Published in The 2001 International Conference on Imaging Science, Systems, and Technology (CISST 2001)

June 25-28, 2001, Monte Carlo Resort & Casino, 3770 Las Vegas Blvd., South, Las Vegas, Nevada, USA.

Paper ID: 360CT
Paper Title: Circular Technology; Technological Management of Technology
Paper Author(s): Claire Knight and Malcolm Munro

Circular Technology; Technological Management of Technology

Claire Knight and Malcolm Munro
Visualisation Research Group
Research Institute in Software Evolution
Department of Computer Science
University of Durham
Durham, DH1 3LE, UK

Abstract *The use of computers, systems, and software is prevalent in many industries. The reliance on such electronic infrastructures should not be underestimated by managers or computer scientists. In order to provide the variety of services sold to customers a company must ensure that their systems provide such functionality in a time that is considered acceptable for customer service, and in the context of business faster than their competitors can. With the advent of component (and possibly distributed) systems this functionality is often created out of various compositions of base components. The provision of services through these systems then has to be managed and marketed from a "user" perspective, as well as internally in the form that the products and processes are understood by those who work with them daily. The provision of a visual management aid is the focus of this research. It utilises company data that already exists and then aims to display it in various forms to enable both an overview and detail where required to be visible at any one time. This already captured information can be integrated with system and software visualisations to provide a more complete large picture, and also at a lower level to offer a richer picture.*

Keywords: Visualisation, Comprehension, Knowledge Management, Software Systems

1 Introduction

The speed with which one captures a market today is critical in this current climate of fast-paced technologically driven innovation combined with a need to capture customers to whom loyalty doesn't mean what it once did. Part of this problem has been caused by the organisations themselves with their ever more inventive customer attraction promotions, but whatever the underlying reasons both system and organisational data are important in mapping out internal administration and supervision before integrating with any customer data and analysis.

The aim of this research is to empower the user of any system to enable them to find the information that is likely to be of most use. This follows the tenet of Intelligence Amplification (IA), whereby users are considered to be the powerful parts of the computer-human combination, and that the judgement and intuition comes from them. The computer provides the facilities for effectively making these decisions and interpreting the data [8, 3, 1, 2].

Data mining is a technique that is often considered to be of use in situations such as these, but this requires that the data be suitable for some statistical technique to be applied. This usually means that the data needs to be cleaned. A process whereby outliers in any dataset are considered to be anomalies that will distort the final result and are hence deleted. For the purposes of this analysis and visualisation, and from experience with the industrial dataset that motivated this research, such outliers can be very useful to managers and therefore techniques are necessary that leave them intact [6].

This paper will present a visualisation that attempts to provide this level of support for management through incorporating more than just software visualisation aspects. It will also emphasise where such visualisations can be enhanced to embody new (but related) knowledge that may be part of an organisational culture or exist in someone's own knowledge but is not part of any formal capture mechanism. Essentially it is about using technology to manage technology.

2 Background

Software visualisation grew from the information visualisation field. Software engineering and maintenance have long made use of node and arc graph style representations, but with the increase in graphical capabilities on standard machines, and the efforts made by those in fields such as information visualisation it was seen how different techniques could be incorporated. This led to many different ways of viewing source files and lines of code, but very little has been done in the way of adding *information value* to the visual displays. The beauty of visualisation techniques is the number of things that can be embodied at once to create a comprehensive view of the subject. The use of auxiliary sources of information is one that is used daily by many, but by providing linked information, and linked views, allows for a greater range of information to be analysed, and theoretically for more informed analysis and decisions [9, 4, 5].

Customisation of visualisation is also important. Being able to select the information of interest is vital to allow for many tasks to be accomplished with the same tool, especially if the range of supported data is large. Customisation is also related to usability; if a tool does not aid a user then they are likely to resort to using older less efficient methods. Hindering users, rather than just not being more effective, is even worse. Whilst any new tool and/or representation suffers in that it is not familiar to the user it is expected that once initial use has occurred that it will then become an effective part of their working process [7]. Customisation supports the IA principle of aiding rather than replacing users. It supports the personal choices of users and can therefore better support an individual in utilising the data presented. This supports personal preferences as well as the specific sub-tasks necessary for task completion. It is an often forgotten fact that there are two variables to consider in such situations.

3 Management Requirements

Obviously the normal management requirements of a system still exist; uptime, planned maintenance, ability to produce correct output given the right data, and so on. What has changed is the nature of the systems and therefore what can and is achieved with them; even from a business/user perspective. Distributed heterogeneous systems are more and more common. The provision of services rather than specific products is also more prevalent as organisations seek to give their customers what they think they want. The provision of such services may well originate amongst several systems within an organisation.

There is therefore a need to manage not only the individual systems but also the interactions between the systems, and the services which combinations of system functionalities provide. Regardless of the underlying system structures (monolithic C or COBOL, or distributed heterogeneous components, or combinations of both) these issues remain at a higher level. They may also exist within systems so visualisations (and other support tools) need to be capable of reflecting this interconnectivity and complexity.

4 Historical Events & Future Deadlines

There is often lethargy within the large organisations in responding to change, which smaller companies tend to address earlier – not least because their business will depend on it. A recent, admittedly over hyped, example is the case of the Year 2000. There was a need to change many existing systems, and what made this unique for many organisations was that it was very definitely a deadline that could not be moved to accommodate slippage further down the process. This meant that much planning had to go into the evolution of systems and anything upon which those systems depended. It is also the case that from the data collected during that process and at the time of the deadline, the management of similar future deadlines could be better dealt with based on learning from effects of decisions made then.

This process of using past events as a guide to future ones is not new, and indeed is applied very much in other fields such as economics. It is however, a phenomenon that has been under utilised in the context of systems and software.

It is true that new process and analysis methods have emerged from software engineering studies, and that there are management methods, but combining the two and utilising organisational knowledge (as an example of another very useful source) has been very much ignored. As organisations capture more and more information, much of which is left dormant in data warehouses, it provides just the impetus to extract this day-to-day information and utilise it in the context of managing systems. The combined sources can then be viewed over time in order to work with upcoming deadlines, and incorporate ideas and lessons learnt from previous ones.

5. Knowledge Capture

Knowledge capture, especially within the software engineering field, has long known to be a problematic issue. Programmers do not like process (where much documentation stems from), and therefore system knowledge is mostly contained within individual's heads. Even if documentation is generated, and there is traceability from requirements through to development any maintenance is often confined to the system code. Documentation is rarely updated, and even a comprehensive log of change requests/implementations may not be kept for many systems [2].

Annotation, and comments added during comprehension, can be useful for both the originating author and others who come to view a system. This information may not be entirely accurate, and usually will not be subject to formal review processes, but nevertheless it provides a valuable resource. Annotation of the underlying data without affecting that data can be achieved with appropriate visualisations. This then allows someone else's knowledge to be utilised to good effect by those less experienced, and also provides an informal way of capturing the sort of knowledge that may well be lost if an employee leaves a company. In the same way, being able to view the actual data items when in a visualisation allows for annotations to refer to the underlying data. This can then be used by others to augment their analysis and thus decision making surrounding the same data items.

6. Visualised Knowledge Management

This section presents visualisations that support the higher-level management of the systems and services in question. It does not deal with visualising the individual systems that are part of the whole organisational technology base as these can be of many forms. Indeed the design of this visualisation was to allow for just such a variety within the system level visualisation.

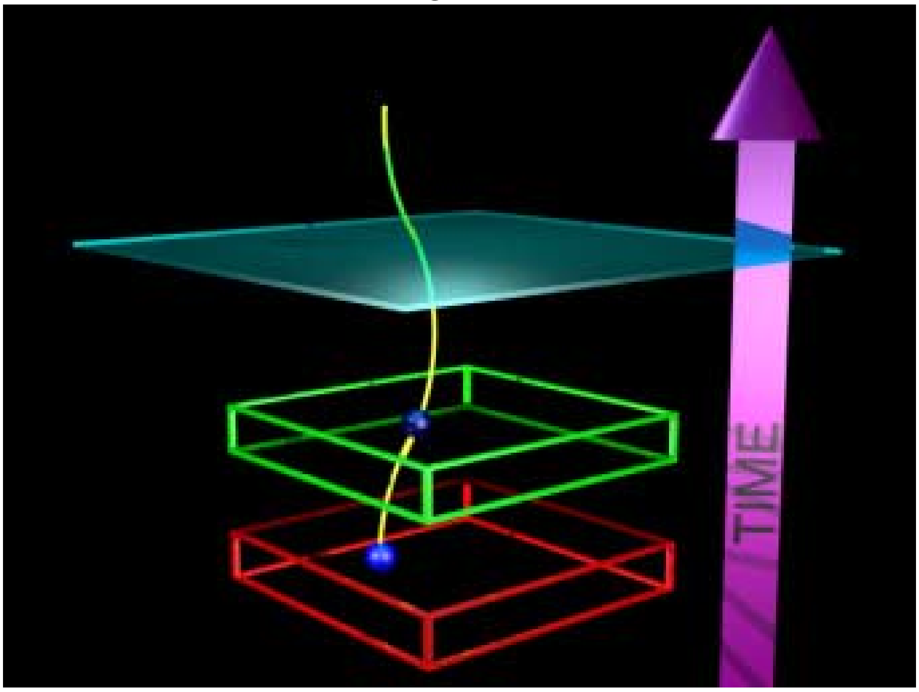

Figure 1 - System visualisation with one component of interest

This provides customisation to suit the user and task, and thus relives any visualisation at that level of the problem of trying to be all things to all data.

Figure 1 demonstrates the various ideas encompassed within this visualisation. Since the concept of time is very important for both systems evolution, and for the concept of deadlines, delivery and the management of the combination of all of these it plays a major part of the visualisation structure. The Z (upwards) dimension is time in relation to the systems, services, and deadlines. The two wire framed boxes represent a collective visualisation of system(s) related to this analysis at two distinct points in time (such as a major release). They are displayed as wire framed boxes to indicate that this is an abstracted view, and that detailed system visualisation of just that information is available. The spheres within the boxes represent a component-part of interest. The first instance of the sphere is also the starting point of a traced path through which the component is expected to travel. The second instance (within in the green box) is also along this same path to show that intentions met reality at this system release. The plane at the top of the diagram represents a fixed deadline. The projected path of the component of interest has been estimated beyond this time, and the intersection of the two can provide information about whether that component is likely to meet such a deadline.

Figure 2 shows an expanded version of Figure 1 to demonstrate some of the more advanced concepts that the visualisation is capable of incorporating. System data is never removed from the warehouse, which allows for the darker paths to exist. These are the ones that have originating spheres but no more along their length. These were projected paths for that component at the time of first (visible) system release. As time progressed only one of the three possible paths for the leftmost component was realised, and therefore the others remain for information and guidance with data behind them indicating reasons why they existed. The path that is taken is lighter in colour, and only those that cease to be valid are darkened.

Figure 2 also has several more components of interest visible at the same time. It is highly likely that there will be more than one for any given analysis task. The customisation of such views allows the components of interest to be selected and also allows for selective viewing of the predicted and actual paths. This customisable part of the visualisation also prevents the problem if all pieces of information

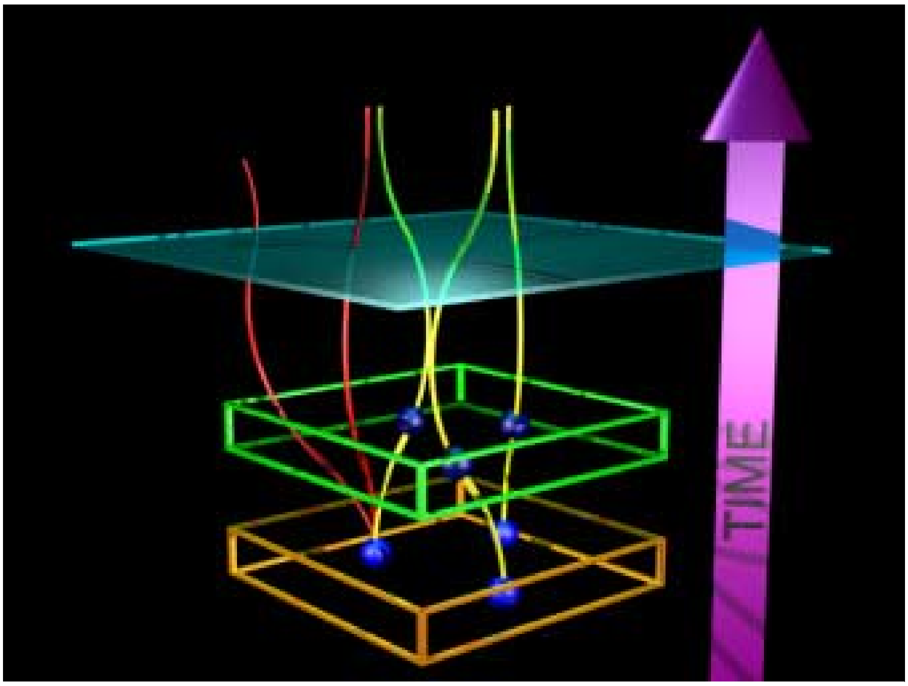

Figure 2 - System visualisation showing various components and both real and failed paths

were always visible; that of visual overload through visual clutter.

In order to be able to best utilise the underlying data and encapsulated knowledge, annotations are possible. These can be added along the component paths (with time a distinguishing feature thus allowing for several), at the components at each release, the releases themselves, and also any fixed deadlines. These annotations at conceptually higher levels can be stored in a data warehouse of their own and thus then do not interfere with the original data. This is a useful feature because the visualisation is constructed through the aggregation of a large amount of related data. Not only does the visualisation allow for effective viewing of this at once, but it also allows for the capture of extra information that can only be inferred by someone who has all of the data available to them and the domain knowledge in which to make judgements.

7. Conclusions

This visualisation aims to provide a way of viewing, understanding, analysing, and managing software systems. Essentially using software to cope with the demands of modern software. The visualisation has a potentially wide use base. This is because of its flexibility in incorporating high and low level detail in the visual display (before displaying an entirely new visualisation) and the integration of annotation and organisational information. This means that the visualisation is able to provide purely information and data provision services to one user whilst acting as a guidance tool to those who require more information for future decision making and planning. It also serves as an understanding aid to both managers and lower level system analysts. This visualisation, it should be stated, is not intended to make programmers managers or vice-versa, but to integrate everyone's knowledge about a system to enable the best analysis and knowledge utilisation possible.

The visualisation utilises data that already exists, which is likely in many organisations with software systems hence this work has a wider applicability. It provides a means of documenting history to help the future in the specific context of an organisation. This reduces system knowledge loss, which is a documented problem for software maintenance. The visualisation enables users to find their own evidence in the dataset and supports various sources for that evidence. It provides a way of integrating information into a coherent whole view and thus can be considered to be of benefit as a visualisation oriented knowledge aid.

Acknowledgements

This work has been supported by the EPSRC project VVSRE; **V**isualising **S**oftware in **V**irtual **R**eality **E**nvironments, and influenced by contact with industrial partners.

References

[1] M. Crossley, N. J. Davies, R. J. Taylor-Hendry, and A. J. McGrath, "Three-dimensional Internet Developments", *BT Technology Journal*, Vol. 15, No. 2, pp179-193, April 1997.

[2] M. Crossley, N. J. Davies, A. J. McGrath, and M. A. Z. Rejman-Greene, "The Knowledge Garden", *BT Technology Journal*, Vol. 17, No. 1, pp76-84, January 1999.

[3] R. Hubbold, A. Murta, A. West, and T. Howard, "Design Issues for Virtual Reality Systems", *Presented at the First Eurographics Workshop on Virtual Environments*, 7 September 1993.

[4] C. Knight and M.Munro, "Comprehension with[in] Virtual Environment Visualisations", *Proceedings of the IEEE 7th International Workshop on Program Comprehension*, pp4-11, May 5-7, 1999.

[5] C. Knight, "System and Software Visualisation", To appear in the *Handbook of Software Engineering and Knowledge Engineering (2 Volumes)*, mid 2001.

[6] Data Management, Exploration and Mining at Microsoft, available online at http://research.microsoft.com/dmx/DataMining/default.asp

[7] E. P. D. Pednault, "Representation is Everything", *Communications of the ACM*, Vol. 43, No. 8, August 2000.

[8] G. Walker, "Challenges in Information Visualisation", *British Telecommunications Engineering Journal*, Vol. 14, pp17-25, April 1995.

[9] P. Young and M. Munro, "Visualising Software in Virtual Reality", *Proceedings of the IEEE 6th International Workshop on Program Comprehension*, pp19-26, June 24-26, 1998.