Abstract

Modular Visualization Environments (MVEs) have recently been regarded as the de facto standard for scientific data visualization, mainly due to adoption of visual programming style, reusability, and extendability. However, since scientists and engineers as the MVE principal user are not always familiar with how to map numerical data to proper graphical primitives, the set of built-in modules is not fully used to construct necessary application networks. Therefore, a certain mechanism needs to be incorporated into MVEs, which makes use of heuristics and expertise of visualization specialists (visineers), and which supports the user in designing his/her applications with MVEs.

The Wehrend’s goal-oriented taxonomy of visualization techniques is adopted as the basic philosophy to develop a system, called GADGET, for application design guidance for MVEs. The GADGET system interactively helps the user design appropriate applications according to the specific visualization goals, temporal efficiency versus accuracy requirements, and such properties as dimension and mesh type of a given target dataset. Also the GADGET system is capable of assisting the user in customizing a prototype modular network for his/her desired applications by showing execution examples involving datasets of the same type.

This paper provides an overview of the GADGET guidance mechanism and system architecture, with an emphasis on its knowledge base design. Sample data visualization problems are used to demonstrate the usefulness of the GADGET system.

CR Categories and Subject Descriptors: D.2.m [Software Engineering]: Miscellaneous – Reusable software, D.1.7 [Programming Techniques]: Visual Programming; I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods and Search – Heuristic methods; H.5.2 [Information Interfaces and Presentation]: User Interfaces – Interaction styles.

Additional Keywords and Phrases: visualization systems, Modular Visualization Environments (MVEs), dataflow paradigm, visineers’ heuristics and expertise, taxonomy of visualization techniques, knowledge base, object-oriented design

1 INTRODUCTION

The process of scientific data visualization is generally divided into the following three consecutive stages:

1. filtering which refines given numerical data;
2. mapping the data to graphical primitives; and
3. rendering the primitives to generate an image.

The dataflow [10] (or information flow [20]) paradigm has resulted in a class of visualization software systems, which provide an easy-to-use visual programming environment. These environments allow the user to follow a rough guideline to select and connect built-in functional units, called modules, and construct his/her own visualization applications, called networks. The user is also allowed to incorporate his/her own modules into the system’s module pool; and to share proprietary module-centered networks with colleagues as common basic resources for further applications.

The above-mentioned availability, extendability, and reusability are the primary reasons why such a class of systems, termed Modular Visualization Environments (MVEs), nowadays can be regarded as the de facto standard for scientific data visualization. MVEs are especially suitable for solving application problems with a short lifetime, and hence they are capable of serving as a rapid prototyping environment. Indeed, in this respect, MVEs have been making a great contribution to the increase in the population of visualization end users in various scientific/engineering disciplines. Such representative systems include apE, AVS, IRIS Explorer, IBM Data Explorer, and Khoros. For detailed historical and technical perspectives of MVEs, see references [30, 24, 3].

In practical situations, however, since most of the MVE users are not visineers (visualization specialists) [25], they are not always familiar with how to map their datasets on hand to proper graphical primitives in the second dataflow stage described above. Consequently, it has been often reported that the powerful set of built-in mapping modules is not fully used to construct necessary application networks for realizing the intrinsic nature of their datasets. Furthermore, since module connectivity in current MVEs depends only on the inter-module communication data types, a high degree of freedom in module connection presents difficulty for novice users who construct their own networks from scratch using the module pool whose cardinality is around a couple of hundreds. Therefore, a certain mechanism needs to be incorporated into MVEs, which makes use of heuristics and expertise of visineers to offer support for the users in designing their applications with MVEs.

The Department of Information Sciences, Ochanomizu University has recently begun development of a system called GADGET (an acronym for Goal-oriented Application Design Guidance for Modular Visualization Environments). The early version of the system was presented in [19]. The goal of the GADGET system is to interactively help the user design appropriate applications according to specific visualization goals, temporal efficiency versus accuracy requirements, and such properties as dimension and mesh type of a given target dataset. Also the GADGET system is designed so as to be capable of assisting the user in customizing a prototype module network for his/her desired applications by showing execution examples involving datasets of the same type.
The next section provides an overview of the GADGET guidance mechanism, with an emphasis on the Wehrend’s taxonomy of visualization techniques [32, 14] as the primary component of the underlying visineers’ heuristics and expertise. Section 3 is devoted to the GADGET’s knowledge base, including the conceptual schema design using an object-oriented modeling methodology. A translated relational schema and an example operation with a set of occurrences are also described. Section 4 presents the main flow of design guidance and the entire architecture of the GADGET system. In section 5, two sample data visualization problems are used to demonstrate the usefulness of the GADGET approach. Section 6 discusses the further potential of the GADGET system with reference to other existing visualization system approaches.

2 GUIDANCE MECHANISM

As the primary component for the visineers’ knowledge representation, the GADGET system relies heavily on the goal-oriented taxonomy of visualization techniques, which was originally proposed by Wehrend, et al. [32]. The feasibility of this approach was demonstrated by indexing a hundred of collected visualization pieces in [14]. First, Section 2.1 gives a formal review of the taxonomy-based method for selecting visualization techniques. Then, Section 2.2 specifies other requirements for the GADGET’s user guidance capabilities.

2.1 Taxonomy-Based Selection Method

The Wehrend’s taxonomy hypothesizes that identification of a visualization goal (what the user expects to extract from a given dataset) suggests appropriate techniques to use in achieving that goal. GADGET takes advantage of this taxonomic knowledge at the first stage of interaction with the user to determine proper techniques based on the visualization goal.

In the approach, visualization goals are identified simply by pairing words (verbs and objects) from two types of vocabulary lists, i.e., actions and targets. The classification of actions distinguishes problems in terms of the representation goals, while the classification of targets groups techniques based on the nature of objects in the target domain [32]. Herein, specific vocabulary lists used in [14] are adopted for the set of actions $A$ and the set of targets $D$, namely, $A$ includes nine actions: {"Identify", "Locate", "Distinguish", "Categorize", "Cluster", "Rank", "Compare", "Associate", "Correlate"}, and $D$ includes seven targets: {"Scalar", "Nominal", "Direction", "Shape", "Position", "3ERO", "Structure"}. An example of a visualization goal is to “Identify Direction.”

Since several actions such as “Distinguish”, “Compare”, “Associate”, and “Correlate” require more than one target, the set of visualization goals $P$ is defined generally as follows:

$$P \subset A \times 2$$  \hspace{1cm}(1)

In addition to the visualization goals, properties of the target dataset such as dimension (the number of independent variables) may be used as another important criterion for eliminating visualization techniques as candidates. Hence, the combination of goal identification and target dimension specification can be used together to select a set of more appropriate visualization techniques.

2.2 Other User Guidance Requirements

The visineers’ technique classification and module network selection methods described in Section 2.1 will be represented in the GADGET knowledge base, which is the main focus of the next section. In order to refine the knowledge for augmenting the GADGET capabilities for user guidance, the following three items must be considered in the design of the knowledge base:

Additional requirement 1: For the user’s convenience in selecting techniques, the technique classification function $K$ should return the sorted list of techniques from $T$ according to empirical effectiveness.

Additional requirement 2: In general, there exist many algorithms that embody a single visualization technique. Temporal efficiency versus accuracy trade-offs can be found among these algorithms. For allowing the user to judiciously choose a usable module network, the network selection function $K$ should account for the nature of an algorithm employed in a principal mapping module, and should include the information related to such trade-offs for each module network in the resulting list.

Additional requirement 3: The GADGET system should handle not only the explicitly-organized knowledge of visineers, but also module network execution examples as implicit disordered knowledge of practical importance. Consider the case that a dataset cannot be visualized sufficiently with a prototype module network which the GADGET system recommends. In such a case, the system should allow the user to

\[K : (P \times \Sigma) \mapsto 2\]  \hspace{1cm}(2)

where $\Sigma$ and $\mathcal{T}$ denote the sets of dimensions and techniques, respectively. For example, the set value for $K$ ("Identify", "Direction", 3) includes "3D hedgheogs", "3D glyphs", and the like.

Extending the above-mentioned Wehrend’s approach makes it possible to select appropriate MVE module networks. In general, there are more than one known algorithm for each of the visualization techniques. Algorithms can be distinguished from one another by the applicability to mesh type of a given target dataset, and the set function $K$ for choosing module networks can be devised so as to have type

$$K : (\mathcal{T} \times \mathcal{C}) \mapsto 2$$  \hspace{1cm}(3)

where $\mathcal{C}$ denotes the set of mesh types, and includes “regular”, “rectilinear”, “curvilinear”, “unstructured”, or “general” (no topology)5. Figure 1 depicts these five cases in 2D. $\mathcal{M}$ is the set of modules offered by an MVE, and $G(M)$ is the set of visualization networks, or a class of directed graph with the elements of $\mathcal{M}$ as nodes. The ease of formulating the range of $K$ is one of the strengths of MVEs, leading to the ease of knowledge base construction.

![Figure 1 GADGET mesh types of target dataset.](image)

Since not all currently available MVEs support polymorphism, the present idea still has a practical merit.

---

5As for diversified analysis of visualization-related concepts, see reference [4], for example.

6In the original paper [32], these terms correspond to operations and objects, respectively.

7Acronym for Spatially Extended Region or Object

8The number of targets is conceptually different from that of dependent variables.
3 KNOWLEDGE BASE DESIGN

3.1 Conceptual Modeling with OMT

This section describes an object-oriented design methodology, called OMT (Object Modeling Technique) [27, Chapters 3 & 4], which is utilized to perform structural modeling of the GADGET knowledge base. OMT is one of the most popular tools for logical database/knowledge base design, and the use of OMT object modeling allows one to capture the structural skeleton of the GADGET knowledge base easily. Figure 2 depicts an OMT class diagram for the GADGET knowledge base.

The object model consists of nine classes, each of which is depicted with a rectangular box. The name of class is depicted in the upper part of the class box, and a lower part contains pairs of name and domain of class attributes.

Following the definition shown in Equation 1, the Goal class is designed using two types of OMT data abstraction concepts; that is, aggregation and recursive aggregation. The Goal class is an aggregation (the Cartesian product) of two classes Action and Target. A variable list of targets appearing in the Target class is aggregated recursively from the base Primitive target class of targets. Note that the Target class possesses an attribute dimension.

The set function $K$ (Equation 2) to classify visualization techniques is modeled with a many-to-many association between Goal and Technique. Note that to meet additional requirement 1 stated in Section 2.2, a single link attribute called priority is added to the association. An attribute description of Technique is to hold the related technical information.

The set function $K$ (Equation 3) to select module networks is modeled with two consecutive associations between Technique and Module network via an intermediate class called Algorithm. The link multiplicity depicted with the solid circle at Algorithm side indicates that it is possible for a single technique to be associated with more than one algorithm. The Algorithm class has an attribute mesh type to store the information about the most general class of datasets to which each algorithm is applicable (see Figure 1). Here, to realize the user guidance specified in additional requirement 2, time and accuracy are incorporated as link attributes for the association between Technique and Algorithm. The expression beneath the bottom line of the link attribute box is a constraint to qualify the relationship between actual occurrences for time and accuracy link attributes.

There can be found a bijection between Algorithm and Module network. The network fileID attribute of Module network holds an index to the corresponding module network file. Since Example network is defined as a sub-class of Module network, network fileID attribute is inherited and used together with its own attribute datafileID to generate example images, thus being devoted to the fulfillment of additional requirement 3. Explanation of examples is retained in the attribute description of Example network.

3.2 Relational Schema Design

As will be described in Section 4, the GADGET knowledge base is finally implemented on a relational data base management system, primarily because the relational database technologies are matured, dominate in the marketplace [27], and have good code portability. The OMT object model designed in Section 3.1 is then translated to a corresponding relational schema, mainly by using the OMT design guidelines to map OMT concepts to relational tables [27, Chapter 17]. Figure 3 shows a resulting conceptual schema used in the current GADGET system.

<table>
<thead>
<tr>
<th>table name</th>
<th>attribute name</th>
<th>domain name</th>
<th>#bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>actionID</td>
<td>varchar(4)</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>action_name</td>
<td>varchar(16)</td>
<td>16</td>
</tr>
<tr>
<td>Target</td>
<td>targetID</td>
<td>varchar(4)</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>target_name</td>
<td>varchar(32)</td>
<td>32</td>
</tr>
<tr>
<td>Goal</td>
<td>goalID</td>
<td>varchar(4)</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>algorithmID</td>
<td>varchar(4)</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>mesh_type</td>
<td>varchar(32)</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>modular_net_name</td>
<td>varchar(64)</td>
<td>64</td>
</tr>
<tr>
<td>GoalAndTech</td>
<td>goalID</td>
<td>varchar(4)</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>techID</td>
<td>varchar(16)</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>description</td>
<td>varchar(32)</td>
<td>32</td>
</tr>
<tr>
<td>Algorithm</td>
<td>algorithmID</td>
<td>varchar(4)</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>algorithm_name</td>
<td>varchar(32)</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>network_fileID</td>
<td>varchar(64)</td>
<td>64</td>
</tr>
<tr>
<td>TechAndAlgorithm</td>
<td>algorithmID</td>
<td>varchar(4)</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>techID</td>
<td>varchar(16)</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>time</td>
<td>real</td>
<td>4</td>
</tr>
<tr>
<td>Example_network</td>
<td>example_netID</td>
<td>varchar(4)</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>algorithmID</td>
<td>varchar(4)</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>ex_description</td>
<td>varchar(32)</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>data_fileID</td>
<td>varchar(32)</td>
<td>32</td>
</tr>
</tbody>
</table>

Figure 2 OMT class diagram for GADGET knowledge base.

Figure 3 Relational schema for GADGET knowledge base. Attribute names with solid and dashed underlines indicate primary and foreign keys, respectively.
The relational schema consists of eight tables. A specific ID is required to identify tuples for each table. Introducing compound target occurrences simplifies the definition of the recursive aggregation concept around the Target class. Also for simplicity, it is assumed that all the techniques and algorithms are designated for 3D target datasets, thereby omitting dimension attribute from the Target table. The TechAndAlgorithm table is left intact, instead of being merged into Algorithm, for ease of future extensions, while Module network class is degraded into two corresponding attributes of the Algorithm table. The IS-A relationship between Example_network and Algorithm is represented implicitly via the foreign key algorithmID in the Example_network table.

The relational schema is stable and compact, because it can be shown in Boyce-Codd normal form [5]. Figure 4 shows sample occurrences in the GADGET knowledge base. As an example of internal knowledge processing executed in the GADGET system, consider the following sample query on the knowledge base:

```sql
QUERY: Select the most accurate algorithm embodying the most appropriate technique to "identify shape" of a given 3D dataset, and print the name of the corresponding module network file.
```

Below is a sequence of SQL statements to evaluate the query:6

6These statements are formulated in Transact-SQL\textsuperscript{3d}, SYBASE\textsuperscript{3d} enhanced version of the SQL relational database language [18]. The "into #tmp" clause specifies that the result tuples are inserted into a newly created temporary relation named #tmp. Note also that these statements could be formulated in one statement, even though the statement would be less readable.

select algorithm_name, modular_net_name into #tmp from Action, Target, Goal, GoalAndTech, TechAndAlgorithm, Algorithm
where Action.action_name = 'identify'
and Target.target_name = 'shape'
and Action.actionID = Goal.actionID
and Target.targetID = GoalAndTech.techID
and Goal.goaID = GoalAndTech.goaID
and GoalAndTech.techID = TechAndAlgorithm.techID
and TechAndAlgorithm.algoID = Algorithm.algoID
and priority = 1

select algorithm_name, modular_net_name, from #tmp
where accuracy = (select max(accuracy) from #tmp)
```

The evaluation result with the sample occurrences shown in Figure 4 is as follows:

<table>
<thead>
<tr>
<th>algorithm_name</th>
<th>modular_net_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interval volume (GCH: QF)</td>
<td>iv_qf.net</td>
</tr>
</tbody>
</table>

As for the details about interval volume, which is retrieved above, see Section 5. In the next section, the focus will be shifted to the architecture of the entire GADGET system.

Figure 4 Sample occurrences in GADGET knowledge base.
4 FLOW OF APPLICATION DESIGN

Figure 5 illustrates the main flow of GADGET application design guidance. The behavior of the GADGET system can be described with six states, which are grouped into three parts: specifying visualization goals; suggesting appropriate techniques and algorithms; and supporting the browsing of examples.

Visualization techniques and algorithms are assumed to be classified in advance in terms of visualization goals and properties of datasets. These classification indices make techniques orderly, and are useful for selection of the effective categories of MVE visualization module networks. Specifically, the system first allows the user to specify his/her visualization goals characterized by the indices (state1), and shows the sorted list of the recommended visualization techniques (state2). Then, the system accepts a requirement on target dataset mesh type, and returns the list of applicable visualization algorithms (state3). The user is allowed to re-sort this list according to his/her preference concerning temporal efficiency or accuracy of algorithms. Each of the visualization algorithms presented in state3 is related to a recommended MVE module network implementing the algorithm. The network, once loaded to an MVE, is executable and modifiable (state4). The recommended network is “full fledged”: it is comprised of a principal module that implements the mapping algorithm, and supporting modules that implement pre/post processes such as data input, filtering, resizing, rendering, and image display. Hence, the user can apply the suggested network to his/her own dataset on the spot.

In addition to recommended principal network, each algorithm is associated with networks exemplifying specific cases (state5). On user demand, each of the module networks is automatically executed, and the resultant image is displayed (state6). Browsing such examples can provide the user with empirical knowledge to customize the recommended network in a trial-and-error manner.

The GADGET system consists of an MVE, a GUI, a knowledge base system, a querying subsystem, and a dedicated system manager loosely coupling these components, as shown in Figure 6.

The system manager is the kernel to control the transition of the system, such as retrieval from the knowledge base, displaying of the results through the GUI, and transfer of module networks to the MVE. The knowledge base system is consulted through the querying subsystem at the transition from one state to another. The GUI and the MVE are the accessible components for the user. These components are associated with the states depicted with rectangles and rounded-edge rectangles, respectively, in Figure 5. The GUI passes input data to the system manager, and shows the retrieved knowledge back to the user, and so on. The MVE allows the user to modify a network and its properties.

The current preliminary version of the GADGET system has been implemented on a Sun SPARCstation10, using Sybase as the knowledge base management system, and AVS 4.9 [1] as the MVE. All the GUI codes are written with a builder called OpenLook.

5 APPLICATIONS

In order to demonstrate the usefulness of the GADGET system, two execution examples are described herein. One is to confirm the mechanism to assist the user in selecting appropriate MVE module networks to solve a simple problem of volume visualization [13] (Section 5.1). The other is to demonstrate the process to visualize a dataset from CFD research, called bioconvection (Section 5.2).

5.1 Volume Visualization Example

Suppose here that as a visualization goal “Identify Shape” of 3D curvilinear volume data is given.

To start with, the visualization goal is specified (state1). In this example, action and target are “Identify” and “Shape”, respectively (Figure 7(a)).

According to the input goal, the set of appropriate techniques is retrieved from the knowledge base, and shown in the technique list window (state2). In this case, solid fitting, Surface fitting, and Direct volume rendering are retrieved and shown in this order (Figure 7(b)). The concept of solid fitting was presented by the authors as a generalization of traditional surface fitting [7]. The concept uses a geometrical volume data model, called interval volume, which represents a subvolume for which the associate scalar values lie within a user-specified interval.

Then, selecting Solid fitting, followed by specifying mesh type from the same window gives the system a trigger to make it retrieve and show the list of available related algorithms in the next window (state3). Here, if the user pushes the buttons time or accuracy, the listed algorithms are re-sorted in order of the specified preference (Figure 7(c)).

It is assumed here that the most accurate version of interval volume SCHOF is selected. This version takes advantage of an auxiliary algorithm called quadratic fitting. This is known as the most promising method among the set of Gradient Consistency Heuristics proposed by van Gelder, et al. [8] to alleviate topological ambiguities in determining boundary surface patches created by the curvature.

7SPARCstation10 is a registered trademark of SPARC International.
8Sybase is a registered trademark of Sybase, Inc.
9AVS is a trademark of Advanced Visual Systems, Inc.
10OpenLook is a trademark of Sun Microsystems, Inc.
Marching Cubes isosurfacing algorithm[17]. For more details, see reference [29].

If the user pushes the button Example, several preloaded sample data can be chosen (state5) to be applied to the corresponding MVE network as in Figure 7(d), and resultant images are automatically displayed (state6). On the other hand, if the user pushes the button ReadModule, the corresponding MVE network becomes available to the user in the MVE window, and the user is allowed to input his/her own data to the network (state4). When the result is not as good as expected, the module network can be adjusted freely.

Figure 7(e) shows a final image of iso-electron density interval volume extracted from the High-Potential Iron Protein volume data [21], using the module network shown in Figure 7(d).

5.2 Case Study: Bioconvective Data Visualization

Bioconvection, named by Platt[22], forms characteristic aggregation patterns, like fingers, beneath the surface of cultures of aquatic microorganisms (Figure 8). In recent years, several researchers have attempted to visualize simulated data of the phenomena [11].

Since the essence of the phenomena considered is the interaction of microorganisms with surrounding water, the main goal of the bioconvection simulation is to correlate distribution of microorganisms with the structure of fluid flow field. Suppose that datasets available here include 3D regular multiscalar volume data for density of microorganisms and the magnitude of velocity of water in which microorganisms are present. The main goal can be abstracted with the present vocabularies into “Correlate (two) Scalars.”

Figure 9 shows the result of the GADGET’s user guidance for visualizing the bioconvective data. The user selected Colored isosurfaces from the list of recommended techniques for correlating scalars. For fast display, the user then chose the original Marching Cubes algorithm[17] in the upper-right window. The corresponding module network was used to generate the final image in the lower-right window, where the isosurface for the magnitude of velocity of water is visualized, whose color represents the density of microorganisms. The result reveals a certain number of cyclic flow patterns and the fact that a relatively large number of microorganisms tend to accumulate in downward flows.

6 FUTURE WORK

The system called GADGET has been proposed as an enhanced MVE to allow the user design visualization networks based on the methodology for classifying visualization techniques and for showing visualization examples. The taxonomic knowledge contributes toward reducing the cost of initial visualization network design, while the examples are useful for the user to fine-tune the prototype module networks suggested by the system. The GADGET system can be viewed to show a new possibility to integrate visualization with database/knowledge base management [16].

However, from the viewpoint of supporting the user, the current GADGET system is still in the early stages. The following are left as future R&D issues:
Very limited cardinality of the vocabularies for action and target lists makes it difficult for the user to abstract practical visualization goals. It is a challenging theme to enrich the vocabularies so as to cover the whole lifecycle of scientific simulation, which the GRASPARC system [2, 3] attempts to treat, and a newly-emerging field called information visualization [9].

In addition, the system should be able to offer a wider range of candidates for visualization techniques and algorithms to achieve a variety of goals. The concept of multi-sensory re-

alization [25], including sonification [6] and haptization [12], is expected to provide a rich set of powerful techniques and algorithms.

In general, the user tends to represent more than one analysis result within a single image. This means that GADGET should have a certain capability to assist the user to design module networks to achieve multiple goals simultaneously. To this end, GADGET should adopt higher-level visineers' heuristics used in another existing knowledge-based visualization assistance system called Vista [28], and Visage and SAGE[26]. Also, some hints could be learned from Glyphmaker [23], which is directed to finer-grain design guidance that allows non-expert user to customize his/her own graphical representations to capture more meanings from the same datasets.

In the current system, browsing is an only way to evaluate the query results. When a large number of results are retrieved, it is difficult for the user to select an appropriate visualization module network. Therefore, the method to evaluate the retrieved results quantitatively should be investigated [3, 31].

In the network modification state, GADGET gives suggestions for improving the network only through the display of similar examples. Thus, the quality of the results depends heavily on the user’s “reasoning” capability. In addition to such indirect suggestion, attempts should be made at automatic construction of modification proposals by an additional intelligent subsystem like an inference engine based on case-based reasoning [11] [15].

Acknowledgements

This work has been supported in part by the Ministry of Education, Japan under grant 09680330, and Japan Radio Co., Ltd. The authors would like to thank the reviewers for their useful comments. Ken W. Brodlie at University of Leeds kindly offered valuable information on recent trends in visualization research and education. The authors are grateful to Karen M. Vierow at NUPEC for her patient proofreading of English. Special thanks are due to Xiaoyang Mao at Yamanashi University for her technical suggestions and encouragement.

References


[11] It is a paradigm that derives the solution through directly using past examples (success or failure), which is similar to a given problem, followed by adjustment of the examples.