing plane defined as a plane of constant $z' = d$. A 3-D rotation can be defined in terms of Eulerian angles [13] with full details given in [2]. These mathematical details are not essential to the understanding of the rest of the paper, but highlight the point that the computation of a 2-D section requested by a user query involves the retrieval of 3-D data from the database as well as some image processing on that data.

For efficient browsing, it is necessary to get the entire voxel image into the main memory so disk accesses are avoided. For very large reconstructions, i.e., images of 1–3 GB, this is impractical for typical laboratory-based machines, particularly since users may wish to access many such reconstructions concurrently. Even with the entire image in memory, most CPUs will be slow. While CPU and memory specifications will be steadily improved, image processing requirements are also expected to increase; e.g., future image volumes might well reach 10–100 GB.

There are many aspects of the Mouse Atlas, which are not directly relevant to the discussion that follows and, therefore, omitted from this paper. The interested reader can refer to [1], [2], [11], and [12].

III. SYSTEM'S ARCHITECTURE

The proposed design consists of four main components: client, directory, mediator, and image servers (Fig. 2). The client consists of a graphical user interface (GUI) that provides the stage selection menus, controls for section parameters, and is responsible for the generation of user queries while the mediators provide a transparent view onto the image server data sources. The mediators encapsulate the knowledge of the data sources (metadata), such as the objects which are stored

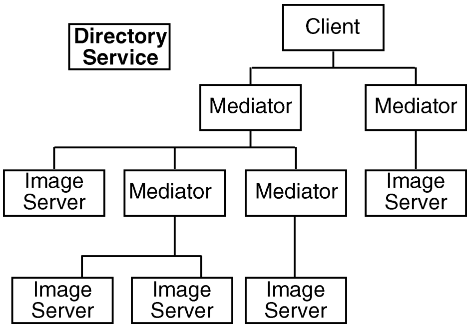

Fig. 2. Layered design based on the system's requirements.

in addition to their structural properties and the image servers, where objects can be found, providing transparent access to the image data. Image servers provide direct access to the underlying image storage, e.g., files and databases. It is only an image server that loads and directly queries a Woolz object. Due to the volume of the 3-D image data, declustering is expected to be introduced. By declustering, we mean the process of cutting an original 3-D voxel into smaller dimensions placing them at different sites. A key difference in the concept of declustering between our approach and the one that is normally in use is that our declustering performance relates to the CPU speed-up that can be gained; current declustering schemes are concerned with the disk input–output speed-up [14], [15]. Our system assumes that for each request, the declustered 3-D image data can be loaded into the main memory of each image server. Finally, the directory servers enable access to mediators and image servers by providing their connection details. Note that clients can only acquire information about a mediator while a mediator can acquire connection details related both to other mediators and image servers.

A mediator's task is to provide a unified view of the declustered distributed Woolz, transparent access to heterogeneous systems and their data, monitoring of the user access patterns and prediction of future requests so as to precompute related sections. The successful completion of these tasks depends on a metadata model that links image servers to Woolz image objects and Woolz objects to object properties, such as their structure and size.

An image server on the other hand provides a logical view onto the Woolz data and a Woolz query-processing model to access them. The object-oriented database in which Woolz are saved can be only accessed by appropriate image servers.

In a mediator–wrapper approach, a mediator encapsulates a global data model to serve a number of heterogeneous data sources. Such an approach has been adopted by many distributed systems such as DISCO [16], GARLIC [17], and Informia [18]. In comparison to other bioinformatics implementations where the mediator approach is used to integrate heterogeneous data sources (Raj and Ishii [7], and Kemp *et al.* [8]), our design is similar because it also encapsulates a schema of its integrated image servers. However, these systems typically employ a centralized mediator serving a number of data sources (Fig. 3), whereas our mediators are fully scalable (Fig. 2) and the data saved in the data sources are homogeneous.

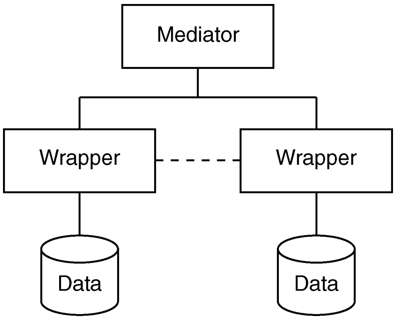

Fig. 3. Mediator–wrapper approach.

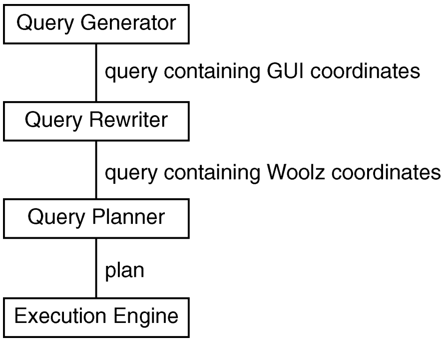

Fig. 4. Query processing architecture.

Numerous mediators can be dynamically created according to the performance needs of the system at run-time. As a consequence, our design can result in both centralized and distributed topologies depending on performance requirements. More precisely, a mediator might not only control a number of image servers but also a number of mediators as well. Hence, a mediator's schema might be based on the schema of its lower-level mediators. The concept of such a composable approach is similar to AMOS II [19], [20], a lightweight OODBMS, where mediators regard interconnecting mediators as data sources. The purpose of AMOS II, however, is a federated multidatabase system that combines heterogeneous data sources and views. On the other hand, our main aim is to reduce the time needed to fetch process and display large image objects, not to integrate diverse data sources. Apart from that, our proposal is based on standard and flexible solutions that facilitates scalability and maintenance, and is language and platform independent. Furthermore, our mediators perform additional tasks such as the monitoring of the user access patterns and the precomputation of future requested sections so as to improve the overall performance of the system. These facilities can be seen as an extension of our query processing mechanism that is described in Section IV. Details of this speculative computation model are beyond the scope of this paper and will be presented elsewhere.

The client component consists of a GUI that provides the tools to query and process Woolz image objects. Initially, clients are unaware of the connection details of the mediators they are connected to. Such information is acquired from the directory server (Fig. 4). Another important issue is that clients are not aware of the underlying Woolz technology [2]. Their only intelligence relates to the display of the 2-D sections, which are acquired as bitmap arrays, and their understanding of the 3-D bounding box of the saved Woolz objects. An important property of the technology underlying the GUI is that it results in asking only the visible part of every reconstruction. As a consequence, only a small part of the whole virtual object is requested, minimizing the data that needs to be transmitted over the net.

We are aware of the overhead that the layered approach might introduce. At this prototype level, our emphasis was on functionality. Our aim is to justify the necessity to alter the number of mediators due to performance requirements and to establish the system's capability to process very large reconstructions. The layered approach enables integration of additional services in a simple and open way and it further enhances system's scalability to adjust to future requirements. The advantages of such an approach are under investigation.

IV. QUERY PROCESSING

Kossman [21] has extensively described query processing optimization techniques. Assuming that Woolz internal image processing is optimal, our efforts are focused on accessing the Woolz data sources as fast as possible producing the appropriate query for them. Our query processing architecture comprises of a query generator, a query rewriter, a query planner, and a query execution module (Fig. 4). A query generator is found on the client side and involves the generation of a Woolz query based on the visible coordinate system of the GUI. The query rewriter module transforms the query from GUI coordinates into Woolz coordinates. The generation of a query can be also introduced on the mediator's side as a consequence of the precomputation process algorithm that monitors user activity. The query planner, that includes a query rewriter module, selects only the data servers that can respond positively to a particular query. To optimize system performance, only the image servers that can respond to queries are sent modified user requests. A query rewriting process starts only after the data servers are selected. Its aim is to split the initial Woolz query and request only the subregion to which each data server can respond. The query execution model that is found both in the client and mediator is used to transform either a user or a mediator query into a compatible IDL query. On the other hand, query execution on the image server side relates to the use of Woolz libraries so as to physically execute a query.

V. EXPERIMENTS

The experiments were run on a SUN Netra array, which includes ten servers running SOLARIS 8. Each server has the same capabilities and the interconnection between them is provided via the use of dedicated switches whose speed is 100 Mb/s. Every server has 500 MB of RAM, 500 MHz of UltraSPARC-II CPU, and 256 Kbytes of e-cache. The developed programs were written according to the requirements of the system's design in Java.[4] A benchmark that requests 1000 random 2-D sections for a given viewing orientation has been developed. After a client acquires a requested 2-D section, a generation of a new query takes place immediately. The window size is set up to 800 × 400 pixels while initially a centralized mediator is used. Every image server responds to a mediator query by sending only the part of the section that it can generate. The requested parts are determined by the window size of the client's GUI and the section parameters. The observed network costs correspond to those of a local area network (LAN). It was not the purpose of this study to include wide area network optimization, which will be subject of later work.

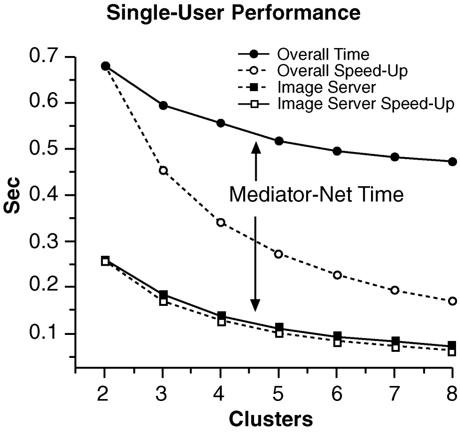

Fig. 5. Detailed analysis of the times spent throughout the experiment. Declustering and parallelism result in decreasing the average response time of the system when eight clusters are used.

The first experiment examines the overall advantage gained by the introduction of data parallelism and declustering. TS20, a Woolz object sizing 685.5 MB, was cut into a number of clusters (two to eight) of approximately equal size and placed in different locations over the network. Cluster volume varied from 342.75 MB (two clusters) to 85.69 MB (eight clusters). The benchmark described previously was used to measure the system's performance and it was run three independent times before the derived results have been averaged. For this particular experiment, the network time to send a user request and the time to deliver the requested section is omitted

$$\frac{2\mathrm{AVG}(2)}{\mathrm{clusterno}}. \tag{1}$$

In Fig. 5 the obtained results are plotted. The average image server response time represents near optimal performance assuming linear speed-up with respect to the two-computer case (1), where $\mathrm{clusterno}$ is the number of clusters and $\mathrm{AVG}(2)$ is the average response time when two clusters are used. Those results reflect the improvement introduced by the process of declustering and in addition to the parallel process of the responded image servers 2-D sections at the mediator side, the overall system's performance is accelerated. The gap between the overall mediator time and the average image servers time represents the network time and the time to process the requested sections. Due to the almost equal processed amount of data, such time is roughly constant and, therefore, the overall response time does not represent optimal performance with respect to the linear speed-up.

Our next experiment establishes the capability of our design to process very large objects. More precisely, TS20 and Theiler Stage 16 (TS16) have been scaled by a factor of two across their

---

[4] Java. [Online]. Available: www.java.sun.com.

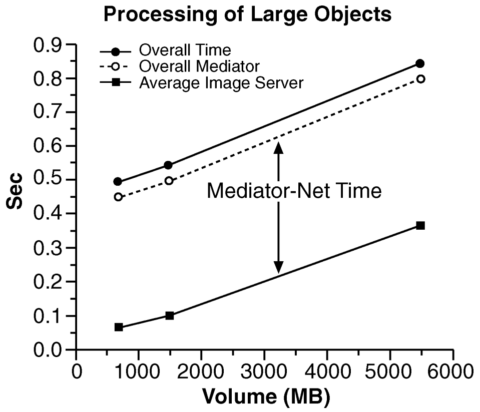

Fig. 6. The larger the volume and the dimensions of the objects are, the slower their response time. The gap between the two diagrams represents the mediator time in addition to network and idle time.

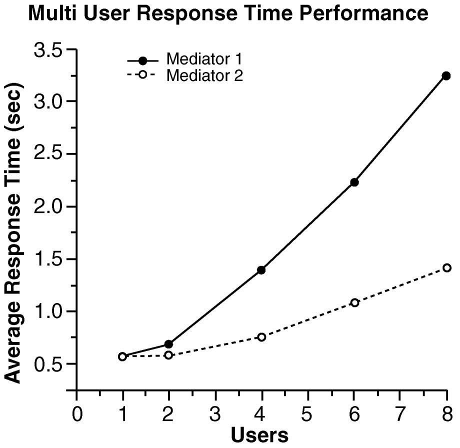

Fig. 8. Introduction of another mediator significantly reduces the average response time.

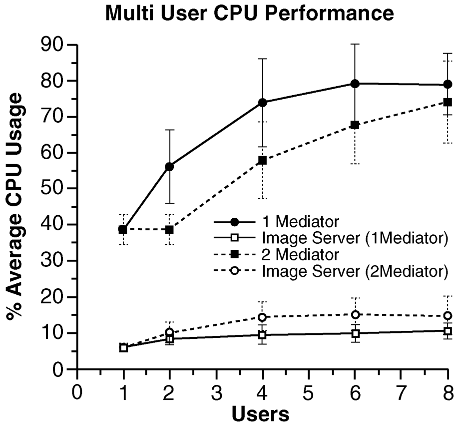

Fig. 7.Introduction of another mediator reduces CPU utilization at the mediator side while it slightly increases the image servers' cost.

x-y-z axis resulting in objects sizing 5.5 and 1.47 GB. Those objects have been declustered across their y axis producing in each case eight clusters that are distributed to the available eight image servers.

As pointed out in Fig. 6, the larger the volume of the processed clusters the slower the system's response time. Such behavior is explained by the higher observed cost to process larger volume objects at the image server side. On the other hand, the same window size that is used results in processing almost the same amount of data at the mediator side and, thus, its cost is roughly constant.

To provide evidence of the scalability and the necessity to dynamically alter the configuration of the mediators, a number of simultaneous users has been simulated. Initially, CPU usage has been examined for the case of the declustered TS20 object whose clusters have been placed across eight image servers and the number of users varied from one to eight. Each client performed 1000 random queries across a given viewing orientation. The obtained results are plotted in Fig. 7. The black line represents CPU usage for the single mediator case while the dashed line corresponds to the image server usage. As shown in Fig. 7, the process of the requested sections at the mediator side consumes the available resources. The mediator becomes rapidly the bottleneck of the system, while on the other hand, CPU utilization increases at an insignificantly small rate at the image server side. Those properties suggest a scalable and potentially a dynamic design that will alter the available number of mediators in order to balance resource utilization and improve performance.

To test this hypothesis, another mediator has been added and the directory service directed users to the appropriate mediator in a round robin fashion. The overall system's performance has been improved by a factor of 2.3 in the case of eight users (Fig. 8). At the mediator side, a better resource utilization is observed while the increased cost at the image server side due to the higher rate in which queries are now able to be sent is insignificant (Fig. 7). Those results signify the importance of introducing a number of mediators to balance load and, therefore, improve performance. A dynamic mediator configuration at run-time can potentially be achieved, however, the performance metrics of such system remain to be seen.

VI. CONCLUSIONS AND FUTURE WORK

This paper has discussed the capabilities of a distributed approach to efficiently process large biomedical 3-D image data. In the Edinburgh Mouse Atlas, user queries do not directly correspond to the 3-D objects saved in the database but rather to 2-D sections, which need to be calculated at run-time from the 3-D image data. This creates an interesting context in which new analysis related to different declustering schemes is required.

Some preliminary experiments reveal that user navigation patterns strongly affect the optimum cluster generation procedure and, hence, further details concerning optimum declustering and placement approaches is required. The required analysis will examine different placement strategies to minimize the cost of distribution and optimize the performance of the system.

In addition, our mediators will be enhanced with prediction capabilities to precompute future requested queries based on the monitoring of the user access patterns. Initial studies here suggest that time series prediction approaches such as the Autoregressive Integrated Moving Average [22], [23] and the Exponential Weighted Moving Average [24] adapt fast and efficiently to changing user behaviors. However, the cost of the algorithms and the improvement of the system need to be evaluated. Apart from that, a unified prediction policy should be suggested to decide how, when, and where precomputation will be introduced. Under this concept, the identification and the properties of different user access patterns are of great importance, so as to apply prediction only when needed.

In summary, this paper has presented a scalable design to handle large bioinformatics 3-D images, which represent different stages of mouse embryo development. Such a design is based on the mediator approach and exploits the advantages gained by parallel processing and data distribution. The work thus far has shown the validity of the approach and already provides an acceptable response time for user interaction with a very large-scale image (5.5 GB). Data distribution and parallel processing are essential not only to provide faster response times but also to enable processing of large objects unable to fit into the main memory of a single machine. Such features have been presented not only by processing a very large-scale biomedical image, but also by examining the results of declustering and data distribution for a smaller scale object (685.5 MB) in which the overall response time has been decreased by more than 30% when all eight available distributed CPUs have been used. Finally, the necessity to alter the available number of mediators due to performance requirements has been evaluated. The cost of processing the requested reconstructions at the mediator side is high and, therefore, the system's performance can be significantly improved by the introduction of additional mediators, providing a better resource utilization.

REFERENCES

[1] R. A. Baldock, F. J. Verbeek, and J.-L. Vonesch, "3-D reconstructions for graphical databases of gene expression," in *Seminars in Cell and Developmental Biology*, 1997, pp. 499–507.

[2] R. A. Baldock, C. Dubreuil, B. Hill, and D. Davidson, *The Edinburgh Mouse Atlas: Basic Structure and Informatics, Bioinformatics Databases and Systems*, S. Levotsky, Ed. Norwell, MA: Kluwer, 1999, pp. 102–115.

[3] G. Wiederhold, "Mediators in the architecture of future information systems," *IEEE Computer*, vol. 25, pp. 38–49, Mar. 1992.

[4] R. Orfali and D. Harkey, *Client/Server Programming with Java and Corba*, 2nd ed. New York: Wiley, 1998.

[5] A. Spiridou, "A view system for CORBA-wrapped data sources," in *Proc. IEEE Advances in Digital Libraries 2000 (ADL 2000)*, Washington, DC, pp. 228–240.

[6] J. Hu, C. Mungall, D. Nicholson, and A. L. Archibald, "Design and implementation of a CORBA-based genome mapping system prototype," *Bioinformatics*, vol. 14, pp. 112–120, 1998.

[7] P. Raj and N. Ishii, "Interoperability of biological databases by CORBA," in *Proc. 1999 Int. Conf. Information Intelligence and Systems*, Rockville, MD, Mar. 31–Apr. 3, 1999.

[8] G. J. L. Kemp, N. Angelopoulos, and P. M. D. Gray, "A schema-based approach to building a bioinformatics database federation," in *Proc. IEEE Int. Symp. Bio-Informatics and Biomedical Engineering (BIBE 2000)*, Arlilngton, VA, Nov. 8–10, 2000.

[9] D. C. Schmidt and S. Vinoski. Objet Interconnections: CORBA and XML-Part3: SOAP and Web Servises, C/C++ Users Journal. [Online]. Available: www.cuj.com

[10] K. Theiler, *The House Mouse: Atlas of Embryonic Development*. New York: Springer-Verlag, 1989.

[11] R. A. Baldock, J. Bard, M. Kaufman, and D. Davidson, "A real mouse for your computer," *BioEssays*, vol. 14, no. 7, pp. 501–503, 1992.

[12] M. Ringwald, R. Baldock, J. Bard, M. Kaufman, J. T. Eppig, J. E. Richardson, J. H. Nadeau, and D. Davidson, "A database for mouse development," *Science*, vol. 265, pp. 2033–2034, 1994.

[13] H. Goldstein, *Classical Mechanic*, 2nd ed. Reading, MA: Addison-Wesley, 1950.

[14] Y. Zhou, S. Shekhar, and M. Coyle, "Disk allocation methods for parallelizing grid files," in *ICDE 1994*, 1994, pp. 243–252.

[15] C. Chen, R. Sinha, and R. Bhatia, "Efficient disk allocation schemes for parallel retrieval of multidimensional grid data," presented at the 13th Int. Conf. Scientific and Statistical Database Management, Fairfax, VA, July 18 – 20, 2001.

[16] A. Tomasic, L. Raschid, and P. Valduriez, "Scaling access to heterogeneous data sources with disco," *IEEE Trans. Knowledge Data Eng.*, vol. 10, pp. 808–823, Sept./Oct. 1998.

[17] L. Haas, D. Kossmann, E. L. Wimmers, and J. Yang, "Optimizing queries across diverse data sources," in *Proc. 23rd Int. Conf. Very Large Databases*, Athens, Greece, Aug. 1999.

[18] M. L. Barja, T. A. Bratvold, J. Myllymäki, and G. Sonnenberger, "Informia, a mediator for integrated access to heterogeneous information sources," in *Proc. Conf. Information and Knowledge Management (CIKM'98)*, Gardarin, French, Pissinou, Makki, and Bouganim, Eds., Washington, DC, Nov. 3–7, 1998, pp. 234–241.

[19] V. Josifovski, T. Katchaounov, and T. Risch, "Optimizing queries in distributed and composable mediators," in *Proc. 4th IECIS Int. Conf. Cooperative Information Systems*, Edinburgh, U.K., Sept. 02–04, 1999.

[20] T. Risch and V. Josifovski, *Distributed Data Integration by Object-Oriented Mediator Servers Concurrency and Computation: Practice and Experience J*. New York: Wiley, 2001, vol. 13(11).

[21] D. Kossman, "The state of the art in distributed query processing," *ACM Computing Surveys*, vol. 23, pp. 422–469, Sept. 2000.

[22] C. Chatfield, *Time Series Forecasting*. London, U.K.: Chapman & Hall, 2001.

[23] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1994.

[24] C. D. Lewis, *Industrial Business Forecasting Methods*. London, U.K.: Butterworth, 1982.

Konstantinos Liakos received the B.Sc. degree in informatics from Athens University of Economics and Business, Athens, Greece, in 1999. In 2000, he received the M.Sc. degree (with distinction) in distributed and multimedia information systems from Heriot-Watt University, Edinburgh, U.K. He is currently working toward the Ph.D. degree at the MRC Human Genetics Unit, Edinburgh.

He is currently a Research Assistant for the MRC Human Genetics Unit, Edinburgh. His research interests are focused on bioinformatics, speculative computing, distibuted systems, and databases.

Albert Burger received the B.Sc. degree from the Fachhochschule Augsburg, Germany, the M.Sc. degree from the University of Missouri, Kansas City, and the Ph.D. degree from the University of Edinburgh, U.K., all in computer science.

He works as a Bioinformatics Research Scientist for the MRC Human Genetics Unit, Edinburgh, and holds the position of a Lecturer at the School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh. Originally working in distributed and database systems, his research work now focuses on bioinformatics, specifically biomedical ontologies and distributed bioinformatics systems.

Dr. Richard Baldock received the B.Sc. degree in mathematics and physics from the University of Bristol, U.K., in 1975. He received the Ph.D. degree in theoretical physics from the Australia National University, Canberra, in 1980.

He is currently a Senior Research Scientist at the MRC Human Genetics Unit, Edinburgh, where he is working on 3-D reconstruction, a digital atlas of mouse embryo development, and a spatio–temporal database of gene expression and other spatial data in the mouse. His research interests include knowledge-based systems, segmentation of biomedical images, 3-D reconstruction and visualization, and databases.