Process Visualization with Levels of Detail

Krešimir Matković, Helwig Hauser,
VRVis Research Center
in Vienna, Austria,
`http://www.VRVis.at/`
{ Matkovic, Hauser } @ VRVis.at

Reinhard Sainitzer,
Imagination
Computer Services
Vienna, Austria
sainitzer@imagination.at

and M. Eduard Gröller
Institute of Computer
Graphics and Algorithms
TU Vienna, Austria
groeller@cg.tuwien.ac.at

Abstract

In this paper we demonstrate how we applied information visualization techniques to process monitoring. Virtual instruments are enhanced using history encoding *– instruments are capable of displaying current value and the value from the near past.* Multi-instruments *are capable of displaying several data sources simultaneously.* Levels of detail *for virtual instruments are introduced where the screen area is inversely proportional to the information amount displayed. Furthermore the monitoring system is enhanced by using* 3D anchoring *– attachment of instruments to positions on a 3D model –,* collision avoidance *– a physically based spring model prevents instruments from overlapping –, and* focus+context rendering *– giving the user a possibility to examine particular instruments in detail without loosing the context information. Two applications were developed, a prototype application and a commercial process monitoring tool.*

Keywords: process monitoring, process visualization, information visualization, levels of detail, focus+context visualization, virtual instruments.

1. Introduction

Information visualization (InfoViz) has emerged as an important field of research in the past few years. Process visualization as one of its sub-fields is directly related, but nevertheless much older. Processes have been visualized and monitored using traditional, analog instruments since the early days of industrialization. Analog instruments were followed by the use of digital instruments, and finally, computers are increasingly used to visualize and monitor processes.

There are a lot of software tools available, offering a comfortable and intuitive way for process visualization. Examples of widely used instrument libraries and tools are products by Global Majic Software [5] and Quinn-Curtis Inc. [9]. A wide-spread monitoring tool also is LabVIEW System from National Instruments [8]. There are hundreds of instruments available within this software. Another offerer in this field is GE Fanuc Automation with the DataViews system [3] and a large base of virtual instruments as well.

Most of these tools and libraries try to mimic conventional instruments, without taking advantage of well-known InfoViz methodology. Virtual instruments are much more flexible than conventional settings with real instruments. The virtual instruments are easier customized, but essentially they are still just an electronic version of the real instruments.

In this paper, we describe how process visualization can benefit from information visualization. We have improved features of singular instruments, as well as global monitoring features, and finally we developed two applications which illustrate the methods described in this paper.

Features added to virtual instruments are: history encoding, multi-instruments, and levels of detail (LOD). Including history encoding allows to display the current value and the values from the near past simultaneously. We achieve this by visually augmenting the common gauge and bar instrument. The historic values are displayed using less saturated colors and smaller needles or bars. In this way history values can be distinguished from the current value on a first sight, but still give an additional information about sensor behavior in the near past.

Multi-instruments display more than one data source simultaneously, making it easier to compare them. A multi-gauge, for example, has several needles, each of them in a different color and length. Values corresponding to the needles can be easily compared. In fault-tolerant systems often several sensors measure the same phenomenon. In this case multi-instruments are well-suited in emphasizing a malfunctioning sensor.

Levels of detail is an interesting choice in structuring the concurrent display of many instruments. Various types of instruments of different sizes are suitable for a LOD approach. If, e.g., four redundant sensors should be visualized, an instrument which covers a small area and which displays only the collective state of all sensors (ok/not-ok) can be considered as the first (and roughest) LOD. The next level could be an instrument which displays the average value of all the four sensors. Such an instrument covers more area on the screen, but gives more information as well. The following level could be a multi-instrument, displaying all four values, and finally the highest LOD would be represented by four single instruments. The highest level occupies the largest area, giving the most information at the same time.

Besides augmenting the instruments themselves, the global monitoring system was improved as well. Global features added are focus+context rendering [6] and collision avoidance.

Focus+context (F+C) approaches make a huge information space manageable for a user by coarsely representing the entire information in the available screen space. The user is enabled to display selected parts (focus) enlarged but embedded into the surrounding environment (context). In the example described above, it is possible to choose the level of detail of an instrument depending on the degree of interest. The degree of interest can be determined either explicitly by the user – the user points to the focus –, or it can be data-driven. Here the degree of interest might increase whenever a specific event occurs, e.g., a data value leaves the admissible range.

Increasing the level of detail, increases the required display area of the instrument, and it potentially occurs that some of the instruments overlap after a level change. In order to avoid instrument overlap, a collision avoidance algorithm based on a physically based spring model was implemented.

The above described features will be explained in more detail using gauge, bar, LED, and numeric instruments. The bar instrument is an instrument where the length of a solid bar corresponds to the measured value. It's analog counterpart is a mercury thermometer, or a tube manometer. The LED instrument mimics a single LED and can represent two states in our case. The numeric instrument is an instrument that displays measured data on a numeric display. Data can be displayed as a decimal, hexadecimal or binary number.

All of the new features are implemented in two applications. A placement prototype application [10] and a commercial TTPView 3.0 application [14].

2. New Features for Virtual Instruments

The idea of using virtual instead of classical instruments is not new. However, most of the virtual instruments try to mimic real ones, and offer a faster, more comfortable and more flexible way of arranging a new set of measuring instruments. We improve virtual instruments by adding some new features and techniques which are well-known from information visualization. We not only improve the instrument management but also augment the virtual instruments as compared to their real counterparts. We made virtual instruments capable of displaying the history of data (history encoding). Our virtual instruments are also able to simultaneously display several data values within one instrument, so that divergences in redundant information are easily recognizable (multi-instruments). Finally, we introduce a LOD representation for virtual instruments.

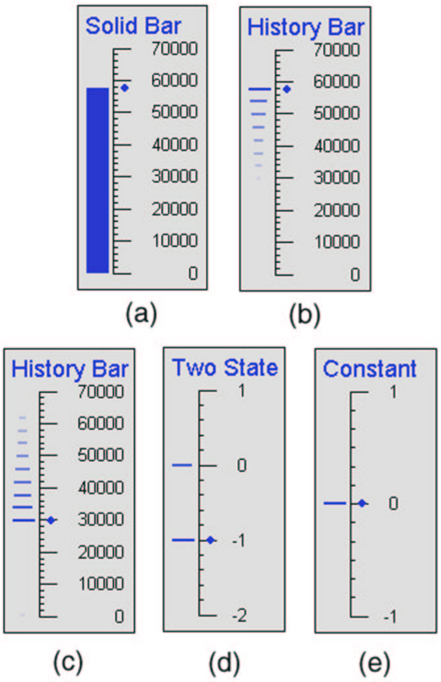


Figure 1. Solid bar (a), history bar with increasing (b) and decreasing (c) value, two state (d) and constant value (e) history bar

2.1. History Encoding in Virtual Instruments

It is often important not only to read the current value of a sensor, but also to get a clue what was the value in the near past. Of course it is possible to use some kind of oscilloscope instrument or time chart, but their size is often a limiting factor. We added history encoding to the bar and gauge instrument. Such an augmented instrument takes no more place than a common bar/gauge instrument, but gives far more information to the user. In the case of the bar instrument, history is encoded through horizontal lines on the left of the numerical scale. Length and saturation of a line encode the ”recentness” of a value, i.e., longer and more

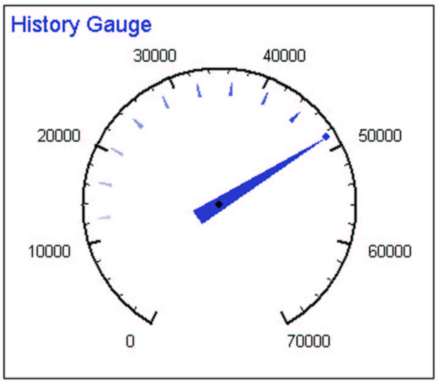


Figure 2. History gauge depicting a uniformly increasing value

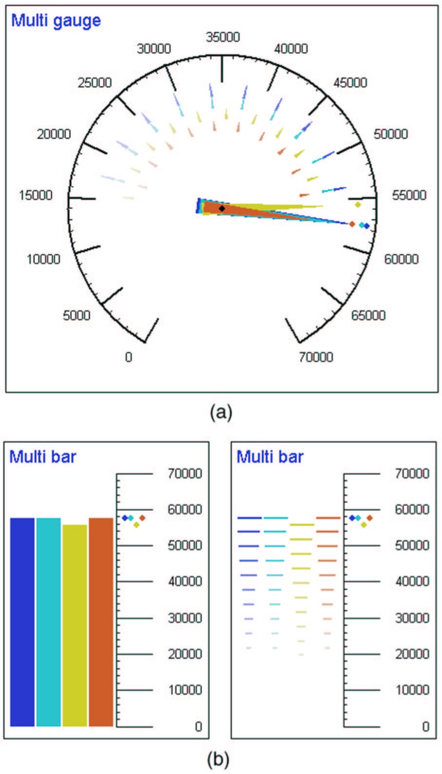
saturated lines correspond to newer data values. Figure 1(a) illustrates a bar instrument without history encoding. Just the current value is depicted, redundantelly through a bullet on the numerical scale as well as a solid bar to the left of the scale. In figures 1(b)–1(e) data history is depicted. Figure 1(b) and (c) show an over time monotonically increasing, respectively decreasing data value. Figure 1(d) shows a quantity that is switching between two values and figure 1(e) shows a quantity that is constant over time. A simple solid bar instrument would not be able to capture these vastly different temporal evolutions.

The same idea has been applied to the gauge instrument. The history needles in the gauge instrument are reduced to the needle tips in order to make the whole instrument more clear. Figure 2 illustrates the history gauge instrument for a uniformly increasing data value.

History encoding also relates to the visualization of gradient information in the data, similar as, for example, often used in scientific visualization (compare to gradients in volume visualization, for example). By showing not only the immediate values, but also a notion of temporal derivatives, the user gains more information about the sensor data.

2.2. Multi-Instruments

Common practice in fault tolerant systems, e.g., planes, locomotives, modern cars, etc., is to install several sensors for the same data. Important system decisions must not be based on a single measurement alone. In a faultless situation all sensors deliver the same data. If a sensor is out of order, its measurements should be neglected, and the maintainer of the system should get the information that the sensor is not working properly. If a separate instrument is used for each sensor, it is very hard to see if there are any deviations in the measurements, unless one value is significantly different from the others.



(a)

(b)

Figure 3. Multi-gauge and multi-bar

In order to make it easier to visualize such cases, we introduce multi-instruments. The multi-bar and the multi-gauge instrument, display several distinct data sources simultaneously, but using the same scale. The multi-bar instrument uses more bars beside each other, and the multi-gauge instrument uses more needles. The needles have different length, and shorter needles are placed on top of longer needles, so that the user immediately sees all the values. If there is a deviation in the values, one (or more) not matching needle(s) will be easily spotted. Figure 3 illustrates a multi-bar and a multi-gauge instrument. The user easily notices that a sensor represented by the yellow bar/needle obviously malfunctions. The multi-instrument approach has its drawbacks as well. Although the needles in the multi-gauge instrument are color-coded and have different lengths, it is not so easy to perceive separate needles when they overlap. Several, single instruments are certainly easier to read but require much more space.

Multi-instruments can also be used whenever it is necessary to visualize related data which does not have to be identical. An example is the visualization of a distribution of a data value using three needles. One needle displays the average, and other two needles the standard deviation of the distribution.

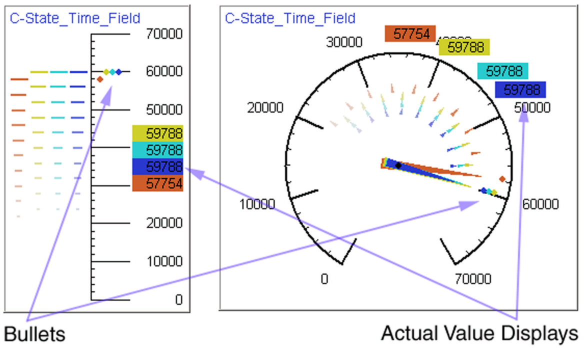


Figure 4. Bullets and actual value displays in the multi-bar and multi-gauge instruments

In order to make it easier to read the instruments, two more features are added. First a bullet on the scale indicates the current value. This is especially useful for multi-instruments where the numerical scale is not next to all bars/needles. Furthermore, an actual value display is added as well. An actual value display shows the current value as decimal number in a numeric display. The position of the actual value display changes and closely follows the current data value. The display is positioned using a running average making it easier to read the numerical value in case of fast changes or oscillating values. Furthermore, actual value displays never overlap, which makes them easier to read as well. Figure 4 shows these two features.

2.3. Levels of Detail

Multi-instruments have not only the advantage of making it easier to compare related values, but they save screen space as well. A multi-bar instruments with four bars occupies a far smaller area than four single bar instruments. This led us to the next new feature in process visualization, i.e., levels of detail. An example is the following. Four single bar instruments are the finest level of detail, the multi-bar instrument representing the same four values is the next lower level, a numeric instrument which displays the average value and a LED instrument are further levels of detail. Figure 5 illustrates the example, and gives the area usage for each level (highest level = 100%). In this way, it is possible to use the LED instrument if everything is running OK, and than automatically switch to a higher level if some data does not fulfill predefined conditions. There are other possibilities how the levels can be switched, and some of them will be described in the following section.

3. F+C Process Visualization

The LOD idea itself leads to a F+C process visualization. The focus+context approach is a well-known technique in information visualization. The surrounding environment, i.e., the context, makes this approach different from simple zooming-in, where an area is displayed in another scale but the connection to the environment is lost.

The F+C principle is partially based on the fact that humans can best perceive a circular area with a limited diameter in the center of their view [2]. The important information should be placed in this focus, but the information outside the ”most interesting” area, is not unimportant. This peripheral area (context) helps us to orient, and brings surrounding information for the instruments in focus.

According to Kosara et al. [7] the F+C methods can be divided into three groups. The first, and widest spread group of F+C methods are distortion-oriented or spatial methods. The idea is to distort the display to allow a magnification of interesting areas without losing the context which will be displayed on a coarser scale. A lot of metaphors relating this kind of F+C techniques to real world objects have been introduced like fisheye views [11, 4], stretched rubber sheets [12], etc.

The second group of methods are dimensional methods, which are suitable for objects with a lot of data associated with them. Objects in focus will be displayed using more dimensions of the data. An example of such a method is the magic lens [13].

Finally, the last group of F+C methods is called cue methods. These approaches allow the user to select objects in terms of their features, and not their spatial relations. An example is the semantic depth of field approach [7] which blurs different parts of the scene in dependence of their relevance. In this way the context information is still present, but the important information is in depicted sharp, and can be easily recognized by the user.

Our idea is to use the LOD principle introduced in the previous section for our F+C approach. The degree of interest corresponds to the level of detail of the instrument. The larger the degree of interest is, the higher level of detail is used. This can be considered, according to the above introduced categorization as a dimensional F+C method. Increasing the level of detail, however, increases the size of an instrument, and instrument overlapping possibly would occur. Some kind of rearrangement of the instruments should be done in this case (assuming there is enough free space on the screen) which makes our method a spatial method too. Therefore, our approach represents a combination of a dimensional and a spatial F+C method.

3.1. 3D Anchoring and Interaction

While process visualization usually is done in 2D, the data often comes from a 3D environment. Of course it would

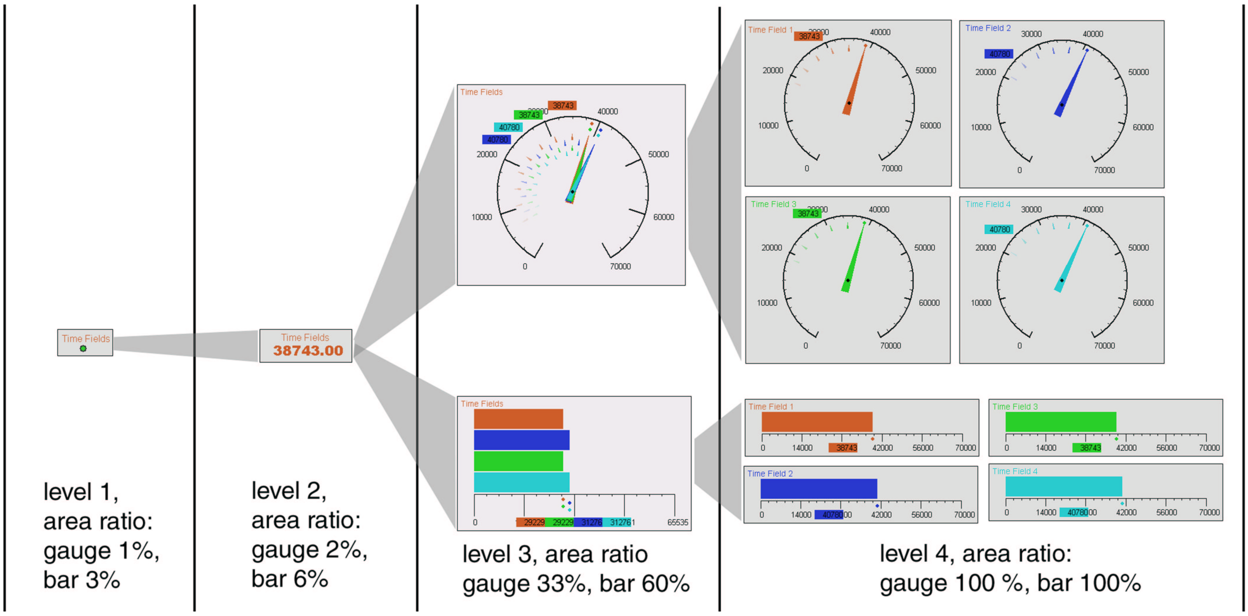


Figure 5. Various LODs, building up a tree of instruments, with corresponding area sizes

be possible to use 3D visualization with its advantages like similarity to real-world situations and flexibility which is offered by the third dimension. On the other hand 3D visualization has some drawbacks as well. The computing power needed for 3D visualization is significantly higher than the computing power needed for 2D, the objects are often occluded by other objects, and there should be some navigation mechanism in order to visually access all objects. The fact that common output devices (monitors) and input devices (mouse) are actually 2D devices, makes manipulating and using 2D visualization more intuitive for a typical user.

Virtual instruments represent real measuring sensors, that are placed in a 3D environment, and send data to the system. The user positions instruments on the screen, specifying the instruments current and desired position (which can but must not be the same). Since the real sensors exist in a 3D environment, e.g., a racing car, virtual instruments could be attached to the 3D model of the real environment and then mapped to the screen. This attachment of an instrument to a 3D position of a 3D model is called anchoring. If data that is going to be visualized comes from the wheel sensors of a car, the corresponding virtual instruments can be anchored to the wheels on the 3D model. In this way, the user can immediately recognize which data is represented by which instrument. The user can for example rotate the 3D model, and the assigned instruments will follow the movement.

The instruments are not simply placed in the 3D world and then mapped, because they would be easily unreadable after applying projection. Instead, after projection the instruments are placed at the positions of anchoring points (projected now). In this way the instruments are always visible, and easily readable. Figure 6 shows an example of four instruments assigned to the four wheels of a schematic car model. The anchored instruments follow changes of the underlying 3D model.

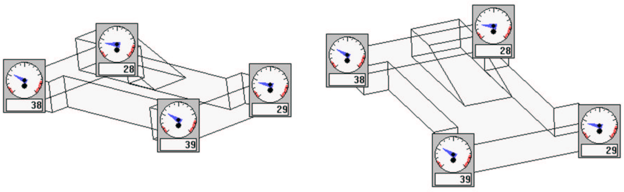


Figure 6. 3D anchoring: virtual instruments follow the 3D model

3.2. Collision Avoidance

In principle, anchored instruments, which follow a 3D model, often would overlap. It is possible to let the user solve the overlapping problem by arranging the instruments himself. It would however be far more comfortable if the system accomplishes this task.

In this case, it is necessary to develop a collision avoidance system which will prevent the instruments from over-

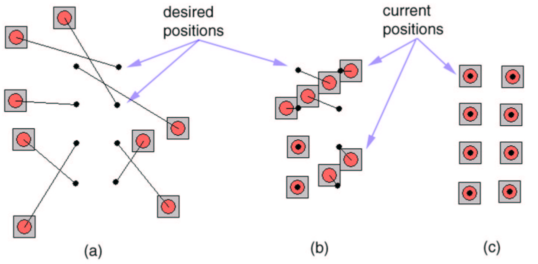


Figure 7. Collision avoidance of virtual instruments: (a) start configuration, (b) rearrangement is going on, and (c) final solution

lapping. The algorithm implemented [10] uses a physically based spring model [1].

A system of objects and springs is defined, such that there is a spring between each instrument's current and its desired position. The desired position is the projected anchor point. The tension of a spring is proportional to the distance between the instrument's current and desired position. Furthermore, a minimal allowed distance between two instruments is defined. If two instruments come closer than the minimal allowed distance, repelling forces are added to the system. This keeps instruments apart from each other, and avoids overlap. As soon as the instruments are far enough from each other the repelling forces are not acting any more.

In our prototype application the user determines the initial position by placing instruments and specifying their ideal positions (anchor points on the 3D model). When all instruments are placed, the spring model tries to assume a minimum-tension state by incrementally moving the instruments in the spring force directions. The process is repeated until a stable state is found, or until a user-defined time elapses. During the rearrangement, the stiffness of springs successively increases. This simulates annealing to avoid oscillations between two stable solutions (instruments jumping between different positions). The process of automatic placement is illustrated in figure 7.

3.3. Focus+Context Rendering

The LOD principle and automatic instrument placement are used for focus+context rendering. The idea is to use different levels of detail for different degrees of interest. The instruments with the highest degree of interest will be represented using the highest LOD, and the level of detail will decrease with decreasing degree of interest. After a level changes it is sometimes necessary to rearrange the instruments. Automatic collision avoidance described in section 3.2 is used in such a case. Figure 8 illustrates the change of levels, and rearrangement of the instruments.

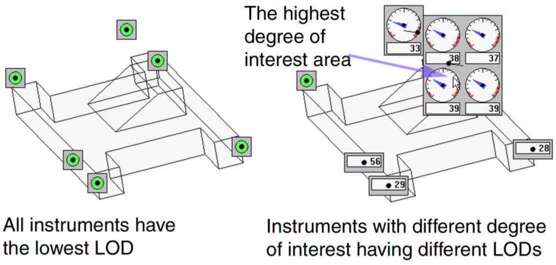


Figure 8. Changing the degree of interest

An instrument's degree of interest can be increased either explicitly or implicitly. By explicitly changing the degree of interest the user selects the instruments that are more interesting. This can be done in two ways.

One way is to equip the user with a sort of magic lens [13] which can be coupled to the cursor. In this way the area around the cursor is considered to have a high degree of interest, and the degree decreases with the distance from the cursor.

Another way, could be to define a fixed area on the screen which represents the region with the highest degree of interest. This can be the area around the center of the screen, or area around the golden cut point, or any other user selected area. The closer the instrument is to the center of the area, the higher level of detail is used to represent the instrument. In this way the user can examine a particular instrument by bringing it into a predefined screen area.

Implicitly changing the degree of interest is coupled with predefined, undesired system states. If an unwanted state occurs, e.g., a value exceeds a predefined range, the instrument's degree of interest is increased automatically, to notify the user and get his attention.

4. Application and Evaluation

The principles described in this paper are implemented in a placement prototype [10] and in a commercial application TTPView 3.0 [14]. The placement prototype realizes focus+context visualization as well as collision avoidance as described in section 3. The TTPView 3.0 application is far more complex. It is a commercial product which is intended to be used on a daily basis.

TTPView is a high-speed bus monitoring tool for the TTP system. TTP stands for Time Triggered Protocol, which represents one of two available technologies in the field of real-time control systems. One is event-triggered, and the second one is time-triggered. The TTPView 3.0 is easily connected to the system, and gets all information about existing data sources from the system. The data sources are logically organized in trees, and the user can

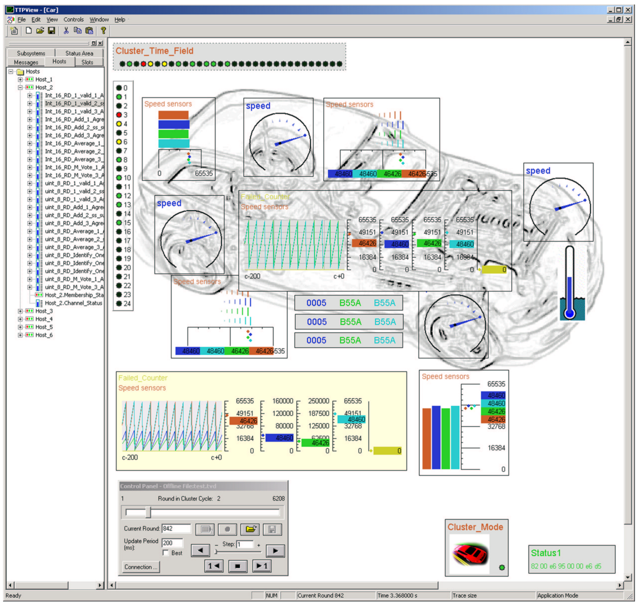


Figure 9. A screen shot of TTPView 3.0

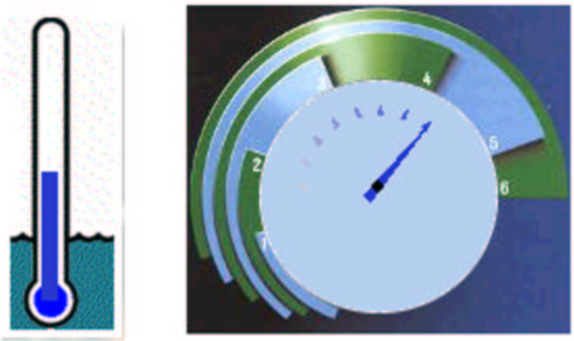


Figure 10. The bar instrument with a thermometer background and the gauge instrument showing RPMs

easily drag-and-drop wanted items into a view. Each tree item has a default instrument assigned to it, but the user can switch to another virtual instrument. All available instruments can be configured by: adding a caption string (description of data source) and unit string (physical unit of the data like RPM, MPH,..), adding an instrument background image, switching the scales and actual value displays on or off, changing size and color etc. It is easy for the user to configure the views according to his or her needs. The user can create several views as well. The views themselves may have a background image and a user-selected background color. Figure 9 shows some of these features.

The feature to add an instrument background has turned out to be very useful. Using cleverly designed background images, it is often immediately obvious what type of data the instruments display. The bar instrument in figure 10 shows temperature and the gauge instrument in the same figure shows revolutions per minute (RPM). Both data types are obvious from the design of the background image of the instrument. Note that the instruments in figure 10 are the same bar and gauge instruments as shown throughout the paper, but with scales switched off, and instrument background images added.

TTPView supports the following instruments: bar, gauge, trace, led, numerical, icon, and text instrument. The bar, gauge, LED, and numerical instruments were described in previous sections. Here we just briefly describe the other instruments.

The trace instrument displays the temporal evolution of a data value as a polyline in a time vs. data-value plot. The trace instrument supports more data sources, which makes it a multi-trace instrument. It has a separate scale for each data source (note that multi-bar and multi-gauge instruments have only one scale). This can be useful to study correlations between different sources, e.g., correlation of speed, RPM, and engine temperature.

The icon and text instruments are designed to visualize discrete sets of values. An icon or a string is assigned to individual intervals of the input data. An example could be to assign traffic-light icons with green, yellow and red light switched on according to a three state input.

The user can define a range of valid values for all instruments. If a value exceeds the admissible range a warning sign is added to the instrument. A message describing when the error occurred, and what was wrong is printed on the screen.

5. Conclusion

This paper presents several novel approaches to process visualization. We did not only try to mimic real instruments, but added features to the virtual instruments that can not be realized using their real counterparts. Furthermore we have enhanced the whole process visualization system, and we have implemented two applications.

The new features added to the virtual instruments can be summarized as:

History encoding, which makes it possible to display the values from the near past together with the current value. This allows to see the evolution of the value without increasing the display size of the instrument.

Multi-instruments are capable of displaying several values simultaneously. A multi-instrument makes it easier to compare redundant values, and saves screen space compared to single instruments which display the same data.

The levels-of-detail approach makes it possible to use instruments of different sizes to represent the same data. Increasing the level of detail will increase the amount of information shown, but will take up more screen space.

The advanced features in instruments made it possible to improve the global monitoring system by applying:

The focus+context principle, which allows the user to explore particular instruments in more detail (using higher levels of detail) without loosing the overall context. The method implemented is a combination of a dimensional and a spatial F+C method. The degree of interest can be explicitly set by the user, or can be data driven, increasing the degree if some special, predefined data state occurs.

3D Anchoring was used since our data originates from a 3D environment. Instruments are anchored on the 3D model, but special attention is taken when instruments are projected to 2D screen space in order to make them always visible.

Collision Avoidance was implemented using a physically based spring model. This was necessary to avoid instrument overlapping. Overlapping can be avoided only if the total instrument area is not larger than the total screen area.

The methods described in this paper were implemented in two applications: a placement prototype with F+C visualization and collision avoidance, and TTPView 3.0, as a commercial application, developed with the TTTech company [14], which is used on a daily basis.

This paper shows how InfoViz techniques can be applied to process monitoring and visualization. We did not try only to mimic the real instruments, but also to enhance them, and the whole monitoring system, as well. As a result, a system offering more information, using not much more screen space, and giving the user more flexibility is realized.

Acknowledgements

The authors thank Wolfgang Rieger for his inputs and his help in implementing the software, Herbert Buchegger for his help in implementation and Wolfgang Meyer for valuable discussions and ideas. Furthermore we would like to thank the TTTech company from Vienna (`http://www.tttech.com/`). TTTech helped us in understanding the needs in process visualization, and the TTPView 3.0 was made for TTTech. Parts of this work were carried out in the scope of applied and basic research at the VRVis Research Center in Vienna (`http://www.VRVis.at/`), Austria, which is funded by an Austrian governmental research program called K plus.

References

[1] D. Baraff and A. Witkin. Phisically Based Modeling: Principles and Practice. In *Siggraph '99 Course Notes*, 1999.

[2] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization Using Vision to Think*. Morgan Kaufmann Publishers, Inc., 1999.

[3] Data Views by GE Fanuc Automation, http://www.dvcorp.com.

[4] G. W. Furnas. Generalized Fisheye Views. In *Proceedings of the ACM CHI '86 Conference on Human Factors in Computing Systems*, pages 16–23. Association for Computer Machinery, 1986.

[5] IAL by Global Majic Software, http://www.globalmajic.com.

[6] T. A. Keahey. The Generalized Detail-In-Context Problem. In *Proceedings of the 1998 IEEE Symposium on Information Visualization (InfoVis 1998)*. IEEE Computer Society Press, 1998.

[7] R. Kosara, S. Miksch, and H. Hauser. Semantic Depth of Field. In *Proceedings of the 2001 IEEE Symposium on Information Visualization (InfoVis 2001)*, pages 97–104. IEEE Computer Society Press, 2001.

[8] LabView by National Instruments, http://www.natinst.com/labview.

[9] Real-Time Graphics Tools by Quinn-Curtis, http://www.quinn-curtis.com.

[10] W. Rieger. Neue Strategien für Interaktive Processvisualisierung. Master's thesis, Vienna University of Technology, 2002.

[11] M. Sarkar and M. H. Brown. Graphical Fisheye Views. *Communications of the ACM*, 37(12):73–83, Dec. 1994.

[12] M. Sarkar, S. S. Snibbe, O. J. Tversky, and S. P. Reiss. Stretching the Rubber Sheet: A Methaphor for Visualizing Large Layouts on Small Screens. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 81–91, 1993.

[13] M. C. Stone, K. Fishkin, and E. A. Bier. The Movable Filter as a User Interface Tool. In *Proceedings of the ACM CHI '94 Conference on Human Factors in Computing Systems*, pages 306–312. Association for Computer Machinery, 1994.

[14] TTTech Computertechnik AG, http://www.tttech.com.