
SUPPORTING SYSTEMS ANALYSIS AND DESIGN THROUGH FISHEYE VIEWS

Demonstrating the usefulness of integrating context-based views
into the systems analysis and design process.

BY OZGUR TURETKEN, DAVID SCHUFF, RAMESH SHARDA, AND TERENCE T. OW

With the emergence of integrated, enterprisewide information systems, the task of systems analysis has become increasingly complex. Systems that used to be confined to “functional silos” must now cross organizational boundaries. Consequently, there is an abundance of information contained not only within an organization’s data stores, but also in the schemas and blueprints of the systems themselves. Analysts who must learn how a system operates may face an information overload problem. On the other hand, a “gestalt” approach to system development argues that in order to fully understand a detail, one must understand the context in which it exists, but this approach is often ignored when individual systems are planned [3].

One way of coping with information overload is to create a grouping structure within the collection of information [6]. Accordingly, complex graphical models representing information systems are often visualized through a grouped, hierarchical structure. However, as the systems get larger and the hierarchies become more complex, understanding the system details within the context of the overall system

becomes a daunting task. The hierarchical structures force the analyst to mentally integrate detail and context, hence causing disorientation [8].

The traditional visualization support of zooming and panning the whole graphical model does not address the problems of information overload and disorientation [7]. A visualization technique that could be useful for understanding graphical system models is the fisheye distortion, which shows how high-granularity details are related to their context by combining them within the same diagram. Here, we present two potential applications of fisheye views for systems analysis and design. After briefly summarizing fisheye views, we propose its use in Data Flow Diagrams to aid in understanding processes, and in entity-relationship diagrams to better understand data models. We also report on an experiment that tested the efficacy of the fisheye-based DFD visualization using a preconfigured prototype.

Fisheye Views

The fisheye view technique [2] is inspired by the nonlinear (distorted) way in which people perceive their environments. We pay greater attention to details that are in our immediate surroundings while



acknowledging only major landmarks that are farther away. Such landmarks constitute the context within which details have meaning. A “fisheye view” is the application of this idea to graphical interfaces where details are shown within their context.

Creating a fisheye view of a diagram involves assigning a “degree of interest” to each object [2] in the space examined. The degree of interest of an object is determined by its original importance and its proximity to the current focus of interest. Proximity can be based on physical distance such as on a geographical map, or a semantic distance such as the time between two events. The calculation of the degree of interest is based on the following equation:

$$\text{Degree of Interest} = \text{Original Importance} - \text{Distance from Focus}$$

The views are constructed such that the size of each object is proportional to its degree of interest. Accordingly, the size of the in-focus elements would be larger relative to the out-of-focus elements. This is different from the traditional display of multilevel diagrams where the details are either visible or not visible [8].

Using variants of the basic principle of assigning

“degrees of interest” to different objects in a diagram, fisheye views could be defined in a number of different structures [5]. For example, they have been used in graphical presentations of hierarchies in domains such as groupware [4], hypertext [1], and Web search [10]. Figure 1 shows a Web search application, which imposes a fisheye view upon a set of clustered Web search results (the upper graphic of Figure 1). When a cluster is selected, it is made the focus. The focus cluster’s contents are shown in detail, while the remaining clusters are made smaller (the lower graphic of Figure 1). This allows the user to more fully investigate the selected cluster by “exploding” it, while still maintaining a view of the broadest level of the original search. The intention is to keep the user from losing track of the original search results.

Application of Fisheye Views to Systems Analysis

Viewing the metadata of a system is another important application of fisheye views. The application of visualization in general is not new to systems analysis: the entity-relationship diagram, the data flow diagram, and class diagrams are all visual models. However, these diagrams are not “context-aware.”

CAREN ROSENBLATT

They can represent an entire system, or a specific part of a larger system, but not both at the same time. As a result, system designers must often use a complex diagram of a system, or a series of many less-complex drawings. This causes the designer to be overloaded with information and/or lost within a series of drawings. Either way, understanding the role an individual system serves within an enterprisewide system can be lost.

To demonstrate how the fisheye view's context-orientation can be used to aid context-oriented understanding for the systems designer, we present two potential applications. The first applies this technique to process modeling by presenting a fisheye view of data flow diagrams. The second example, a data modeling application, combines the fisheye view with the entity clustering technique [9] to represent entity-relationship diagrams.

Process Modeling: Data Flow Diagrams

A data flow diagram (DFD) is a commonly used model to represent the flow of data between an organization's business processes. The DFD is a good candidate for the application of the fisheye view technique because the possibility exists for information overload and disorientation. The amount of information conveyed through a DFD—every business activity with its associated data flows—is overwhelming. To avoid this, these diagrams are organized in a hierarchical structure. A set of DFDs for a system is commonly comprised of a single high-level diagram (called a “level 0” diagram) that shows all major processes, and a series of more detailed diagrams that breaks down the processes into subprocesses of increasing levels of granularity (see Figure 2, Part A). On the other hand, the complex navigation between multiple diagrams at different levels of such a hierarchy could cause disorientation.

To create fisheye views of a series of DFDs, the detail of the subprocess (the focus) is embedded into a higher

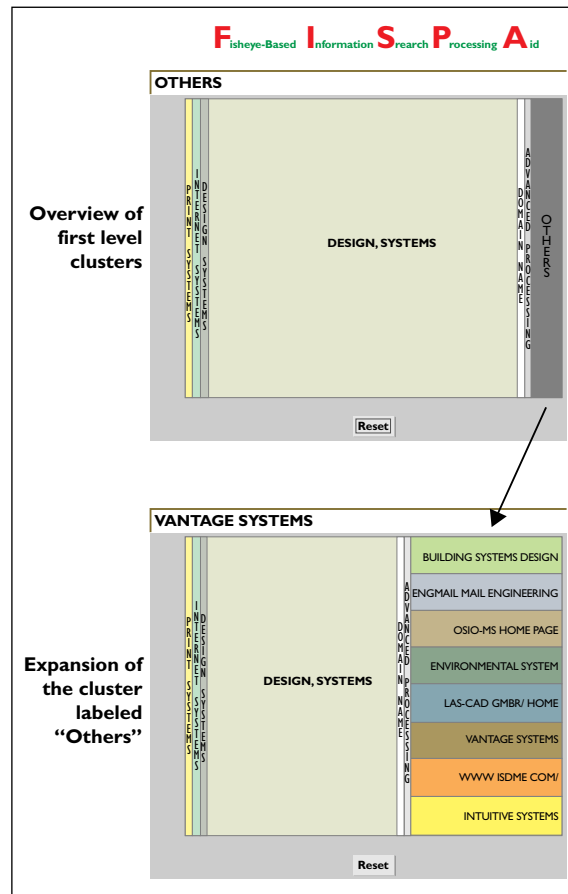


Figure 1. Application of a fisheye view to Web search.

level diagram (the context). Heuristics to determine the size and position of the elements inside the focus relative to elements in the rest of the diagram is based on the “degree of interest.” This can be formalized by assigning specific values to the degree of interest (DOI) function. Using “1” as the level of original importance for all level 0 elements, we arrive at the following equation:

$$DOI = 1 - k * DF$$

where DF is the number of levels in the hierarchy between the element and the focus. The coefficient k (a percentage) represents the amount of reduction in the degree of interest as the distance from the focus increases. A series of relative sizes for DFD elements can be derived from this equation.

For example, if k were 0.2 and the focus of the DFD were a level 2 subprocess, the level 0 contextual elements would be 60% of their original size, yielding a larger proportion of the diagram for the details of the focused elements. Elements where the DOI is zero or less may be aggregated with other elements, or be excluded from the diagram altogether.

In order to test this application of the fisheye technique, we developed a series of preconstructed, static images on linked HTML pages that facilitated movement between the Level 0 diagram and subprocess diagrams [7]. Each image was either a full-zoom (regular) view, or a fisheye view, of a process. The individual processes, as well as the other elements in the DFD (data stores and external entities), were weighted with an equal level of original importance. Therefore, the size and position of an element were determined only by the distance from the focus.

The diagrams had two levels of detail: “level 0,” and one additional level of detail for each subprocess. This resulted in two levels of the degree of interest: that of in-focus elements, and that of out-of-focus elements. The degree of interest for out-of-focus objects can be predetermined or may be variable and controlled by the viewer. Accordingly, elements that are part of the focus become larger, and others

become smaller (see Figure 2, Part C). Users navigated between these images using a series of hyperlinked Web pages.

An experiment was conducted in which subjects either navigated the “traditional” series of diagrams (with subprocess detail shown in separate diagrams from the level 0 diagram), or the fisheye view diagrams. Subjects were asked to identify processes (as shown on the DFDs) that were associated with a specific business event. It was found that subjects given the fisheye diagrams were able to correctly identify more processes than those given the traditional diagrams, providing support that the fisheye view is effective in enhancing understanding. It should be noted that the “two-level” prototype system is a relatively simple example. In practice, there would likely be many more levels. However, it is reasonable to assume that in those more complex systems, there is an even greater chance of information overload and disorientation. Therefore, an even greater potential benefit from the fisheye presentation could be anticipated.

Data Modeling: Clustered Entity-Relationship Diagrams

As DFDs are used to model processes, entity-relationship diagrams (ERDs) are used to model data. For even a moderately complex database, the amount of information contained in the corresponding ERD can become large. In order to reduce the complexity of the ERD, Teory and colleagues [9] developed a technique to group related entities. This method reduces overload through the creation of “clustered ERDs.” However, the designer may be required to switch frequently between levels of cluster detail and a top-level overview diagram in order to understand the entire data model, potentially creating disorientation.

The trade-off between information overload and disorientation makes the clustered ERD another candidate for the application of fisheye views. As with DFDs, entity clustering creates a “built-in” hierarchy to which the view can be applied. The fisheye view is created by expanding one or more clusters, while leaving the remaining clusters collapsed.

As in the case of the DFD application, the original importance of each entity cluster in the ERD is considered equal; therefore the “degree of interest” in elements in the fisheye view is based on their semantic distance from the focus. In ERDs, a natural choice for the semantic distance is the strength of relationship between entities. “Distance” would increase depending on how many other clusters are needed to establish the indirect relationship. The equation for determining relative element sizes for clustered ERDs

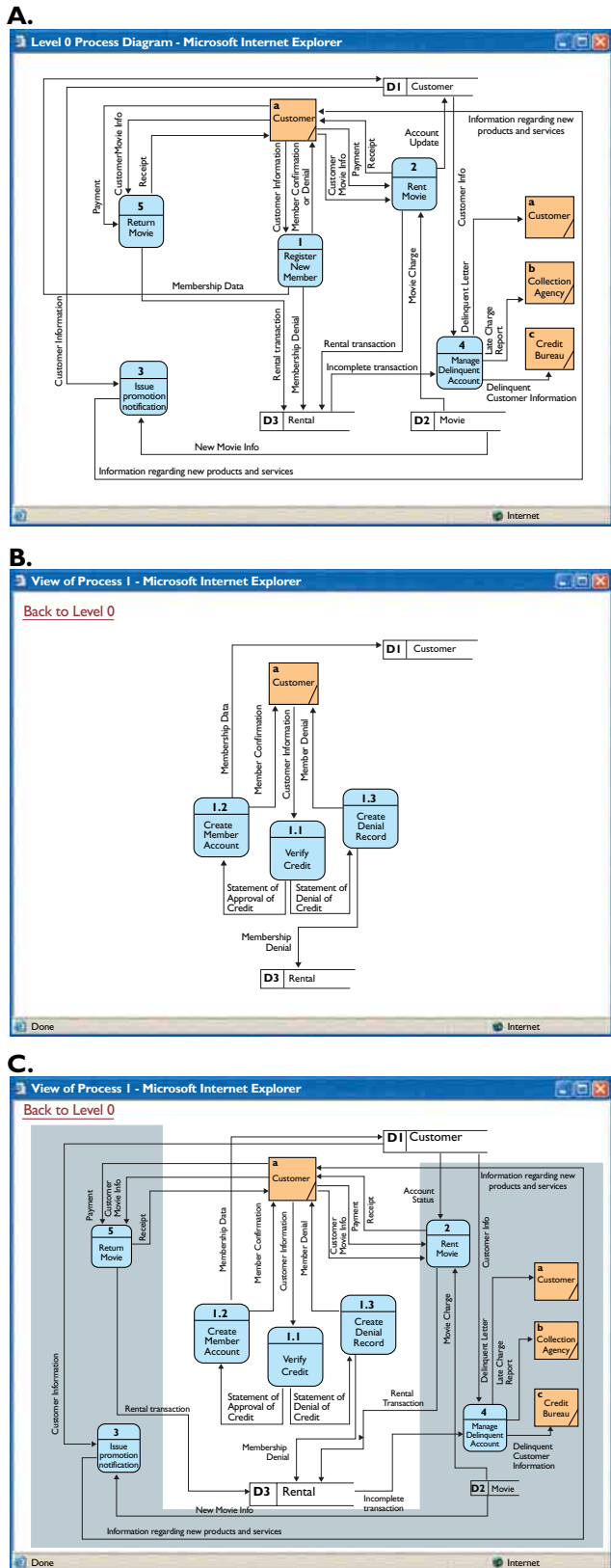


Figure 2. Part A: A level 0 DFD; Part B: The details of subprocess 1 without context; Part C: The fisheye view, with the details of subprocess 1 within its context.

Figure 3. Part A: A clustered ERD; Part B: The fisheye view, showing details of the “customer” cluster.

is essentially the same as the one used for the DFDs. Given a uniform level of original importance (1):

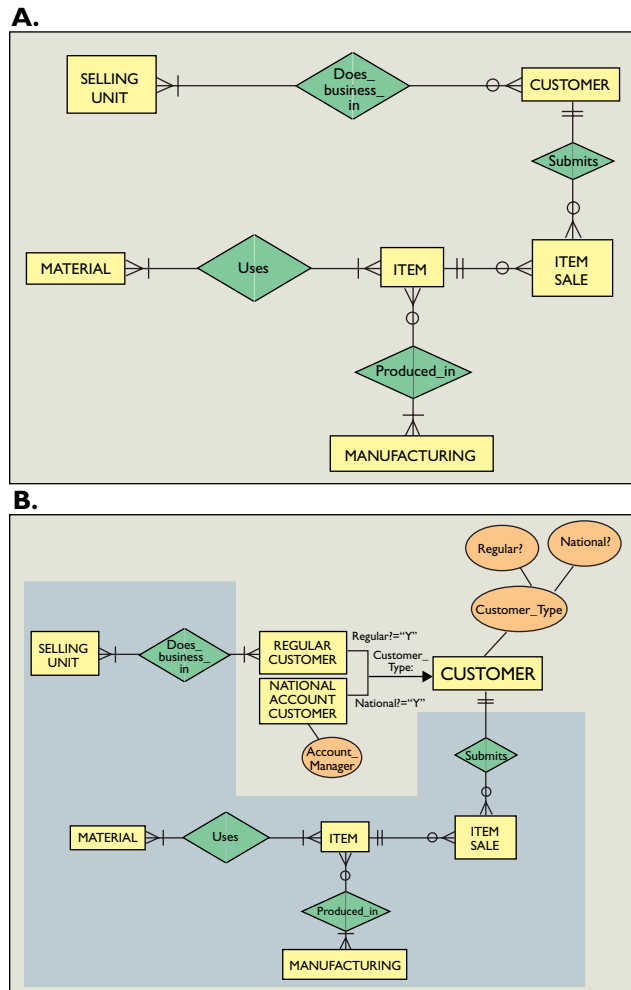
$$DOI = 1 - k * DF$$

In this case, the distance of a cluster from the focus (DF) is defined by the number of clusters needed to establish the relationship with the focus cluster.

Figure 3 shows a clustered entity-relationship diagram that is created using the clustering rules reported in [9]. The figure also displays a fisheye view of the same diagram with the “customer” entity cluster in focus. As seen in the figure, all the details of the entity cluster in focus are visible while those clusters that are not in direct relationship with customer are smaller in size and located further away.

The ability to see detail of a specific portion of an ERD in the context of the entire data model has implications for database design. First, the fisheye view may help designers optimize the physical arrangement of the tables in the schema implied by the ERD. Identifying where the relationships exist between entities can help a database administrator decide on which disk drives tables should be placed. Second, this method could provide insight into the development and implementation of distributed databases, where each cluster represents a portion of the distributed schema. Therefore, it closely represents the way people perceive and use the data in a distributed database, where data is placed closest to the users to whom it is most important. We are currently in the process of designing user studies to test such expected benefits from the fisheye views of ERDs.

As in the example of clustered logical models (ERDs), the application of fisheye views to a clustered table schema could also yield benefits for database administrators. A fisheye view of the schema could facilitate the construction of queries that span tables



across multiple clusters. This visualization technique would be particularly effective if integrated with a Query by Example (QBE) tool.

Conclusion

The two applications described in this article demonstrate the potential usefulness of the fisheye view as an aid to the systems analysis and design process. To take advantage of this technique, the fisheye view can be incorporated into existing Com-



DEVELOPMENT OF CONFIGURABLE FISHEYE VIEW ALGORITHMS AND THEIR IMPLEMENTATION AND INTEGRATION IN EXISTING TOOLS IS A MAJOR OPPORTUNITY FOR DEVELOPERS OF SYSTEM TOOLS.

puter-Aided System Engineering (CASE) tools. Versions of a context-aware user interface are currently in use. For example, Visible Systems' Visible Analyst integrates "local context" into its data flow diagrams by showing related incoming and outgoing data flows. We believe the fisheye view is a significant improvement over this type of view (as well as traditional views) because it allows the designer to see a subprocess in the context of the entire system, while still emphasizing the subprocess by enlarging the scale for the subprocess in detail. Therefore, development of configurable fisheye view algorithms and their implementation and integration in existing tools is a major opportunity for developers of system tools.

From a managerial perspective, there are two important potential benefits of using fisheye views. First, the use of fisheye views can increase the effectiveness of system design due to the ability to recognize and eliminate redundancy and create effective linkages between subsystems. These views also facilitate greater efficiency in system enhancements due to the quicker navigation of the system model and identification of interrelated components. These values will be realized for systems that are complex. In addition, the subsystems of a particular project should be interrelated, making it necessary to see an individual portion along with its context. Finally, the system should be hierarchically organized, so that it can be modeled using a hierarchical diagramming technique.

For organizations interested in assessing the value of these tools, there are several options. The most straightforward and traditionally effective way of determining the relative value of this visualization method is a controlled experiment (for example, two development groups would complete the same development project using different tools). However, such studies are ultimately impractical in a corporate setting. An alternative would be to have the development group work on a project with the new tool, and compare the time to complete the project to an estimate of the amount it would take using existing tools (based on historical data). This, coupled with anecdotal feedback from the development team, can provide significant insight into the benefits of using the technique on future development projects.

Considering both the inherent difficulty in conducting empirical tests in an organization and the power of such tests, some academic research opportunities emerge. Comprehensive academic studies incorporating controlled experiments would provide insights for practitioners who would potentially benefit from our proposed solutions and would help establish a foundation for future studies. One major

challenge in conducting such empirical research is identifying experimental tasks that are realistic enough in mimicking the system development tasks performed by practitioners. Another major challenge is to find good measures of task success. We believe addressing these challenges, in addition to developing solid designs that incorporate the visualization ideas we have described in this article, will lead to the successful integration of context-based views in system development tools. **C**

REFERENCES

1. Egan, D.E. et al. Formative design-evaluation of SuperBook. *ACM Transactions on Information Systems* 7, 1 (1989), 30–57.
2. Furnas, G.W. Generalized fisheye views. In *Proceedings of the Conference on Human Factors in Computing Systems*, (Boston, MA, 1986), 16–23.
3. Gottschalk, P. Strategic information systems planning: The implementation challenge. In *Proceedings of the American Conference on Information Systems*, (Indianapolis, IN, 1997).
4. Gutwin, C. and Greenberg, S. *Interactive Fisheye Views for Groupware*. Department of Computer Science, University of Calgary, Calgary, Alberta, 1997.
5. Leung, K.Y. and Apperly, M.D. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction* 1, 2 (1994), 126–160.
6. Roussinov, D. and Chen, H. Information navigation on the Web by clustering and summarizing query results. *Information Processing and Management* 37, 6 (2001), 789–816.
7. Sarkar, M. and Brown, M.H. Graphical fisheye views of graphs. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 92)*, (Monterey, CA, 1992), 83–91.
8. Schaffer, D. et al. Navigating hierarchically clustered networks through fisheye and full-zoom methods. *ACM Transactions on Computer-Human Interaction* 13, 2 (1996), 162–188.
9. Teory, T.J., Wei, G., Bolton, D.L. and Koenig, J.A. ER Modeling clustering as an aid for user communication and documentation in database design. *Commun. ACM* 32, 8 (Aug. 1989), 975–987.
10. Turetken, O. and Sharda, R. Development of a fisheye-based information search processing aid (FISPA) for managing information overload in the Web environment. *Decision Support Systems* 37, 3 (2004), 415–434.

OZGUR TURETKEN (Ozgur.Turetken@temple.edu) is an assistant professor of MIS at the Fox School of Business and Management at Temple University in Philadelphia, PA.

DAVID SCHUFF (David.Schuff@temple.edu) is an assistant professor of MIS at the Fox School of Business and Management at Temple University in Philadelphia, PA.

RAMESH SHARDA (sharda@okstate.edu) is a Regents Professor, Conoco Phillips Chair of Management Science and Information Systems, and the director of the Institute for Research in Information Systems at Oklahoma State University in Stillwater, OK.

TERENCE T. OW (terence.ow@marquette.edu) is an assistant professor of MIS at the College of Business Administration at Marquette University in Milwaukee, WI.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.