Design of a User Interface for Process Control in a Medication Production Facility

Harry Cooney

A dissertation submitted to the University of Dublin,
in partial fulfilment of the requirements for the degree of
Master of Science in Health Informatics

2011

Author(s) Declaration

I declare that the work described in this dissertation is, except where otherwise stated, entirely my own work, and has not been submitted as an exercise for a degree at this or any other university.

Signed:

____

Harry Cooney

September 19, 2011

Permission to Lend and / or Copy

I agree that Trinity College Library may lend or copy this dissertation upon request.

Signed:

____

Harry Cooney

September 19, 2011

Acknowledgements

My thanks go to Jackie and my family for supporting me while I completed my study and, yes, it did take a little longer than expected. I would also like to thank the staff at Trinity College, and Lucy Hederman and Bridget Kane in particular, for their advice and guidance. Finally my employers encouraged me to continue developing myself, thanks Tony and Martin.

Summary

This study demonstrates how computer systems that are used to control production processes can be enhanced to provide greater ease of use and help increase the users' productivity. In a healthcare setting these benefits can translate into greater patient safety and improved access to care through lower costs.

Today charts are heavily used in dashboards and business intelligence applications but their use is still not common in line of business applications. The conditions may be right for this to start changing. Recent improvements in computer hardware and software have greatly reduced the effort required to include graphical data representations in the interfaces of computer systems. In addition, methods are emerging that can guide designers to create increasingly effective graphical representations of data.

At the site where this study is set medications are prepared by combining off-the-shelf drugs in doses prescribed individually for each patient. The production process is controlled by computer systems but there is still a significant manual effort in preparing the medications. This study generated a new user interface design for a module of the existing production control system. The new design incorporates graphical data representations designed using state-of-the-art tools and methods.

The new design was compared against the current design and the results indicate that the new design presents data more effectively to the users and increases the ease-of-use especially for new users.

Table of Contents

Table of Contents vi

List of Figures viii

List of Tables viii

List of Abbreviations ix

1 Introduction 1

1.1 Background 1

1.2 Aims and Objectives 3

2 Setting for the Study 4

2.1 Introduction to Production Management 4

2.2 The Production Environment 5

2.3 Challenges facing the Production Supervisor 8

2.4 The Existing Solution – The Web Planner 10

2.5 The Web Planner – Areas for Improvement 10

3 Summary of the Literature 14

3.1 Overview 14

3.2 Information Visualisation 14

3.3 Communicating through Graphics 21

3.4 State of the Art in Chart Design 24

3.5 Interface Design Methods 29

3.6 Selected Tools 40

4 Methodology 41

5 Analysis and Design Phases 42

5.1 Eliciting Requirements 42

5.2 Designing the Proposed Solution 44

5.3 Confirming all requirements are met 52

6 Results 54

7 Discussion 57
7.1 Lessons from Using the Tools 57
7.2 Source of the Benefits Achieved 59
7.3 Weaknesses of the Study 60
7.4 Opportunities for Future Work 60
8 Conclusion 63
9 References 66
Appendices 72
A. Extract from Hierarchical Task Analysis 73
B. Extract from the Abstraction Hierarchy 77
C. Extract of the Function and Data Requirements 79

List of Figures

Figure 1. Screenshot of existing Web Planner 13

Figure 3. Pictograms (Mulhall, 2009) 19

Figure 2. Coxcomb (Nightingale, 1887) 19

Figure 4. Statistical Control Chart (van Reeuwijk, 1998) 20

Figure 5. Isotype Graphics (Burke, 2009) 23

Figure 6. Sparklines (Tufte, 2006) 26

Figure 7. Pareto Chart (Sookman, 2010) 27

Figure 8. Bullet chart (Few, 2006) 27

Figure 9. Box plot (Kirkman, 1996) 27

Figure 10. Scatter plot (Rezaeian, 2007) 28

Figure 11. Tree map (Shneiderman 2001) 28

Figure 12. Simple Interaction Design Lifecycle Model (Sharp Rogers & Preece, 2007) 30

Figure 13. The 5 phases of CWA (Jenkins 2006) 32

Figure 14. Layout of Final Design 47

Figure 15. Monitor panel 49

Figure 16. Temporal Mosaic (Luz & Masoodian, 2007) 61

List of Tables

Table 1. AH 5 Levels of Abstraction (Jenkins, 2006)............................. 33

Table 2. Comparison of output from HTA and AH................................. 45

Table 3. Coding of Icons Representing Jobs 51

Table 4. Referencing the Design to the Requirements 53

Table 5. Extract from Hierarchical Task Analysis.................................. 74

Table 6. Abstraction Hierarchy (Extract).. 78

Table 7. Function and Data Requirements (Extract) 80

List of Abbreviations

AH	Abstraction Hierarchy
CIS	Computerised Information Systems
COTS	Commercial off-the-shelf software
CWA	Cognitive Work Analysis
EHR	Electronic Health Records
EID	Ecological Interface Design
HTA	Hierarchical Task Analysis
IT	Information Technology
KBB	Knowledge-based behaviour
NIMIS	National Integrated Medical Imaging System
RBB	Rules-based behaviour
SBB	Skills-based behaviour

1 Introduction

1.1 Background

The Health-care industry is adopting computer systems to gain increased operational efficiency and improved quality of outcomes in the areas of clinical, administrative and managerial-strategic decision support (Wang, Wan, Burke, Bazzoli, & Lin, 2005). This adoption is being accelerated in the US, for example, where President Obama announced that US$19 billion will be spent on Healthcare Information Technology (IT) and especially on Electronic Health Records (EHR) systems as part of the American Recovery and Reinvestment Act 2009 economic stimulus package. Beyond EHR, clinicians use IT systems to support all aspects of their work. Computerised Physician Order Entry systems include extended checks and alerts to reduce the incidence of errors when prescribing medications (Ammenwerth, Schnell-Inderst, Machan, & Siebert, 2008). In Ireland the Health Service Executive is also investing in technology, for example driving the implementation of a country-wide Picture Archive and Communication System and Radiology Information System systems as part of the National Integrated Medical Imaging System (NIMIS). NIMIS will create a paperless national system for the capture, storage and sharing of medical images and associated assessments.

As the healthcare industry matures it is following other industries in adopting commercial off-the-shelf (COTS) software packages instead of building systems using their own resources. There are 3 primary considerations in the arguments for and against buying or building software and these are (i) the cost of building/acquiring the software (including the risk of failure) and on-going maintenance effort for the applications (ii) the degree of fit between the delivered system and the organisation's processes, and (iii) the duration of the project from the time where the business need is identified until the system is deployed. Healthcare has additional influencers – government may influence decisions by supporting options financially to drive integration or inter-operability across the national healthcare market as with NIMIS in Ireland. The rule of thumb among software developers is to buy COTS systems where they are available to satisfy standard requirements and invest in tailored applications that give an organisation a competitive edge (Traylor, 2006). If this logic carries through to the health-care market then the market will see increased investment in the back-office systems built to support non-specific processes at a variety of organisations.

Commercial computerised information systems (CIS) are designed and built to be used by users of varying abilities in diverse settings and so they cannot be expected to meet the specific needs of an organisation' processes or its users. One technique used to satisfy the varied requirements is to adopt a design methodology of *Design for Availability* as described by (Woods, 1985), (Eccles & Groth, 2006). The key principle of design for availability is that the internal model of the system is made visible to the users and they use this model in planning how they work with the CIS. In practice, design for availability can be implemented lazily by presenting much of the application data in table or grid layouts. This presentation method ensures that users have access to the data needed to support varied processes but it can result in systems that present volumes of data to the user without any process context. When the users are presented with data with no context they must refer to an internal or external model of the business process and then relate the CIS data and dialogs to the process steps without any direct support from the CIS. This places a load on the users. There is also the expectation that the users are able to flexibly work around short-comings in the design or implementation of the system because the humans "*often also implicitly have the vicariant role of fixing a technical system that was built imperfect*" (Lahlou, 2007).

The result is a situation where the organisation reaps benefits from acquiring and implementing COTS systems balanced by an increased mental load placed on the users of the system. In a health-care setting putting additional load on the users can lead to errors and impact on patient safety.

Studies have identified a number of possible user errors that reduce the users' effective performance in cases where their attention is divided or there is data overload (Woods, 1985). These errors include tunnel vision and users becoming fixated on insignificant data. While these risks may not be critical for users of business systems they present additional risks to patient safety in a healthcare environment.

Our eyes are the sensory channel that presents the highest capacity for gathering and interpreting information. Much work has been done over the past 40 years to define principles that can be used by software designers to create computer systems that support knowledge workers more effectively in their day-to-day work. While the sciences have adopted visualisation as a way to deal with the exploding volumes of data being generated by scientific instruments but the use of visualisation in business applications is lagging by 10 years (West, 1995). A more recent study specifically assessing the use of data visualisation with electronic health records reported that "*Many EHR systems lack features that support important end-user tasks. Exploratory analysis, effective representation, and temporal queries are but a few that are often found lacking even in state-of-the-art systems ....*" (Wang, Wongsuphasawat, Plaisant, & Shneiderman, 2010).

This study demonstrates how an existing module of a software application can be redesigned to be a much more effective tool by presenting key data graphically.

1.2 Aims and Objectives

This purpose of this research is to demonstrate and measure the benefits that can be expected when designers include improved data representations in applications used to manage health-care processes. The study takes the form of a project to redesign a module of an existing bespoke software application incorporating improved graphical presentation of key data. The software module is used by production supervisors to control an operation that produces patient-specific medication in low volumes.

The study begins by analysing the current process and the users' objectives. When the operational and user contexts are documented a new interface will be designed. The study follows guidelines and approaches to requirements elicitation and software design that have emerged over the last 40 years. When the prototype of the new module has been created the users' response to both versions are compared.

2 Setting for the Study

2.1 Introduction to Production Management

The hands-on phase of this study concerns itself with the design of an improved user interface using graphical presentation of data to support and improve production scheduling activities in a small scale make-to-order production site.

Graves (1981) distinguished *production scheduling* from the related topics of *inventory management* and *production management* as follows: production scheduling is focussed primarily on allocating resources to production tasks; it assumes that the necessary materials and resources have been foreseen by the Inventory and Production Planning processes. This introduction will not review or analyse the theory of production scheduling in detail but some background will help people unfamiliar with the environment to relate to the subject.

In a make-to-order environment production activities are undertaken in direct response to customer orders; in contrast, a make-to-stock environment is one where production is planned based on forecast aggregate demand and the finished goods are stored in warehouses until the anticipated customer orders are received.

The production manager’s primary aim is to maximise customer service levels while minimising cost. Customer service levels are typically measured as accurate and on-time fulfilment of the customer order while costs are measured financially as inventory (of raw materials, work in process and finished goods) and the cost of unused production capacity. It is easier to meet customer expectations if a supplier manufactures and stores extra buffer stock or if the company maintains a high level of unused capacity to cope with unexpected orders. Both of these options cost money and will only be rewarded if unexpected orders are placed and if the price that can be charged is sufficient to cover the costs of carrying the excess inventory or unused capacity. If the additional stock or capacity are unused, or if a premium price cannot be charged for the unplanned orders then the company suffers the cost.

There are a number of options available to the manager to cope with a mismatch between the required level of output and the level of output being achieved. In the short term production capacity can be increased by adding overtime or additional shifts; conversely it can be reduced by eliminating shifts or through temporarily reassigning people to other activities. For medium term adjustments the load can be adjusted increasing or decreasing production batch sizes or giving/taking work with external suppliers. External suppliers can take on self-contained activities such as sterilisation or they can be contracted to supply finished goods or sub-assemblies that would normally be produced in-house.

Operating in a health-care environment can limit a producer's flexibility to respond to changes in demand. For example, operations that have an impact on the safety or efficacy of medical products can only be out-sourced to third parties following a stringent assessment of the ability and controls of the third party.

2.2 The Production Environment

The production environment which forms the basis for the study is a homogenised model based on a number of units that are operational in Ireland and the UK. The product sets are the same at all sites and the production processes are similar but these sites work closely with their clinical customers to deliver personalised medications and so there are variations in production capacity, layout, and product mix and in some of the technologies used.

The units produce medications in low volume under sterile conditions for administration to individual patients. The majority of orders are received for same day or next day delivery and most of the medications produced have a shelf-life measured in hours or days.

It is important to understand the relationship between prescriptions, orders and jobs to avoid confusion as we discuss the process. The process begins when a clinician issues a *prescription* for a course of treatment for a patient. A course could be a one-time dose of a medication, a series of 10 treatments with the same medication repeated daily or in a more complex case it could be a series of 8 weekly treatments where each treatment is a different combination of varied doses of prescription medicines. A pharmacist at the treatment centre reviews the prescription takes the decision to prepare medications in-house or to order some or all of the medications externally; where medications are ordered externally the pharmacist places the appropriate *orders*. A single order may be for one or more treatments; at the production site the order results in one *job* being placed on the production control system for each treatment included in the order. The jobs are produced according to the required dates, labelled with identifying information including the patient identification and delivered to the pharmacist or ward at the treatment centre.

The medications produced are in three main product families – i) oncology treatments for cancers, ii) parenteral nutrition for intravenous feeding of patients who are unable to take food orally or digest normal food and iii) high-dose antibiotics. The value-added element that makes the service interesting for health-care providers is the tailoring of the delivered medication to match the characteristics of the patient and the stage of their condition. It is proven through clinical trials that cancer treatments are more effective when they are tailored to the stage of the disease and the age, size and other characteristics of the sufferer. An oncology treatment may be a combination of 2 or more commercially available drugs proven to give better results against a specific condition. Similarly, the prescribed dose of parenteral nutrition must reflect the age and development stage of the patient and also the elements of their digestive system that are impaired. Finally, high-dose antibiotics are part of the strategy used in Ireland to deal with the limited facilities available for cystic fibrosis sufferers. The aim is to avoid the risk of infection in hospital settings by delivering tailored doses of antibiotics to the patient for use in the home.

2.2.1 Personalised Medicine

During the 20th century medicine made great advances through the use of tightly controlled compounds, randomised testing on groups of patients and assessment of evidence-based outcomes. Personalised medicine is a model for diagnosis and treatment which attempts to build on the positive lessons learned to date and make further progress by recognising that it is an over-simplification to treat all patients as a homogenous population. Patients form sub-groups that will respond differently to treatments, based on ethnic background or family history for example, and the best outcomes can be achieved by treating each patient as an individual. The existence of the production units described in this study show that clinicians are practising personalised medicine today based on knowledge of the patient and their condition. Since the completion of the Human Genome Project it is becoming ever more practical to include genetic information in the drug testing process and in the treatment decisions for individual patients (Piquette-Miller & Grat, 2007, pp. 311-315). This will be the enabler for widespread adoption of the personalised medicine model and that will ensure continued demand for facilities that can create tailored combinations and doses of standardised medications such as those described in this study.

2.2.2 The Production Process

At the production unit each job goes through a number of steps beginning with the initial entry of the order until it is shipped to the customer. All jobs pass through the same steps of:

Order Entry	Data entry including automated checks on drug volumes, checks on the stability of the resulting compound and calculation of the shelf life of the resulting medication
Review	A pharmacist checks the order in the system for errors and also compares it against the original order received
Release to Manufacture	The production supervisor releases jobs to be manufactured so that they can be prepared and delivered to meet the customers specified delivery date and time
Stock Picking	The components for the job are retrieved from storage and brought to the production area
Sterilisation	Ensure that the components are free of contamination
Manufacture	The medication is produced by combining the

	necessary components into a final container, e.g. syringe, infusor or intravenous solution container. The complexity of the job depends on the number of components, the number of operations to be performed to combine them safely and whether particular care is needed e.g. to avoid needle stick injury
Labelling	The containers are labelled with the customer delivery details, patient identifiers, batch number, expiry date and control information
Inspection	The physical product and the production records are compared to the original job details
Despatch	The courier delivers the medications from the production site to the customer

The order entry and review tasks are triggered by an incoming order. The remaining steps follow after the production supervisor has selected jobs and released them for production.

2.3 Challenges facing the Production Supervisor

The supervisor decides when to release jobs based on the overall capacity-demand balance at the production site and on certain attributes of the individual jobs.

Production capacity depends on the number of workstations available for each activity and on the technical characteristics and throughput of each workstation. In addition, capacity may be reduced at the site due to employee absence or equipment maintenance activities.

On the demand side the work-load may be higher than normal in anticipation of up-coming public holidays and in these cases the supervisor will attempt to release jobs at the earliest opportunity. This means utilising capacity as early as possible provided that the resulting products have sufficient stability and shelf-life to be fit for use when administered by the clinicians following delivery to the customer. In this way spare capacity is held in reserve to meet any urgent orders that come in during the shift.

In addition the supervisor must consider the attributes of the individual jobs to decide when they can be released into the workload for a shift. Jobs have a delivery date and time requested by the customer and an internally defined measure of complexity assigned by the system, the batch equivalent (BEQ). The BEQ reflects the amount of work required to produce the medication and therefore the duration of the manufacture activity. The earliest time that a job can be released to manufacture depends on the required delivery date/time and the shelf-life. If the medication is produced too early then it will have expired before it can be administered. In this case the medication is discarded, the patient is not treated, the treatment will be rescheduled and the medication will have to be produced again. The latest time that the job can be released depends on how long it takes to produce the medication – time to get the appropriate components picked, sterilised and transferred into the isolator or cabinet where manufacture will take place and the elapsed time for manufacture, labelling and inspection. Delivery runs are scheduled to each customer in the morning and the afternoon and the supervisor targets those delivery times when planning the production.

The production plan and the delivery schedules can be broken for urgent orders or to clear a backlog that may have built up but it is difficult to recover from interruptions during a shift. For example, the sterilisation cycle may take an hour. If sterilisation for a batch of components is interrupted to facilitate an urgent job that will have an impact on the manufacture and delivery times for all of the jobs depending on the interrupted components. As such, the operation runs much more smoothly if the supervisor has access to all of the information needed to maintain an overview of the schedule and progress against the plan.

The supervisor must build some flexibility into the schedule to handle unexpected jobs. This can be done by including additional components in the steriliser so that there is excess available for unexpected jobs. This solution is not guaranteed to work and it does introduce waste into the process. If there is a clash between the scheduled production and an urgent order then the supervisor may discuss the relative priorities of the jobs with the customer and come to an arrangement that minimises disruption by agreeing changed delivery times for one or more jobs. This approach is likely to leave the customer at least partially frustrated and it can also take a lot of time.

2.4 The Existing Solution – The Web Planner

The Web Planner is the name given to the existing module that supports the supervisor in maintaining the scheduled level of production and in dealing with unplanned urgent orders. Figure 1 (p. 13) shows a screenshot with the customer and patient identification hidden to preserve data privacy.

Looking at Figure 1 it is clear that the Web Planner was primarily designed to help the supervisor choose the next job(s) to release to production and therefore most of the screen area is used to display a list of jobs (see Block D). The concept is that of a Departure Board as seen at airports or train stations. Any urgent jobs are displayed at the top of the list.

Block A is a calendar allowing the supervisor to select the date. This can be used to preview the jobs to be produced on the following day.

Block B contains a number of options. The supervisor can choose to include or exclude jobs from the previous day that are not yet complete. To make best use of the screen space the user may show/hide the menu bar and show/hide the BEQ Totals (Block C).

Block C, the BEQ Totals, is a measure of progress on the day's scheduled production. It displays the current total of the BEQ values for all jobs by job status. At the start of the shift the totals indicate the workload to an experienced supervisor. During the shift the supervisor can see how the scheduled production is progressing by checking these totals. The sample screen shot it taken close to the end of a shift and so we see that the workload is 96% complete.

2.5 The Web Planner – Areas for Improvement

Initial interviews with the users of the module identified weaknesses or opportunities for improvement that can be classified in two categories. Some of these weaknesses are due to the limited scope that was defined for the Web Planner module during the original design and so we can identify opportunities that broaden the scope of the module. Other weaknesses reflect features that were within the original design scope and that have been implemented in a way that makes the module inefficient or difficult to use in certain situations. For example, the module relies on the user having knowledge of the operating environment that is not available to inexperienced users.

Increased Usability

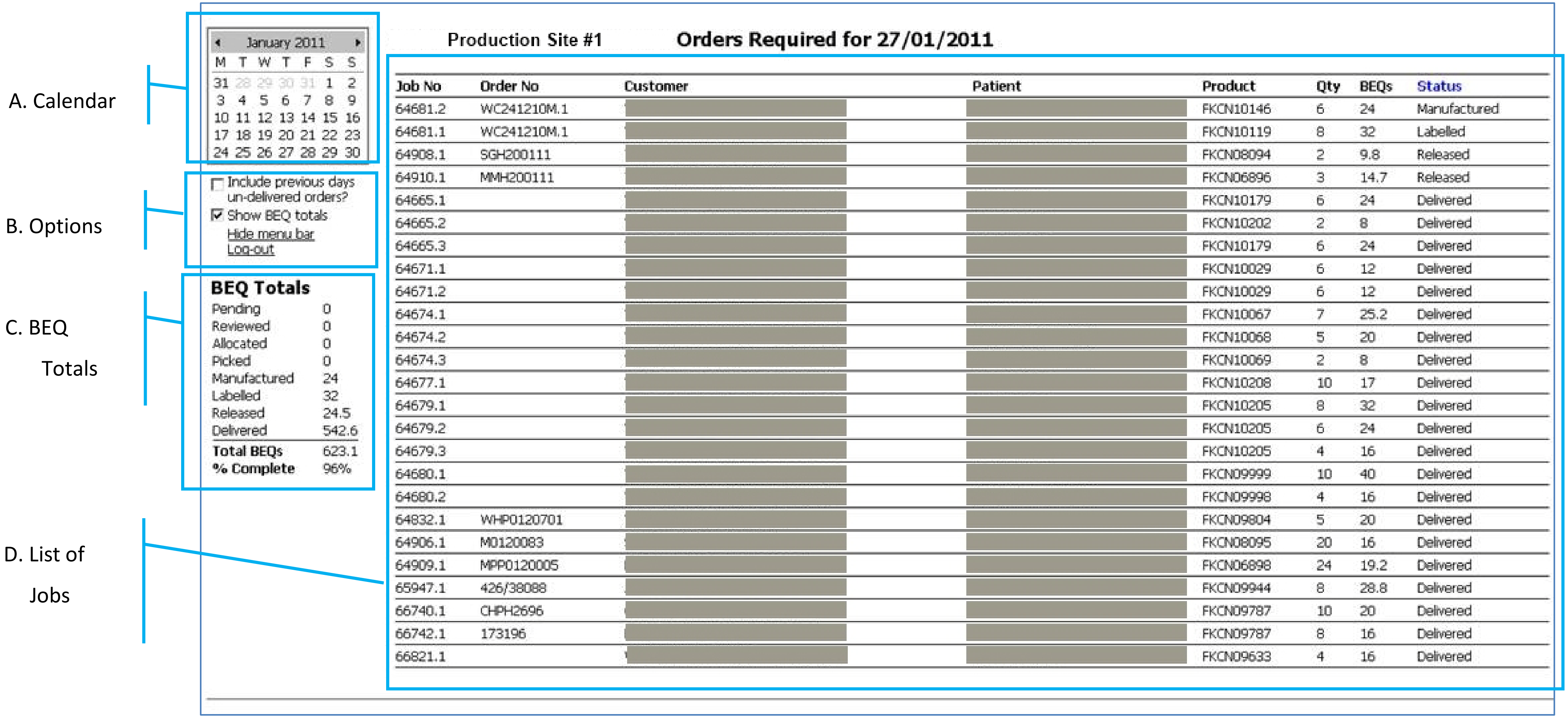
- Most of the screen is given over to an output-only list of Orders/Jobs. Uses requested that list should be more interactive e.g. filtering, especially by urgency, customer or status.
- It is not possible to take action from the list of Orders/Jobs. The commonest action taken is to release one or more jobs for production and users requested that this action should be available from the planner.
- Early in a shift a supervisor can tell how large the workload is by interpreting the BEQ Totals and the number of urgent orders based on their experience. The Web Planner could provide more context to enable less experienced supervisors make useful judgements
- The BEQ Totals are useful to see progress during the day but they are a snapshot at a point in time. It would be better if the user sees progress up to the time the system is consulted as this may be used to predict the pattern for the remainder of the shift.

Broader Scope

- The production units produce medications in three product families. For historical reasons the Web Planner was designed to give visibility to products from only one product family. There are obvious benefits to being able to see the full range of production; however, the product families are important in resource allocation and so it should be possible to filter the view by product family.
- There are a number of production units in the UK and Ireland. As the business becomes more complex it is possible that orders placed against one production unit may be manufactured as jobs at a different unit and shipped to the customer. This makes it important to

identify which production unit a job is assigned to and to be able to filter on the production unit.

- Eventually the Web Planner could enable the supervisor to issue an estimate on the likely delivery time for a newly placed order.



Job No	Order No	Customer	Patient	Product	Qty	BEQs	Status
64681.2	WC241210M.1			FKCN10146	6	24	Manufactured
64681.1	WC241210M.1			FKCN10119	8	32	Labelled
64908.1	SGH200111			FKCN08094	2	9.8	Released
64910.1	MMH200111			FKCN06896	3	14.7	Released
64665.1				FKCN10179	6	24	Delivered
64665.2				FKCN10202	2	8	Delivered
64665.3				FKCN10179	6	24	Delivered
64671.1				FKCN10029	6	12	Delivered
64671.2				FKCN10029	6	12	Delivered
64674.1				FKCN10067	7	25.2	Delivered
64674.2				FKCN10068	5	20	Delivered
64674.3				FKCN10069	2	8	Delivered
64677.1				FKCN10208	10	17	Delivered
64679.1				FKCN10205	8	32	Delivered
64679.2				FKCN10205	6	24	Delivered
64679.3				FKCN10205	4	16	Delivered
64680.1				FKCN09999	10	40	Delivered
64680.2				FKCN09998	4	16	Delivered
64832.1	WHP0120701			FKCN09804	5	20	Delivered
64906.1	M0120083			FKCN08095	20	16	Delivered
64909.1	MPP0120005			FKCN06898	24	19.2	Delivered
65947.1	426/38088			FKCN09944	8	28.8	Delivered
66740.1	CHPH2696			FKCN09787	10	20	Delivered
66742.1	173196			FKCN09787	8	16	Delivered
66821.1				FKCN09633	4	16	Delivered

Figure 1. Screenshot of existing Web Planner

3 Summary of the Literature

3.1 Overview

This study concerns itself with the design of a user interface for a process control software application. The literature review covers two distinct areas. It begins with a review of the principles of user interface design and the graphical representation of data because these are necessary for the effective presentation of data in the user interface. However, it is also necessary to identify the functional and data requirements of the users before the interface design can begin and so the second section covers the tools and methods used to capture requirements and create a design.

3.2 Information Visualisation

"The drawing shows me at one glance what might be spread over ten pages in a book" (Turgenev, 1991)

Turgenev's character in *Fathers and Sons* (1991) clearly expresses the idea that pictures and other graphical representations use the brain's capacity for visual processing to convey information rapidly. It has been claimed that humans acquire as much information visually as we do from all of our other senses (Ware, 2004). Given the capacity of images to carry information it is not surprising that there is evidence of a long history of visual communication among people from very early in the history of human culture.

The most famous earliest drawings are cave art but maps were the most economically important early visualisations. During the 1700s as human knowledge became codified through the disciplines of science and statistics data visualisation grew in importance as a way to analyse data and to communicate information. These visualisations were stylised maps and statistical charts. Until the 1960s the tools for data visualisation were refined through practice without an overarching theory to guide their development. It is in the last 40 years that theorists have codified this practice and developed principles for the design of data visualisations based on knowledge of human perception and also the affordances of the presentation media.

Computer technology has been a significant enabler of data visualisation. The rapid development of software tools and powerful hardware has allowed larger data sets to be visualised with less effort. The internet provides potential data sources, as data-sets are shared online, and an unparalleled publishing and collaboration medium providing large audiences with access to the resulting images. As a result the designers of commercial packaged software and their end-users are exposed to extensive use of data visualisations on web-sites and in traditional newspapers and magazines. They see the way that well designed graphics can help analyse large volumes of data and communicate important information. It is only natural that both groups will expect to see this technology incorporated into future generations of packaged software.

Technology makes it easy to generate graphics from data but it takes much design and planning to use data visualisation effectively in an IT system. For graphics to support the end-user in their job it is necessary to create a model of the task being supported, to extract the data and the functions that the system must provide to the user and then to select the most effective way to present these to the user. The tools required for this analysis and design are based on a long history of previous work. Industrial engineering had established principles of design focussed on the domain of physical labour before the use of interactive computer systems. As the use of computer systems became widespread researchers were driven to develop methodologies to design applications that support knowledge workers in their work. The 1960s saw the emergence of the first theoretical frameworks to guide the assessment and development of charts and data graphics. Over the subsequent half century the frameworks have matured and been improved with developments based on theoretical advances and based on real world use of graphics in software packages, periodical and newspapers.

3.2.1 A Brief History

This section provides an overview of the evolution of information bearing graphics through history. The early wave of exploration and conquest drove the development of map-making techniques. As governments became more organised the discipline of statistics emerged and required charts to represent the data economic and demographic being gathered. Over time the diagrams developed to present more complex pictures of the data.

3.2.1.1 Map-making

Kings, emperors, military forces and navigators had a need for a geographical representation of the world to support the planning and recording of taxes, military campaigns and voyages of discovery. The oldest surviving maps are the World Maps from Babylonia from the 9th century BC and there are also surviving geographical documents from China in the 5th century BC. Over time, map-making advanced to represent the landscape more accurately with the development of projections to take account of the earth's curvature, the use of stylised symbols to represent artefacts, and the reliable use of scales. The 18th century saw the introduction of contours and isolines to identify common features on the map, for example Buache's geographical contour map of France and Galton's weather map. Buache, 1752 and Galton, 1861 are both cited in Friendly & Denis (2001).

While the primary use of maps was to aid and record navigation it was a natural extension to use maps as an aid to analysis and explanation. A landmark example in the early development of the science of epidemiology occurred in London in 1855 (Snow, 1855). Dr. John Snow used a map to document the deaths resulting from an outbreak of cholera in Broad Street, London in 1854 and to support his argument that the disease was spread by contaminated water.

3.2.1.2 Statistical Graphics

Statistics is the science of collecting, organising and interpreting data (Best, 2001). According to David (1995) the term "Statistics" itself is referenced in the Oxford English Dictionary to a translation by W. Hooper of Bielfeld's *Elementary Universal Education* in 1770. The initial focus of statistics was measuring economic and demographic data for the emerging states. Through the early 19th century the scope of statistics broadened and the relationship with probability theory developed; see for example the work of Laplace (1749-182) on mathematical astronomy and statistics. Simple diagrams had been used to represent data and concepts through the 17th and 18th centuries and from the end of the 18th century we see the emergence of recorded examples of data representations that are still in use today.

In *The Commercial and Political Atlas*, originally published in 1786, William Playfair introduced the line chart and the bar chart to represent economic data. The earliest printed example of a pie chart appeared in 1801 in Playfair's book *The Statistical Breviary*. Both books have been republished recently as Playfair, Wainer and Spence (2005). Playfair used the pie charts to represent the land areas of countries in Europe. For example, the pie chart representing Turkey was divided into 3 coloured segments representing the proportion of the land mass in Europe, Asia and Africa. In the same volume Playfair also introduced a diagram with intersecting circles which he referred to as "Eulerian Circles'' in reference to earlier works attributed to the Swiss mathematician Leonhard Euler (1707-1783). Today these charts are referred to as Venn diagrams reflecting their use by Venn working with Boole's system of logic in 1880. Thus the basic set of statistical graphs had been introduced by the early 1800s and their key features have remained valid until today (Biderman, 1990).

3.2.1.3 Information-bearing Diagrams

The use of diagrams and graphics to convey information developed organically as practitioners learned from each other. Maps had evolved in to include information on population and economic activity toward what is now called thematic cartography. Statistical graphics had developed a toolbox of techniques that could be used to illustrate attributes of data collections, find relationships within the data and illustrate features not easily identified in the raw data. In the mid-1800s the elements were in place to allow more complex diagrams to be created and understood. Here we will consider two charts from the late 19th century that are considered to be classics in this field.

In 1857 Florence Nightingale and Dr. William Farr conducted an investigation into the high mortality among the British military during the Crimean war. Their analysis concluded that the causes of death were due to preventable disease and poor medical treatment rather than fatalities in combat (Nightingale, 1857, republished online). Their report includes a series of coxcombs, or polar area graphs, that highlight the relative importance of three causes of death – *wounds*, *all other causes* and *zymotic diseases* (or infectious diseases). Contrary to expectations wounds led to fewer deaths than the other two categories – the peak is clearly visible on the November arc radius of the chart. The shading shows that infectious diseases were by far the greatest contributor to fatalities.

Figure 2 shows an example of a coxcomb. The arcs, or wedges, in the diagram represent particular time periods for which data had been collected. The concentric circles indicate the number of deaths from a cause in that particular period. More recently, it has been found that people have difficulty in accurately assessing the relative sizes of angular measures (Tufte, 1983, p. 178) and so the polar graph and the more popular pie chart have fallen out of favour among professionals.

Mulhall's *Dictionary of Statistics*, first published in 1884, contains early usage of pictograms to represent data values using icons whose size varies in direct proportion to the value being represented (Mulhall, 2009, p. 77). See Figure 3 for an example comparing livestock headcounts to human population for a number of countries. Pictograms are engaging for the reader and are used in the infographics of contemporary publications e.g. the *U.S.A. Today's* Snapshot feature.

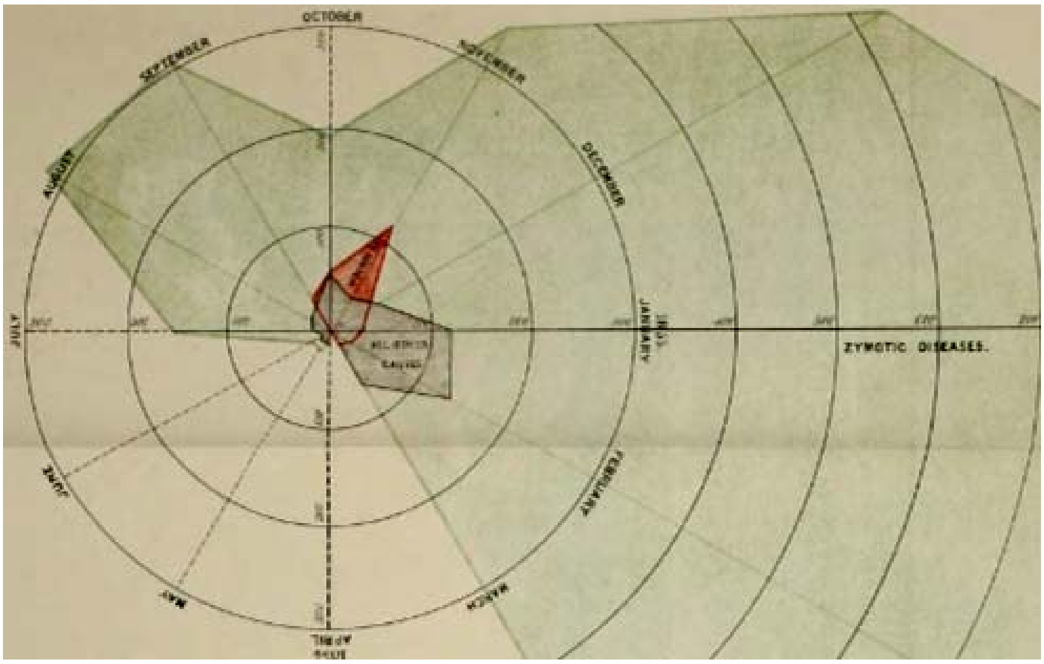


Figure 2. Coxcomb (Nightingale, 1887)

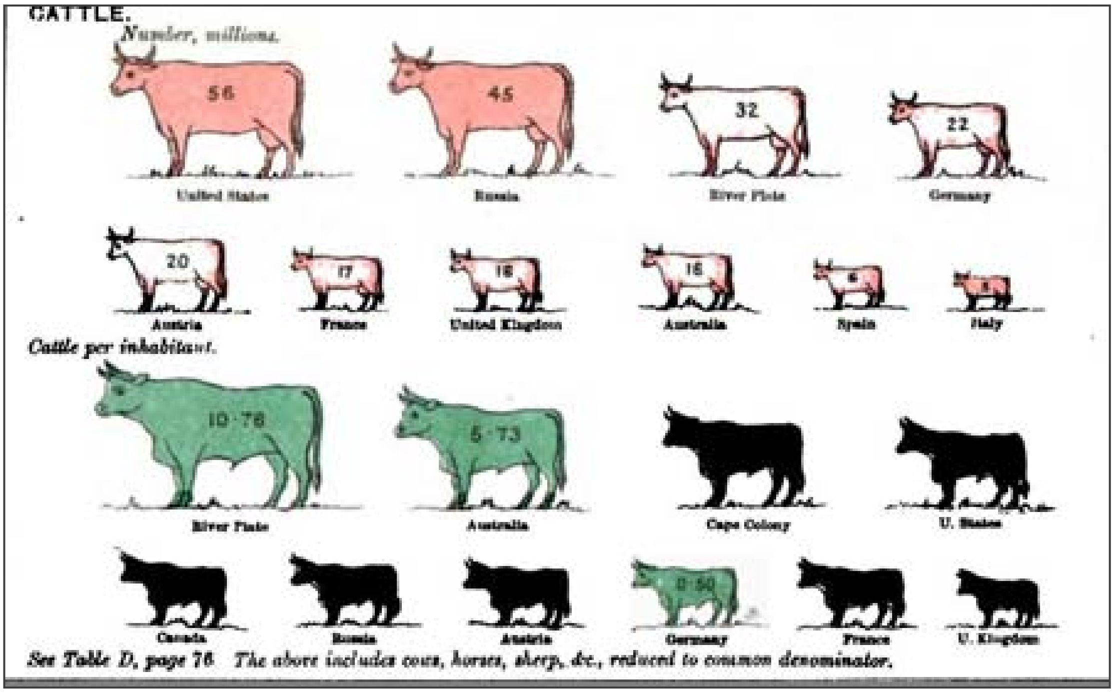


Figure 3. Pictograms (Mulhall, 2009)

Note that the quality of the graphic is poor in both originals since these are scanned versions of the original documents.

As an interesting example from the 20th century, in 1925 Walter Shewhart introduced the Statistical Control Chart for the control of industrial processes. The chart tracked significant indicators from the process against pre-defined warning and action limits to signal when corrective actions should be taken. The example in Figure 4 is from the UN Food and Agriculture Organisation guidelines (van Reeuwijk, 1998).

Perhaps the most significant addition in the control chart is the inclusion of context with the data. This echoes one of the primary principles of ecological interface design which is explained in section 3.5.3.1. The control chart plots the operational indicators against a background of pre-defined limits. If the actual measurement lies within all limits then the process is behaving acceptably. If the measurement lies between the upper warning limit (UWL) and the upper control limit (UCL), or if the measurement lies between the lower warning limit (LWL) and the lower control limit (LCL), then the process may be behaving abnormally so future measurements should be monitored closely and the process owner should be ready to take corrective action. If a reading falls above the UCL or below the LCL then the process is no longer behaving as required and correction action must be taken.

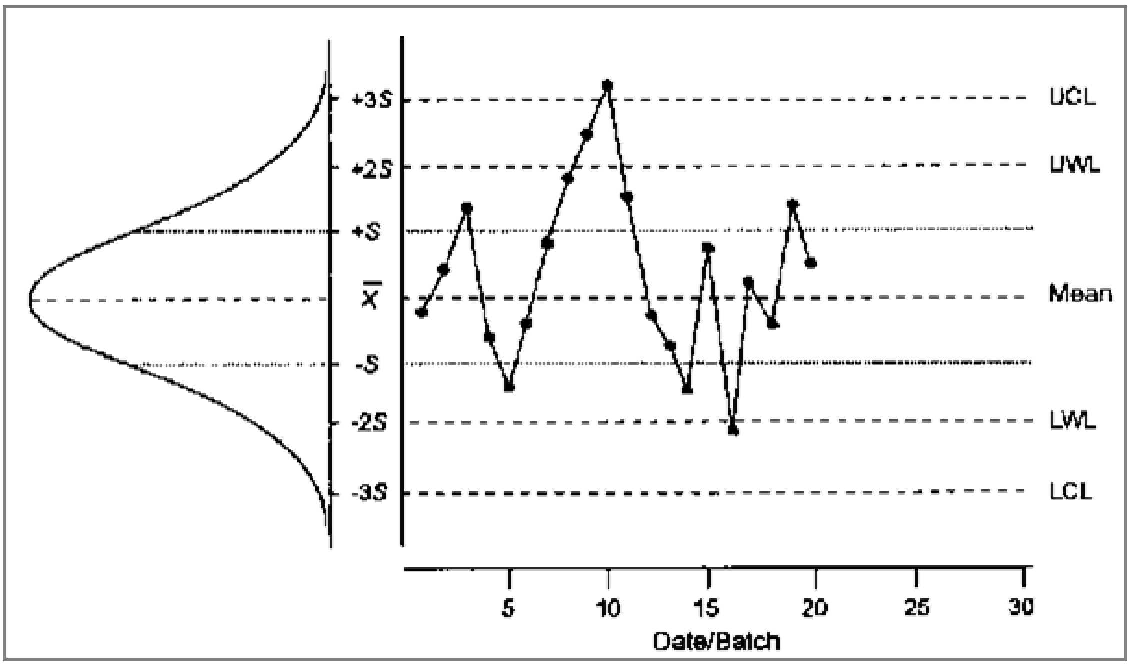


Figure 4. Statistical Control Chart (van Reeuwijk, 1998)

3.3 Communicating through Graphics

3.3.1 Development of Principles of Chart Design

Through the centuries of these developments there had been no formal theory of communication that practitioners could use to assess approaches and drive progress forward. In 1967 Jacques Bertin's *Sémiologie graphique* was published and this volume aimed to codify a 'grammar of graphics'. The 1983 translation made it accessible to the English-speaking readership and is referenced in this study. Bertin's (1983) addresses the topic of communication through graphics by considering the concrete aspects of the available 2D medium and then builds on that to derive a theory of how to maximise effective communication of data through graphical representations.

Spoken language permits humans to transmit symbols through the ear and it has 2 channels available to encode data – the audio and temporal. The individual sounds are significant and in addition the stream of sound is linear so that the sequence or timing of the sounds also conveys information. In contrast, Bertin's *Image Theory* contends that a graphic image has much higher bandwidth because it has three variables available to encode data. The three variables of the graphic image are the 2 axes of the plane and the representation of data at each point or region on the plane. The data can be represented by point, line or area markings using the 6 *retinal variables* of size, value, texture, colour, orientation and shape of the markings. Bertin (1983, p. 3) contends that writing and mathematics are special subsets of spoken language when considered as communications methods. These temporal communication methods can convey a sound at each instant of time; in contrast the image can convey the relationship among three variables across the full plane instantaneously.

A number of researchers have built on Bertin's theoretical approach to graphical representation to create a true grammar of graphics which could support the automated generation of graphical representations (graphics or visualisations) from data sets. Mackinlay (1986) did original work on representing relational data in his APT system. Cleveland (1993) expanded on Bertin's work adding knowledge of statistics and human perception to build a set of guidelines and tools most notably the Trellis display. The trellis display presented multi-dimensional data in a series of small charts shown in a grid on a panel (a page or computer screen). All charts on the panel have the same 'primary variables' assigned to the X and Y axes. Each chart shows a plot of the values of the primary variables for a combination of values of 'conditioning' variables. For example, if the data set contains height, weight, gender and age each chart could show a scatter-plot of height against weight for one combination of gender and age. Cleveland also recognised that human perception is best able to compare the slopes of lines when they are at an angle of approximately 45° and he recommended scaling data or manipulating the aspect ratio of charts to align regression lines appropriately in a procedure he called *banking to 45 degrees*. Cleveland believed that these techniques and tools supported the case for visualisation as a viable alternative to the use of statistical analysis to explore and understand data sets.

Wilkinson (2005) expanded on Bertin's work to create an algebra for the specification of graphic types with the aim of enabling users to easily create effective representations from data sets. A similar result has been achieved in the commercial software product *Tableau* which has features that allow users to create effective visual representations of data based on an algebraic specific language called VizQL (Mackinlay, Hanrahan, & Stolte, Show Me: automatic presentation for visual analysis, 2007).

3.3.2 Communication using Pictograms

It is important to recognise that effective communication is only possible if the transmitter and receiver are using an agreed meaning for each symbol that is used. Language, including mathematics, permits the transmission of precise information because the available vocabulary is large and well defined. Maps transmit information about features such as landscape contours, roads and water features because cartography has a well-developed standard symbology. The standard representations may be extended to permit a map to include novel features by providing a legend describing the symbols used. In contrast, figurative imagery such as photography, cinema and art can convey emotions and general meaning but not exact messages.

The evolution of pictograms can help in the design of icons in an applications user interface.

The work that has been done in the area of graphic communications owes much to the pioneering efforts of Otto Neurath, a sociologist and economist born in Austria. During his life he was committed to the ideal of education of people regardless of social class as a way to increase social stability. His motto was "words divide, pictures unite" and he relied on graphics to clearly communicate information to people regardless of their level of education. He founded a *Museum of Social and Economic Sciences* whose primary exhibits were displays of data presented through stylised graphics called Isotype (International System of Typographic Picture Education). The Isotype graphics were developed by Neurath and his colleague and illustrator Gerd Arntz and they are now recognised as the forerunners of today's pictograms. Figure 5 is part of a reference sheet from the Isotype 'Picture Dictionary' showing how separate symbols could be combined to provide more detailed information in the same way that an adjective modifies a noun (Burke, 2009). In this case the sacks are different sizes to represent volumes and the icons represent commodities or products.

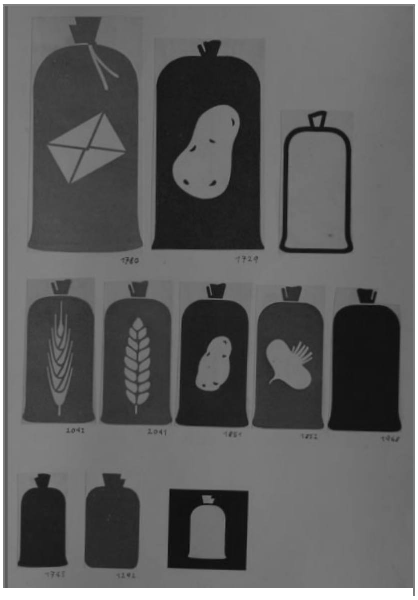


Figure 5. Isotype Graphics (Burke, 2009)

The most visible descendants of the Isotype symbols today are the stick figures and symbols used to indicate services at transportation hubs, large events and buildings. The 1972 Munich Olympics organising committee commissioned the graphic designer Otto Aichl to create a set of 180 pictograms representing the Olympic sports and the necessary supporting services to make it easy for the international athletes and press to navigate the city and the event locations. In 1974 the US Department of

Transport and the AIGA (American Institute of Graphic Art) created the first set of 34 passenger/pedestrian symbol signs. The set, which has been extended since then, is freely available for use and these symbol signs have now become a de-facto international standard (AIGA, 2011).

3.4 State of the Art in Chart Design

Bertin's (1983) *image theory* links attributes of data to the properties of graphic systems in order to define principles that can be used to create graphics which communicate information efficiently. He considered reading a graphic to be the acting of attempting to answer a question and said that reading can happen at the Elementary, Intermediate and Overall levels (Bertin, 1983, p. 140). By considering the graphic from the reader's point of view Bertin explains that a graphic can be designed to answer questions relating to the number of attributes represented in the graphic at the three levels of reading. In a healthcare setting extracting a patient's temperature at a certain point in time is elementary reading; understanding the fluctuations in a patient's temperature for the last 48 hours would be intermediate reading; overall reading would be attempting to understand this patient's temperature pattern against other patients with the same condition.

A motivated reader can extract much information from a graphic but the act of design makes the more relevant answers more accessible than others. In the Production Control scenario the chart will be designed to provide quick answers to the questions most frequently asked by the production supervisor. For example, *is the team's progress on track to have all jobs ready for dispatch at the scheduled time?* To design the chart we can learn from other charts that are optimised to answer pre-defined questions. In the late 20th century these were primarily control charts but advances in data processing technology have enabled the creation of more complex dashboards in the data-driven world of business.

Until recently dashboards have been associated with the periodic presentation of strategic business measures to senior management with a weekly or monthly frequency. Few is a leading supporter of the position that dashboards can add value at different levels in the workplace (Few, 2006).

In his *Information Dashboard Design* Few identifies this opportunity emerging due to the evolution and combination of technological facilitators such as high-resolution computer graphics, a growing understanding of how the power of visual processing can be harnessed for information acquisition and the emphasis on performance management and metrics (Few, 2006). Few proposes a taxonomy of dashboards and in one arrangement he separates them by function into strategic, operational and analytical categories. The graphical design of our process control solution can be guided in part by his principles for operational dashboards:

- Simplify and condense
- Maximise the data to ink ratio
- Overview first, zoom and filter, then details on demand
- Choose appropriate chart types

Simplify and condense

The first principle is to simplify and condense the data so that only the significant details are being communicated. These significant details will normally include the exceptions and so exceptions should be highlighted.

Maximise the data ink ratio

Few's next guideline reiterates Tufte's principle that the designer should "maximise that data ink ratio, within reason" (Tufte, 1983). That is, to consider all of the ink used to create the chart and ensure that most of the ink is being used to represent data. This is done by reducing and de-emphasising the amount of non-data content while making best use of the ink used to represent data. Reducing non-data ink is achieved by removing all decorations such as icons, shading, colour blocks and colour gradients that do not carry any meaning, 3-D effects, unnecessary grid-lines, etc. Some non-data ink to annotate the chart with headings, titles, legends, essential grid-lines and so on but these elements should be presented in muted colours so that they are available when needed for reference but they should not take attention away from the data. The designer enhances the data ink by removing unnecessary data from the chart and then representing the remaining data as points, lines or areas decorated using Bertin's retinal variables of location, size, value, texture, colour, orientation and shape. Data should be grouped so that it suggests relationships and encourages comparisons that are meaningful in the context.

Overview first, zoom and filter, then details on demand

Few states that the dashboard should not be static but should be a launch-pad for investigation of the underlying data. This echoes Shneiderman's (1996) visualisation information-seeking mantra of "Overview first, zoom and filter, then details on demand". The user should be presented with an overview of the data initially so that they can understand the big picture and context. Zooming and filtering allow the user to explore the data set, to ask and answer questions. Finally, giving access to the details permits the user to check hypotheses and to being formulating courses of action based on the results of their investigation.

Choose appropriate chart types

Regarding the choice of standard statistical chart types Few strongly encourages the use of sparklines, Pareto charts, box plots, scatter plots and tree maps. In *Information Dashboard Design* (2006, pp. 140-150) he presents his arguments and sets out the particular strengths of each.

Sparklines are useful to show the trend for measured values using a minimum of space. Figure 6 is an example showing the current value for several parameters with sparklines to the left showing the recent history. The normal range is indicated by the grey bar (Tufte, Beautiful Evidence, 2006).

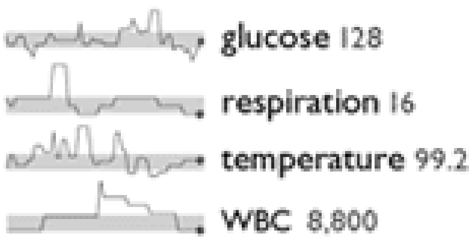


Figure 6. Sparklines (Tufte, 2006)

Pareto charts combine a bar chart and a cumulative line chart to highlight the most important items in a set of factors. The bars are shown in order of decreasing value to highlight the important factors. Figure 7 is an example with 5 bars showing sales value and a line indicating percentage of totals sales (Sookman, 2010).

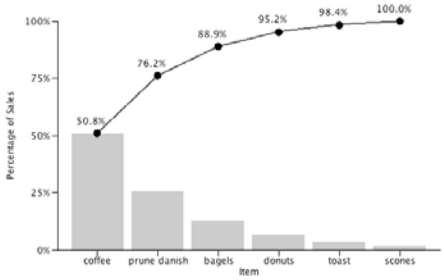


Figure 7. Pareto Chart (Sookman, 2010)

A *bullet chart* is a combination of a gauge and a bar-chart. It shows an actual value against a target and, optionally, against ratings scales to give context to the measurement. Figure 8 is an example from (Few, 2006)

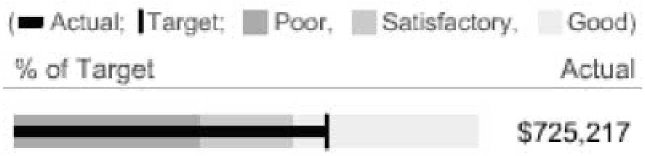


Figure 8. Bullet chart (Few, 2006)

A *box plot* is a chart which presents the distribution of a set of numbers by showing the median, the first and third quartiles and the minimum and maximum values. Outliers and suspected outliers are represented when they are present. Figure 9 shows an example from Kirkman (1996).

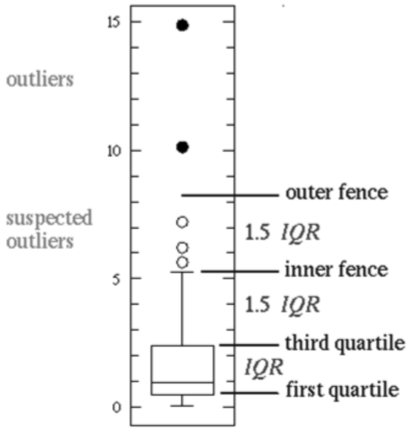


Figure 9. Box plot (Kirkman, 1996)

A *scatter plot* charts pairs of data values as points. This visualises the distribution of the values and clearly shows clusters and outliers. It also shows the correlation between the independent variable on the x-axis and the dependent variable on the y-axis. Figure 10 shows a plot of suicide rate (SRR) against deprivation index (LC) for the boroughs of London (Rezaeian, Dunn, St Leger, & Appleby, 2007).

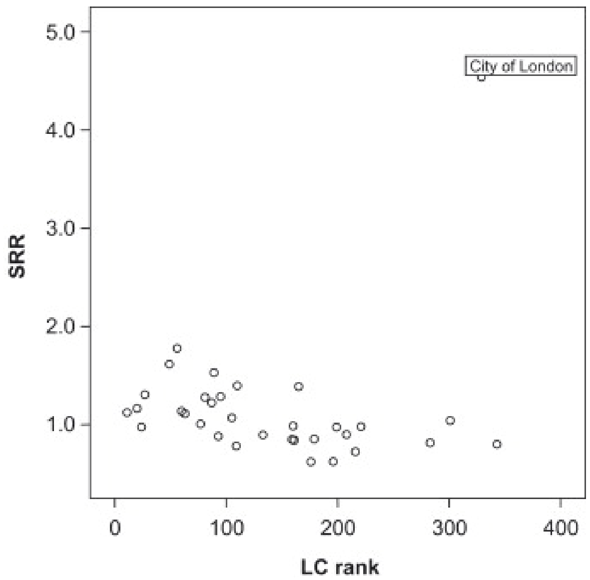


Figure 10. Scatter plot (Rezaeian, 2007)

Tree maps are used to represent hierarchical data by allocating regions of a rectangular area to elements of a data collection. The area of a section represents an appropriate measure of the element, e.g. cost. Colour may be used to represent a second attribute. Figure 11 is an example from (Shneiderman & Wattenberg (2001).

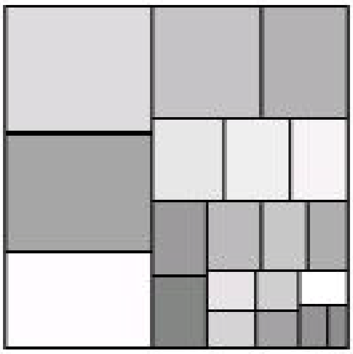


Figure 11. Tree map (Shneiderman 2001)

In addition to these positive recommendations Few advises that designers avoid the use of pie graphs and radar graphs as both are less effective substitutes for bar graphs.

3.5 Interface Design Methods

We have reviewed the history and current principles of creating graphics to communicate data effectively. In the next section we present the current research on tools and techniques to capture user requirements and derive a corresponding interface design.

3.5.1 Interaction Design

Sharp, Rogers and Preece have described interaction design (ID) as being a broader discipline than human-computer interaction (HCI) (Sharp, Rogers, & Preece, 2007). HCI is "*concerned with the design, evaluation, and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them*" (Hewett, et al., 1992). Interaction design has a wider focus "*concerned with the theory, research, and practice of designing user experiences for all manner of technologies, systems, and products*" (Sharp, Rogers, & Preece, 2007, p. 10). While ID has a scope that goes beyond computer systems it describes a process which can help designers of interactive computer applications create an improved user experience for their users.

The interaction design process consists of four key activities. First the designer must understand the user needs and identify the requirements. Understanding the user characteristics, expectations and abilities is essential for a successful design. Second, the designer takes the knowledge about the users and the requirements and creates a number of alternative designs. Third, the designer creates interactive versions, or prototypes, of the designs so that they can be evaluated and assessed with the users. The fourth activity is the formal evaluation of the designs ensuring that the design for the final product meets the user requirements.

Figure 12 shows how the four activities relate to each other in an iterative process (Sharp, Rogers, & Preece, 2007, p. 448). This is not the only possible ID lifecycle but it clearly shows the process as iterative with multiple links between then activities. It also shows that the evaluation step has three possible outcomes – (i) ideally it leads to a final product but it can also lead to (ii) a redesign based on the requirements that are already identified or (iii) the evaluation may identify new requirements and so bring the cycle back to the first activity. The evaluation is the formal measurement that the design meets the user requirements but user feedback is solicited during the design and when building the interactive models. The constant feedback serves to minimise miscommunications and gives the users a sense of ownership of the final product.

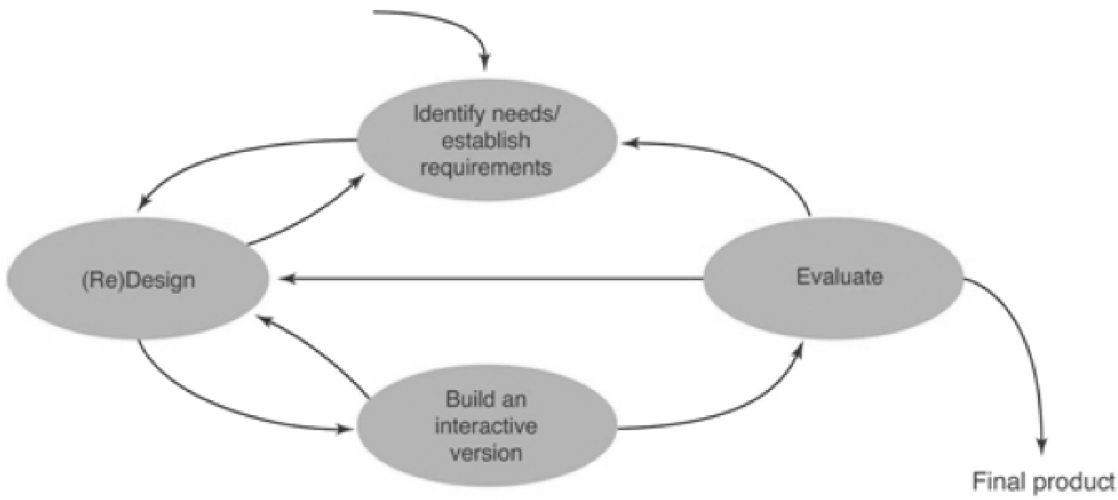


Figure 12. Simple Interaction Design Lifecycle Model (Sharp Rogers & Preece, 2007)

In this study the users are a clearly identified group of shift supervisors. One of the weaknesses of the study is that the pool of users was very limited and so it was expanded by including business analysts with a strong knowledge of the work environment. This was necessary to ensure access to knowledgeable people when needed during the project. The potential designs were created on paper initially to explore design ideas. The interactive versions were created using software design tools that supported user interface prototyping with sample data (Microsoft Expression Blend and Sketchflow). The Microsoft tools were chosen because they fit well with the other tools being used by the development team. This integration allows the software developers to use the prototype screens directly in the programming toolset and so leverage the work has been done already. When prototyping tools are not well integrated the prototypes are often discarded when development begins.

3.5.2 Requirements Capture and System Design

When dealing with software applications one of the key challenges is to correctly identify and define the user requirements. A number of techniques have been developed to identify the underlying work and the information requirements needed to support complex cognitive tasks. When Crystal and Ellington (2004) reviewed the state of task analysis they concluded that "*Task analyses are too difficult to perform, and when they are performed they are too difficult to understand and use*" and the authors point to some avenues for possible progress. Considering the available techniques today the Cognitive Work Analysis (CWA) and Hierarchical Task Analysis (HTA) are among the most commonly used.

These techniques have also been the subject of several studies comparing their strengths and effectiveness and the consistent result is that they are complementary and combining them delivers better outcomes than relying on either technique in isolation (Miller & Vicente, 2001) (Jamieson, Miller, Ho, & Vicente, 2007) (Salmon, Jenkins, Stanton, & Walker, 2010).

3.5.2.1 Cognitive Work Analysis (CWA)

Cognitive Work Analysis (CWA) is a framework that seeks to create a formative description of a system by identifying the boundaries and constraints of the system, the repeating activities, decision points, social organisation of the agents and worker competency requirements. CWA concerns itself with the high-level purpose of the system and its model incorporates the physical and social constraints that apply to the system. As such CWA produces descriptions of systems that are designed to allow their users to adapt to new or unfamiliar situations (Vicente, 1999).

Figure 13 shows the 5 phases of CWA with the key outputs coming from each phase (Jenkins, Stanton, Walker, Salmon, & Young, 2006, p. 5). The framework is comprehensive but each of the levels requires a different complementary set of techniques and so a full analysis is very time-consuming. Jenkins et al contend that for this reason most documented case studies have confined themselves to the first one or two phases and consequently when people talk about CWA in practice they are often referring to Work Domain Analysis or its key output, the Abstraction Hierarchy (AH) or Abstraction-Decomposition Space (Jenkins, Stanton, Walker, Salmon, & Young, 2006).

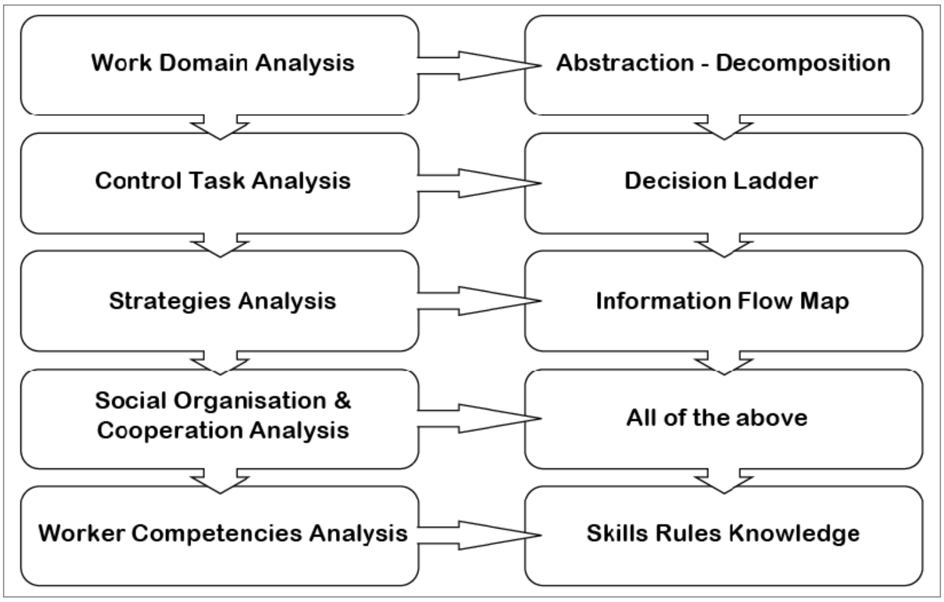


Figure 13. The 5 phases of CWA (Jenkins 2006)

The CWA does not consider goals; it considers the purpose of the system at the top level and then relationship between the different levels of the system. Jenkins et al. (2006) have summarised the original labels assigned to the levels by Rasmussen and a more recent revision of the titles, see

Table 1. The AH is a hierarchical means-end decomposition and so the items at the lower level provide the means for the higher level items. The Description column indicates the type of information that should be captured at each level.

Table 1. AH 5 Levels of Abstraction (Jenkins, 2006)

"Old Labels"	"New Labels"	Description
Functional Purposes	Functional Purposes	The purposes of the work system and the external constraints on its operation
Abstract Functions	Values and Priority Measures	The criteria that the work system uses for measuring its progress towards the functional Purposes
Generalised Functions	Purpose-related Functions	The general functions of the work system that are necessary for achieving the functional purposes
Physical Functions	Object-related Processes	The functional capabilities and limitations of physical objects in the work system that enable the purpose-related functions
Physical Form	Physical Objects	The physical objects in the work system that afford the object related processes

3.5.2.2 Hierarchical Task Analysis (HTA)

Hierarchical Task Analysis (HTA) is a task analysis technique which analyses an existing process to create a description of the goals to be performed through a cascade of sub-goals, operations and plans. The 'task' in the title is a misnomer that may indicate HTA's long pedigree. HTA was developed in the 1960s at a time when it became important to properly understand the more complex cognitive work that was emerging in the workplace (Annett, 2004). HTA went beyond analysing the physical activity and included the goals and the related cognitive aspects of the work. HTA begins with a user goal and proceeds by identifying tasks that must be completed to satisfy the goal. Analysing the tasks that are identified may result in sub-goals or sub-tasks being identified and these must be analysed in turn. As sub-tasks are identified they are documented in plans which describe the trigger for the plan and the rules describing the execution of the sub-tasks. The sub-goals and sub-tasks are analysed and the process continues recursively until all sub-goals, operations and plans have been identified. The data collection can be done in many ways – user interview, observation, cognitive walkthrough, reviewing documentation, etc. The main weakness of HTA is that its results are based on the existing process (Stanton, 2006). While HTA can identify areas for improvement it does not generative in the same way as CWA. In addition, Sharp, Rogers and Preece (2007) contend that HTA is limited in that it cannot effectively model parallel or overlapping tasks and that it is unable to model interruptions which are a common feature of cognitive work. Despite these limitations HTA is still the most widely used analysis technique (Sharp, Rogers, & Preece, 2007, p. 515).

HTA is an approach to task analysis rather than a prescriptive modelling framework and so its practitioners can, and have, proposed many formats to present the goals, operations and plans that its primary outputs. Stanton (2006) presents three formats: hierarchical diagrams, hierarchical lists and tabular. HTA is based on decomposition of goals into sub-goals with their related plans and operations and so the hierarchical representations are a natural fit. However the hierarchical representations can be unwieldy for larger analyses; the tabular format is more compact and has the additional advantage that additional information can be added to the goals such as frequency, duration or likely sources of error for example.

The use of HTA as a requirements elicitation tool for a Hospital Information System has been demonstrated successfully in the requirements engineering phase of a software development project (Teixeira, Ferreira, & Sousa Santos, 2010). The authors used HTA with object-oriented systems analysis and prototyping to design a system to manage clinical data in a haemophilia setting.

3.5.2.3 Combining CWA and HTA

Several authors recommend using both techniques to generate an analysis that benefits from the strengths of both (Miller & Vicente, 2001) (Jamieson, Miller, Ho, & Vicente, 2007) (Salmon, Jenkins, Stanton, & Walker, 2010). When the two techniques are combined in a project CWA can be more suited to the early design activities that identify requirements. HTA is more suited to the analysis and refinement of an existing process or application and so it can used to assess the resulting designs and applications. (Salmon, Jenkins, Stanton, & Walker, 2010, p. 526).

In addition, HTA is focussed on the normal expected work pattern whereas CWA can help produce designs that effectively support the operator when unexpected situations arise. Using both forms of analysis identifies a more comprehensive set of requirements than using either method alone.

Based on the literature review and the specific problem selected for this study the decision was taken to perform a hierarchical task analysis and to generate the abstraction hierarchy (AH) from the cognitive work analysis. The HTA has a lot to offer in this study - the subject is the redesign of an existing artefact and so we can and should learn from the as-is situation. However we want to explore new possibilities rather than being guided towards incremental changes to the current design and so we should take advantage of the generative capabilities of CWA.

3.5.3 Application and Interface Design

3.5.3.1 Ecological Interface Design

Vicente and Rasmussen (1992) proposed Ecological Interface Design (EID) as a framework to aid in interface design in complex work domains. The creation of EID was motivated by the desire to enable system operators to cope with errors and unexpected events especially in complex control tasks. A system created through EID will have a user interface that incorporates the internal constraints in the system model and also represents the relationships between the data elements used to display its internal state.

EID proceeds from a domain model that generates an Abstraction Hierarchy (AH). Vicente (2002, p. 63) contrasts AH with a task analysis which can only describe goals and tasks that have been included in the analysis by the analysts. An interface based on task analysis can only support the user through activities foreseen by the designers whereas an interface based on EID also supports the users when unexpected conditions arise.

EID is grounded in Rasmussen's (1983) model which classifies human behaviour into categories of skills-based behaviour, rules-based behaviour and knowledge-based behaviour. Skills-based behaviour (SBB) is automatic or unconscious activity at the sensory-motor level which takes place without conscious attention or control. Rules-based behaviour (RBB) is a goal-oriented activity where actions are executed from sets of learned skills following some rule. A rule is selected and fired consciously or unconsciously in response to an external stimulus; if the selection is unconscious then the goal may be implicit rather than explicit. The rule itself may have been pre-defined as part of a procedure or it may have been discovered and learned by the individual in previous situations. The SBB and RBB can overlap depending on an individual's experience - the main difference is that SBB happens without conscious control and the person may not even be able to explain what they are doing whereas in RBB there is some conscious input and the person can normally explain their actions. When SBB and RBB behaviours are not sufficient to meet the needs of a situation then the person moves to Knowledge-based behaviour (KBB). This entails conscious assessment of the environment and the generation and selection of plans of action that will help achieve certain goals.

Significantly for computer system design, when a user is faced with an unfamiliar situation and moves to KBB then their plans will depend on their internal model of the computer system as they try to predict how the system will respond to the actions proposed in their plans. To give the user the best possible chance of identifying a good solution it is important that the user has an accurate model of the computer system and in addition the system should expose as much detail about its internal state as possible.

EID has a number of principles of good design to guide the designer in creating an effective interface that aims to support behaviour of a specific type:

- KBB is supported where the system provides an external model of its design by representing the functional hierarchy of its state in the user interface. In this case the user draws on their knowledge to supplement the information being presented by the system

- RBB is supported where the interface maintains a 1:1 mapping between elements of the interface and the physical or social constraints that are modelled in the system. The user can more easily choose the appropriate course of action by reacting to the information presented
- SBB is supported where data from the system is presented logically in the interface and direct manipulation is supported where feasible. If the system supports a direct manipulation interface the user relies on lower-level skills to interact with the system by choosing from the limited set of appropriate actions presented by the system

The EID design principles influence interface design by seeking to push tasks to a 'lower' level of behaviour. For example, presenting raw data on the workload remaining for the current shift requires a supervisor to incorporate knowledge about capacity and time remaining to understand if the shift is on target to meet its goals – this is knowledge-based behaviour. If the system integrates the capacity constraint, with tolerances, alongside the remaining workload then the supervisor exercises rules-based behaviour by comparing the numbers. If the interface represents the remaining workload and remaining capacity graphically then the supervisor can understand whether the goals are achievable by visual comparison which is a skills-based behaviour.

EID has its roots in the attempt to improve control systems for complex systems such as nuclear power plants and chemical processing plants and some authors have identified weaknesses in the because it is so strongly linked to these industrial origins. For example, Moise and Robert propose replacing the AH by their Constraints Model (2007). Given this discussion it is fair to question if EID can be usefully employed outside of its original domain of large-scale complex industrial systems. Upton and Doherty (2008) have presented a study where EID was extended to the creation of a process monitoring system rather than a process control system. The authors addressed the switch from hard process control to the management of information systems by supplementing the work domain with task modelling. Separately, in a health-care study Sharp and Helmicki (1998) used EID in the design of an interface to monitor patient status in a neonatal intensive care setting. The authors modified the abstraction hierarchy to their domain by using it as a tool to identify a hierarchy of concepts that physicians use during diagnosis and used this to drive their decisions on data presentation. The authors also found that sensors play a different role in healthcare monitoring compared to industrial monitoring. When monitoring patients the need for patient comfort and dignity limit the number of sensors that may be used and the locations where the sensors may be placed. In addition limitations on sensor technology can restrict the data that may be collected. These limitations mean that data identified as important in the AH must be inferred from the limited set of sensors that is available. In both cases the studies produced results that were superior to traditional interface designs and so we can conclude that it is valid to use EID outside of its original domain.

3.5.3.2 User Interface Design

The graphical presentation of data to the users is one feature of user interface design that is appropriate in some circumstances. However, it is important that this opportunity does not blind the designer to the important lessons that have been learned about the more general issue of interface design. The principles of graphical design sit within the large body of pre-existing literature and guidance on user interface design and user experience design as set out by Norman in *The Psychology of Everyday Things* (1988), Nielsen in *Usability Engineering* (1994), Shneiderman et al. in *Designing the User Interface* (2009), and others. For example, Shneiderman's eight principles of interface design:

- Consistency – the system should support the same actions through the same command sequences where appropriate. The same terminology should be used on all dialogs, menus, documentation etc.
- Shortcuts – the system should present shortcuts so that experienced users can access commands more quickly. Where appropriate the shortcuts should be consistent with the larger software ecosystem, e.g. a manufacturer's style guide

- Feedback – the user should receive feedback for all actions taken in the systems
- Closure – sequences of actions should have a beginning, a middle and an end; it should be clear when the end is reached and what result has been achieved. This allows the user to clear that task from their mind and move ahead to the next task without any doubts or concerns
- Error handling – the system should prevent errors as much as possible and where errors do occur feedback should be clear and include guidelines on how to correct the error
- Undo – providing a way to reverse actions give the user the freedom to experiment and so it encourages the user to use the full functionality of the system
- Control – experienced users want to feel that they are in charge and that the system responds to actions or requests that they initiate
- Reduce Short-term Memory Load – the system should let the user complete tasks without having to gather details from a number of dialogs or navigate through many screens

3.5.4 Test Data Management

The study will progress by eliciting user requirements and building prototype user interfaces and data representations to support the users' in their daily work. The final deliverable is effectively an enhancement to an existing computer system and so there is a production database available for reference. It has been proposed that where possible the prototype database should be built by sampling the production database because the resulting prototype database "*with domain-relevant data values and semantics, is expected to better support the software development process*". (Bisbal, Grimson, & Bell, 2005) Their *Consistent Database Sampling* technique results in an internally consistent and complete test data-set in the prototype database. Data privacy is a major consideration when working with healthcare data and so details that could lead to identification of individual patients or clinicians are set to synthetic values as part of the database sampling process. This is in line with current best practices around data privacy and data management.

3.6 Selected Tools

The study will proceed following the principles of interaction design. The analysis phase will use hierarchical task analysis and the abstraction hierarchy of cognitive work analysis to identify requirements. The design activities will follow the ecological interface design methodology and the general user interface design principles espoused by Shneiderman, Nielsen and Norman. The design phase will also incorporate the guidelines for representing data graphically proposed through the work of Bertin, Tufte and Few.

4 Methodology

The study was executed as a form of participatory action research. The primary methodology used was Interaction Design with its four activities of identifying the users and the requirements, creating alternative designs, building interactive models of the design and evaluating them with the users (Sharp, Rogers, & Preece, 2007).

Requirements elicitation was performed through interviews and observation. The process was guided and the results were documented using the Hierarchical Task Analysis (Annett, 2004) and Cognitive Work Analysis (Vicente, 1999) methods.

After the requirements were identified potential designs were generated following the Ecological Interface Design (EID) methodology (Vicente & Rasmussen, 1992). While EID provides general direction about the content of the user interface the specific details of the screen layout and graphical elements is based on the principles of user interface design (Norman, 1988), (Nielsen, 1994), (Shneiderman, Plaisant, Cohen, & Jacobs, 2009) and graphical design (Few, 2006), (Tufte, 1983), (Bertin, 1983). The initial proposals were created on paper and reviewed with the users. The most promising candidate designs were then built as interactive models using appropriate software tools. The prototypes included test data which was extract from the live environment. Great care had to be taken in managing the test data because of the data privacy and confidentiality considerations when working with personally identifiable medical records.

The evaluation of the final design with the users was performed based on an interactive prototype using cognitive walkthroughs and interviews driven by scenarios. The software application is used in a strictly controlled production environment and so it was not possible to complete programming, testing and formal validation of the final design and have it promoted to the live environment in the time available.

5 Analysis and Design Phases

5.1 Eliciting Requirements

5.1.1 Hierarchical Task Analysis

As noted in section 2.4 the existing Web Planner module is used by the supervisor to gain a high-level view of the jobs being produced during a shift. The module was first designed more than 5 years ago and the business has changed significantly since then. The business has expanded to include new product families and customers' expectations have increased in terms of the ability to request urgent deliveries. As the business has grown the size of the units and there teams have become larger. This has lead to a number of informal requests for the module to be expanded and updated. For example senior management believe that the supervisors and their managers need to have a more comprehensive view of activity at the site. The supervisors want all product families to be visible through the module.

Rather than rely on these ad hoc statements of requirements the study incorporated a hierarchical task analysis (HTA) to ensure that the opportunities for improvement were correctly identified. One weakness of HTA is that its results can strongly reflect the as-is situation and the analyst must be aware of the need to identify opportunities that are not considered to be in scope by the users. The initial informal feedback was used to guide the interview process to ensure that the broader requirements were being considered.

During the HTA data was collected through user interviews, observation and cognitive walkthroughs. The observation and cognitive walkthrough provided the opportunity to incorporate goals or tasks being performed by the supervisors which the supervisor may not have considered as part of the Web Planner scope.

There are a number of assumptions and scope definitions supporting the task analysis. The HTA deals with the operation of a single shift. Management tasks such as scheduling staff training or longer-term resource planning to take account of business growth are not concerns at this level. Material planning and resource scheduling for future shifts happen in parallel and are not considered in this HTA. Manufacturing efficiency is handled such as minimising waste by combining jobs where possible are taken care of in the production system logic.

Table 5 (p. 74) shows the key high-level goals and sub-goals identified. The Information Requirements column identifies features or tasks that drove decisions in the interface design process.

5.1.2 Abstraction Hierarchy

Table 6 (p. 78) shows the resulting AH. The AH has made a number of constraints explicit. Since this is a healthcare domain there is a clear constraint all products must be fit for use when they are delivered to the customer. The system calculates the expiry date/time by adding the shelf life of the compound to the production date/time. The product is considered fit for use if the expiry date/time is less than the required date/time plus a pre-defined grace period. The grace period represents the time that can elapse at the treatment centre between them receiving the delivery and them using the compound.

The delivery schedule is a constraint on the system's ability to meet the purposes of maximising efficiency and including new jobs in the schedule. Jobs delivered on the scheduled deliveries are cheaper than unplanned deliveries. A new job may be received too late to produce it and include it on a scheduled delivery in time to meet the customer's Required By deadline. In this case the Supervisor may negotiate with the customer to see if the Required By deadline is flexible; if not, then and additional delivery must be scheduled for this job and cost-sharing must be agreed with the customer.

The overarching economic requirement is that all opportunities should be taken to minimise costs. For example, unused resources should be reallocated to other work where possible; urgent and unplanned deliveries should be avoided by negotiating the delivery schedule where appropriate; where demand in the next days is increasing jobs should be produced if possible without incurring overtime, etc.

5.1.3 Validating the Outputs

The HTA and the CWA/AH provide two different views of the same process. It is possible to cross-reference the purposes and constraints identified in the AH against the operations and plans in the HTA at a high-level to satisfy ourselves that the analysis phase has generated consistent outputs. Table 2 (p. 45) is a cross-reference showing the mutual support between the two outputs. Table 2 contains eight rows, one for each of the five purposes and the three constraints identified in the AH extract (Table 6, p. 78). The first column contains the purpose or constraint and the second column contains the corresponding components identified in the HTA extract (Table 5, p. 74). The third column contains additional comments.

This consistency demonstrated in this case is achieved because the design is a new revision of an existing solution and the output of the HTA is grounded in the as-is situation. The consistency gives the analyst confidence that the purposes and constraints in the AH combined with the checks and actions in the HTA reflect the business requirements. The designer can then use tools from either technique to help generate a concrete interface from the requirements.

5.2 Designing the Proposed Solution

The previous phase identified system requirements at an abstract level. To deliver an interface that meets those requirements the designer followed a three-step process. First, the functional requirements were identified by parsing the output of the HTA to generate a list of required functions along with the data required to support those functions. Table 7 (p.80) presents an extract that will be used to explain the process followed to generate the interface.

When these more detailed requirements had been defined the EID principles were used to propose effective display elements combining status information with the relevant constraints and targets. Finally the principles of good interaction design were applied to address layout and visibility functions, interface behaviour, consistency with other applications, etc.

Table 2. Comparison of output from HTA and AH

Abstraction Hierarchy	Hierarchical Task Analysis	Comments
Purpose: On-time delivery of jobs	2.3: Monitor progress of production 1.4: Balance the resources, workload and delivery schedule 1.4.1: Check if the schedule is balanced 1.4.3: Negotiate revised delivery with the customer 1.4.4: Arrange special delivery	The HTA addresses on-time delivery several times because it is such a fundamental purpose/ requirement
Purpose: Respect product shelf-life	1.4.5.1: Identify jobs available for release to production (Not shown)	(1.4.5.1 is not shown in the extract in Table 5)
Purpose: Include new jobs in the schedule	Plan 2: If [new jobs are received] do 2.1 2.1: Add new jobs to the schedule	
Purpose: Maximise efficiency	1.4: Balance the resources, workload and delivery schedule 1.4.2: Adjust the available resources up or down 2.2: Respond to resource change	
Purpose: Manufacture drugs	Outside of scope of the monitor and control tasks	Basic deliverable of a manufacturing unit
Constraint: Delivery schedule	1.4.3: Negotiate revised delivery with the customer 1.4.4: Arrange special delivery	The HTA components indicate potential solutions if a constraint limits the ability to achieve a purpose
Constraint: Shelf-life	1.4.5.1: Identify jobs available for release to production (Not shown)	The HTA has a lower level component to ensure the shelf-life requirement is being met (not shown)
Constraint: Throughput	1.4: Balance the resources, workload and delivery schedule 2.2: Respond to resource change	The supervisor considers throughput when balancing the schedule at the start of the shift and when resources or workload change

Turning a set of requirements into an effective design is still more art than science. In this section we consider the selection of two significant elements of the final design. A graphic based on the *bullet chart* was chosen to show the progress of production during the shift and a *heat map* was used to enable the supervisor to see and manage the overall state of the delivery schedule.

5.2.1 Key Elements of the New Design

Figure 14 shows a prototype layout of the new design. It is a single screen representation of all functions required by the shift supervisor. The screen contents are laid out in two rows.

The top row presents three elements to the user.

The left-most element is the *Monitor/Resources* panel which is a multi-tab display; at any time either the Monitor panel or the Resources panel is visible while the other is concealed. The Monitor panel provides an overall picture of production during the shift and the balance between the workload and the capacity of the team. The Monitor panel is presented in more detail in section 5.2.2 (p. 48). This is a key feature of the display and so it is placed in the top left corner of the display for prominence. The Resources panel is used at the start of the shift to confirm the available resources and it may also be accessed during the shift if the supervisor deems it necessary to change the available resources. Since the Resources panel is used only occasionally it is normally concealed to minimise the screen space used while ensuring that the user is aware that the function is available.

In the centre right of the top row we have retained the BEQ Totals as shown in Block C of Figure 1 (p. 13). This feature was carried over from the existing design to retain continuity and to provide quantified measures of the progress presented graphically in the Monitor panel.

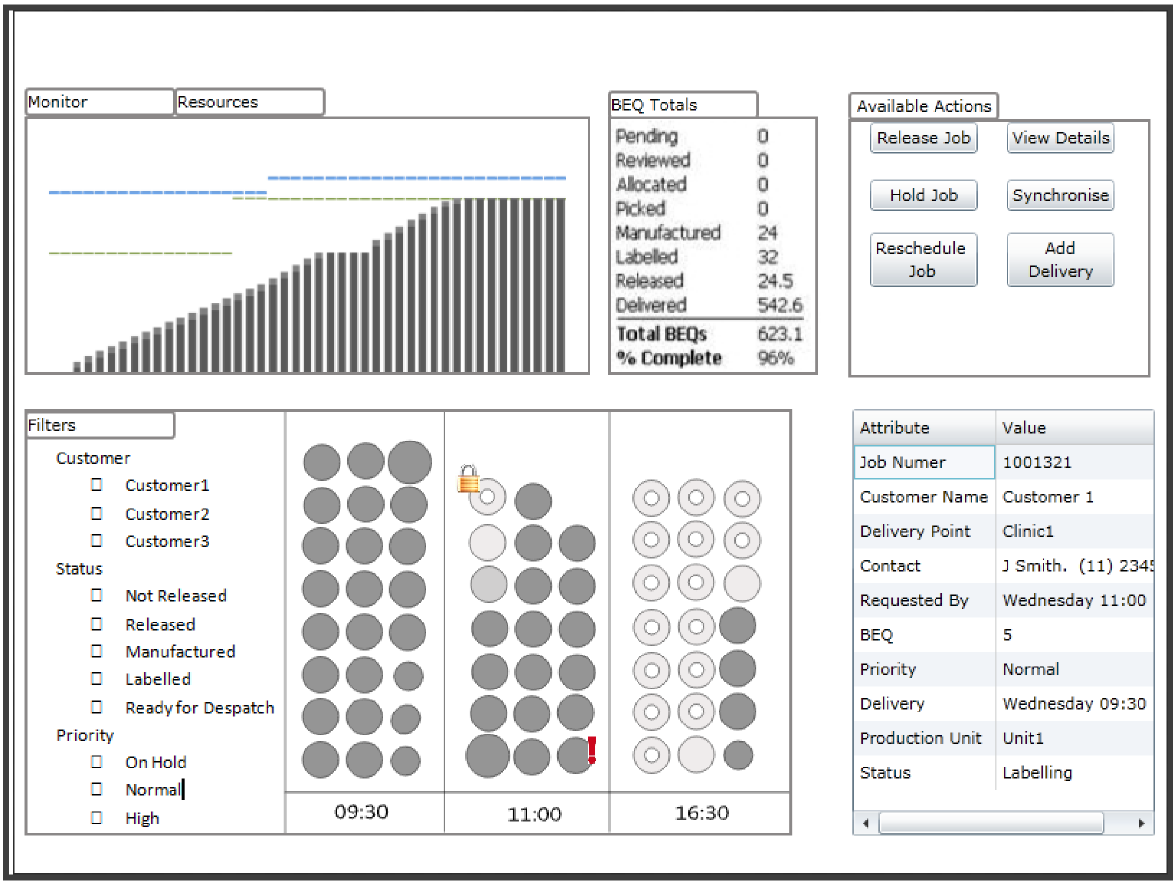


Figure 14. Layout of Final Design

The final element on the top row is a panel with buttons to access the actions that are available to the user. The aim to keep the full list of options visible to act as a reminder of all functions available to help manage the process. Only subset of the actions will be active and available at any time depending on which element on the screen is selected. This presentation satisfies a number of principles of good user interface design by providing feedback when an element is selected on the display, by making the available options clearly available which reduces the load on the user's memory, and helping the user to feel that he/she is in control of the application.

The second row of the display is dedicated to a representing the deliveries and jobs scheduled for this shift. There is a heat map in the centre of the row and it has two supporting panels on the left and right hand sides. The heat map is a variation on the pixel bar chart (Keim, Hao, Dayal, & Hsu, 2001); the data of interest, in this case the jobs, are represented by icons rather than individual pixels. The heat map itself is broken into rectangular subsections representing the scheduled deliveries. Each job scheduled in a delivery is represented by an icon and the shape and shading of the icon present information about the job – see Table 3 (p. 51) for a description of the information coded in the icons. Using this layout the heat map presents a picture of the scheduled deliveries and for each job scheduled in a delivery it indicates the size of the job and its progress through the manufacturing process.

The user interface elements are standard Microsoft Windows controls and so the environment presents familiar interaction patterns to experienced Windows users. This fits the profile of our target user population.

5.2.2 Monitoring the Progress of Production

The *Monitor* panel was created to monitor the progress of production as identified in component 2.3 of the Hierarchical Task Analysis.

The existing Web Planner design (see Figure 1, p. 13) presents a summary of the BEQs by their state which indicates which jobs are entering production, ready for labelling, ready for despatch and so on. The 'flight board' display also shows the list of jobs. An experienced supervisor can combine this data with knowledge about the resources available, the time of day and the cut-off times for despatch to understand if the target production can be met. This is clearly knowledge-based behaviour.

Following the principles of ecological interface design we aim to include the constraints and targets with the point-in-time status of the production. The status information displayed in the *Monitor* panel is the cumulative production of jobs and this was chosen because that is the key deliverable of the production process. Following the EID principle of combining status and context information the production target and capacity constraints are added to the view.

It is also useful to include a representation of where the shift is in terms of time elapsed and remaining so that the supervisor can understand the options that may be available to respond to unexpected events. Figure 15 is an example of the resulting design for the *Monitor* panel.

In order to represent an actual value against a target we have selected the bullet chart. Few (2006) introduced the bullet chart as a display element which can show a variable against one or more background measurements. See Figure 8 (p. 27) for a fuller description. The Monitor panel uses a vertical arrangement of bullet charts to show the actual output against the target (Line 1) and capacity (Line 2).

The full horizontal width of the panel represents the duration of the shift. Each column is a bullet chart representing the status and status updates are provided every 15 minutes by adding a column to the chart. The way that the panel is constructed means that the new update will appear as a new column to the right of existing entries.

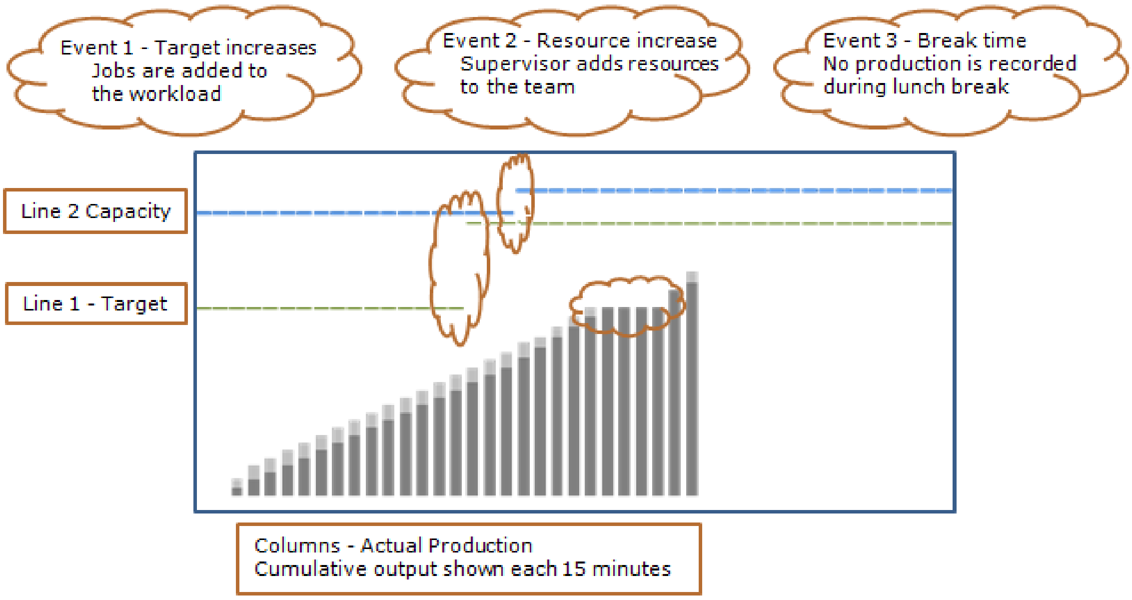


Figure 15. Monitor panel

This example has been annotated as follows - the rectangular boxes contain descriptions of the screen elements and the 'clouds' indicate events that are made visible through the display.

The dark segment of the bullet chart represents those jobs that are ready for despatch; the light segment represents jobs where production is complete but labelling is not. Labelling is a quick and low risk operation and so it is useful for the supervisor to consider those jobs as almost complete.

In this example we can see that initially the capacity was comfortably above the target production for the shift. Event 1 (the left-most cloud) shows that a group of jobs was added to the production schedule which brought the target close to the capacity. Event 2 (the central cloud) shows that additional resources have been added to increase the capacity in response to the increased workload. The increased resources are also visible as a slightly increased rate of production following the addition of the resources. Event 3 (the right-most cloud) shows that no output is produced during the team's break.

5.2.3 Available Actions

Creating an interface that supports direct manipulation interaction is necessary if the system is to support skills-based behaviour in experienced users. Direct manipulation is best achieved where the user is using controls such as sliders and drag-and-drop handling to manipulate objects that represent domain objects. However, it takes time to develop skills using a software application and so the interface should allow novice users to identify the options and features that are available at any point in time.

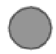
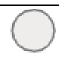
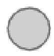
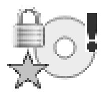
The new design provides visibility to the available options by presenting all options as buttons in a dedicated area of the display which is referred to as the Available Actions panel. When the user selects an object in the display the buttons for the relevant actions are enabled while the buttons for the actions that do not apply to the selected object are disabled. In this way if the user is unsure about actions they know where they can use the panel as a guide.

In addition to the Available Actions panel the system supports context-sensitive menus accessed by right-clicking on each object in the display. The context-sensitive menu provides experienced MS Windows users with a familiar method to access the functions of the system. Context-sensitive menus have the additional benefit that the user can access a function without taking their attention away from the object they are focused on while they search for and activate a menu option or a button.

5.2.4 Controlling the Delivery Schedule

The scheduled deliveries and jobs are represented using the heat map and supporting panels on the second row of the display. The individual deliveries are represented as rectangular boxes containing icons which represent the jobs scheduled for that delivery. Selected information about the current state of job is indicated visually in the icon as described in Table 3.

Table 3. Coding of Icons Representing Jobs

Attribute	**Meaning**	**Values**
Size	Effort for this jobs i.e. BEQ rating	Relative size – small, medium and large 
Shape	Released for manufacture or not released for manufacture	Doughnut – not released  Disc – released 
Shading	Lighter shades indicate early steps; darker shades show progress towards completion	35% Released  65% Manufactured  90% Ready for despatch 
Modifier	Additional icons carrying more information about a job These icons are combined with the icon representing an individual job. This echoes the approach taken in the original Isotype graphics of	Star - New job Exclamation mark – Priority job Lock – Job is held  

The Filter panel to the left of the heat map is used to select which jobs to display in the heat map. The list can be filtered on priority, customer and current status of the job. The filter can be used to focus attention on a sub-set of the workload to quickly answer questions. For example, if a special delivery is to be planned for a customer the filter can select all jobs to be delivered to the same customer so that deliveries can be optimised. When the user selects an icon the corresponding job's details appear in the Job Details panel to the right of the heat map.

The icons representing the jobs are sorted by state of progress through the process with the results that the lighter icons are to the left of the delivery box while those that are ready for despatch are to the right. As more jobs are ready for despatch the pool of icons progress through to the darker shades. When a delivery has been picked up by the courier the entire box is greyed out to indicate that it is not longer active. The jobs within that delivery may still be filtered and the details seen in case there are queries from the customer or the courier.

The Scheduled Deliveries and Jobs panel supports direct manipulation interaction where possible. For example, rescheduling a job from one delivery to another can be accomplished by dragging and dropping the icon to the appropriate delivery. Some actions cannot be performed through direct manipulation because there is no appropriate object visible in the interface. For example it is necessary to scheduling additional deliveries from time to time. In this case there is no object in the interface that the user can interact with and so this function is made available as a button in the Available Actions panel and also on the context-sensitive menu in the Scheduled Deliveries and Jobs panel.

The design incorporates additional features to satisfy the principles of good design. For example, the Filters panel has an associated timer such that if a filter has been applied to the view then after five minutes the background colour on the Filters panel blinks. This is to act as a reminder to the user that the view on-screen has been filtered.

5.3 Confirming all requirements are met

When the design is being assessed it is useful to return to the principal data, constraints and functions that were identified during the analysis phase (see Table 7, p. 80) to ensure that those features are present in the final design. Table 4 presents the outcome of this review. The designer can be satisfied that all requirements that were identified have been included in the design.

Table 4. Referencing the Design to the Requirements

HTA Component	Function	Interface
1.1	Calculate workload for a shift	BEQs panel
1.2.1	View planned resources; Edit resources; Confirm resources are correct; Calculate capacity;	Resources panel
1.3.1	(Verify schedule with Shipping department)	Off-line activity
1.3.2	View delivery schedule; Retrieve planned delivery schedule from the production system; Add a new (initially empty) delivery; Confirm schedule is correct;	Scheduled Deliveries and Jobs
1.4.1	View and compare resource capacity, workload and delivery schedule; Filter view by production unit (optional);	Monitor panel
1.4.2	Edit resources; from 1.2.1	Resources panel
1.4.3	(Negotiate revised delivery with the customer)	Off-line activity
1.4.4	Add new (empty) delivery; from 1.3.2	Context-sensitive menu on the Scheduled Deliveries and Jobs panel
1.4.5.1	Identify jobs available for release;	Scheduled Deliveries and Jobs panel
1.4.5.2	Release job to production; Notify production system;	Context-sensitive menu on the Scheduled Deliveries and Jobs panel
2.	Recognise and alert on new jobs;	Scheduled Deliveries and Jobs panel
2.1	Retrieve new jobs from production system	Not applicable - the production system is the master so jobs are added there
2.3	Monitor the progress of production	Monitor panel

6 Results

During the study a prototype was created and evaluated by users. To capture a baseline for later comparison a number of users were interviewed at the start of the process to capture their views on the existing solution. Following the creation of the prototype a second evaluation was performed by interview and cognitive walkthrough, asking the users to perform specified tasks and then interviewing them to capture their opinions of the improved design. The resulting evaluation is a qualitative assessment. While Sharp, Rogers and Preece (2007), and others, give good guidance on how to execute quantitative assessments by timing users as they work through scenarios, measuring user interface interactions, and so on, it was not possible in this study. This software module is used to monitor production control by a small group of users and those users did not have the time available to run the kind of repeated scenarios under controlled conditions that are necessary to perform a statistically acceptable assessment.

The users responded positively to the new design for a number of reasons. The initial response was based on the attractiveness of the graphical design and this was to be expected since the original design was basic. While this was the first response for all users the improved aesthetics are the least significant benefits of the new design.

After studying the display with no introductory explanation the users recognised that the interface presents significantly more data and functions than the existing design. While they could see that more data was being presented they had concerns that this new design would require significant interactive assistance ("*tooltips to show what each element means*") or at a minimum a formal training session for new users. The users appreciated that all of the functions available in the module were available on the primary screen and not embedded in main menus or right-click menus. They felt that this would act as a reminder of the possible actions for inexperienced or occasional users.

The setup work at the start of a shift to ensure the capacity and schedule are in balance was seen as weak-point in the interface. It reflects activities that the supervisors do today but in formalising the process it lost some of the detail and became a simplification ("*the capacity calculation does not take account of all factors affecting throughput*", "*the delivery schedule has to be represented more strongly when looking at capacity*"). Overall it was seen as a useful contribution but the area of the new design that needs to be reviewed for possible improvement.

Users immediately recognised the power of Monitor panel presenting the progress of production against the target and the capacity for the shift. All users felt that this allows for assessment on progress at a glance and would also be extremely useful for new or inexperienced users. There was discussion about the method used to calculate the capacity for a shift. Some users felt that it was too simplistic because it did not recognise the variation in productivity among technicians. Users recognised that the baseline is really only useful at the start of shift. During the shift they assess progress against the actual Target and the Capacity becomes less relevant.

The heat-map was identified as a visually striking element of the new design. Some users questioned its usefulness during normal operations ("*it takes up a lot of space on the screen*", "*it's pretty but should we be looking a list of jobs*"). The users found the filtering on the heat-map (and text view of the jobs list) to be very powerful and a definite aid to issue resolution (e.g. "*if I need to see the customers affected when a number of jobs are delayed*"). The users also recognised the benefits in having the available options linked to the entries on the heat-map and jobs list. This was seen as a reminder of what is possible and also an ease-of-use function which avoids the need to remember details and enter them into another system.

Users at one of the smaller production units questioned the need for this module. The supervisor at that unit can assess the state of production by walking through the production area and talking to the technicians. The feedback is valid given the current scale of that unit however as units grow larger or the supervisor's span of control grows the need for assistance from the system becomes apparent.

Interestingly, the social and communication aspects of the supervisor's role had not emerged as strong requirements for the tool during the analysis.

When it was assessed new display was seen as providing two important communication benefits. Currently, there is no formal output from the start of shift whereas in the new design the resources are confirmed and the capacity and targeted production are visible. In addition, the Transport team can easily see the workload and progress which cuts down on communication overhead as deadlines approach.

The overall response was very positive. All users saw benefits in the monitor and control tasks; this is a reflection of the strengths of the EID approach. The graphical presentation was also very well received using newer controls and techniques that users have seen on modern web-sites and applications. The density of the data being presented and especially the combination of status information, targets and limits was also well received. The estimation of capacity for a shift was designed as a simplification since so little of the necessary data is available in the underlying production system and this is the area that most users singled out as a concern.

7 Discussion

7.1 Lessons from Using the Tools

The approach in the hands-on phase of the study followed the principles of Interaction Design (ID). The tactics of involving the users heavily and using interactive prototypes to explain and assess designs was well accepted by the team and they brought a lot of interactivity to the analysis and design phases of the project.

The hierarchical task analysis (HTA) was straight-forward to apply because it felt like a standard requirements capture technique. Observing and interviewing the users is a standard approach to systems analysis especially when a project requires redesign of an existing piece of software. The HTA was an effective tool to document the existing process and it did produce significant findings. The most important influence of the HTA on the new design was to change the focus from the flow of jobs through the production process to the delivery schedule. Going in to the requirements phase it was felt that the flow of jobs through the production activities would be most important factor to visualise. The analysis revealed that the cut-off times for collection are of equal importance in understanding if schedules would be met and so the HTA transformed the scheduled deliveries from their role as the final step in a process to the defining measure of the process. The detail of each job's progress through its life-cycle is not normally important unless an individual job is to be expedited.

Generating the Abstraction Hierarchy (AH) from Cognitive Work Analysis (CWA) was found to be more difficult. Researchers have pointed out theoretical and practical issues with the AH especially when it is applied outside its original domain of automated large-scale industrial process control, see for example Lind (1999) and Upton & Doherty (2008). While the AH has shown its usefulness even its creators have not defined a clear process to generate an AH - their articles on the subject have presented case studies and examples but not a defined process. In this study the AH was generated after the HTA based on many of the same inputs. Using the HTA and CWA on the same process in this way requires the analyst to be disciplined in lifting their attention away from the details of the current process.

It was re-assuring to be able to cross-check the outputs of the two analysis techniques - the HTA and the AH. While each tool has its own strengths the visible overlap in the outputs gave the analyst the confidence that the key requirements had been captured. Interestingly, the initial scope of the analysis was focused on the shift supervisor's responsibility to maximise the efficiency of the production unit. As such, no requirements were identified related to communication with teams outside of the production area. During the evaluation of the prototype additional benefits of the new design were recognised because the new design clearly shows the readiness of jobs to be included in the scheduled deliveries for the shift. This visibility will allow the Transport team to track progress and answer many of their own queries without having to speak to the production supervisor.

Applying the ecological interface design (EID) principles delivered real benefits in the final design. The inclusion of constraints such as the available resources and the delivery cut-off times in the final graphical design appears to give real assistance to the supervisor, especially to less experienced supervisors. These benefits had a cost because it became clear that some cases this contextual information is not readily available. While an experienced supervisor can gauge the resources at his/her disposal creating a model to quantify the resources is not simple. The complexity comes from (i) the need to consider the balance between resources at different stages of the production flow (e.g. sterilisation units versus technicians for manufacture) and (ii) the variation in productivity between individuals in a largely manual process. In some cases this deficiency was addressed by mapping available data to constraints using heuristics. In other cases, for example when estimating the capacity for a shift, a rough-cut solution was created which gives the supervisor a baseline to assess the schedule against. The accuracy of the rough-cut model is considered to be acceptable since the Monitor panel provides feedback every 15 minutes on progress against the target. This provides sufficient opportunity to recognise and respond to a situation where capacity is insufficient or where throughput is lower than estimated.

This lack of access to accurate data deemed necessary by the requirements analysis, and the actions taken to address it, also reflect the experience of Sharp and Helmicki (1998) when they faced similar limitations in a patient-monitoring scenario. Their study used the data from sensors applied to patients in a neonatal ward to build an enhanced patient-monitoring system. They found that some of the data identified as necessary in the abstraction hierarchy could not be gathered because the necessary sensors were not available or could not used with the patients for reasons of comfort or safety. Their solution was to infer the missing data from measurements that could be taken and their results showed that working with approximated data does yield benefits.

7.2 Source of the Benefits Achieved

The primary benefits of the new design come from adding context to the data and this was seen as bringing benefits especially to users with less experience. This mirrors the findings of the medical team using EID in neo-natal care (Sharp & Helmicki, 1998). For example, using the current processes if a supervisor wants to increase the workload by adding jobs from the next day's production then he/she must be aware of the shelf-life restriction. In the new design the system only allows selection of jobs that meet this restriction and in this way an activity that required knowledge based behaviour with the old design is transformed into a skills-based behaviour.

A number of benefits of the new design were achieved by expanding the scope of the tool, for example to include more product families, while remaining with a one-screen layout. The resulting design is a more effective aid to the supervisors than the original design was despite this increased scope. This outcome supports the argument that a well designed graphical presentation allows more data to be communicated effectively without requiring proportionally more space for the additional data.

7.3 Weaknesses of the Study

Some features of the production environment had a negative influence on the execution of the real-world activities in this study.

The small number of experienced users and the short time available limited the effectiveness of many of the tools. There was not time to fully explore the possibilities of two analysis techniques because the users were unable to give that amount of time to the project.

When the design was complete it was not possible to implement the prototype in the production environment. This was due to the validation effort necessary in the heavily-controlled production environment and competing project priorities at the production sites. This limited the evaluation that could be done on the new design.

7.4 Opportunities for Future Work

7.4.1 Visualise Changes to the Production Schedule

The hierarchical task analysis identified a number plans and operations for the shift supervisor to identify and incorporate extra customers' jobs being added during a shift to the production schedule. See Plan 2 in Table 5 (p. 74) for the details. While the new design provides an overview of the existing workload it does not help the supervisor to assess the impact that adding this job into the schedule would have on the existing jobs. The supervisor would have to consider whether the required components are already available as buffer stock in a steriliser or an isolator; if not, will the time needed to pick and sterilise components and manufacture the job require other jobs to be delayed or planned for a later delivery? If jobs must be re-planned, then which jobs should be selected to minimise the impact to customers and disruption to the work in progress in the production area?

This is a significant challenge and the aim of an initial solution would be to assist the supervisor in assessing the situation and presenting the information necessary to make a good decision. Vrotsou, Ynnerman and Cooper (2008) showed how the technique of work sampling can be used to create a database of historical information that can be visualised and data mined to reveal patterns in the behaviour. There does not appear to be a path to allow the tool to be used to analyse future tasks and manipulate the proposed schedule. The work of Luz and Masoodian (2007) on *temporal mosaics* could provide the basis for further research in this area. The authors introduced the temporal mosaic - as a two-dimensional graphic to represent the interaction between parallel activities. The mosaic was created as a visualisation to simplify the analysis of the participants' activities in multimedia meeting recordings. The chart is effectively a rectangle with time on the horizontal axis. The height is fixed regardless of the number of activities occurring at any particular point in time. See Figure 16 for an example of a temporal mosaic.

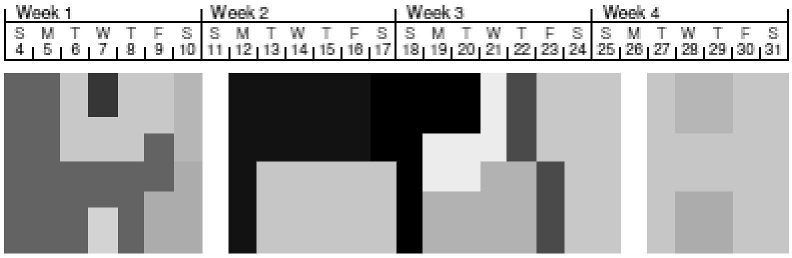


Figure 16. Temporal Mosaic (Luz & Masoodian, 2007)

The temporal mosaic provides an overview of the full set of activities for the timeline in question and it represents overlapping and concurrent events very effectively. A comparison against the standard timeline visualisation showed that users could answer questions relating to the activities on the timeline more quickly and more accurately using the temporal mosaic. One significant weakness compared to a Gantt chart is that the temporal mosaic does not represent the effort consumed by an activity – since the height of the chart does not vary the space allocated to an activity depends on the number of concurrent activities as the graphic is generated and not on any property of the activity.

In their later paper *Comparing static Gantt and mosaic charts for visualization of task schedules* Luz and Masoodian went on to apply the mosaic chart to represent static task schedules. The authors indicated that further work is on-going to make the mosaic an interactive tool. If the temporal mosaic can be adapted to create an interactive representation of the production schedule then it becomes possible for the supervisor to visually assess the impact of adjustments that would be necessary to include additional jobs.

7.4.2 A Visual Design Methodology

Considering the challenges of the design process generally there is still a need for a visual design methodology that designers can use when creating systems. The work done by Upton & Doherty (2008) has shown a way forward. In their own words their use of work domain and task analysis to generate information requirements combined with ecological interface design to guide the design process "falls short of being a step-by-step methodology" and is best considered as a framework of activities. More work is needed to create a methodology that can be used by novice designers and skilled designers alike.

8 Conclusion

The study set out to investigate how graphical data presentation can improve the usability of a process control computer system. The chosen method was to redesign the user interface of a software module used to control a small-scale make-to-order production facility.

The tools and techniques required to build an effective interface exist today although the learning curve is steep and results are uncertain for the inexperienced practitioner. This study used the following tool-set. While these are not the only options available they do represent an effective combination as demonstrated in this study but also by others e.g. Upton and Doherty (2008).

Component	**Function**
Interaction Design (ID)	Overall framework
Hierarchical Task Analysis (HTA)	Requirements – current process
Cognitive Work Analysis (CWA)	Requirements – work domain
Ecological Interface Design (EID)	Interface Design – interface content and presentation
Principles from Nielsen, Norman and Shneiderman	Interface and Interaction Design – layout, controls, interaction, etc.
Principles from Bertin, Tufte and Few	Interface Design – graphical elements
MS Expression Blend and Visual Studio	Software design and development

The experience of using the tools is described in section 7.1 but it is worth reinforcing that the designer must remember to ground the outputs of EID in the wealth of guidance and advice about the design of good user interfaces and user interaction that has been published over the past 3 decades by thought leaders such as Nielsen, Norman and Shneiderman. In addition, any designer planning to use graphical representations of important elements of the user interface is advised to prepare by reviewing the works of leading authors such as Bertin, Tufte and Few.

This study reinforces one of the risks of working with small teams that have operational concerns as their primary consideration – the possibility of limited access to the users. While it was possible to gain co-operation for the design work and support during the review phase it was not possible to do a quantitative comparison of the two designs. A qualitative comparison based on walkthroughs and interviews showed that the new design is seen as a significant improvement over the existing interface. Users responded well to the fact that the control data was placed in the context of the overall operation. This was especially so for less experience users. This result reflects the findings of several published studies, for example Sharp & Helmicki (1998).

This research makes a number of contributions. The study generated qualitative results that support the use of graphical representations in the user interface of process monitoring systems where the underlying production or execution system is not highly automated. This scope includes many opportunities in the Irish healthcare environment. For example, there are several small scale production units similar to the one described in this study operating in acute care settings. Other opportunities are found in the scheduling and management of resources such as operating theatres and homecare nurses. Using graphical representations it is possible to present more data to the users and to present the data in an integrated fashion which supports the user more effectively in doing their work. The research also demonstrates the practicality of using Ecological Interface Design in a small project team which should encourage others to consider adopting the practice.

Two opportunities for future work are identified. First, the practice of visual design is still lacking a clearly defined methodology that combines the available tools so that even inexperienced designers can use them. Second, within the scope of production scheduling it would be useful to investigate the creation of an interactive temporal mosaic representing the production schedule so that proposed changes to the schedule can be assessed visually.

In conclusion, graphical representation has much to offer in the area of process control even where the underlying processes are not heavily automated. While the methods and tools to build effective graphical interfaces continue to improve choosing the most appropriate options and using them well remains more art than science. When researchers evolve the existing guidelines into a comprehensive methodology that makes the process more predictable all software designers and development teams will be able to consider incorporating powerful graphics in their designs.

9 References

AIGA. (2011). *Symbol Signs*. Retrieved May 17, 2011, from AIGA - Tools & Resources: http://www.aiga.org/symbol-signs/

Ammenwerth, E., Schnell-Inderst, P., Machan, C., & Siebert, U. (2008). The Effect of Electronic Prescribing on Medication Errors and Adverse Drug Events: A Systematic Review. *Journal of the American Medical Informatics Association , 15* (5), 585-600.

Annett, J. (2004). Hierarchical task analysis. In D. Diaper, N. Stanton, & eds, *The handbook of task analysis for human-computer interaction* (pp. 67-82). Mahwah, NJ: Lawrence Erlbaum Associates.

Bertin, J. (1983). *Semiology of Graphics. Diagrams Networks Maps.* Madison: The University of Wisconsin Press.

Best, J. (2001). *Damned Lies and Statistics: Untangling Numbers from the Media, Politicians, and Activists.* University of California Press.

Biderman, A. (1990). The Playfair Enigma: The development of the schematic representations of statistics. *Information Design Journal* (6), 3-25.

Bisbal, J., Grimson, J., & Bell, D. (2005). A formal framework for database sampling. *Information & Software Technology , 47* (12), 819–828.

Burke, C. (2009). Isotype: Representing social facts pictorially. *Information Design Journal , 17* (3), 211-223(13).

Cleveland, W. S. (1993). *Visualizing Data.* Summit, NJ: Hobart Press.

Crystal, A., & Ellington, B. (2004). Task analysis and human-computer interaction: approaches, techniques, and levels of analysis. *Proceedings of the Tenth Americas Conference on Information Systems, New York.* New York.

David, H. (1995). First(?) Occurrence of Common Terms in Mathematical Statistics. *The American Statistician , 49* (2), 121-133.

Eccles, D. W., & Groth, P. T. (2006). Problem Solving Systems Theory: Implications for the Design of Socio-technological Systems. *Technology, Instruction, Cognition and Learning , 3*, 323-343.

Few, S. (2006). *Information Dashboard Design.* London: O'Reilly.

Friendly, M., & Denis, D. J. (2001). *Milestones in the history of thematic cartography, statistical graphics, and data visualization.* Retrieved 05 11, 2011, from DataVis.ca: http://www.datavis.ca/milestones

Graves, S. C. (1981). A Review of Production Scheduling. *Operations Research , 29* (4), 646-675.

Hewett, T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., et al. (1992). *ACMSIHCHI curricula for human–computer interaction. Association for computing machinery.* New York: ACM.

Jamieson, G., Miller, C., Ho, W., & Vicente, K. (2007). Integrating Task- and Work Domain-Based Work Analyses in Ecological Interface Design: A Process Control Case Study. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS , 37* (6), 887-905.

Jenkins, D., Stanton, N., Walker, G., Salmon, P., & Young, M. (2006). *Creating interoperability between the Hierarchical Task Analysis and the Cognitive Work Analysis Tools.* London: UK MoD Scientific Research Programme.

Keim, D., Hao, M., Dayal, U., & Hsu, M. (2001). Pixel bar charts: a visualization technique for very large multi-attribute data sets. *Infromation Visualization* , 1-14.

Kirkman, T. (1996). *Display of Statistical Distribution.* Retrieved 08 15, 2011, from Statistics to Use.

Lahlou, S. (2007). Human Activity Modeling for Systems Design: A Trans-Disciplinary and Empirical Approach. In D. Harris, *Engineering Psychology and Cognitive Ergonomics, HCII 2007* (pp. 512-521). Heidelberg: Springer-Verlag.

Lind, M. (1999). Making sense of the Abstraction Hierarchy. *Proceedings of cognitive science approaches to process control,* (pp. 195–200). Velenciennes, France: European Association of Cognitive Ergonomics.

Luz, S., & Masoodian, M. (2007). Visualisation of parallel data streams with temporal mosaics. *Procs. of the 11th International Conference on Information Visualisation* (pp. 196-202). Zurich: IEEE Computer Society.

Mackinlay, J. D. (1986, April). Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics* , pp. 110-141.

Mackinlay, J. D., Hanrahan, P., & Stolte, C. (2007). Show Me: automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics , 13* (6), pp. 1137-1144.

Miller, C., & Vicente, K. (2001). Comparison of Display Requirements Generated Via Hierarchical Task and Abstraction–Decomposition Space Analysis Techniques. *INTERNATIONAL JOURNAL OF COGNITIVE ERGONOMICS , 5* (3), 335–355.

Moise, A., & Robert, J. (2007). Model Driven Ecological Interface Creation: The Constraints Model. In J. Sprinkle, J. Gray, M. Rossi, & J.-P. Tolvanen (Ed.), *Proceedings of the 7th OOPSLA Workshop on Domain-Specific Modeling (DSM'07).* Jyväskylä, Finland: University of Jyväskylä, Finland.

Mulhall, M. G. (2009, June 12). *Dictionary of Statistics.* Retrieved July 07, 2011, from Open Library: http://openlibrary.org/books/OL23402517M/Mulhall's_dictionary_of_statistics

Nielsen, J. (1994). *Usability Engineering.* San Francisco: Morgan Kaufmann.

Nightingale, F. (1857, republished online). *Mortality of the British Army.* Retrieved 06 07, 2011, from Open Library: http://openlibrary.org/books/OL23320658M/Mortality_of_the_British_army

Norman, D. (1988). The Psychology of Everyday Things. New York: Basic Books.

Piquette-Miller, M., & Grat, D. (2007). The Art and Science of Personalised Medicine. *Clinical Pharmacology and Therapeutics , 81* (3), 311-315.

Playfair, W., Wainer, H. (., & Spence, I. (. (2005). *Playfair's Commercial and political atlas and Statistical breviary.* New York: Cambridge University Press.

Rasmussen, J. (1983). Skills, Rules and Knowledge; Signals, Signs and Symbols, and Other Distinctions in Human Performance Models. *IEEE Transactions on Systems, Man and Cybernetics , SMC-13*, 257-266.

Rezaeian, M., Dunn, G., St Leger, S., & Appleby, L. (2007). Do hot spots of deprivation predict the rates of suicide within London boroughs? *Health & Place* , 886-893.

Salmon, P., Jenkins, D., Stanton, N., & Walker, G. (2010). Hierarchical task analysis vs. cognitive work analysis: comparison of theory, methodology and contribution to system design. *Theoretical Issues in Ergonomics Science , 11* (6), 504-531.

Sharp, H., Rogers, Y., & Preece, J. (2007). *Interaction Design* (2nd ed.). New York: Jonh Wiley & Sons.

Sharp, T., & Helmicki, A. (1998). The application of the ecological interface design approach to neonatal intensive care medicine. *Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting* (pp. 350-354). Human Factors and Ergonomics Society .

Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. *Proceedings of the 1996 IEEE Symposium on Visual Languages (VL '96)* (pp. 336-343). IEEE CS Press.

Shneiderman, B., & Wattenberg, M. (2001). Ordered Treemap Layouts. *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01).* San Diego, USA: IEEE.

Shneiderman, B., Plaisant, C., Cohen, M., & Jacobs, S. (2009). *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* Boston: Addison Wesley (Pearson).

Snow, D. J. (1855). *On the Mode of Communication of Cholera.* Retrieved 06 25, 2011, from Google Books: http://books.google.com/books?id=-N0_AAAAcAAJ&printsec=frontcover

Sookman, J. (2010, 4 28). *Effectively Communicate Mission-Critical and Relevant Data.* Retrieved 08 01, 2011, from Startup Life.

Stanton, N. (2006). Hierarchical task analysis: Developments, applications, and extensions. *Applied Ergonomics , 37*, 55-79.

Teixeira, L., Ferreira, C., & Sousa Santos, B. (2010, Available online 13 November.). User-centered requirements engineering in health information systems: A study in the hemophilia field. *Computer Methods and Programs in Biomedicine* , In Press, Corrected Proof.

Traylor, P. S. (2006, 02 13). *To build or to buy IT applications?* Retrieved 07 11, 2011, from InfoWorld Applications: http://www.infoworld.com/d/applications/build-or-buy-it-applications-676

Tufte, E. R. (2006). *Beautiful Evidence.* Cheshire, CT, USA.: Graphics Press.

Tufte, E. R. (1983). *The Visual Display of Quantitative Information.* Cheshire, CT: Graphics Press.

Turgenev, I. S. (1991). *Fathers and Sons.* Oxford: Oxford University Press.

Upton, C., & Doherty, G. (2008). Extending Ecological Interface Design Principles: A Manufacturing Case Study. *International Journal of Human Computer Studies , 66* (4), 271 - 286.

van Reeuwijk, L. (1998). *Guidelines for Quality Management in Soil and Plant Laboratories.* Rome: Food and Agriculture Organization of the United Nations.

Vicente, K. (1999). *Cognitive Work Analysis: Toward Safe, Productive and Healthy Computer-based Work.* Mahwah, New Jersey: Lawrence Erlbaum Associates.

Vicente, K. (2002). Ecological Interface Design: Progress and Challenges. *Human Factors , 44* (1), 62-78.

Vicente, K., & Rasmussen, J. (1992). Ecological Interface Design: Theoretical Foundations. *IEEE Transactions on Systems, Man and Cybernetics , 22* (4), 589-606.

Vrotsou, K., Ynnerman, A., & Cooper, M. (2008). Seeing Beyond Statistics: Visual Exploration of Productivity on a Construction Site. *Proceedings of the International Conference on Visualization Vis 2008* (pp. 37-42). London: IEEE.

Wang, B. B., Wan, T. T., Burke, D. E., Bazzoli, G. J., & Lin, B. Y. (2005). Factors Influencing Health Information System Adoption in American Hospitals. *Health Care Management Review , Vol. 30(1)*, 44-51.

Wang, T. D., Wongsuphasawat, K., Plaisant, C., & Shneiderman, B. (2010). Visual information seeking in multiple electronic health records: design recommendations and a process model. *Proceedings of the 1st ACM International Health Informatics Symposium, IHI '10* (pp. 46-55). New York: ACM.

Ware, C. (2004). *Information Visualisation - Perception for Design.* Elsevier.

West, T. (1995, November). Forward into the Past: A Revival of Old Visual Talents with Computer Visualization. *Computer Graphics* , 14-19.

Wilkinson, L. (2005). *The Grammar of Graphics.* Heidelberg: Springer.

Woods, D. (1985). Knowledge-based development of graphic display systems. *Proceedings of the Human Factors Society 29th annual meeting.* Human Factors Society.

Appendices

A. Extract from Hierarchical Task Analysis

Table 5. Extract from Hierarchical Task Analysis

Super-ordinate	Task Component – operation or plan	Information Requirements
0	*Manage production during a shift* Plan 0: Do 1 Then do 2 and 3 in parallel until the end of the shift Then exit ____ 1. [Supervisor] Prepare the schedule by balancing the resources and workload 2. [Super.] Monitor progress and ensure that the delivery schedule is met 3. [Super.] Maximise the productivity of the team	
1	*Prepare the schedule by balancing the resources and workload* Plan 1: Do 1.1, 1.2, 1.3 in parallel Then do 1.4 Exit ____ 1.1 [System] Calculate total effort for the scheduled jobs based on their complexity 1.2 [Super.] Confirm the available resources 1.3 [Super.] Confirm the delivery schedule 1.4 [Super.] Balance the resources, workload and delivery schedule	
1.1	*Calculate total effort for the scheduled jobs based on their complexity (assigned at order entry)*	
1.2	*Confirm the available resources* Plan 1.2: Do 1.2.1 then 1.2.2 Exit ____ 1.2.1 [Super.] Confirm which people and production facilities are available 1.2.2 [Super.] Enter production resources into the system	

Super-ordinate	Task Component – operation or plan	Information Requirements
1.3	*Confirm the delivery schedule* Plan 1.3: Do 1.3.1 If [there are changes] then do 1.3.2 Exit ____	
	1.3.1 [Super.] Confirm the delivery schedule with the transport group	
	1.3.2 [Super.] Revise the delivery schedule in the system if necessary	Change delivery times for customer delivery locations
1.4	*Balance the resources, workload and delivery schedule* Plan 1.4: Do 1.4.1 If the schedule is not in balance select an action and revise the schedule: Do 1.4.2 Do 1.4.3 Do 1.4.4 Do 1.4.5 Repeat until the schedule is balanced Exit ____	
	1.4.1 [Super.] Check if the schedule is balanced	View capacity utilisation and despatch schedule. Filter jobs by resource
	1.4.2 [Super.] Adjust the available resources up or down	Change available resources Filter jobs by customer
	1.4.3 [Super.] Negotiate revised delivery with the customer	Change Required By date/time
	1.4.4 [Super.] Arrange special delivery	Change delivery schedule
	1.4.5 [Super.] Include jobs from future shifts in the schedule	View Shelf Life and Required Date/Time

Super-ordinate	Task Component – operation or plan	Information Requirements
2	*Monitor progress and ensure the delivery schedule is met*	
	Plan 2: If [new jobs are received] do 2.1	Alert for new jobs
	If [available resources change] do 2.2	View Progress indicator
	Do 2.3	
	If [progress is not acceptable] then do 1.4	Alert on progress
	Exit	
	____	
	2.1 [Super.] Add new jobs to the schedule	
	2.2 [Super.] Respond to resource change	
	2.3 [Super.] Monitor progress of production	
	1.4 [Super.] Balance the resources, workload and delivery schedule	
3	*Maximise the productivity of the team*	
	Plan 3: Do 2.3	View Progress indicator
	If [there is excess capacity] then do 1.4	Alert on progress
	Exit	
	____	
	2.3 [Super.] Monitor progress of production	
	1.4 [Super.] Balance the resources, workload and delivery schedule	

B. Extract from the Abstraction Hierarchy

Table 6. Abstraction Hierarchy (Extract)

Level	System	Sub-system: Production Schedule	Sub-system: Delivery Schedule	Sub-system: Production Unit	Components
Functional Purposes	Run an efficient shift	Include new jobs in the schedule Constraint: Delivery schedule Respect product shelf-life Constraint: Shelf-life - (Expiry date/time > Required by date/time + Grace period) Maximise efficiency (Constraint: Shelf-life, Throughput)	On-time delivery of jobs	Manufacture drugs	
Values and Priority Measures		Identify new jobs not yet released; Utilisation of resources; Balance the resources against the workload; Production progress - on schedule to meet despatch?			
Purpose-related Functions		Negotiate delivery for jobs. Release order to manufacture. Calculate workload in BEQs. Calculate earliest time for release-to-manufacture	Schedule delivery		
Object-related Processes		Jobs (Job attributes: customer, contact details, current status, normal/urgent, required by, drugs/expiry, BEQ complexity, etc.)	Deliveries	Manufacture drugs. Calculate shelf-life/expiry date. Calculate capacity of resources in a unit.	
Physical Objects					Technicians, isolators, workstations, sterilisers, components, vans, drivers, etc.

C. Extract of the Function and Data Requirements

Table 7. Function and Data Requirements (Extract)

HTA Component	Function	Data	Comment
1.1	Calculate workload for a shift	Job Identifier; BEQ	
1.2.1	View planned resources; Edit resources; Confirm resources are correct; Calculate capacity;	Resource: Identifier; Type; Capacity/Hour; Availability during shift; Total capacity;	Standard resource allocation is stored as setup to avoid data entry
1.3.1	(Verify schedule with Shipping department)		Off-line activity
1.3.2	View delivery schedule; Retrieve planned delivery schedule from the production system; Add a new (initially empty) delivery; Confirm schedule is correct;	Delivery schedule: Date Repeating: Delivery	A Delivery consists of: Delivery id Cut-off time Repeating: Destination Delivery time
1.4.1	View and compare resource capacity, workload and delivery schedule; Filter view by production unit (optional);	Capacity from 1.2.1; Workload from 1.1; Delivery schedule from 1.3.2	
1.4.2	Edit resources; from 1.2.1		
1.4.3	(Negotiate revised delivery with the customer)	Customer contact details	Off-line activity
1.4.4	Add new (empty) delivery; from 1.3.2	Delivery; from 1.3.2	Delivery is arranged off-line with Courier
1.4.5.1	Identify jobs available for release;	Job id;	Job contains: Required By; Shelf-life;
1.4.5.2	Release job to production; Notify production system;	Job id;	
2.	Recognise and alert on new jobs;		
2.1	Retrieve new jobs from	Job:	The production system

	production system	Job id; Delivery location; Customer contact; Required by date/time; Assigned to unit; Assigned to resource; Shelf-life BEQ;	is the master so jobs are added there
2.3	Monitor the progress of production	Production workload; Delivery schedule; Job status	