The Life Cycle of a Visualization Case Study

Can you think of a domain application where you haven’t seen a visualization technique used? You might think only a few such applications exist because scientific visualization has benefited applications from many disciplines. This is evident at the IEEE Visualization conferences, where each year we learn about many new visualization techniques and applications. On the other hand, why have many application areas still not adopted scientific visualization techniques for data analysis? While new visualization techniques disseminate quickly through publications and professional meetings, applying one to a new application takes some effort. We believe that it doesn’t take magic to develop techniques for new applications. It just takes commitment from the domain scientists and the visualization specialists.

In general, two possible pathways lead to developing a new case study:

1. applying a known technique to a new application with requirements similar to well-developed applications and
2. developing a new technique from scratch for a new application with unique characteristics.

We want to look at the steps of the latter path. Specifically, we’ll discuss the life cycle of a visualization case study, which has a general pattern, although the actual steps can be unique for each specific case study.

Life cycle of a case study

Our experience has taught us that the life cycle of a case study application generally consists of the following stages:

1. Problem description
2. Data preprocessing
3. Solution formulation
4. Implementation
5. Testing, evaluation, validation

Every case study has its special requirements. The life cycle described here is a general one. Stages 3, 4, and 5 are cyclic and may take several iterations before the results are satisfactory. We now discuss each of these stages in depth.

1. Problem description

The domain scientist brings a data analysis problem to a visualization specialist (developer). For example, “Here’s what I have. Could you show me how to visualize my data?” Sometimes, a developer may approach the scientist and demonstrate possible visualization techniques. The developer wants to find out if the visualization technique would be useful to the scientist. Often, it’s not straightforward for the developer to understand the physics of the problem without the necessary background in the discipline. Here the scientist plays an important role in giving a clear description of the problem. It’s also the developer’s responsibility to do some homework in that discipline. The developer needs to understand the terminology and the technical aspects.

2. Data preprocessing

Next the developer must understand the data format. What’s the current format? How are data stored? The data may be in some physical form rather than stored online. Would the developer need to write a translator to convert the data? Does the developer need to define a data format? These are just a few questions addressed during this stage. This stage of the case study can be challenging and time consuming because usually the data isn’t in a form suitable for the developer.

3. Solution formulation

Now that the developer has a good understanding of the data format and knows how to access it, the developer must work with the scientist to reach agreement on the technique for visualizing the data. The scientist may not always find it easy to decide what’s best for the application, making this one of the most critical stages of the case study—it requires good communication between the developer and the scientist. Both need to give their best effort to make the case study successful.

The developer must suggest the type of visualization techniques and algorithms most useful to the case study. If a new visualization technique is needed, then the developer should create the new algorithm relying on

David L. Kao
NASA Ames Research Center

Kwan-Liu Ma
University of California at Davis
past experience and expertise in visualization. Always interactive, this stage of the case study may occur over several meetings. Often, new visualization techniques are developed during this stage because of the application’s special requirements. At the same time, this stage can be the most rewarding part of the case study because many great ideas could result from fruitful discussions between the scientist and the developer.

4. Implementation
Once the previous stage has produced an acceptable data format and the participants have formulated an approach to solving the problem, the developer can start to implement the algorithms. The developer should keep the scientist informed of progress so that the scientist may point out any potential problems early in the implementation.

5. Testing, evaluation, validation
The ideal person to test the new visualization technique is the scientist who would also evaluate and validate the visualization results. This may be as easy as simply looking at the output image or as tedious as trying out all sorts of test cases with the new algorithms. The algorithms should be validated with at least several test data sets.

What makes a successful case study? Is it measured by how widely its application community accepts it? Is it determined by how quickly it helps scientists find the relevant information in the data? Or is it the lessons learned from the case study? We believe that all these factors measure a successful case study. One of the most important goals is to gain insights obtained by using the tools/techniques developed.

Four case studies
This special issue of IEEE Computer Graphics and Applications features the four best case studies presented at Visualization 99. Each case study introduces variations into the life cycle’s common path.

Free Flight testbed system
Travel, whether for business or pleasure, has become part of many people’s lifestyles. Azuma et al. have developed a testbed system with a suite of visualization tools for a Free Flight environment, which gives pilots the flexibility to change their flight routes in real time. Free Flight would reduce the costs and increase the capacity of the existing air traffic control system. The authors propose a testbed system for constructing and evaluating conflict detection, resolutions, and visualization tools. The authors believe that such tools are a critical part of making Free Flight a reality. The results described here represent the second iteration of this testbed around the case study life cycle.

Situational awareness in military operations
A military training center had a large amount of tactical data in digital form and a very large amount of digital geo-spatial data. To improve training of commanders in military situations, they sought a visualization that showed military units in context with terrain, maps, and imagery. Feibush et al. proposed several visualization techniques for a situational awareness application. They discuss how to visualize very large amounts of terrain and tactical data for rapid understanding and to avoid information overload. The techniques were implemented in a system called the Joint Operations Visualization Environment (JOVE), which is being developed to assist in top-level military decision making.

The typical “playbox”—the area of interest—covers one million square miles with a variety of military units to be displayed in the air, on land, and at sea. Conventional polygon-based terrain modeling would have exceeded the capacity of current display systems. To overcome this limitation, the authors texture mapped very large images onto polygons using either clipmaps or a paging scheme that requires only a small amount of texture memory. Their spatial navigation model makes it easy for users to explore the geo-spatial features and the tactical environment. They also describe their information representation techniques for effective presentation of military units.

Not all user data come in a ready-to-use form. In this case study, Feibush et al. had to convert the user data to a form that would take advantage of the graphics hardware in the data preprocessing stage. They also had to observe several military exercises to see existing visualizations and determine what wasn’t being visualized in the solution formatting stage of the life cycle. They assembled and discussed prototypes with the users. Also in this stage they acquired domain knowledge on representing the attributes of military units with symbols and combining maps with terrain.

A challenge for Feibush and his development team arose when the users requested the visualization on lower performance platforms, while they are working with large amounts of geo-spatial and tactical data. Hence, in the implementation stage they chose to use an image-paging scheme and a Web browser for collaboration between the command center and remote users in the field.

User feedback and evaluation were a critical part of JOVE’s success. Feibush and his team understood the importance of this stage of the life cycle:

We operated the visualization system at several exercises and incorporated the users’ feedback into subsequent versions. Some exercises required the system to be operated only by military personnel, so we had to make the system reliable and easy to use. The exercise control staff used JOVE to resolve conflicts about the location of airplanes and submarines. The playback facility allowed participants to review significant events after the action occurred.

Visualizing sparse gridded data
Meteorologists at the Naval Postgraduate School in Monterey asked Djurcilov and Pang to look into ways to visualize outputs from a new generation of radars called Nexrad, a weather radar producing 3D output whose results are currently only visualized in 2D. While preprocessing the radar data for visualization, Djurcilov...
and Pang were surprised to learn that the data set is nearly empty in comparison to the number of points it could have if all the grid-points were filled. Furthermore, it seemed like a waste of possibly valuable information not to use the third (vertical) dimension, which could lead to discovery of more complex patterns otherwise not noticeable in 2D.

Data acquisition systems often generate gaps in the output data. Djurcilov and Pang proposed solutions to deal with missing gridded data in visualization. Typical approaches either replace the missing values with an out-of-range number and proceed with standard visualization methods, or fill in the gap with data from some interpolation algorithm. Though both approaches seem straightforward, the first may produce misleading or wrong visualization, and the second may hide the fact that the data set now includes a large region of interpolated data.

A more effective approach, proposed by Djurcilov and Pang, is to adapt and modify visualization techniques to handle large gaps of missing data. Specifically, they proposed modifications to contour lines, where solid contours denote valid data and dashed contours denote interpolated or missing data. Similarly, for volume rendered images they add speckles where missing data is encountered. They also proposed two postprocessing methods to smooth isosurfaces with missing data.

Sometimes, the solution formulating stage yields no single well-defined solution, and it’s only by trying different approaches that a good solution arises. This situation occurred in this case study:

At this stage we didn’t know what to expect, so we decided to try out a variety of techniques and make our decisions along the way. Each visualization method taught us something new about configuration, quality, and distribution of the data.

**Volume data mining**

Many techniques address volume rendering. Fujishiro et al. extended the Reeb graph to a hyper Reeb graph (HRG), to capture the topological skeleton of volumetric data. Their overall objective was to provide automatic setting of parameter values for volume data mining. Based on the HRG results, they constructed multiple semitransparent isosurfaces and decomposed the volume into nonoverlapping interval volumes. They also proposed two principles for designing transfer functions to accentuate the topological change in the volume data.

Fujishiro et al. demonstrated the effectiveness of their approach on a hydrogen ion-atom collision problem data set. By using their proposed volume data mining approach, they could identify the collision without going through the entire $10^4$ time steps of $61^3$ volume data. Furthermore, they obtained a better understanding of the inner structure of distorted electron density distribution.

In this case study, Fujishiro et al. assumed that the data was to be described in a volumetric form and that adequate data sets were available. Hence, their case study spans stages 3 to 5 of the life cycle. Furthermore, because the acquired knowledge about the topological structure of volume data sets was represented in HRG form, both the scientists and developers could easily select the proper parameters for visualizing the volumetric data. This reduces the number of iterations of the life cycle stages.

Fujishiro and his collaborators sincerely believe that their case study is quite successful:

With the new term “volume data mining,” we would like to say that the proposed method could enrich the traditional volume visualization techniques, to provide the scientists and developers with the “serendipity” for visual exploration of large-scale volume data sets. It is just like a benevolent wizard living in the mountains who can [point] with his magic cane to let the village people know the first position to dig, where an unknown fountain never fails to appear.

**Conclusion**

These case studies demonstrate the iterative nature of the scientific visualization process. They also show the need to adapt existing techniques to new applications. Designing a visualization solution/system therefore requires taking the iterative and adaptive processes into consideration.

David Kao works in the Data Analysis group of the Numerical Aerospace Simulations (NAS) Systems Division at NASA Ames Research Center. His research interests include numerical flow visualization, scientific visualization, computer graphics, and NASA’s Earth Observing System (EOS) data visualization.

Kao is an adjunct faculty member in the Computer Engineering Department at Santa Clara University and is a research advisor for the National Research Council. He served as a case studies co-chair for the IEEE Visualization conferences in 1999 and 2000. He received his PhD in computer science from Arizona State University in 1991.

Contact Kao at Nasa Ames Research Center, M/S T27A-2, Moffett Field, CA 94035-1000, davidkao@nas.nasa.gov.

Kwan-Liu Ma is an associate professor of computer science at the University of California, Davis, where he teaches and conducts research in the areas of computer graphics and scientific visualization. His career research goal is to improve the overall experience and performance of data visualization through more effective user interface designs, interaction techniques, and high-performance computing.

Ma received his PhD from the University of Utah in 1993. He served as co-chair for the 1997 IEEE Symposium on Parallel Rendering, the IEEE Visualization Conference Case Studies in 1998 and 1999, and the first NSF/DOE Workshop on Large Data Visualization.

Contact Ma by e-mail at ma@cs.ucdavis.edu.