CHAPTER 7

CONCLUSION

This chapter summarizes the research conducted, provides comparisons to previous research, identifies contributions of the research, and suggests avenues for future work.

7.1 Thesis Summary

We initially set out to develop and realize a systems model for database exploration that integrates database and visualization components. This integration would serve a pressing need, since emergent database technologies such as knowledge discovery in databases, on-line analytic processing and data warehousing are overlooking (in our view) the important contribution that visualization can make. Furthermore, the visualization community seems to be overlooking these emerging fields, by focusing on rendering algorithms for physical science data sets with little data interaction.

Thus, there is a definite void to be filled that we found both interesting and worthwhile to address. Our solution, detailed in Chapter 3, is based on a relational database-iconographic visualization *data exploration scenario* having a defined *task repertoire* of the prominent activities that a data analyst would undertake during a *data exploration session*. To support these tasks, we exploit a concept that unifies both database and visualization systems, the *view*.

The *local database view* represents database query results and, like the traditional database view, it has an input interface and can be queried further. The *visualization view* encapsulates the visualization and additional user input facilities that support interaction by the analyst. We contend that visualization is more than an output transformation; it is an input interface for continued exploration, due to its visual nature. The systems integration model maps the database query result to the visualization domain, and visual interactions over the visualization back to the database domain. This is largely accomplished by the visualization view.

The *Exbase* database exploration system is the realization of our systems integration model. Exbase is an object-oriented database client with visualization capabilities that performs the requisite mappings between database and visualization systems. The local database view class maintains query results and supports simple range queries over tuples in the view. The visualization view is spread out across a class hierarchy, with several cooperating classes that communicate to the local database view, maintain data projections (i.e., data-to-visual primitive mappings) in base classes, and maintain display parameters in leaf classes.

Having this ability to perform integrated database and visualization operations at the visualization display makes it easy to rapidly explore a database. This new functionality raises more fundamental questions concerning the nature of database exploration, and how entire exploration sessions might be described, instead of individual tasks. Since the point of any systems architecture is to support user tasks and interactions, we felt it necessary to model the database exploration process in order to provide a context in which to feed back information about the user as a data explorer.

Thus, the research then shifted focus to modeling and measuring the database exploration *process*, the contents of Chapters 4 and 5, respectively. In the Generalized Data Exploration (GDE) model, *data entities* model the data objects, and *data derivations* model the mappings between data entities. *Metadata* is an integral component of these definitions, and we identify four primary categories: *identification*, *structural*, *process* and *knowledge* (or *semantic*). Identification and structural metadata are common types in use today, while process and knowledge metadata are less common. It was a goal of this phase of the research to enhance the process metadata that describes database exploration sessions. Though metadata is a type of knowledge structure, knowledge metadata contains analyst notations, insights and descriptions of the data, in addition to summaries that describe data values (such as statistical summaries).

Comparisons between data entities are based on their data and metadata components. Distinct data entities having the same data are differentiated by their metadata, specifically their knowledge metadata: some derivations compute knowledge metadata (such as statistical summaries) while not changing the data. We defined *congruence* and *similarity* classes of data entities, that are equivalence classes, to aid in comparisons and help highlight important patterns in the data exploration session. The result of each data derivation is a set of data entities, called a *data exploration state*. Thus, data derivations transform data and create new states. Three types of derivation components are identified: *forward* (those that create a new state), *backward* (those that recreate a previous state) and *identity* (those that recreate the input state).

The data exploration session has two primary structural configurations. The first is the temporally-ordered *data derivation sequence*, the second is the derivation-ordered *data derivation graph*. A special kind of data derivation graph is the *forward derivation*

component graph, which has no back edges (from backward derivation components) or self-loops (from identity derivation components). The forward derivation component graph, also called the *session graph*, is a directed acyclic graph that shows the forward progress of the session, from raw, knowledge-poor data entities close to the data source vertex (i.e., the database) to more refined, knowledge-rich data entities further away from the data source vertex. Among the relevant operations identified over the forward derivation graph are *connected components* (to identify descendent vertices) and *transitive closure* (to compact the graph).

An intermediate structure between individual data entities and the entire forward derivation component graph is the *forward derivation path*, which is a generalization of a simple path that accounts for multiple edges per derivation. A forward derivation path can be any connected subgraph of the forward derivation graph. It shows a particular thread of inquiry, having a derivational ordering, in the exploration.

The GDE model is structural, while its realization, the session graph, captures a dynamic process. To capture the dynamics of database exploration, we developed a number of data-independent *metrics*. These metrics are enhancements to the process metadata of data entities. As a framework for defining the metrics, we first limited ourselves to the database exploration scenario that drove the systems part of the research. We then defined the *data exploration space*, a metric space that is the Cartesian product of all data attributes and configurable visualization parameters, to support some descriptions. We then developed several graphical representation schemes for session graphs, to aid in their comprehension: *detailed*, *summary*, *compressed* and *compressed summary*. Summary graphs were used extensively in the examples. Completing the

framework is the concept of *data exploration scope*, which is a region of the session graph (local or vertex, path and session) that places the user within a specific context .

We developed two types of structural metrics: vertex-based and path-based. Vertex-based metrics use local scope: vertex indegree and outdegree values to indicate if the vertex was used primarily as a derivation result or source, respectively. *Explicit state changes* (those that do not result from a derivation), along with indegree, indicate vertex visitation and abandonment. Important types of vertices described by these metrics include *sequential*, *landmark* and *terminal*. *Sequential* vertices indicate a single input-output usage of a vertex. *Landmark* vertices have high visitation values, and serve to orient the analyst during the session. *Terminal* vertices are either "dead ends" that mark the end of an unpromising derivation path, or "tagged" to indicate a promising result that is close to extracted knowledge. Tagged vertices are expected to contain large amounts of knowledge metadata that are analyst notations.

An important result of the vertex-based metrics is the *Conservation of State Change Property*, in which the sum of all state changes (derivations plus explicit state changes) into a vertex equals the sum of all state changes out of a vertex. Additional vertex based metrics to categorize vertices as being *convergent* or *divergent*.

Path-based metrics use derivation path scope to describe both forward derivation paths, and exploration session graphs, in general. The path *depth* metric is associated with linear structures, and is used to indicate the amount of data refinement. The path *breadth* metric is associated with branching structures and is used to indicate the amount of data exploration space coverage. Two Postulates, the *Data Refinement Semantic* and the *Data Coverage Semantic* were formulated to advance these concepts. The *cyclomatic complexity* metric applies a well-known measure of software complexity to data

exploration, as forward derivation graphs are very similar to program control flow graphs. Branching structures, if viewed as creating distinct connected components of the session graph, increase the complexity of the session. They also make the session graph more difficult to render clearly.

We then developed a data exploration calculus to account for the temporal aspect of the session. By approximating the sequential application of data derivations with a piecewise continuous 3rd order polynomial, rates of change within the session can be measured. The simplest rate of change is the number of derivations per second. Smaller rates indicate the analyst is slowly issuing derivations, while larger rates indicate the analyst is rapidly issuing derivations. Though the underlying reasons are many, the calculus yields more information about the dynamic nature of the session than the structural representation. Incorporating the *data derivation vector* containing the data attributes and visualization display parameters of a derivation that undergo change adds some data-dependent insight into the workings of a derivation. This unit vector does not say how much each independent dimension is changing, only the general direction within the data exploration space in which the analyst is traveling.

We then developed several notions of continuity for the forward derivation graph. While the data exploration process is temporally continuous (by our approximation), it is not necessarily continuous with respect to the data being explored. Branching structures attest to this fact. We showed how branching or revisitation graph patterns, or piecewise temporal continuity reduce the overall derivational velocity.

In Chapter 6 we describe how the GDE model and metrics can be applied to the data exploration scenario of Chapter 3. This requires a specialization of the model for the relational database and iconographic visualization domains, and the addition of a new

data-dependent metric, the *derivation granularity*. This metric measures the data resolution of a derivation in terms of the data structures it manipulates and how they are manipulated. We outlined a general approach, and gave a detailed example of how the metrics are applied.

The GDE model specialization for the relational database and iconographic visualization domains identifies two similar graph patterns. The first, *panning*, is a data coverage operation that creates a branching graph pattern. Data panning is accomplished by partitioning a database relation or view on its attributes. Visual data panning is accomplished by altering the data-to-visual primitive mapping associated with a visualization. Spatial panning is accomplished by a sliding range query on the axis-mapped data attributes. The second fundamental graph pattern, *zooming*, is a data refinement operation that creates a linear graph pattern. Data zooming is accomplished by progressively more restrictive queries over tuples. Spatial data zooming is accomplished by specifying a region of interest within a visualization, either through a direct manipulation over the display, or via some control widget.

We show how the specializations are combined into a database-visualization environment, and also show a canonical exploration session graph. The canonical graph structure is based on the semantic Postulates developed in Chapter 5: exploration session graphs have high breadth at smaller depths (more data coverage operations exist closer to the database), and low breadth at larger depths (more data refinement operations exist further away from the database). A complementary pattern shows where refinement precedes exploration, to suggest that sometimes data must be refined to a point where it has a compelling enough visualization to warrant further exploration.

Finally, we show how these patterns are realized in Exbase, which requires the introduction of a new derivation edge label to account for the querying of visualizations. We show the canonical Exbase session graph, and identify the general structuring of derivations, based on Exbase's task repertoire. We then compare Exbase exploration with the dynamic query approach, which is similar, except for the visual representation type and the rapidity of interactions. The model and metrics picked up both of these differences.

7.2 The Research in Context

Our systems research can be considered a hybrid solution to the database-visualization impedance mismatch problem, that selectively borrows from previous database and visualization research. In incorporates what we consider are the most important qualities of each domain.

We differentiate our work from previous database research by integrating a database with iconographic visualization. This greatly increases the amount of data displayed simultaneously, and harnesses the human visual system's texture discrimination capabilities. The relation data structure maps naturally to the iconographic visualization technique. Since the systems research was used as a springboard for the process research, we did not address other important issues such as view management, dataflow architectures (which seem to be common in the literature), or query language improvements.

Our integration strategy departs from those identified at the IEEE Database Issues for Data Visualization Workshop. It is a hybrid strategy that is partially *visualization as a database front-end* and partially *database as a visualization back-end*. The local database

view is a queryable object, not just a visualizable object. Visualizations can be queried via direct manipulation and widget controls.

As a knowledge discovery system in the KDD sense, the Exbase system has very low autonomy and high versatility (see Figure 2-18). This is quite the opposite of most knowledge discovery systems available today. Exbase could be effectively used as a KDD front-end, to orchestrate the usage of database, data mining and visualization tools.

The database-visualization integration research primarily focuses on new data models and query languages for spatio-temporal data, which is not our focus. We are more concerned with effective interaction support for database exploration, through the visualization interface. This requires the mapping of visual interactions to the database system. As with the database research, we can distinguish our research by our provisions for iconographic visualization. Compared to the visual data exploration system research, Exbase has the same high degree of user-data interaction, but is coupled to a relational DBMS.

Finally, in comparing our systems research with previous research, we differentiate ourselves by stressing the need for supporting a specific task repertoire, as opposed to simply mapping some database model to some visualization model. Taking a user-centered approach towards defining the system architecture, we feel, is a vital undertaking. Most of the related database research focuses on scientific data modeling, while most of the visualization research refrains from visualizing relational data. We give both database and visualization components equal weighting, and give special treatment to the mapping of interactions at the visualization interface to database operations.

As for the process research, the GDE is distinguished by the metrics. Not only does it define a data structure, it also defines numerous measures for describing the data structure. While some earlier research efforts do have graph representations of similar processes, none have gone so far as to formalize their models, or try to describe the process. Our descriptions are both static (i.e., structural metrics based on vertices and paths) and dynamic (i.e., temporal metrics based on data entity changes over time). All of the previous research takes a predominantly structural view of the process, in a domain-dependent manner, while our abstract model is domain- and task-independent. Concrete specializations of the model add the data dependencies.

Two interesting and similar research efforts are the Gaea and IMACS systems. The Gaea system does provide a derivational semantic layer to model the domain under investigation, but this is based on a dataflow model, which can be considered a dual of the GDE model. They do not attempt to perform any measurements of the session, however, and it seem like sessions are pre-planned. The IMACS system incorporates knowledge metadata that is user extensible, as does Exbase and the GDE model. Our model, however, places the knowledge metadata into a specific context with respect to the session, such that insights into the process can be made. The GDE model also provides process metadata to further describe the database exploration process.

7.3 Thesis Contributions

This research makes the following contributions to the fields of databases and visualization:

1. The systems research provides a solution for the relational database-iconographic visualization impedance mismatch that is based on the *view* concept. This solution is generalizable to any visualization domain, not just iconographic.

2. The systems integration model strikes a balance between database and visualization components: queries can be visualized and visualizations can be queried.

3. The systems research improves upon the dynamic query paradigm by visualizing a database as an iconographic texture instead of a simple scatterplot of points.

4. The process research defines a generalized model of the data exploration process, devoid of any data or database dependencies. This GDE model is based on *data entities* and *data derivations*, organized into sequences or graphs.

5. The GDE model presents a framework in which to describe data exploration sessions consisting of *data exploration state*, *space* and *scope*, and a taxonomy of various representation schemes.

6. The process research defines a suite of data-independent *metrics* over the GDE model that can be used to describe data exploration sessions. The metrics statically describe the history of data entity usage and aggregate graph structures.

7. The process research defines a *data exploration calculus*, and the related concepts of *derivational direction* and *continuity* to describe dynamic aspects of the process.

8. The process research defines canonical session patterns common to both specializations, and canonical graphs based on the patterns and metric semantics.

7.4 Future Work

Due to the interdisciplinary and broadly-focused nature of the research, numerous avenues for future research present themselves. Some address the current limitations of the research, while others extend the research in interesting ways. Future work falls into two main categories: improving the systems integration model and the Exbase prototype, and enhancing and extending the GDE model.

7.4.1 Improving the Systems Integration Model and Exbase

Given the limitations described in Section 3.3.6, we can greatly improve the robustness of the Exbase prototype. Exbase can be immediately improved by creating a client-side local database view cache, having a known size and replacement policy. The database would need to be updated with those local database views that are used frequently, and the GDE model process metrics can be beneficial. It would be important to store landmark and terminal vertices, while it may not be worthwhile to store sequential vertices. When a local database view must be replaced, a good first attempt would be to implement an LRU algorithm, with special preferences for retaining local database views that create landmark vertices. Landmark and terminal vertices would need to be actively inserted into the database, either under user control, or via some ranking algorithm based on the process metrics. The data exploration calculus is also useful, because it models the temporal nature of the process, and can be an arbiter of what gets inserted into the database, and what local database view to replace.

Since database exploration sessions may span days, weeks or more time, the entire session graph must be written to the database. Due to the potentially huge number of views and derivations that are created (consider the dynamic query approach), efficient

graph compression schemes must be developed. One approach could be to store only the metadata associated with each view or derivation. Another approach could be to exploit *closure* wherever possible, such that only the essential views and derivations are stored, such as would exist in a compressed session graph (Section 5.2). Perhaps only "delta" derivations can also be stored, such that the session graph may be rebuilt as needed. It is important to retain landmark vertices, tagged vertices (those with additional knowledge metadata that the analyst directly adds such as annotations) and terminal vertices, and the paths that lead to tagged vertices.

A further enhancement is to exploit the session graph in real time. An active database methodology can be developed to sense when the analyst is either panning or zooming on the data, for example. This information can then be used to instruct the DBMS to fetch additional tuples having different attributes than those in the current view, but along the same derivational direction (Section 5.5.3). This capability requires the data exploration calculus to be used in determining rates and vectors within the data exploration space.

A critical avenue for future work is to conduct experiments of actual database exploration sessions, and tabulate the process metrics. This would help validate the semantic Postulates of Section 5.4.2, and also give important information concerning the interaction styles and data management requirements of the systems integration. A prerequisite for this activity is the implementation of the GDE model and its specializations. This requires a methodology for comparing data entities to determine whether and where they should be included in the session graph. View comparison schemes based on metadata are required, since it may be prohibitively expensive to compare based on actual data values and still support interactive rates.

In order to make Exbase more useful, the session graph must be presented to the analyst, allowing the analyst to deal with known objects (views) through an intuitive interface (the graph itself). The view caching scheme would shelter the analyst from the actual location of the data. Effective graph visualization schemes must be employed, which, in and of itself, is a fertile research area. The taxonomy provided in Section 5.2 is only a high-level categorization, and we have chosen to use textual labels for edges. Considering all of the possible information that can be mapped to an edge, better visualizations can be developed.

7.4.2 Enhancing and Extending the GDE Model

One enhancement to the GDE model would be to develop a *weighted derivation path breadth metric*. Assuming Postulates 5.1 and 5.2 to be true, we can combine them, and assume branches closer to the reference vertex cover more of the data exploration space than branches farther away (i.e., at a greater data derivation depth). In this way, we can define a *weighted* derivation breadth value, $W(\mathbf{B}_{\text{path}})$, defined as:

$$W\left(B_{path}\right)=\sum_{i=1}^{N}\frac{b_i}{i},$$

where i is the depth where b_i is the breadth at depth i, and N is the maximum path depth. This may more accurately account for the effect of branching at different depths. Empirical data is required to properly validate this metric.

Extending the GDE model requires adding new database exploration domains to the current set. Interesting additions include additional visualization techniques, data mining techniques and world wide web browsing. We discuss each briefly.

There are many interactive visualization techniques to probe the visualized data for additional information. One dimensional probes attached to the cursor to probe for single data values are equivalent to point queries. Two dimensional cutting planes swept through a volume to portray a colormapped cross-section of the data set are equivalent to a 2D range query and additional visualization of the query result, along with the visualization of the enclosing space. An n-dimensional data brush selecting a 2D region of interest in a scatterplot matrix is equivalent to a 2D range query and possibly independent visualizations of the linked displays. The Magic Lens technique is a direct manipulation generic data filter that is driven by a 2D range query over a visualization, either displaying data not in the current visualization, or a new visualization within the lens geometry. Parallel coordinate display interactions are very similar to Exbase interactions, though the visual representation and graphical display settings are different.

Thus, at a first glance, it seems as if the GDE model can already model these techniques. Most seem to produce a query result visualization along with a visualization of the data being queried, something not done in Exbase. This might create a new kind of visual interaction pattern, where multiple derivation results are created. The primary differences are in the visual representation schemes and associated graphical display settings, which would be useful in correlating to data sets, analysts and exploration sessions in general.

Most data mining approaches utilize some algorithmic pattern extraction, such as decision trees, association rules and neural networks. A single data mining derivation may encompass numerous database-visualization derivations; it automates the discovery process. The derivations often will not modify the underlying data, but rather directly compute knowledge (semantic) metadata. In a larger context, database queries can

precede data mining derivations, to prepare the data subset for automated analysis. Data visualizations would be tables, textual rules, or simple graphs. Session graphs stand a good chance of being much smaller using data mining approaches, and might be equivalent to database-visualization compressed graphs. This would be a very interesting area for future work.

In world-wide web browsing, the web is a network database. Each vertex contains a measurable amount of multimedia-based knowledge, and each edge is a single type of data derivation, a navigation operation. All derivations are single-edged, and the session graph can have cycles, self-loops and multiple edges per vertex pair. The GDE model and metrics can be used to extract navigation paths, which can be compared against the knowledge stored at each vertex. In addition, a single web site can be monitored to determine the “prototypical” user, and other categories, based on the metrics and calculus. The data exploration calculus can be used to determine, for example, the temporal profile of a user, what pages were dwelled upon for a long time, if the user becomes lost due to an incorrect link, or has some other difficulty getting to a page deemed critical by the webmaster (such as an order form). Since the data exploration space and the information content on each page is known, they can be measured and factored into the metrics and their application.

After enough empirical evidence is gathered about the database exploration process, it may be determined that it indeed consists primarily of pan and zoom operations, where zooming follows panning (as in the canonical session graph example). If this is the case, then the session graph may resemble a *fractal*. The pattern of pan/zoom would be evident in any portion of the session graph, and a *fractal dimension* can be calculated.

One possible future enhancement to the GDE model is to advance a theory of database exploration as a physical system, where data entities are equivalent to physical objects and data derivations are equivalent to physical forces. Newton's laws are directly applicable to the basic forces found in nature: gravity, electromagnetism and the strong and weak nuclear forces; it would be interesting to see if Newton's laws can apply to GDE model, as a way to describe the data exploration process. For example, Newton's second and third laws are stated as:

II. **Law of Mass and Force** - The acceleration of a body is inversely proportional to its mass, and directly proportional to the force acting on it.

III. **Law of Interaction** - Forces always occur in pairs: for every action, there is always an equal and opposite reaction.

Force and mass are properties of physical objects that are defined in terms of Newton's laws. A force is an agent of change, often associated with a physical action. Mass is the amount of material that composes a physical object. Mass is synonymous with weight, but only under the force of gravity, and if the objects under consideration are weighed at the same place. Force and mass in physical domains are defined in terms of each other, related by some constant as indicated by Newton's second law (acceleration in the case of physical motion).

We may be able to define *data exploration force* as the knowledge base of the data analyst. This includes perceptual abilities, intuition and access to adequate reference materials. The data exploration force is manifested as derivations applied to data entities. We may also be able to define *data exploration mass* as the data under investigation. Not merely its schema and extension, but its meaning - the trends, patterns and embedded

knowledge the analyst is trying to uncover. Moreover, data exploration mass relates to the complexity of the data and the difficulty it poses to the analyst to unlock its secrets.

If we are able to do this, we may be able to dramatically increase the description and measurement of the data exploration process.

7.5 Concluding Thoughts

This research develops a data-independent model of the data exploration process, and an implementation to support a concrete specialization of the process. The systems portion of the research is highly dependent on the user-modeling portion of the research. Any improvements to the system model must come as a result of implementing and testing the user model further, and while we have defined a complex model with metrics, we have but scratched the surface of describing the entire database exploration process. Further implementation and testing is warranted, and this research provides a firm foundation for subsequent stages of the description process.