Designing Usable Charts for Complex Work Settings

Connor Upton & Gavin Doherty
Trinity College Dublin
{connor.upton@cs.tcd.ie, gavin.doherty@cs.tcd.ie}

Abstract

Advances in graphing applications, plug-ins and toolkits means that integrating charts and graphs into software is easier than ever before. However, selecting the optimal graphing technique for a workers task remains a difficult challenge. Information visualisation experts draw on research from cognitive engineering, perceptual psychology and human computer interaction when designing displays. For the increasing number of developers who are integrating visual displays into applications, there is for a lack of a general methodology that pulls together key activities from these diverse fields. In the absence of such a methodology, it is very difficult for software developers to identify if their choice of representation satisfies both the user's tasks and perceptual limitations. We describe the approach taken in the redesign of an interactive chart used in a High Volume Manufacturing environment. We show how analyses of the work domain, the data and the users' tasks are all crucial steps in the design process.

1. Introduction

Advances in sensor and communication technologies have lead to a data explosion in industrial environments. Information Visualisation provides us with a means for accessing and understanding this data. Graphic representations of data can improve user performance for a range of cognitive tasks [1] . As a result, Information Visualisation is becoming an integral part of control systems in a wide range of industries.

In parallel with this, many data processing applications are starting to provide advanced graphing capabilities. MS Excel 2003 offers 14 basic chart types, each with multiple subtypes. These can be customized to produce relatively sophisticated interactive graphs, often through a wizard interface. Charting toolkits have made it easier for developers to integrate dynamic graphs into their software. As a result information visualisation is no longer the domain of a small group of experts. These technologies are popularising information visualisation in the same way that desktop publishing popularised graphic design. Unfortunately, while toolkits can help automate some of the technical aspects of graphing, the knowledge that is required to support usable visualisation design is not so easily transferable.

1.1. Establishing Methodologies

As a design practice, information visualisation is relatively young. Unlike architectural, industrial or graphic design, it lacks clear methodologies that can guide a practitioner towards a successful solution. At the same time, it is very much a design activity in the truest sense of the word. It involves real world problems that need to be solved through visual solutions. Currently information visualisation falls between being an art and a science. It is often seen as a craft carried out by multidisciplinary experts who draw on past experience and domain knowledge. This has lead to a strong focus on new interface solutions rather than the design process that led to their creation.

This poses a problem for developers who need to incorporate interactive charts into software. Their issue is how to select visualisation techniques that will best support the tasks posed by their particular work domain. Without a general methodology to draw on, they have no way of knowing that their choice is appropriate or even whether an appropriate solution exists?

It has been shown, that novice graphic designers working through a computer are more likely to take the path of least resistance through the design process rather than following established design methodologies (sketching, reviewing, prototyping etc.) [2] . This results in a compromised design solution. In order to avoid similar mistakes when designing interactive charts, it is important that we establish methodologies and processes that capture the full range of requirements for information visualisation design. It is also critical that we present these in a practical and accessible manner so that they can be used by developers with varying levels of information visualisation expertise.

1.2. Case study: Semiconductor Manufacturing

In this paper we present the methodology taken in the redesign of an interactive chart used in a High Volume (Semiconductor) Manufacturing environment. We show how three activities; Work Domain Analysis, Task Analysis and Data Analysis, are required to inform the design rationale. Semiconductor manufacturing is an extremely complex process. Much of the software is developed in-house by programmers who have extensive knowledge of the system. As a result many of the charts are designed and developed by in-house teams. The complexity of the domain makes it difficult to bring in outside assistance such as visualisation expertise.

2. Work Domain Analysis

The Semiconductor Fabrication Plant (Fab) involves a highly complex process flow, hundreds of machines (known as tools) and hundreds of workers. As an outsider the complexity of the domain can be daunting. Cognitive Work Analysis [3] (CWA) is a framework for researching complex socio-technical systems such as these. A work domain model of the system is generated as the first output of a CWA. This is a design artifact that acts as an external model of the work domain. This model is not based on individual user tasks but rather describes the functional purpose of the overall system and the various constraints under which it operates. It uses an abstraction hierarchy, which combines a physical and functional decomposition of the system. Here we attempt to model a sub-domain of the fab which we describe as the engineering hierarchy.

2.1. The Engineering Hierarchy

Semiconductors are produced by laying down alternating layers of conductive, resistive and semi-conductive materials. Hundreds of different operations are involved in this process and these are carried out by specialised, high precision process *tools*. Tools that carry out the same operation make up a *toolset*. A group of toolsets that share the same general functional activity (i.e. etching, lithography etc.) make up a functional area. The fab consists of several functional areas. A process engineer is responsible for the health and performance of process tools. Generally, he/she will be responsible for a *module*, a subset of ten to fifteen tools in a toolset carrying out the same operation on the same product type. They do not carry out physical maintenance on tools but they set targets, monitor performance levels, and diagnose problems. Most of this activity is carried at remote workstations in an office environment rather than on the factory floor. The engineering hierarchy can be seen in the top half of figure 1.

2.2. Parameters and Control Limits

Parameters are factors within the tool that affect an operation (e.g. pressure, temperature). Semiconductor manufacturing now operates at the nanotechnology level; therefore even minor fluctuations in process parameters can have a deleterious effect on the product. For this reason it is desirable to keep them as stable as possible. Tools have a variety of *sensors* built into them that monitor critical parameters. On average a tool may have twenty to thirty parametric outputs. Each parameter will have a target indicating optimal performance. This target will be shared across a toolset. Some of these targets are stable; they are defined by engineering constraints. Others can change depending on a number of different factors. These targets are statistically derived and may be changed from time to time. A process engineer sets these targets and a series of control limits and assigns them to different monitoring *templates* that are then associated with a particular module.

2.3. Structure of the Work Domain

In previous work [4] we show that the Fab features some extremely complex relationships. Typically an abstraction hierarchy decomposes a domain from system to sub-system to component. However within the fab, components often belong to multiple subsystems and a straightforward hierarchy is not an accurate depiction of the structure. Here again we are faced with elements (the sensors) being the lowest level of granularity in two structures (the engineering hierarchy and the parameter structure). It is difficult to produce a clear model of this structure without multiple meaningless intersections. Inspired by Bertin's illustration of the "totemic operator" [1] we generated a 3D model that accurately described the relationships (fig. 1).

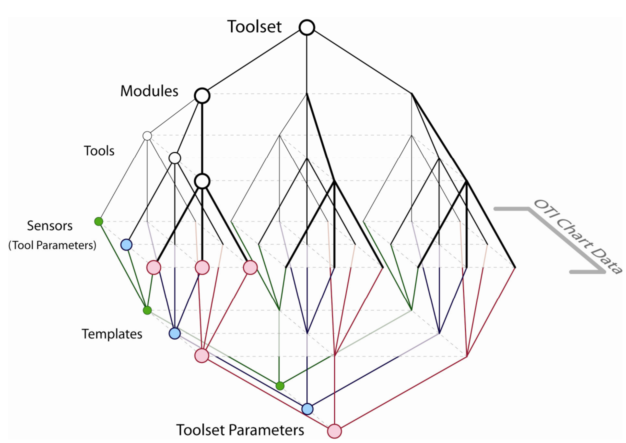

Figure 1 Work Domain Structure

This model provided us with an external representation of the work domain that was invaluable during task analysis interviews. Workers frequently referred to the model when discussing the range of their responsibilities and data requirements. This artifact was a design deliverable from the work domain analysis phase.

3. Task Analysis

A process engineer's goal is to maintain a high level of system health and performance. They carry out a range of different tasks to achieve this goal and a variety of decision support tools have been developed to help them. A series of interactive charts afford different perspectives on the data coming from tool parameters.

3.1. The OTI Chart

An On-Target Indicator (OTI) chart (fig. 2) allows engineers to monitor tools and ensure that they remain within control limits. While any sudden change in a tool parameter will result in the automated control system taking defensive action, a gradual drift from target may go undetected. Because of this, human intervention is required. A graphical representation not only allows engineers to deal with the high volumes of data; it also increases the potential for pattern recognition, whereby experienced engineers will start to recognize the signature of recurring problems based on the shape of the datasets.

Our task analysis was carried out in a number of stages. First using a "think aloud" verbal protocol we carried out a detailed walkthrough with a process engineer interacting with the chart. The user identified the main tasks and also mentioned some aspects of the design they found to be frustrating. The original charts were accompanied by a user manual that supplied us with a task list and a set of sub-task sequences. These backed up the information revealed by the walkthrough. Finally we carried out a number of semi-structured interviews with process engineers. Here we established the level of detail required to complete the tasks and gained a better understanding of the relationship between the work domain structures and the OTI data.

3.2. The Tasks

This chart aims to support situations where a fleet of tools may be drifting as well as detecting "dog" (erratic) tools. These are two very different perspectives on the data with the first taking a parameter based view of the system and the latter, an engineering (tool) based approach. Here we outline three major tasks that the engineer must carry out.

Task 1: Spot off-target tools parameters. Here the engineer must locate any tool parameters (sensors) that lie outside of the control limits set at one standard deviation. These tools need to be returned to an in-control state to avoid damage to product.

Task 2: Detect unmatched parameters. Here the engineer must look at the spread of a parameter reading across the tools. It is important that the tools perform in the same manner. If the sensors for a parameter have a wide range they are said to be unmatched. Unmatched parameters cause variance that may result in problems in upstream process steps or faulty product at end of line. Tools within a widely spread parameter need to be brought closer to target to avoid such problems.

Task 3: Find matched but off-target parameters. In certain situations an entire fleet of tools may be off-target for a parameter. There are two probable causes for this. An incorrect target may have been set in the monitor template, or a change in the product has had a knock on effect on the processing requirements. In either case the target for the process parameter needs to be checked and adjusted.

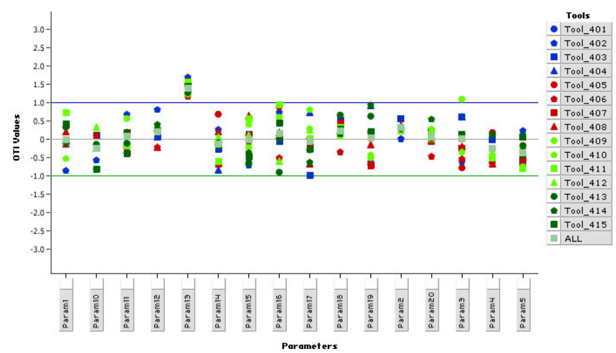

Figure 2 Original O.T.I. Chart Design

3.3. Task Analysis Review

During our analysis we noticed that the user guide explained the tasks in terms of the chart rather than in terms of the goals of the user. More importantly the users themselves explained their tasks in terms of what was achievable through the chart rather than their job needs. The wider work domain was not referred to until it was asked about in the interview. Had we not carried out the work domain analysis we might not have been aware of the overall structure in which these tasks were occurring. As we will see later this structure plays an important role in guiding the design.

The output from the analysis is the list of tasks identified above. These may seem very high level as, unlike Hierarchical Task Analysis [6] , the tasks have not been decomposed into sub-tasks and actions. Such detailed decomposition can only be achieved *after* the representation has been designed. We want to focus on the nature of the tasks before we consider how the data can be converted to visual form. By combining these with the outputs of the work domain and data analyses we can construct a design rationale matching the visual design to the users needs.

4. Data Analysis

Bertin's provides us with guidelines for converting data to visual form [5] . He proposes that data exists on three perceptual scales; Nominative, Ordinal, and Quantitative. He describes the visual variables, the basic elements of graphic composition, namely the spatial variables (position) and the retinal variables: size, value (tone), texture, orientation, shape and hue. These visual variables can be matched to the perceptual data scales. Zhang and Normans [7] demonstrate how data that is correctly matched to a visual variable will dramatically improve performance for certain cognitive tasks. They also note that the spatial variables are unique in that they provide the best support for all data scales. Here we outline the data types involved in the OTI chart and the visual variables that can be used to represent them.

Parameters and Tools: Both of these data types exist on the Nominative scale (category). They can be supported by all visual variables. However the most suitable are colour, shape and spatial position

Standard Deviation Values: This data exists on the Quantitative scale which is the highest perceptual scale and can only be supported through spatial position and size.

Additional Data: Looking at the OTI chart, it appears that the three datasets mentioned above are the only information required to support the tasks. However the control limits are used to judge when a parameter is off-target and their presence on the chart is essential. Another aspect is the structural relationships of the work domain which was revealed earlier. This is non-essential for spotting off-target tools so from a data analysis perspective its display may be considered unnecessary.

5. Current Chart Review

On the original OTI chart design (fig. 2) the quantitative standard deviation values are displayed on the y-axis. The nominative parameter data are displayed in the x-axis. The nominative tool data are encoded through icons by combining colour and shape variables. The original chart design is a version of a scatter plot. This graphing technique is incorporated into a wide range of spreadsheet applications and graphing toolkits. Engineers in the fab are very familiar with these applications and their charts. In the absence of a methodology that takes a wider set of analyses onboard, it would be entirely reasonable to use such a display technique that is familiar to both developers and users.

5.1. Satisfying the Tasks

On initial inspection the chart seems to satisfy the requirements. Task 1; selecting off-target tool parameters, is easy to achieve. The user scans the chart and selects the points that lie above or below the control limits. Task 2; detecting unmatched parameters can be achieved by visually estimating the vertical spread of the readings. Unmatched parameters can also be mathematically detected so further support is provided by colouring in their labels. Task 3; finding entire parameters that are matched but off-target can be estimated by the vertical position of a parameter cluster and can be further clarified by looking at the position of the mean value shown as the ALL icon.

5.2. Problems with the Original Design

Despite the fact that the basic tasks were achievable through the chart, users were having difficulty. Certain additional tasks not mentioned in the user manual were not encouraged by the design and as a result workarounds were used. For example the selection of a specific sensor in the graph based on tool & parameter reference is very difficult. These tasks were not deemed possible to achieve through the original design. During our interviews further problems were revealed. These are listed below.

Problem 1: Selecting icons. Clicking on an icon brings up a more detailed graph showing the specific sensor's (tool parameter) performance history. Users found it difficult to click on the icons as they were very small. Increasing the size would make it difficult to estimate its' y-position and worsen problem 2.

Problem 2: Occlusion of icons. Icons with the same or similar standard deviation readings tend to overlap making it difficult to click on any icon other than the foremost. This meant incorrect selections were frequent.

Problem 3: Detection of "ALL" icon. The ALL icon shows the mean reading for the parameter across all tools. Users had difficulty locating this icon as it generally lay at the centre of a cluster. This was particularly problematic when a parameter cluster lay close to a control limit.

Problem 4: Ability to locate a specific sensor. Sometimes an engineer obtains information about tools from external sources (co-workers, automated systems etc.). The current design makes it difficult to locate icons based on the tool reference rather than the value.

Problem 5: Ability to view performance across a tool. In a situation where an engineer identifies a "dog" tool, it can be useful to see how that tool is performing across other parameters. Again the current design makes it difficult to achieve this.

5.3. Problem Sources

The chart used a correct matching of data scales to variables so where did the design go wrong? The problem lies in the nature of the perceptual tasks being carried out and the added need for interaction. Selective perception is a process by which all instances of a category can be isolated and visually grouped into a single image. This is the activity required when looking for specific icons (the core problem in 5.2.3, 5.2.4 and 5.2.5 above). While shape does allow the encoding of nominative data it does not permit selective perception [8] . Interaction places an additional challenge on displays. Targets have a lower-limit to their area before they become unclickable. Any situation where targets are placed on a numeric scale runs into the risk of target occlusion and the loss of interaction.

During our interviews, we discovered another issue. In the work domain analysis we showed that engineers are responsible for modules of tools, however the OTI chart displayed parameter readings for entire toolsets. This raised the question of how engineers become aware of issues that might only be affecting their modules. On enquiry we revealed that many engineers generated graphs of their own modules for individual parameters.

6. Redesign of Chart

The original OTI chart accurately depicted the data collected by the sensors. The problem lies in the matching of the perceptual scales that the data exists on, and the perceptual scales required for the cognitive tasks. The important step of data transformation [9] has been missed. Data transformation is a process by which data values are transformed into derived values or derived structures. It can be carried out to make data easier to understand. We propose that the information required for a data transformation is revealed though both the task and work domain analyses.

6.1. Data Transformation

The standard deviation values are originally calculated as quantitative data. However, at no stage are quantitative cognitive activities such as addition, subtraction or multiplication required. In the first task the aim is to identify tools that lie outside the control limits. The second and third tasks relate to the spread of tools within parameters. This deals with the relative distance of tools from each other and the control limits. The perceptual scale for this activity is ordinal. By transforming the standard deviation values into a series of ordinal classes we reduce its complexity and permit the use of a wider range of visual variables. The display of ordinal data can be supported by the spatial variables, size and value (tone).

6.2. Visual Scale Selection

While the spatial variable has been shown to be perceptually dominant, we have chosen to use size to encode the Standard Deviation data (fig.3). The spatial variable had additional advantages; it is easy to indicate targets and control limits by drawing horizontal lines, dividing the area up into a series of sub-areas. We have chosen to encode these extra dimensions using multiple visual variables within the standard deviation icon. Distance from target has been encoded using size, plus or minus values have been encoded using two hues (red & blue) and icons outside of control limits are marked by a dramatic change in value (tone). What advantages does this icon have over the previous one?

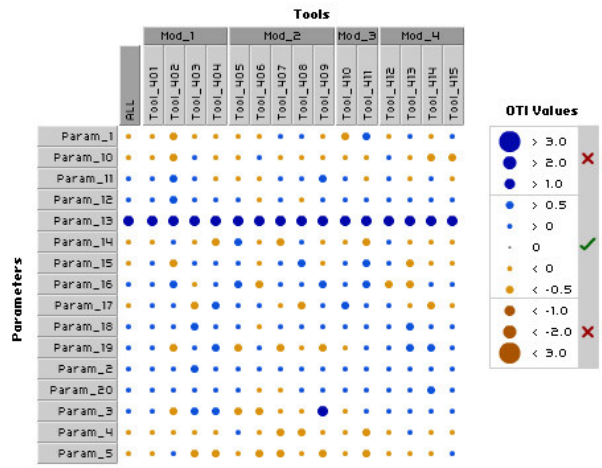

Figure 3: New O.T.I. Chart

6.3. Rationale for New Design

The original chart encoded the tools by combining two visual variables into a single icon. This lead to a redundant visual variable which could be mistaken for another categorization of the data (e.g. yellow icons are off-target?). In the new design (fig. 3) each variable encodes a different perceptual scale relating to the data, as outlined above. Size and Value are dissociative visual variables [8] (i.e. they are perceptually dominant over the other retinal variables). They are used here to indicate distance from target and tool state. Their combination will make the off-target readings highly salient causing them to pop-out of the display. They also present a much larger target making it easier to select individual sensors and drill into further detail. Most importantly, by creating an icon of the standard deviation value, we free up the spatial variable for the encoding of the tool data. This spatial encoding of the two nominative data types creates a matrix display of the dataset. This eliminates the occlusion of icons due to overlapping again improving the selection of icons. It also makes it possible to read the chart from the perspective of locating specific tools or sensors (tool parameters) thus eliminating the last three original problems with the chart.

6.4. Additional Advantages

The changes brought about by the data transformations allowed us to introduce a number of additional features not available in the original design.

Firstly, it is possible to organise the sequence of tools on the x-axis into modules. By introducing a second series of module labels we can embed the structure of the system, revealed by the work domain analysis, into the chart. This means that users can now focus on their own tools while maintaining a view of the wider toolset.

Secondly, the matrix allows us to detect sensors that are not giving out readings. In the original chart these sensors would generally go unnoticed as they may be present but occluded. Because of this, the new design gives the user a better understanding of the system state.

Thirdly, the new design makes a much more efficient use of space, making it suitable for display on small form factor devices. In the old design, the parameters needed to be well spaced out to allow for the reading of the sensor clusters. This sometimes led to the need for horizontal scrolling. Also, the design featured a lot of negative space in the areas outside of the control limits. In the new design each sensor takes up a maximum area of 16 pixels and even the empty spaces have a semantic meaning for the system state. This spatial efficiency could prove to be a critical advantage as there is a lot of interest in supplying decision support information on mobile devices [10] .

7. Evaluation

A second round of interviews was carried out at which the users were presented with a number of paper prototypes of the redesign. Initial feedback was very positive with users being able to identify situations in the dataset following a very brief explanation of the elements. There was enthusiasm for the appearance of modules in the display and the ability to see non-reporting sensors. We have prepared a series of experiments involving fully interactive versions of the old and new designs. The two designs will be evaluated against each other for a range of tasks and different datasets. Users will be measured for efficiency, accuracy and satisfaction on both displays.

8. Conclusions

In this paper we have looked at a real-world example of how interactive charts are being developed and applied within industry. In the absence of practical methodologies, developers are relying on familiar graphing techniques and/or application wizards for the production of charts and graphs. This approach can lead to applications incorporating visual displays that are not well suited to the user's tasks. We propose a three stage approach involving Work Domain, Data and Task analyses as the first steps in information visualisation design methodology. A work domain analysis can provide us with a structure that outlines relationships between data and allows us to chart ranges of responsibility. A data analysis reveals the variable types and the volume of data involved. A task analysis guides us in our data transformation allowing us to select visual variables at the appropriate perceptual scale. Using this approach we can ensure that the information visualisation techniques we use are suited to the cognitive tasks being carried out. In situations where existing techniques do not fully support our tasks, this approach can guide us in the generation of new visual interfaces that fit the user's needs.

Acknowledgements

This work has been funded through an Intel Ireland Research Scholarship. The authors would like to thank Paddy Holland, Gavin D'Arcy, Tom Quilty and Michael McGrath at Intel® Ireland, Leixlip, Co. Kildare, for their valuable assistance.

References

[1] Zhang, J. (1997). The nature of external representations in problem solving. Cognitive Science 21(2): 179-217

[2] Black, A. (1990). Visible planning on paper and on screen: The impact of working medium on decision-making by novice graphic designers. Behavior and Information Technology, 9, 283-296

[3] Vicente, K. J. (1999). Cognitive Work Analysis: toward safe, productive, and healthy computer-based work. Mahwah, NJ: Lawrence Erlbaum Associates

[4] Upton, C. & Doherty, G. (2005) Designing Usable Decision Support Systems for HVM, Proceedings of 10th IEEE International Conference on Emerging Technologies in Factory Automation, Lucia Lo Bello & Thilo Sauter (Eds.), Vol. 1., pp.459-466, IEEE, 2005

[5] Bertin, J. (1983) Semiology of Graphics. The University of Wisconsin Press, Madison. Transl. William J. Berg.

[6] Kirwan, B. & Ainsworth, L.K. (Eds.) (1992). A Guide to Task Analysis. London: Taylor and Francis.

[7] Zhang, J. and Norman, D. A. (1994) Representations in distributed cognitive tasks. Cognitive Science 18: 87- 122.

[8] Mullet, K. & Sano, D. (1995). Designing Visual Interfaces. Sunsoft Press (Prentice Hall).

[9] Card, S. K. Mackinlay, J. D. and Shneiderman, B. (1999). Readings in Information Visualization: Using Vision to Think. San Francisco: Morgan-Kaufmann

[10] Doherty, G & Upton C. (2005). Designing Displays for Mobile Decision Support, Volume 2 of Proceedings of 19th BCS Conference on Human Computer Interaction 2005, pp. 83-88, 2005.