IJAM International Journal Of Agile Manufacturing

Visualization of Cost Information

Christophe Bru, James Scanlan, Peter Hale
Faculty of CEMS, University of the West of England, Coldharbour Lane, Bristol BS16 1QY, ENGLAND

Abstract: In this paper, a prototype package developed within the scope of the Implied Cost Evaluation System (ICES) project at the University of the West of England will be presented. It offers visualization and analysis features for engineering cost data. These tools permit the cost engineer to browse and locate stages in the manufacturing process where requirements, such as time and cost to manufacture, are not met. Once a problematic entity has been identified, the product or parts of it should be redesigned. Any inappropriate manufacturing processes should be either improved or substituted for a more suitable one. Ease of use and efficient information delivery have been particularly emphasized.

1. Introduction

Cost estimation at an early design stage is a vital task for most manufacturers. Nowadays, this activity is considered as an integral part of the design process, and not only as an option. (Gayretli 1999a, b, OuYang 1997, Duverlie 1999), as well as (Rehman 1998) have mentioned that major cost savings can be achieved at the early design stage, when up to 80% of a product's cost is determined. Engineers started to be deeply concerned by manufacturing costs towards the end of Second World War, when large numbers of aircraft were required to be produced in a short amount of time, and at a low cost. This was related in the very interesting (Parametric Estimating Handbook 1999) written jointly with the Department Of Defense. Nowadays, engineering students are taught how to balance such factors as "manufacturability", process, material selection and resources for a product, so as to minimize production costs. The maximum allowable cost depends greatly on customer requirements (e.g. mechanical and physical properties, quality, weight, etc.). Thus engineers have to discern a compromise between quality, durability (in the case of life-cycle cost concerns), feasibility and low cost, in order to develop a balanced product. The product should be at the same time functional (i.e. meet all customers' requirements) and as cheap as possible (which usually is itself a customer requirement). If extra cost savings can be achieved, this will result in increased margins.

Increasing computer power makes it possible to undertake very complex analysis (Dodd 2001, Flood 2001). However, although computers are very efficient in calculations, they still lack intelligence. Consequently, they cannot make all the sensible inferences implied in a task such as cost estimation (e.g. choice of the cheapest manufacturing process for producing a given quality). A computer can be taught via algorithms how to make some educated inferences. However, these algorithms will not be generic, but specific to some particular situations. Therefore, cost estimation is not fully automated, and computers need user input and feedback.

To some extent, it could be argued that cost estimation accuracy is proportional to the volume of information fed in the computer, as mentioned in the (Parametric Estimating Handbook 1999). Conversely, a high volume of output information is difficult to analyze as a whole, or in the best case it will slow the process down significantly. This project was born from the initiative to graphically display cost information in an appropriate manner (i.e. efficient information delivery to users). The tool resulting from this research was aimed at giving users an overall view of the whole process plan for a product, at a glance. The main advantage of this software is that it provides the ability to focus on particular manufacturing processes. Therefore, it permits quick and thorough analysis of specific areas of interest (e.g. high cost, high uncertainty).

Although engineering cost estimation is a task that has been extensively surveyed, visualizing cost information in a clear graphical manner has not been meticulously implemented. Indeed, some cost information is obtained as an output, but no tools are offered to the cost engineer so as to analyze the data thoroughly, efficiently and quickly.

In this paper, visualization methods will be discussed as preliminary background. Then, prototype software developed within the scope of the Implied Cost Evaluation System (ICES) will be reviewed, and its data filtering and visualization capabilities will be presented. Finally, the work achieved to date will be summarized, and possible further work will be proposed in order to fill the remaining gaps.

2. Project Aims

The task was to produce a "Cost Map" that would allow the user to display cost information on a rich picture. The display would include manufacturing information in terms of cost, time, uncertainty and state of the product within the whole manufacturing sequence. This graphical map would hence provide an overview of the procedure that transforms raw materials into finished products. It also highlights possible problematic stages in the mass of information.

3. Visual Representation Background

Although graphical representations have been intensively exploited before the eighteenth century, exhaustive work on the subject was not conducted until William Playfair invented or enhanced most of the common graphical representation techniques (Tufte 1983, Wainer 1997).

Playfair's intensive work on the subject was motivated by the understanding that words are not always the best alternative to deliver information (Flood 2001). Graphical tools overcome any problems resulting from textual means by offering clear representations and cognitive comprehension (Dodd 2001, Tufte 1983, Wainer 1997). Following Playfair's work, graphics were subject to profound utilization during the nineteenth century (Flood 2001).

The twentieth century was beneficial to visual formats, and particularly in terms of tool enhancements. This was driven by the need to manipulate larger data volumes. Another important factor was the rapid introduction and widespread use of computers, especially at the beginning of the last decade (Flood 2001). Computers offer computational power that is suitable for filtering large databases, and thus they ensure efficient information delivery.

4. Defining Suitable Visualization Tools

Defining suitable visual formats is not a straightforward task, as it is difficult to say that one method is fundamentally beneficial compared to another one (Purchase 2000). But every alternative has got particular applications for which it is more effective. On the other hand, each option also has its flaws.

For instance, let's consider displaying on a graph the cost of operations within a Resin Transfer Moulding (RTM) process template. The use of a line chart (Figure 1) to accomplish this task is irrelevant because such a chart should be used to show evolution of an entity in time. On the other hand, a bar chart (Figure 2) displays the same information efficiently, as it allows comparison of similar properties for different entities.

This project makes use of several graphical representation tools that have been compiled in a prototype package so as to extend the range of facilities they provide.

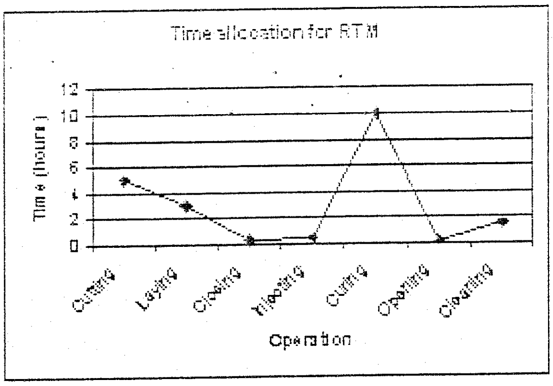

Figure 1. Line Chart.

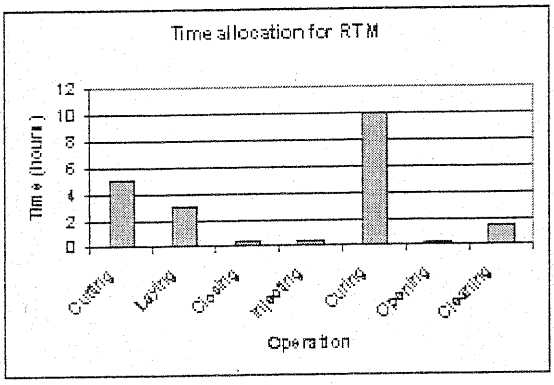

Figure 2. Bar Chart.

5. Overview of the COst INdicating EstimatoR (COINER) Cost Estimation Package

The Cost Map software was initiated by the need to visually represent information generated by an existing cost estimator package developed within the same project (Marsh 2000). This means that the cost estimation facilities are not included in the Cost Map. The existing cost estimation software, written in the Kappa language and called COINER, relies on the description of a product in the form of a bill of materials. The bill of materials consists of a collection of sub-components contained within the product. Sub-components are in turn described as collections of features. Finally, each feature contains slots (which are values, e.g. dimensions) and methods (which are equations, e.g. volume calculation). This hierarchical structure is displayed in the Concepts Browser, as illustrated in Figure 3. The Concepts Browser has a tree structure and does not reveal slots and methods contents.

It could be argued that generating a concept as detailed in Figure 3 cannot be achieved at early design stage. The strength of the COINER model is that although it is a generative cost estimation package (Scanlan 2000), it also includes a database containing historical information, such as product templates. Therefore, users do not necessarily need to describe a new product from scratch but can import an existing suitable template in order to quickly build a draft of a new product. Also, unknown data do not have to be fed in. COINER will either import a default value from historical data, or associate distributions to empty slots. The shape of the distribution will reflect the range of alternatives existing in the database, and passed to the Cost Map as an uncertainty value.

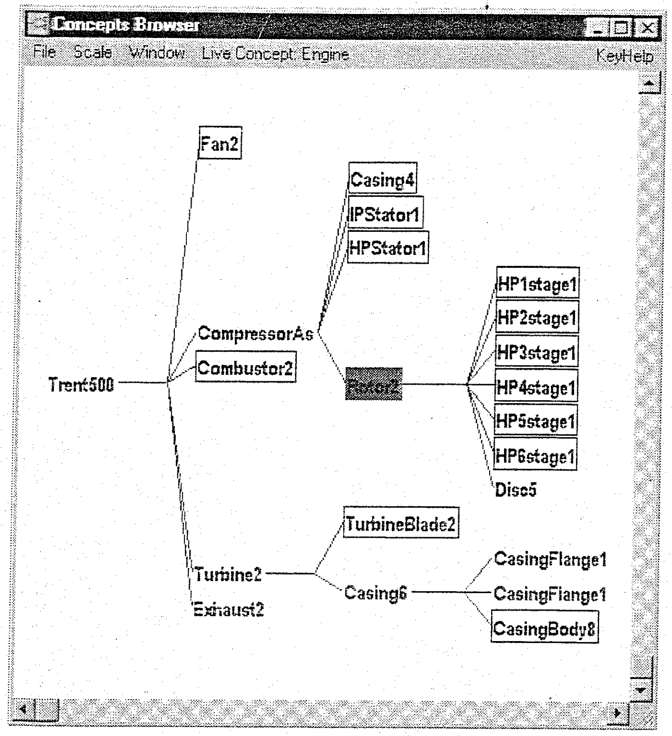

Figure 3. COINER Concepts Browser.

Once the user has described the product down to the features level, each of these features has to be associated with a material, resources, consumables, or any useful related information stored in the proprietary database. Resources (e.g. machinery) contain generic methods that can compute cost, time, and more globally whatever information they have been designed for. By recursively running down the bill of materials and aggregating manufacturing information defined by the associations, an exhaustive "manufacturing build sequence" is eventually obtained and published into a file for further analysis. Figure 4 shows the principles of how the cost aggregation is performed.

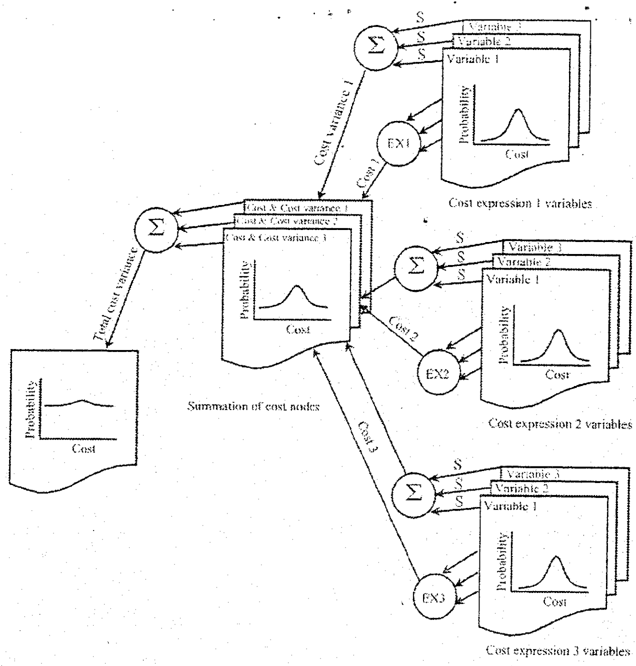

Figure 4. Cost and Uncertainty Aggregation.

6. Transferring Cost Data

Cost estimation packages often generate a large volume of information, and COINER is no exception. Some of this information is not required for the visual presentation that is to be achieved. By filtering and ignoring redundant data, the volume of information is drastically decreased, making future data mining tasks much easier (Shen 1999). In terms of software and resources, this results in increased efficiency (Westphal 1998). Depending on the focus of the data analysis, the information to import may vary slightly. Because data filtering depends on the goals we need to carry out with the filtered data, the filter can be easily customised in the source code so as to meet specific requirements.

Since the amount of data imported from COINER to the Cost Map software has been reduced to the minimum, it is possible to take advantage of the new data structure in order to store it using another format. The goal is to move away from a proprietary format towards a format readable on most computers, and a relational database was chosen because it meets these requirements.

7. Filtering Tool

Now that the data is available in a flexible format, the representation of this information can be considered. However, because of the volume of information, it might be sensible to proceed with some filtering first. Indeed, users will probably want to focus their attention on some specific data. So as to provide such possibilities, the user can manually tag any data item. Filtering capabilities are enhanced by the possibility of building automated customized filters, as shown in Figure 5.

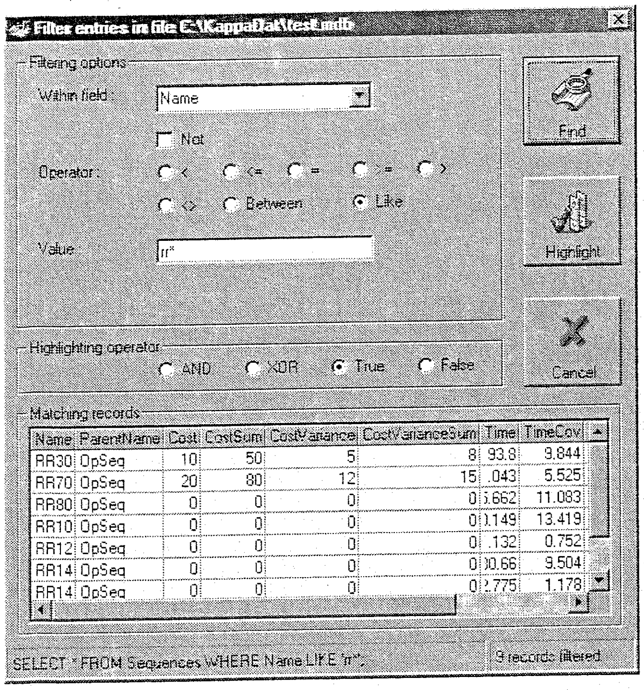

Figure 5. Filtering Tool.

In some respects, the concept is similar to the computer interview, which involves the program asking questions of the user (Peiris 2000). The filtered results can then be flagged into the main screen, which holds the database contents. Each visualisation tool will have a different way of stressing the flagged information within the entirety.

8. The Cost Map

The main visualization tool is the Cost Map itself. It displays a map of the product build sequence in the form of a tree. (Herman 1999) has presented such a tree visualization system dedicated to information visualization.

The specifications of a tree require it to present one single root. The root suitably symbolizes the finished product, which is a unique entity in the map, located at the end of the build sequence. It may have one or several parent branches, making up the sub-components of our product. However, the root cannot have a child branch. Conversely, leaves are nodes that do not have any parent, but have at least one child. These nodes are first stages in the manufacturing template, e.g. raw material or sub-contracted product reception in the workshop. Note that if contractors agree to provide cost information for their product, the Cost Map tree could be extended away from the tree root. This extra information should be added only if the corresponding products can be negotiated; else it would result in swelling the map with static data which cannot be influenced. Some nodes have more than one branch coming in. These nodes are assembly nodes, and therefore they need input from more than one sequence, i.e. as many as there are parts to assemble at the assembly stage. Finally, a node is the smallest entity comprised into the tree. The assembly build state of the product at one given stage implies that the previous operations have been successfully accomplished. The screen capture in Figure 6 helps in conveying the relevancy of the tree structure.

Nodes are used to illustrate sequences. Up to three nodes' physical attributes are available for the user to represent whichever data fields he/she is interested in focusing on. These attributes are size, colour and shape, and can be customized to display any quantitative data field contents (e.g. quality, lead time...).

Let us illustrate a typical scenario in which the user will use node size to represent time, colour for cost, and shape for time uncertainty, as shown in Figure 6. In this case, nodes size is proportional to manufacturing time. Because manufacturing time (or any other data field as a matter of fact) can vary drastically from one sequence to another, a logarithmic scale has been used. Thus, a sequence lasting 300 hours does not have a surface 300 times as large as a sequence lasting 1 hour, which would be cumbersome. The colour gradient reflects the manufacturing cost. The lighter the node is, the cheaper the sequence is. Hence, a light node corresponds to a sequence that is relatively cheaper than a dark one. Two different colour scales are employed. The red scale highlights all the nodes that were flagged using the filtering tool. The grey scale shows all nodes that are not flagged. If no nodes have been filtered, the whole Cost Map will appear in a grey gradient. Finally, the node shape reflects the time variance. The more edges the node has, the less accurate the time is.

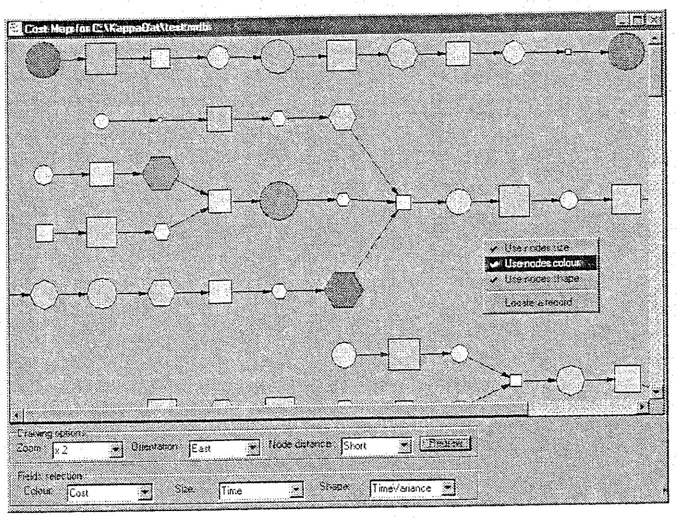

Figure 6. The Cost Map.

It is worth mentioning the fact that the time variance is the only dimension represented in a discrete manner. Because there are only four kinds of shapes employed in the Cost Map (circle, octagon, hexagon and square), the time variance range of values is split into four intervals. Therefore, nodes with the same shape belong to the same interval of values, but do not necessarily have the same exact value. Another consequence is that a square node shows a sequence that is among the most time accurate. This does not necessarily means that it is fundamentally accurate. This last remark is also valid for the colour gradients. A light node will not inevitably be cheap, but it will be cheap relatively to black ones. Thus, it is crucial to understand that shape and colour are reflecting values relatively to the overall minimum and maximum values. In the case of a box of matches, a dark node might indicate a product that costs only few pennies, whereas in an aero-engine, a light node might cost several hundreds of pounds. Similarly, if a design is very well defined, a round node might show an uncertainty on time of a few percent only, when at early design stage and for the same product, a round node would be much more inaccurate.

Once the data is displayed, the user is able to interpret the underlying data, and spot areas which need particular attention. In the previous scenario, a big black node on the Cost Map is expensive and time consuming. Without knowing the nature of the manufacturing process involved, the user knows as a fact that this sequence is a consequent cost driver. Furthermore, time uncertainty is also invaluable information because it indicates if the time figure has been computed accurately. If the large black node is a square, it has been defined accurately. In which case, chances are that this sequence can hardly be optimised. On the other hand, if the node is a circle, then it is uncertain. This is usually the result of poorly defined processes and/or product features. In which case, the user knows that it is necessary to gather additional information in order to increase the accuracy of product and/or processes definition. Once further information has been collected and implemented, the node will change in shape, dimension and colour, eventually allowing the user to determine if the design is acceptable or not.

Finally, the Cost Map, like other tools in the package, provides an interactive visualization as described by (Knight 2001). Not only is information visualized on the Cost Map, but the user can also interact with the nodes. Pointing the mouse to a node will display its name, while clicking the node will display local information (about this node only) as well as global information (such as minimum, maximum, average and variance for the whole map).

9. Benediktine Cell Walls

As stated earlier, one single tool is not likely to meet all our requirements. The second most useful tool is the Benediktine cell walls. The term "Benediktine Space" is coined from the name of its instigator, Michael Benedikt, who conducted research on the structure of Cyberspaces (Palfreyman, 1999). The Benediktine approach states that the attributes of an object can be mapped onto intrinsic and extrinsic spatial dimensions (Lombardoni 1999). In the present case, two extrinsic dimensions are to be selected by the user. Because it employs only two extrinsic dimensions, this visualization can be referred to as a "Benediktine wall". Benediktine Space can represent up to 3 extrinsic dimensions, whereas there is virtually no limit to the number of intrinsic dimensions that can be managed. However, large numbers of intrinsic dimensions requires additional effort from the reader who must concentrate more thoroughly on the graph so as to assimilate all the information (as discussed previously).
Figure 7 illustrates an example of Benediktine cell walls.

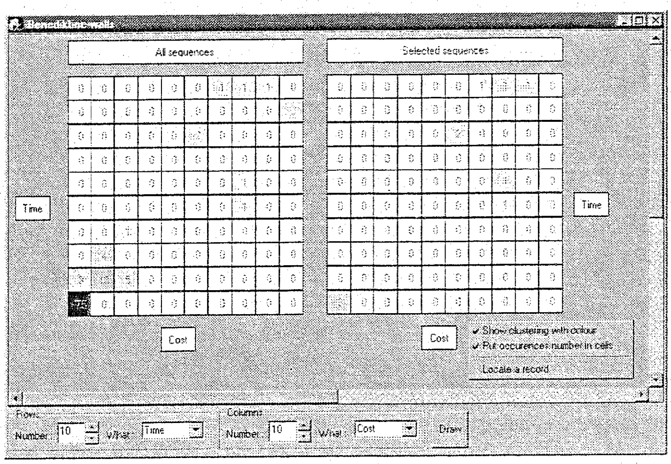

Figure 7. Benediktine Cell Walls.

To the user, Benediktine cell walls will look like matrices. Each cell in a matrix corresponds to an interval in the two dimensions displayed (e.g. time and cost in Figure 7). The lower left cell contains the lowest values in both dimensions, whereas the upper right cell contains the largest values. To some extent, it is similar to a scatter plot in this respect. However, unlike scatter plots, data overlapping is not an issue as for each cell its data density is characterized by the cell colour. The darker the cell is, the denser the data is in this cell. It is similar to the colour scales technique used in the Cost Map, especially since the right hand matrix displays flagged sequences in a red scale, whereas the left matrix displays all sequences in a grey scale. If the user is not comfortable with this technique, the number of occurrences within the cells can be displayed in the form of a number. Another similarity with the Cost Map is the interaction offered. Users can click a cell and have its contained sequences displayed. In turn, clicking a sequence will display its operations.

10. Additional Tools

Both Cost Map and Benediktine cell walls are probably the most useful tools included in the package. This is due to their ability to display the whole data set within a rather small display area, permitting the user to spot specific data of interest at a glance. The first downside of these tools is that data is not exhaustively displayed. Increasing the volume of information on the display inevitably incurs higher data density, leading to decreased readability. The second downside is that because these are graphical tools, exact values are not directly accessible. Again, numbers require some space to be displayed properly, which engenders high information density and poor readability.

Despite the low readability, the user will need to have the detailed information presented in a "text" format. That is why the first visualisation tool offered when a database is opened is a text based tree view accompanied by a grid. The tree can be considered as a very light version of the Cost Map which does not present all the graphical features. It still allows browsing the data, and because the tree structure is identical to the Cost Map one, one can combine both methods to locate certain sequences readily. The grid contains all data stored in the database. When a sequence is selected on either the tree view or the grid, it is simultaneously selected on the other one. The grid can be sorted by any field by clicking on the corresponding column header. For instance, sequences can be sorted by descending time order, clustering all the most time consuming sequences at the top.

Once some sequences have been flagged manually or via the filtering tool, a quantitative data field (e.g. time) can be displayed using a chart representation. Charts are a common technique that has become popularised because indeed it has proved to be efficient. The implementation of the charts makes use of 3D techniques, and it could be argued that simplifying their representation would deliver the information in a more efficient manner (Tufte 1983, Wainer 1997). However, (Tractinsky 2000) states that "what is beautiful is usable". Again, interaction is provided in order to bring forward sequence information when its corresponding chart entity (e.g. bar) is clicked.

Because the Cost Map and Benediktine cell walls approaches display a high volume of information, users may fail to discern specific information of interest. It is very likely that one wishes to locate a sequence in the mass of data. This can be achieved thanks to the implementation of a "locate" tool. It enables the user to highlight a particular sequence in the visualisation tool he/she is currently using.

Finally, any sequence can be displayed on a special kind of bar chart portrayed in Figure 8. There is one bar per quantitative field. The top and bottom of the bar correspond respectively to the upper and lower data field value. An empty red arrow and a filled black arrow alongside the bars indicate respectively selected sequence value and average value. This tool allows locating where the selected node stands within the mass, in terms of fields values. This also helps to circumvent the values relativity as in this graph exact values will be available instead of relative ones (i.e. a black round node in our example will be clearly determined in terms of cost and time uncertainty, making the user able to determine whether the round shape and dark colours are really an issue or just extreme values which are still acceptable).

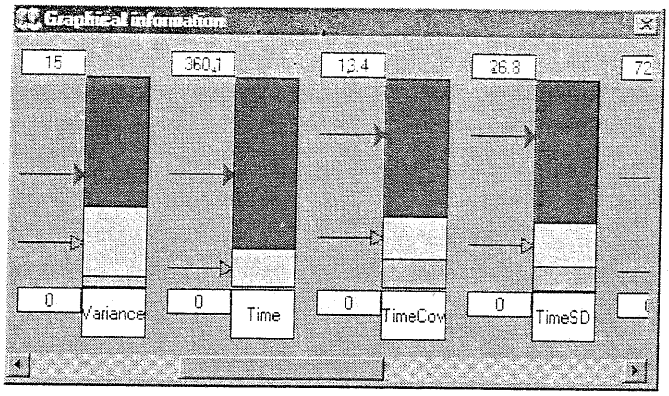

Figure 8. Graphical Information Chart.

11. Conclusions

The collection of tools presented in this paper should prove to be an invaluable package for cost engineers. It allows analysis of a large volume of information at a glance and permits locating and focusing on areas of interest. Because it relies heavily on visual perception, manufacturing sequences presenting potential problems can be spotted quickly. The graphical display does not require any background knowledge from the user in terms of manufacturing processes or product definition (although it is still preferable).

The Cost Map, populated with an aerospace product, has been demonstrated to industrial partners, and reported as interesting and useful. They also mentioned the possibility of using arrow length to reflect lead times between sequences.

However, because of its uncommon nature, the Benediktine walls method has not received the same positive feedback. It has been recognised that this tool does not provide the intuitive comprehension attributed to the Cost Map. This may be due to an interface presenting a novel situation. (Alty 2000) emphasises the need to support the transformation of existing knowledge so as to facilitate the learning of new computer systems. A new interaction approach should be considered and the method re-evaluated by potential future users. The application of (Agah 1999) research may also improve the user interface. Improvements are not limited to the Benediktine walls technique, and the other tools may also benefit from Alty's and Agah's work.

Overall, the Cost Map reached a finished state and handles any DAO (Data Access Object) compatible database. Thanks to basic structure requirements that can be customised easily in order to suit specific needs, and by implementing a comprehensive number of error trapping and tests, the software was able to display successfully several database contents from different sources.

The gap in this software is that, despite its' development in a flexible manner, it does not allow data sharing. This issue could be solved by linking users, tools and knowledge thanks to the Internet facilities (Hale 2002).

This also brings forward another flexibility issue. Because this software has been developed to work jointly with the COINER prototype, part of the same project, compatibility with commercial models has not be implemented. However, because of the way the Cost Map has been designed, it can accommodate any data in the form of a database. The only requirement is to translate information to this format, as it has been done with the COINER model.

When the Internet integration will take place, we will also take advantage of the opportunity to write a COINER cost model clone aimed for Internet integration. It should enable easy data generation, storage and sharing, and should impose a compatible data structure format. Some preliminary work has been done with XML (eXtensible Mark-up Language) files as a step forward in this direction. Because XML files are plain text files, they might make the task of converting data coming from external commercial or proprietary cost estimation packages easier, and thus make of the Cost Map an interesting alternative in any business.

References

1. Agah A., Tanie K. 2000. Intelligent graphical user interface design utilizing multiple fuzzy agents, Interacting With Computers, 12(5):529-542
2. Alty J.L., Knott R.P., Anderson B., Smyth M. 2000. A framework for engineering metaphor at the user interface, Interacting With Computers, 13(2):301-322
3. Davis S., Wiedenbeck S. 2001. The mediating effects of intrinsic motivation. Ease of use and usefulness perceptions on performance in first-time and subsequent computers users, Interacting With Computers, 13(5):549-580
4. Dodd J. 2001. Mapping the mind for knowledge, Information World Review, 174:27-28
5. Duverlie P., Castelain J.M. 1999. Cost estimation during design step: parametric versus case-based reasoning method, The international journal of advanced manufacturing technology, 15:895-906
6. Flood G. 2001. Visualising company data, Information World Review, 174:23-30
7. Gayretli A., Abdalla H.S. 1999a. A feature-based prototype system for the evaluation and optimisation of manufacturing processes, Computers & industrial engineering, 37:481-484
8. Gayretli A., Abdalla H.S. 1999b. An object-oriented constraint-based system for concurrent product development, Robotics and Computer-Integrated Manufacturing, 15:133-134
9. Hale P., Scanlan J., Bru C., Dunkley M. 2002 Preliminary Findings from the DECIDE project, 9th ISPE International Conference on Concurrent Engineering: Research And Applications, Cranfield University
10. Herman I., Melançon G., M. De Ruiter M., Delest M. 1999. Latour – A Tree Visualisation System, Lecture Notes in Computer Science, 1731:0392-
11. Knight C., Munro M. 2001. Visualisations; Functionality and Interaction, Lecture Notes in Computer Science, 2074:0470-
12. Lombardoni A. 1999. Benediktine approach, http://www.inf.ethz.ch/personal/lombardo/archives/da/node7.html, accessed on 8th of April 2002
13. Marsh R., Hill T., Scanlan J., Dunkley M., Cleevely P. 2000. Probabilistic Pseudo-generative Cost Modelling Through Virtual Template Propagation, CEAS Conference on Multidisciplinary Aircraft Design and Optimisation, 25-26 June 2001, Germany:Maternushaus Koln
14. Ou-Yang C., Lin T.S. 1997. Developing an integrated framework for feature-based early manufacturing cost estimation, The international journal of advanced manufacturing technology, 13:618-629
15. Palfreyman K. 1999. End of first year report, http://www.comp.lancs.ac.uk/computing/users/kev/computing/project/year1report/, accessed on 8th of April 2002
16. Peiris D.R., Gregor P. 2000. The effects of simulating human conversational style in a computer-based interview, Interacting With Computers, 12(6):635-650
17. Purchase H.C. 2000. Effective information visualisation: a study of graph drawing aesthetics and algorithms, Interacting With Computers, 13(2):147-162
18. Rehman S., Guenov M.D. 1998. A methodology for modelling manufacturing costs at conceptual design, Computers Ind. Engineering, 35(3-4):623-626
19. Scanlan J., Hill T., Marsh R., Bru C. 2000. Cost Modelling for Aircraft Design Optimisation, Engineering Design 2000 Conference, June 2000, Brunel University
20. Shen L., Shen H., Cheng L. 1999. New algorithms for efficient mining of association rules, Information Sciences, 118:251-268
21. Tractinsky N., Katz A.S., Ikar D. 2000. What is beautiful is usable, Interacting With Computers, 13(2):127-145
22. Tufte E.R. 1983. The visual display of quantitative information, Graphic Press, ISBN 0-9613921-0-x
23. Wainer H. 1997. Visual revelations, Copernicus, ISBN 0-387-94902-4
24. Westphal C., Blaxton T. 1998. Data Mining solutions. Methods and tools for solving real-world problems, Wiley Computer Publishing, ISBN 0-471
25. 25384-1999. Parametric Estimating Handbook, http://www.ispa-cost.org/PEIWeb/newbook.htm, accessed on 8th of April 2002