Designing Visual Decision Support for Sociotechnical Enterprises

Connor Upton

A thesis submitted to the University of Dublin, Trinity College

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Computer Science)

April 2009

Declaration

I, the undersigned, declare that this work has not previously been submitted to this or any other University, and that unless otherwise stated, it is entirely my own work. I agree that Trinity College Library may lend or copy this thesis upon request.

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Connor Upton

Dated: April 6, 2009

Acknowledgements

I would like to thank my supervisor, Gavin Doherty, for the years of advice provided during this research. Gavin's careful and constructive criticism allowed me to formulate my ideas and identify a path through a broad, multidisciplinary topic. His support and patience through the multiple drafts of this thesis are greatly appreciated.

I wish to thank Intel Ireland for providing financial support for this research and for provided access to a complex work environment that ultimately became the focus of the thesis. Special thanks go to Frank Gleeson and Charlie Sheridan whose committed involvement and feedback allowed me to refine the work and whose positive attitude gave me great encouragement.

Sincere thanks to my colleagues in the Distributed Systems Group (DSG). Although my subject area was somewhat unusual for the group, many were willing to discuss my work and help me with my research woes. In particular I'd like to thank Andronikos Nedos, Cormac Driver, Pete Barron, Andrew Jackson, Éamonn Linehan & Eamon Dillon for the conversations over many, many coffees. I'm also sincerely grateful to Dr. Myra O'Regan in the statistics department who provided invaluable help in the analysis of the experimental data.

Thanks to all my friends who provided both moral and practical support during the writing up of this thesis. Kevin Teljeur and Marisa Llorens, who made valuable suggestions on the writing and in particular to Kenn Coleman, who spent many hours reviewing and discussing some of the more complex ideas around design theory.

Finally, I would like to dedicate this work to my family. To my parents, Christy & Marie and my siblings Chris, Vincent, Robert & Kate, who supported me in countless ways and to Amy, I couldn't have done it without you.

Abstract

As automation becomes more pervasive in industry, human decision-makers are becoming increasingly dependent on sensor-data to monitor and interpret performance across large-scale enterprises. At the same time an exponential growth in data volumes, brought about by widespread automation, can make it difficult to transform low-level data into meaningful information. Information visualisation has been proposed as a means for helping humans to cope with data overload, but research in this area has predominantly focussed on data analytics tools rather than control interfaces. To support control, an interface should highlight important information but it must also present this information within the context of the systems functional goals. Graphic displays that achieve both of these objectives can provide visual decision support for human problem solving but their creation poses a serious design challenge.

Cognitive Systems Engineering (CSE) is a discipline that was founded to develop principles, methods and techniques that guide the design of control interfaces for complex work systems. Despite considerable developments in CSE theory over the last two decades, many aspects of design remain poorly defined. In practice, designers bridge these gaps in design knowledge using past experience and tacit skills, but this ad-hoc approach is unsuitable for dealing with the scale and complexity of enterprise-level work systems. This research aims to define a CSE approach that can inform the design of visual decision support systems in large-scale, sociotechnical enterprises.

A literature review specifies analytical, representational and procedural gaps in design knowledge and identifies the limitations of current CSE frameworks in relation to enterprise level systems. This review reveals a fundamental problem facing the CSE discipline; given that design is a deeply contextual activity and work systems can have widely disparate characteristics, how can generic principles inform design practice? In order to resolve this problem a meta-model of the CSE design process is developed. This model recognises design as an ill-defined problem, but one that involves specific problem-structuring and problem-solving phases. Progression through these phases requires an iterative process of concept generation and concept commitment. The model shows how primary design artefacts (models and sketches) support concept generation, while secondary design artefacts (cognitive and perceptual principles) support concept commitment. The sequence of secondary design artefacts used during design practice can be extracted to provide more generic design methodologies. As the complete design process is modelled, the resulting methodology will be comprehensive, providing bridges across the three CSE design gaps.

This meta-model provides a conceptual tool for conducting practice-led research that can extract generic design knowledge from contextual design activity. This approach is applied to two design projects carried out in a semiconductor-manufacturing enterprise. The first project relates to the redesign of a reporting tool used to observe the performance of thousand of sensors within an automated process control system. The methodology generated from this project reduces the representational gap by demonstrating how existing principles can be extended using information visualisation techniques, to provide more explicit design guidance.The second project examines the design of a visual decision support system for remote operations control in semiconductor manufacturing. The resulting methodology demonstrates how multiple analytical frameworks can be combined to describe intentional and distributed aspects of control in enterprise-level systems.

Although both projects were carried out in the same enterprise, their associated cognitive systems have fundamental differences and required separate methodologies. This outcome leads to a further discussion on the re-usability of design knowledge. Several characteristics of cognitive systems that influence the design artefacts used within design methodologies are identified. These characteristics are used to outline a taxonomy of cognitive systems that can be used to develop a catalogue of suitable design methodologies. This approach allows CSE design knowledge to be developed and reported at higher levels of abstraction and supports the re-use of design methodologies across different work domains.

Publications Related to this Ph.D.

Connor Upton, Gavin Doherty, Frank Gleeson and Charlie Sheridan. Designing Decision Support in an Evolving Sociotechnical Enterprise, *Cognition Technology & Work*, accepted 6 July 2008, DOI: 10.1007/s10111-008-0124-1.

Connor Upton & Gavin Doherty, Extending Ecological Interface Design Principles: A Manufacturing Case Study, *International Journal of Human Computer Studies*, 66 (4), pp. 271 - 286, 2008. DOI 10.1016/j.ijhcs.2007.10.007 .

Connor Upton & Gavin Doherty, Integrating Ecological Interface Design with the Visualisation Reference Model, *Proceedings of 25th European Conference on Cognitive Ergonomics (ECCE07)*, pp.175-178, 2007.

Connor Upton & Gavin Doherty, Designing charts for complex work settings, *Proceedings of 10th International Conference on Information Visualisation (IV06)*, pp. 447 - 452, IEEE Computer Society, London, 2006.

Connor Upton & Gavin Doherty, Visual Representation of Complex Information Structures in High Volume Manufacturing, *in IFIP International Federation for Information Processing, Volume 221, Human Work Interaction Design: Designing for Human Work*, ed. T. Clemmensen, P. Campos, R. Orngreen, A. Petjersen and W. Wong, (Boston: Springer), pp 27-45, 2006.

Gavin Doherty & **Connor Upton**, Designing Displays for Mobile Decision Support, *Proceedings of 19th BCS Conference on Human Computer Interaction (HCI 2005)*, Vol. 2, pp. 83-88, L. MacKinnon, O. Bertelsen and N. Bryan-Kinns (Eds.), 2005.

Connor Upton & Gavin Doherty, Designing Usable Decision Support Systems for HVM, *Proceedings of 10th IEEE International Conference on Emerging Technologies in Factory Automation*, edited by Lucia Lo Bello & Thilo Sauter, vol 1., pp.459-466, IEEE, 2005

Connor Upton & Gavin Doherty, Adapting the ADS for High Volume Manufacturing, *Proceedings of 10th IFIP International Conference on Human Computer Interaction (Interact)*, Lecture Notes in Computer Science, Vol. 3585, pp.1038-1041, M.F. Costabile and F. Paterno (Eds.), Springer 2005.

Contents

List of Figures

List of Tables

List of Abbreviations

ADS Abstraction Decomposition Space

AMHS Automated Material Handling System

APCS Advanced Process Control System

AT Activity Theory

CIM Computer Integrated Manufacturing

CLV Control Limit Variation

CSE Cognitive Systems Engineering

CTA Control Task Analysis

CWA Cognitive Work Analysis

DC Distributed Cognition

DSS Decision Support System

EID Ecological Interface Design

FAB semiconductor FABrication plant

FMS Flexible Manufacturing System

FOUP Front Opening Unified Pods

HCI Human Computer Interaction

HTA Hierarchical Task Analysis

HVM High Volume Manufacturing

IC Integrated Circuits

IR Information Requirements

IV Information Visualisation

KBB Knowledge Based Behaviour

KPI Key Performance Indicator

MES Manufacturing Execution System

NGM Next Generation Manufacturing

OC Operations Control

OOC Out Of Control

OTI On-Target Indicator

PCP Proximity Compatibility Principle

PCS Process Control System

PM Preventative Maintenance

QC Quality Control

RBB Rules Based Behaviour

ROCC Remote Operations Control Centre

SBB Skills Based Behaviour

SCADA Supervisory Control And Data Acquisition

TOC Theory Of Constraints

VDSS Visual Decision Support System

WIP Work in Progress

WTU Work,Task,User

Chapter 1

Introduction

1.1 Cognitive Systems Engineering

Cognitive systems engineering (CSE) involves the development and use of principles, methods and techniques for designing complex sociotechnical work environments. As mechanical and control automation becomes more pervasive across all work domains, it becomes increasingly important to understand how technological change influences decision-making and control. Only through this understanding can safe, efficient and effective sociotechnical work systems be designed.

1.1.1 Key Concepts

The core philosophy of CSE is that work environments that combine human and automated agents must be designed using a *systems-based approach* (Hollnagel and Woods, 1983). In the past, occupational accidents involving automation were frequently attributed to "human error", suggesting an inherent weakness of humans to work with highly-technical systems. More recently, investigations have shown that many incidents are the result of a phenomenon known as *automation surprise* (Woods et al., 1994; Sarter et al., 1997). This describes where a functional system responds differently to the intentions of a human operator, due to the unexpected influence of an automated agent. Automation surprise is attributed to a lack of appropriate feedback about the system state. Consequently the root cause of many accidents is not behavioural or human error but in fact *design error*.

The origin of this error has been associated with a "function allocation by substitution" approach to work system design, where human or automated agents can be exchanged depending on their ability to complete individual tasks (Hollnagel, 1999). This approach assumes task independence but in reality complex work involves the *co-ordination* of a range of tasks that are configured to achieve system goals. From this perspective human operators and automated systems should not be seen as autonomous agents but as team members in a *joint-cognitive system* (Hollnagel and Woods, 2005). Collaborative problem solving requires team members to share knowledge about the problem state and each others activities. With human collaboration this information is directly observable but automated systems do not provide natural, perceptual cues. Consequently, human operators in sociotechnical systems are highly dependent on system data for understanding the problem state.

Despite this dependence, ***observability*** is frequently overlooked in the development of joint-cognitive systems. In this context, observability is defined as "the cognitive work needed to extract meaning from available data" (Woods, 1997a). This term captures the relationship between data, observer and context of observation that is fundamental to effective feedback and control. Observability has been identified as a key issue in the design of successful joint-cognitive systems.

1.1.2 Key Challenges

This research deals with the CSE *design process* involved in achieving system observability. Christoffersen identifies two main challenges associated with this (Christoffersen and Woods, 2002). The first, relates to *developing a model of functionality* that describes why and how a work system functions in a particular manner and identifies the information requirements necessary to support control. The development of such a model is a complex task as work environments come as an intricate system of relationships and dependencies. This presents an *analytical challenge* to achieving system observability.

The second challenge relates to *representing system data* in a manner that supports control. Even when information requirements have been identified, if data is presented in a format that demands excessive cognitive work, it is unlikely to be observed (Woods, 1997a). Consequently, it is necessary to design representations that make the relationships and constraints of a work system explicit. However, visual design is notoriously difficult to proceduralise. While a range of visual design principles have been developed, many of these provide conflicting advice (Lin et al., 2006) and few provide the level of detail required to inform visual design decisions. This presents a *representational design challenge* to achieving system observability.

While these two challenges are often discussed independently, together they relate to a more generic problem known simply as the *design gap* (Wood, 1997). The design gap describes the fundamental problem of moving from design research and analysis to the generation of a design solution. Bridging the design gap requires the specification of comprehensive design methodologies that make the relationships between analytical models and representational principles explicit, across an entire design process. This presents a third *design process challenge* for achieving system observability.

These three challenges occur and must be resolved in every CSE design project. However the difficulty in resolving them increases in-line with the scale and complexity of the cognitive work system being studied. The development of pervasive automated control in large-scale industries poses a particular problem in this regard and provides the focus of this research.

1.2 Supporting Control in the Sociotechnical Enterprise

Joint-cognitive systems can be seen as a particular type of sociotechnical system where control is shared between human and automated agents. *Sociotechnical systems* are engineered environments that require the combination of technical and human factors in their design (Cherns, 1976). To date much of the research in this domain has focussed on critical systems where individuals or small teams monitor and respond to real-time events. Large industrial facilities employ multiple joint-cognitive systems, each dedicated to controlling aspects of the overall enterprise. Computer Integrated Manufacturing (CIM) provides a good example this, where complex process flows involving fleets of equipment are controlled using several automated control systems and a large, highly-trained workforce. The complexity of these facilities combined with their large social structures and often conflicting internal goals, allows them to be described as *sociotechnical enterprises* (Upton and Doherty, 2008). The overall control of a sociotechnical enterprise can be described using three basic principles (see figure 1.1).

Firstly, there is *separation of responsibility*. For example, the overall goal of manufacturing is divided into a number of sub-goals which typically include manufacturing operations, process control and equipment engineering. As these are highly specialised activities that take different perspectives on the overall system, each is handled by a dedicated department.

Secondly, there is *distribution of workload* within each department. The scale of these systems is managed by spreading the workload across a hierarchical social organisation. This work distribution ensures that floor-level workers have achievable workloads while management-level workers can co-ordinate their department's activities in relation to the goals of other departments. This social organisation also provides workers with local expertise that is important for training and problem solving.

Thirdly, there is *information processing support*. Large scales and fast production rates mean that CIM produces massive amounts of data. This data must be processed into information that conveys the system state and drives action. Different information systems are used to support workers in different roles and at different levels of management. For example, a manufacturing technician will use a dispatching application to carry out simple scheduling tasks, while a supervisor will use a reporting application to observe productivity within an area. At higher levels of management , Key Performance Indicators (KPI's) are used to describe departmental information and are reviewed during strategy meetings to support decisions about production goals. As data filters up the management hierarchy, the need to balance goals and resolve conflicts means that personal communication and tacit knowledge become increasingly important for interpreting the overall system state and supporting control.

Figure 1.1 depicts how these three principles can be used to model how the control of a sociotechnical enterprise is achieved. However, as manufacturing automation continues to evolve, this model of control is changing.

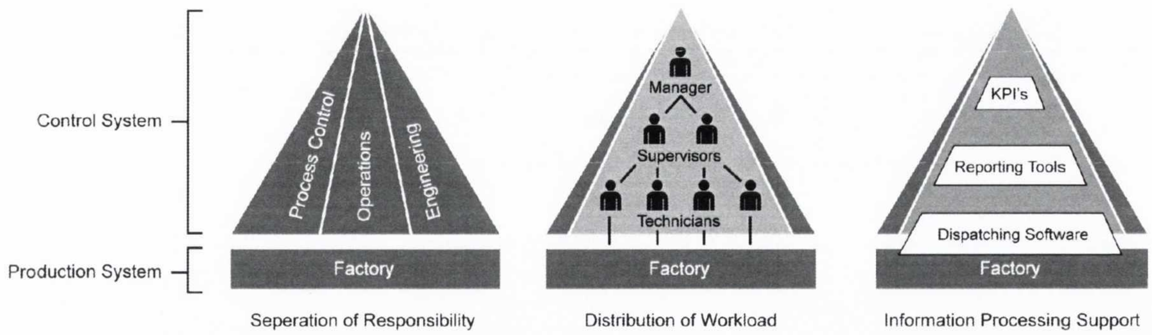


Figure 1.1: Controlling a sociotechnical enterprise

1.2.1 Next Generation Manufacturing Systems

New processes and more pervasive information technology have been identified as key needs facing the future development of Next Generation Manufacturing (NGM) systems (Force, 1997). One sector that is firmly committed to meeting these needs is semiconductor manufacturing. As costs in this sector are tied to the ever-increasing complexity of the product, improving production efficiency is a critical issue (Meieran, 1998). Advanced automation has been identified as a means for continuously increasing production rates and improving the information availability that is necessary for process development (Mouli and Srinivasan, 2004). As mechanical automation becomes more pervasive, it is necessary to match it with improved data automation. For instance, automated material handling requires tighter integration of process control, manufacturing scheduling and equipment maintainence systems. In turn, this integration of data systems allows automation to be extended to higher levels of system control. For example, the integration of advanced process control with fault detection and classification systems has led to the full automation of certain low-level monitoring and diagnosis tasks (Mouli, 2005).

The semiconductor manufacturing sector has predominantly focussed on *technological solutions* for meeting the needs of NGM. However when dealing with sociotechnical systems it is recommended that technological and human systems should be *jointly optimised* for an enterprise to meet its objectives (Paez et al., 2004). A concentration on technical aspects of development, without due attention to human factors issues, can result in sub-optimal performance once systems become operational.

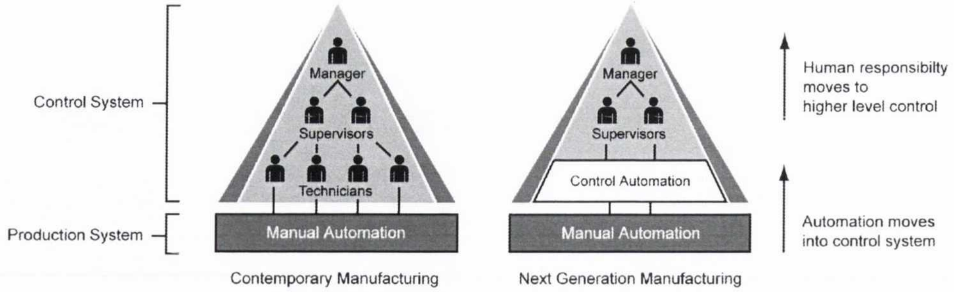


Figure 1.2: Effects of advanced automation on sociotechnical enterprise

The depiction of enterprise control shown in figure 1.1 will be affected in a number of ways by the introduction of advanced automation. Firstly, as automated systems become responsible for low-level decisions, human responsibility turns to; monitoring the performance of automated control systems, diagnosing problems when they arise and resolving issues at a higher level of abstraction. Secondly, the use of automated controllers will allow individual human operators to monitor larger sections of the enterprise and increases their responsibilities. At the same time this reduces the total number of human controllers associated with a department and, in turn, decreases the amount of social support and local expertise available to those remaining. Thirdly, the combination of *more responsibility* with *less social support* will increase the *dependence on information systems* for understanding the system state and responding effectively. These factors are shown in figure 1.2.

From these observations it can be inferred, that the introduction of advanced automation within a sociotechnical enterprise increases the importance of developing information systems that achieve system observability.

1.2.2 The Case for Visual Decision Support Systems

Within the semiconductor manufacturing industry the need to develop new information systems to control an increasingly automated enterprise has been recognised (Mouli and Srinivasan, 2004). A framework based architecture has been developed to improve information access, application interoperability and application extensibilty. Within this architecture the User Interface (UI) and Decision Support System (DSS) frameworks provide the primary means of achieving observability however they are somewhat limited in this regard.

A UI framework provides a platform for developing application interfaces based on re-using standard UI components. While this reflects a common approach to interface development (Myers et al., 2000), it does not not comment on how information requirements should be determined or represented. As such the UI framework acts as a development toolkit rather than a methodology that can guide design.

A DSS framework provides a platform for developing decision support systems using data warehousing and data analysis services. A DSS helps humans to solve ill- or semi-defined problems by using computer processing power to generate Key Performance Indicators (KPI's) from low level system data. However, highly sophisticated DSS often meet resistance from workers due to what the DSS research community calls "people problems" (Carlsson and Turban, 2002). These include claims that people have cognitive constraints in adopting intelligent systems, that they disregard support in favour of past experience and visions and that they believe they get more support by talking to other people. Ironically these problems relate back to the issue of observability. They stem from the fact that the automated decision models that generate high-level KPI's are generally not observable by the user. Unless these processes are made explicit, the user must rely on experience and tacit knowledge to interpret KPI's in relation to system goals.

In an effort to deal with this, another part of this DSS framework recommends using "best of breed" visualization tools to support analysis of system data. While this can make system information more visible, commercial visualisation tools come with their own limitations (Kobsa, 2001). They are primarily focused on data analysis rather than system monitoring and tend not to include performance models that can highlight unusual activity. Their supported tasks are generally limited to sorting, filtering, and correlation of low-level variables. Consequently, their users must be sufficiently competent with visualisation techniques and have enough domain expertise to be able construct meaningful displays.

This framework architecture for NGM has predominantly focussed on technical issues facing information system developers. However, while it supports the rapid construction of dynamic, data-driven displays, it cannot ensure that these representations will achieve system observability. Ideally Visual Decision Support Systems (VDSS) that integrate the advantages of graphic user interfaces, decision support systems and information visualisation are required. Such systems should be designed around a model of system functionality that places information in a context and makes the system state explicit. The development of such systems requires an approach to cognitive systems engineering that encompasses analytical methods for modelling system functionality and concrete visual design guidelines for generating valid representations. However the development and communication of such an approach is a complex task and first requires the analytical, representational and design process gaps in CSE design knowledge to be resolved.

1.3 Problem Statement and Research Questions

The previous sections have outlined the research domain and the motivation behind this work. From these the primary research problem can be stated as follows:

Given the current gaps in cognitive systems engineering theory, how can the analysis of sociotechnical enterprises and the consequent development of visual decision support systems be described in a manner that supports the generation and re-use of design knowledge?

This thesis seeks to answer this problem through a combination of literature reviews, model development, practice-led research and discussion. The problem has been broken down into a series of research questions. The first two deal with structuring existing CSE design knowledge.

1. **What are the limitations of current analytical frameworks for generating a functional model of a sociotechnical enterprise?**

2. **To what extent can generic representational principles inform contextual design practice?** These questions are answered in separate literature reviews in chapters 2 and 3. These reviews identify and specify the gaps in CSE knowledge that must be bridged during design. The presence of these gaps poses the next question.

3. **What role do analytical and representational methods play in the CSE design process?** This question is answered in chapter 4 through reflection on

the practice of design and the subsequent development of a meta-model describing the CSE design process. This meta-model provides a conceptual framework for structuring design practice and generating design theory. The meta-model is flexible, in recognition that design is a highly contextual activity, and this leads to the subsequent question.

4. **Given the contextual nature of design, can generic design methodologies be generated?** This question is answered in chapters 5, 6 and 7 through practice-led research. The design meta-model is used to describe the design process associated with two real-world CSE design projects. Reflection on the characteristics of the targeted work systems leads to a final question.

5. **Given the diversity of cognitive systems, how can design knowledge be re-used?** Chapter 8 provides a discussion of this issue and proposes a potential solution by outlining a taxonomy of cognitive systems that can be used to catalogue design methodologies.

1.4 Research Scope and Contributions

This research focuses on the theory and practice of cognitive systems engineering and aims to generate design knowledge that can guide practitioners in the design of cognitive artefacts for large-scale, complex work environments. The generation of design knowledge is a fundamental problem that lies at the heart of CSE research (Rasmussen et al., 1994). It is an activity that is notoriously difficult to describe but one that is impossible to avoid. As a result, cognitive design remains a predominantly craft-based activity in what purports to be an engineering discipline (Dowell and Long, 1998). This thesis examines the practice of cognitive systems naengineering with the aim of extracting knowledge that can provide a more structured approach to cognitive design. While the general research problem is quite broad, in contrast the fieldwork has been carried out in a highly specialised work domain. This thesis deals with theory development and the generation of design knowledge, however to ensure validity these theories must be applied to real-world systems. Semiconductor manufacturing provides a target industry with enough scale and complexity to test out the concepts put forward in this work.

This research makes a number of methodological contributions to the CSE discipline and practical contributions to the High Volume Manufacturing domain including:

1. The identification and specification of analytical gaps in CSE design knowledge (Chapter 2)

2. The identification and specification of representational gaps in CSE design knowledge (Chapter 3)

3. The development of a meta-model of the CSE design process (Chapter 4) that aims to make the design activity more accessible by providing:

    (a) a generic structure for guiding the practice of design

    (b) a context for understanding analytical and representational methods that can be used to bridge CSE design gaps

    (c) a conceptual tool for conducting practice-led research that supports the extraction of generic design knowledge from contextual design practice

4. A CSE design exemplar that:

    (a) reduces the representational gap by combining existing CSE design principles with information visualisation guidelines.

    (b) develops a visual decision support system for monitoring automated process control

    (c) provides an initial validation of the utility of the meta model for generating design knowledge (Chapter 6)

5. A second CSE design exemplar that:

    (a) reduces the analytical gap by combining multiple analytical frameworks within a design methodology

    (b) develops a visual decision support system for remote operations control in manufacturing

(c) provides further validation of the utility of the meta model for generating design knowledge (Chapter 7)

1.5 Thesis Structure

This chapter has introduced the discipline of cognitive systems engineering, explained the motivation behind the research work and has outlined the problems and questions to be covered in this thesis.

In chapter 2, the prominent frameworks for analysing cognitive systems are reviewed. Their key concepts, main applications and potential limitation are presented and their suitability for modeling large-scale sociotechnical enterprises is assessed. From this review a number of analytical gaps in CSE knowledge are identified and the utility of these frameworks to inform design practice is discussed.

Chapter 3 reviews the current CSE knowledge used to inform representational design. It examines the prominent CSE design theories and principles and identifies their limitations when applied to large-scale systems. Additional visual design guidelines from other disciplines are also reviewed. These offer alternative perspectives on the graphical encoding of information. Again a number of gaps in CSE knowledge are identified.

Chapter 4 begins by identifying how the design process gap between system analysis and system representation forms the crux of the CSE design problem. Current approaches to design reporting and evaluation are shown to perpetuate the design gap and an alternative approach is proposed. A meta-model of the CSE design process is developed which acknowledges design as an ill-defined problem, but one that involves specific problem-structuring and problem-solving phases. Conceptualisation is identified as the contextual activity that progresses design through these phases. Analytical and representational principles are identified as secondary design artefacts that control conceptualisation. It is proposed that this model can be used as a tool for building design theory by extracting generic design methodologies from contextual design practice.

Chapter 5 introduces the semiconductor-manufacturing domain as the target industry for validating this approach. Characteristics of the production process, the physical environment and the organization are outlined to provide background knowledge necessary for interpreting the design projects. Two recent developments within the industry are explained that provide the context for the two subsequent CSE design exemplars.

Chapter 6 traces the development of a visual decision support system for process control health monitoring. In the problem-structuring phase, work domain analysis is augmented with task analysis to provide models of cognitive tasks as well as functional constraints. During design problem-solving, principles from the ecological interface design framework (Burns and Hajdukiewicz, 2004) are extended using a visualisation reference model. This provides guidance for generating representations of large-scale systems. A final prototype is presented and evaluated. The design meta-model is applied to this design process to extract a design methodology.

Chapter 7 traces the development of a visual decision support system for remote operations control. This is a much larger scale project than the previous one. As a result, problem-structuring requires multiple analyses including functional constraints, the configuration of physical, social and information systems as well as the intentional goals and cognitive strategies associated with the work. Problem-solving is again presented using a design rationale that shows how analytical outputs and design principles guide the visual design process. The design model is used to extract a second design methodology from this design process.

Chapter 8 discusses how and why alternative design methodologies were generated. While previous attempts to map cognitive systems placed a strong emphasis on the type of work domain involved, it is demonstrated that a much broader range of attributes should be considered. A taxonomy of cognitive systems is developed around a number of characteristics that influence how cognitive design should proceed. It is proposed that this taxonomy may be used to catalogue associated design methodologies to support the re-use design knowledge across different domains.

Chapter 9 provides conclusions and outlines the contributions made by this thesis. It also points out a number of limitations with the current approach and identifies areas of future research.

Chapter 2

Analysing Cognitive Systems

This chapter reviews a number of frameworks currently used in the analysis of complex work systems. The role of knowledge representation in interface design is initially introduced and the distinction between coherence and correspondence driven work is explained. The characteristics of High Volume Manufacturing (HVM) environments are identified. These systems are shown to be correspondence-driven work domains that require a systems based approach to analysis. Three alternative frameworks used in CSE are reviewed in terms of their core concepts, applications and limitations. As none of these can produce a complete model of HVM system functionality, a gap in CSE analytical knowledge is identified. A further inadequacy relating to their utility for supporting the practice of visual design is also discussed..

2.1 Knowledge Representation and Interface Design

The analytical challenge identified in section 1.1.2 relates to specifying the information required to support problem solving and control. This is a general problem that is shared with human computer interaction. The Human Computer Interaction (HCI) discipline has looked to cognitive science and psychology to develop concepts relating to the analysis and design of interactive systems. Here we outline some of these concepts and show how they can be used to identify information requirements for interactive problem solving.

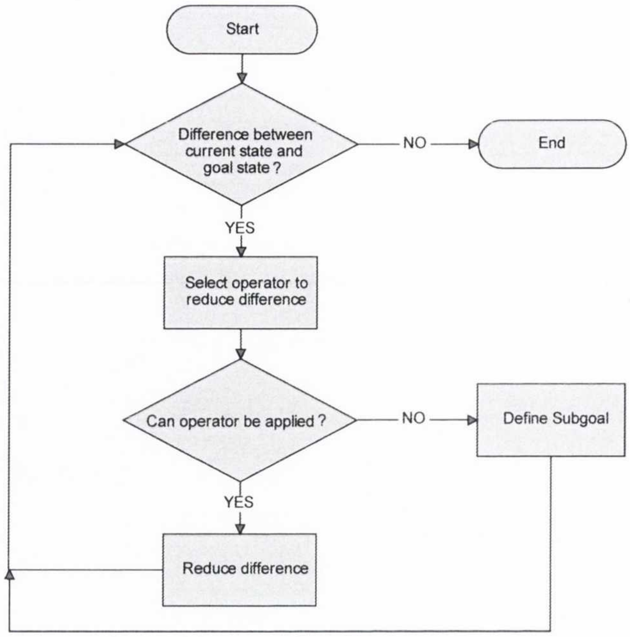


Figure 2.1: Means-ends analysis

2.1.1 Models of Problem Solving

From a cognitive science perspective human problem solving can be portrayed as an *information processing system* involving; *states of knowledge*, *operators* for changing one state into another, *constraints* on applying operators and *control knowledge* for deciding which operator to apply next (Newell, 1972). Problem solving is described as a movement from an initial starting state to a goal state by searching through a *problem space* of knowledge states. It is proposed that humans use heuristic search techniques such as *means-ends analysis* when problem solving (see figure 2.1). This process allows a problem solver to follow a path through the problem space without having to generate every possible outcome and suggests a number of characteristics about intelligent problem solving. Firstly it is goal-directed, secondly goals can be divided into sub-goals that are easier to attain and thirdly it can involve recursive cycles of means-ends analysis. This approach is one of the principal concepts in artificial intelligence and and can be used to resolve well-defined problems, but it also indicates that problem solving in general can be modelled using a hierarchical structure of goals and sub-goals (figure 2.2a).

The concept of a hierarchical goal model forms the basis of Cognitive Task Analysis, a semiformal and systematic method of investigating how users interact with information

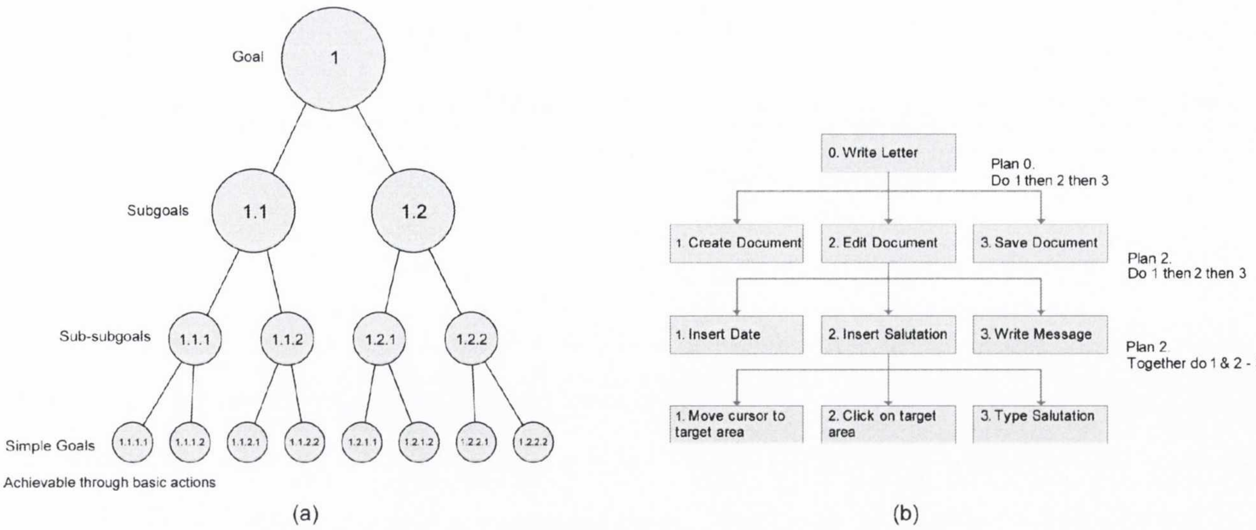


Figure 2.2: hierarchical goal structure (a) and hierarchical task analysis (b)

systems (Card et al., 1983; Kirwan and Ainsworth, 1992). Human-computer interaction frequently involves ill-defined problem solving, where the goal state, operators and constraints are not clear at the outset. For example, using a word processor to write a letter is a complex activity whose specific outcome cannot be pre-determined. Despite this, the user's high-level goal of letter writing can be broken down into sub-goals such as creating a new file, editing the content, printing, saving etc. These sub-goals can be further decomposed until a level of keystroke analysis is reached. The resulting hierarchical goal model provides a view on the problem space that can be used to identify the information requirements necessary to solve the problem (figure 2.2b). By carrying out a cognitive task analysis an application designer can define the interface functionality required for a user to achieve their goals.

While HCI models, analysis techniques and evaluation methods have been developed using the cognitivist perspective, it has been criticised for being too limited (Newell and Card, 1985). It does not comment on a range of important issues including visual displays, the use of natural language, the problems of novice users, the questions of learning and the probability and effects of errors. Issues such as usability, learnability and interpretability are concerns that require an alternative perspective on knowledge representation.

2.1.2 Mental Models and System Image

The mental model theory of thinking and reasoning proposes that humans construct "small-scale models" of reality in the mind that allow relationships in systems to be understood (Craik, 1943). Mental models can be thought of as knowledge structures located in long-term memory that provide the rules and constraints that control movement through a problem space. This provides a reasonable explanation for how humans can operate in a complex world. Mental models are not accurate representations of real world relationships, but it is precisely this lack of accuracy that makes them useful as they allow analogies with simpler functional systems to be used in the construction of models of more complex systems.

Mental models are understood to play a key role in deductive reasoning and this makes them a useful concept when designing user interfaces. Norman suggests that an interface designer develops a mental model of a system's functionality and designs the interface as a conceptual model for communicating this functionality to the user (Norman, 2002).

As computer users do not act on the system directly but only through the interface, a design can be described as a *system image* around which the users develop their own mental models of system functionality (figure 2.3). The overall success of an interactive system is dependent on the user's correct interpretation of the semantics of the system image proposed by the designer. For example the direct manipulation file-managament strategy of dragging files into folders relies on the user's interpretation of the desktop metaphor. The challenge facing the designer is to communicate with the user in a language they understand. This requires a *user-centred design* approach, whereby the *conventions*, *words* and *metaphors* generated by the designer are all familiar and interpretable by the users. To ensure this, the designer must examine the user's behaviour in context (Norman, 2007).

2.1.3 Coherence and Correspondence Driven Work

The two concepts of cognitive task analysis and user-centred design are central themes in HCI research and the wider interface design community. However both of these concepts were originally developed around the desktop computing paradigm. With desktop ap-

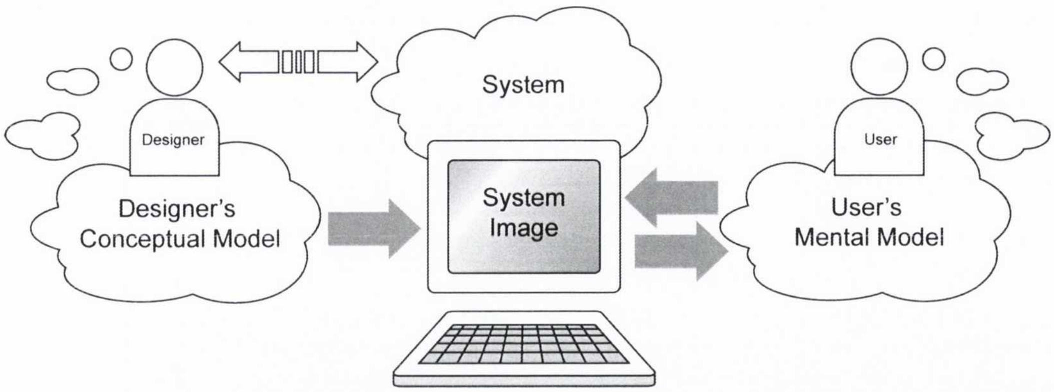


Figure 2.3: Mental models and system image

plications the computer acts as both the medium and the environment in which work is carried out. It is a closed system, whose goals are defined by the user and do not relate to any external events or constraints (fig. 2.4a). As the users' *understanding* of their work is the key factor that defines the interface functionality, these situations have been described as *coherence-driven work domains* (Vicente, 1990). Cognitivist approaches such as task analysis place a strong emphasis on user's tasks and events. These are used by the designer to develop a conceptual model of interaction. The purpose of the interface is to provide a system image that allows users to work fluidly and efficiently. However, as most operating systems have standardised UI components that represent interface functionality, the design of a system image is often limited to selecting icons and words that that are appropriate for a particular user group.

The use of computers to support work as part of a larger sociotechnical enterprise creates much greater challenges for the design of a system image. The goals of an operator controlling a sociotechnical system are defined not by the operator themself but by the functional purpose of the overall work system (fig. 2.4b). For example in a process control task, the goal of hitting a target is established by safety or economic constraints rather than the personal goals of the operator. Consequently it is not enough for the system-image to be coherent to a user's mental model, it must correspond to the goals and constraints of the overall work system. Vicente defines these situations as *correspondence-driven work domains* that impose dynamic, environmental constraints on the goal-directed behaviour of actors (Vicente, 1990). In these work domains the interface carries the double burden of achieving system observability as well as supporting tasks

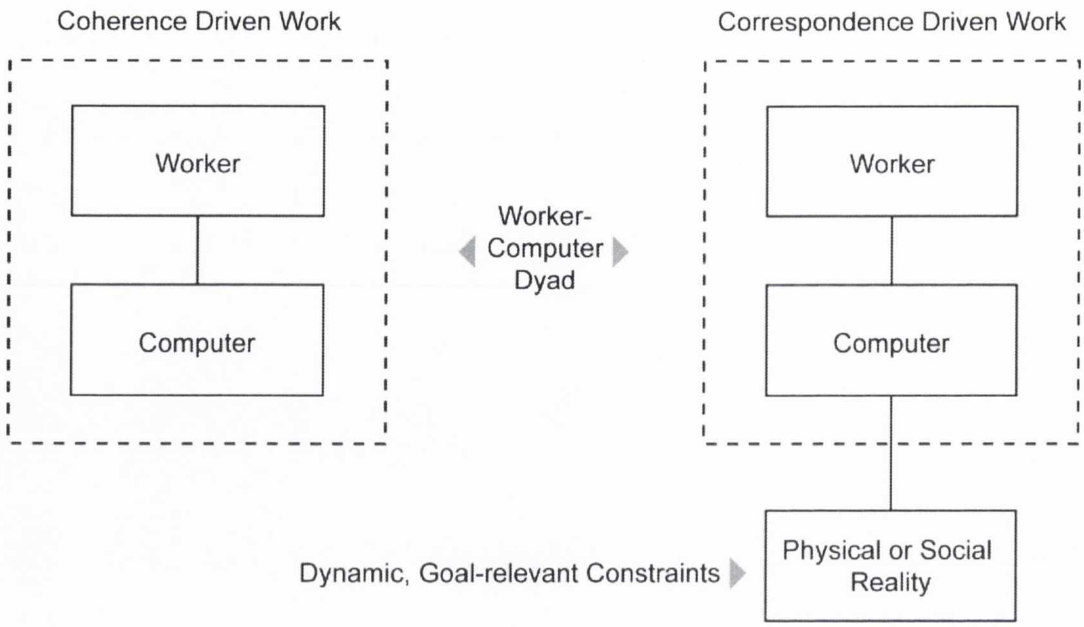


Figure 2.4: Coherence and Correspondence Driven Work (adapted from (Vicente, 1990))

and actions. As the operator will develop their mental model of functionality around the interface, it is essential that this system image corresponds to reality. Failure to achieve this may cause the operator to develop an inaccurate mental model and increase the potential for automation surprise through misinterpretation of the system state.

Ecological psychology provides an alternative perspective on problem-solving from the cognitivist approach. In relation to interpretation and the development of meaning, ecological psychology proposes that humans exist in a "systems" relation to their environment and suggests that complex human behaviour can only be properly understood within the context of this setting (Gibson, 1979). This provides the basis of the CSE approach, which requires an analyst to develop a model of system functionality in order to inform the design of cognitive tools.

2.2 Modelling System Functionality

Early research into the development of functional models focussed on decision-making in man-machine systems (Rasmussen and Jensen, 1974). A number of investigations into fault diagnosis and trouble-shooting were carried out. In each case the problem-solver's mental model of functionality was derived through cognitive task analysis and was compared against the engineering specifications of the faulty equipment. These studies revealed that expert problem-solvers transform the low-level structural relationships between components into a hierarchy of means-ends relationships. This technique allows them to think about the system at different levels of abstraction and supports deductive reasoning. The abstraction hierarchy was proposed as a generic model of functionality that aligned a multilevel representation of causal relationships with the structural composition of a system.

These early investigations established how operators reasoned about functionality in small, physical systems, where the coupling of system components was the primary source of complexity. However, with sociotechnical enterprises, such as manufacturing facilities, there are several factors that contribute to complexity (Vicente, 1999).

- *Large problem space.* There are a large number of variables involved in controlling the system and these are related in a complex network of relationships.
- *Social.* The work is divided across a social organisation where workers are responsible for aspects of the overall problem.
- *Heterogeneous perspectives.* The problem space is multifaceted and features conflicting constraints. As success in one area may cause issues in another, these constraints must be managed effectively within the context of the overall system.
- *Distributed.* Workload is distributed across humans, automated control systems and mechanical automation. In the case of manufacturing facilities, the work is also physically distributed across a large environment.
- *Dynamic.* These are open systems and system data is constantly changing. This requires controllers to maintain a high-level of system state awareness.
- *Potentially high hazards.* The actions executed through the system have real consequences as they affect the real world. Many activities and decisions are irreversible so a high level of procedure and control is required.
- *Coupling.* While task models often use linear depictions of events, the causal network that underlies a manufacturing system means that actions can trigger a range

of responses.

- *Automation.* Automation is pervasive in terms of both mechanical systems that handle processing and transport and control systems that monitor performance.

- *Uncertain data.* A manufacturing systems must balance production goals and engineering constraints against business goals and labour constraints. Many of these goals and constraints cannot be fully specified so a controller will have to rely on tacit knowledge and experience.

- *Mediated interaction.* Human workers do not act directly on the material. Processing and transport are carried out by automated systems which are supervised by human controllers.

- *Disturbances.* As these systems involve the real world they are subject to a range of variables that cannot be controlled. Unexpected events can and do happen and the complexity of the problem space makes it impossible for an analyst to predict all possible situations in advance.

Developing a functional model of such systems requires more than the alignment of an operators mental model with a structural model, it requires an examination of how physical, organisational, informational and technical aspects of the enterprise are co-ordinated to achieve specific goals. The difficulty with carrying out such an examination is that these perspectives are not available as individual streams that can be studied in isolation. They are only available by observing cognitive work in context and this comes in a "wrapped package", as a complex conglomerate of inter-dependent variables (Woods, 2003). Some of the challenges facing the analysis of work in context include:

Work systems are idiomatic. They use specialist concepts, terminology and imagery that an outside analyst must become familiar with before they can understand their use.

Work strategies often exist as tacit knowledge. While many work systems have documentation outlining best practice, workers develop their own strategies for achieving goals. As a result many of the procedures that achieve system functionality exists as tacit knowledge.

Workers are experts. It is difficult for an observer to understand the practices of an expert worker as their fluidity can hide their methods of completing tasks. This is even more problematic with cognitive tasks as an expert may not be able to verbalise thought processes that have become automatic.

Work systems are diverse. Just as joint-cognitive systems require the co-ordination of man and machine, enterprise-level work systems require co-ordination across multiple specialist roles. However, workers within these roles may have alternative or conflicting views on system goals.

2.3 Analytical Frameworks for CSE

Cognitive systems engineering developed out of the related fields of human factors, social science and engineering and each of these fields bring their own methods for examining physical, social and technical aspects of human-work interaction respectively. Currently, hundreds of different analytical methods and techniques are available to a CSE practitioner, ranging from surveys to ethnographic research to task and information-flow modelling (Bonaceto and Burns, 2004). In order to provide a more structured approach to analysis, a number of efforts have been made to construct analytical frameworks for CSE that take a particular stance on how cognitive systems function. These include Cognitive Work Analysis (CWA), Distributed Cognition (DC) and Activity Theory (AT). These frameworks are reviewed below to identify their utility for analysing sociotechnical enterprises and in particular High Volume Manufacturing (HVM) facilities.

2.3.1 Cognitive Work Analysis

Cognitive Work Analysis (CWA) was originally developed out of research into human performance in the control of complex engineering systems such as nuclear power plants (Rasmussen et al., 1990; Vicente, 1999). It is closely aligned with CSE's ecological view of understanding work practice and involves 5 stages of analysis moving between the general ecological constraints imposed by the work domain to the more specific cognitive constraints of the operator.

1. *Work Domain Analysis* involves developing a model of causal relationships between the overall system, its subsystems and their constituent components. This model is a *field description* based purely on the environment constraints imposed on a system that dictate how its goals may be achieved. It defines system goals at multiple levels of abstraction but does not describe events or actions carried out during work. The Abstraction Decomposition Space (ADS) is the modelling tool used carry out the analysis and will be described in more detail below.

2. *Control Task Analysis* defines the information processing required to control a functional system. It differs from other forms of task analysis in that it focuses on *what and why* rather than by whom and how information processing occurs. Recognising whether a goal has been achieved involves the transformation of low-level data into higher-level information and a modelling tool called the *Decision Ladder* is used to trace how this can occur. This approach has the advantage of being agent independent, allowing a designer to examine information processing irrespective of whether it is carried out by a human or automated controller.

3. *Strategies Analysis* is used to describe work activity. While control task analysis defines what needs to occur, strategies analysis explains how events can occur in reality. As sociotechnical systems can involve complex coupling between components there can be multiple ways to complete a task. Information flow models can be used to outline various strategies involved in accomplishing goals.

4. *Social-Organisation Analysis* is used to indicate the relationship between management structures and system functionality. CWA achieves this by superimposing different roles onto the three modelling tools described above.

5. *Worker Competencies Analysis* uses the outputs of the previous stages to define the levels of knowledge required by an operator to control the system. This has important implications in terms of staff training and interface design.

CWA has been applied to a number of different domains but it is most frequently associated with process control systems. Three important concepts; the SRK taxonomy, the integration of functional and physical descriptions and the modelling of control tasks form the basis of the framework and warrant further explanation

2.3.1.1 Application

The Skills, Rules Knowledge taxonomy defines three levels of cognitive control that a user can exert over a system (Rasmussen, 1983).

- *Skills Based Behaviour* (SBB) involves reactive behaviour to real-time system data. It describes how an expert user responds to temporal information about system components to maintain stability. Generally these actions are so fluid that they become instinctive and are not verbalised by users.
- *Rule Based Behaviour* (RBB) relates to procedural tasks that follow a plan of action. The behaviour is triggered by signs describing system performance and guided by rules defined by the constraints of the system. While they can be learned and practiced to the point of fluency, the actions can generally be recognised and verbalised by the user.
- *Knowledge Based Behaviour* (KBB) relates to higher-level decision-making. It generally involves reasoning at multiple levels of abstraction and requires knowledge of the complete relational structures in the system. KBB is used in fault diagnosis and performance analysis.

The ADS provides a model of functionality that describes the system at different levels of abstraction. This allows it to be used to identify the information necessary to support skills, rules or knowledge based behaviour respectively. It combines a means-ends functional abstraction hierarchy with a part-whole physical decomposition of a system (fig. 2.5). These two hierarchies are placed orthogonally in a matrix, essentially mapping function to form at different levels of granularity. The configuration makes the causal relationships between system, sub-systems and components explicit. Figure 2.6 illustrates how this occurs using the ADS for DURESS II, the thermal-hydraulic process control system around which this modelling tool was originally developed. Each cell describes the entire system at a different level of abstraction. The functional purpose of the overall

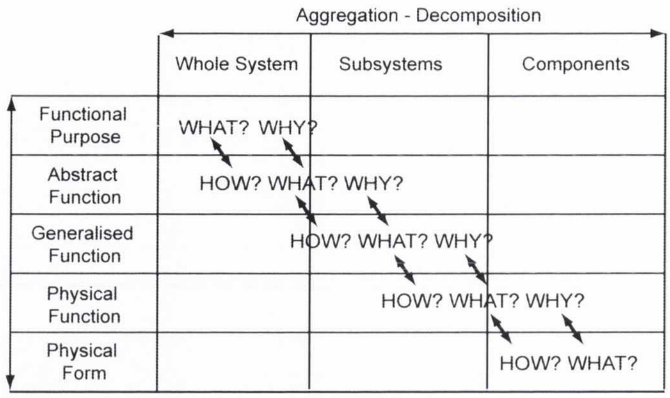


Figure 2.5: Means-ends relationships in the ADS

system is to output water at a specific volume and temperature. The abstract function describes the sub-goals involved in achieving this in terms of energy flows and relates how these flows are handled by subsystems. The generalized function specifies the two energy flows of heat and volume and identifies the components involved in their associated subsystems. The physical function describes the actions carried out by individual components and physical form identifies their configuration in the overall system.

The Decision Ladder models the decision-making process involved in controlling a system (figure 2.7). The left hand side of the ladder represents the activity associated with evaluating a systems state while the right hand side shows the activity involved in executing a response. Routine skills-based tasks involve very little cognitive work as alerts about a state can be immediately responded to with an action. However, as tasks reach higher levels of complexity with more variables and constraints, control moves from skills-based towards knowledge-based behaviour. The decision ladder uses two structures to model cognitive activity involves in supporting this: *states of knowledge* and *information processing activities*. By carrying out information processing data can be transformed into higher-level information, which can be evaluated against system goals. In this way causal reasoning can be understood as a progression up through the abstraction hierarchy, with higher-levels of the ADS corresponding to the higher-level states of knowledge required in the decision ladder. As system operators become experts, their understanding of causal relationships in a system increases. This can allow them to understand the system state without the need to reason about higher level relationships. This understanding is represented within the decision ladder as a *cognitive leap* across the decision ladder

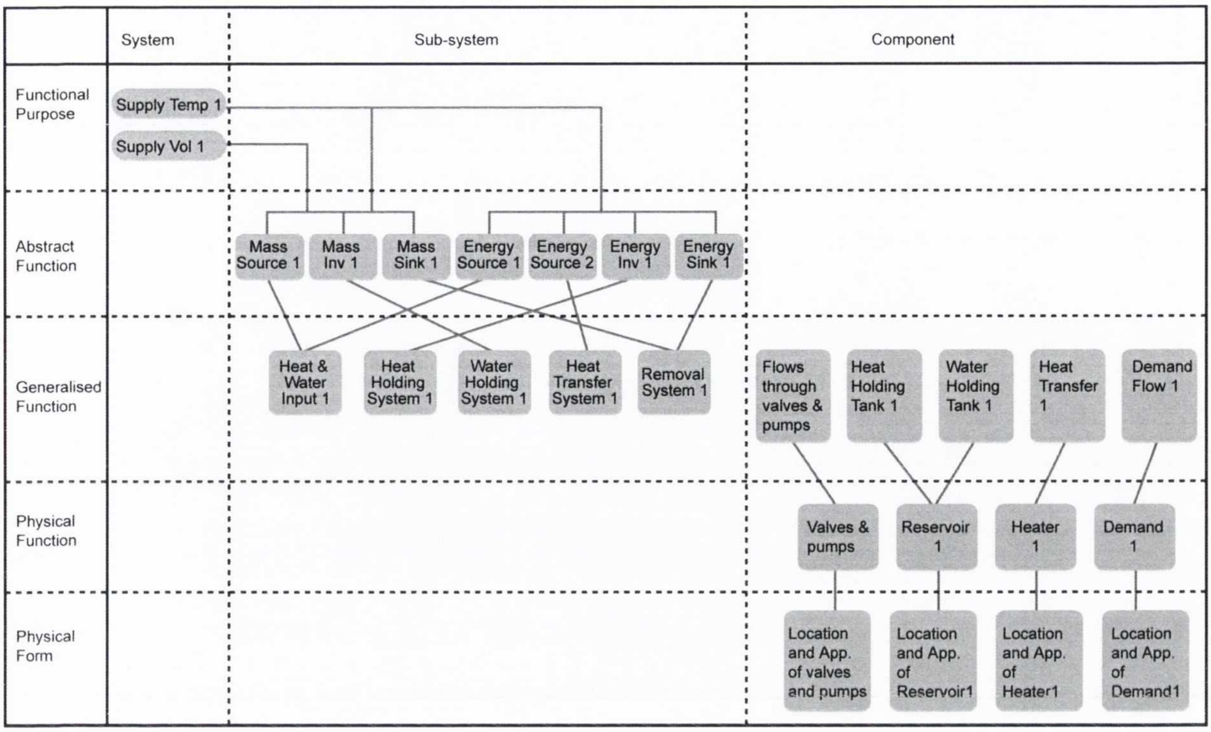


Figure 2.6: The ADS for the Duress II microworld (Bisantz and Vicente, 1994)

between an evaluative state and an executive process.

2.3.1.2 Limitations

CWA is one of the most comprehensive analytical frameworks for cognitive systems but it is subject to a number of limitations that may restrict its utility in relation to a large scale sociotechnical enterprise such as high volume manufacturing.

An initial limitation relates to the deployment of embedded control systems within large enterprises. CWA was developed around a model of process control where

- The process constraints can be described by physical laws
- Control relates to operating within these constraints
- The role of the controller is to maintain equilibrium by balancing constraints under different supply and demand situations

However the HVM domain introduces additional complexity through the use of embedded control systems. Autonomous systems with built in feedback loops are widely applied

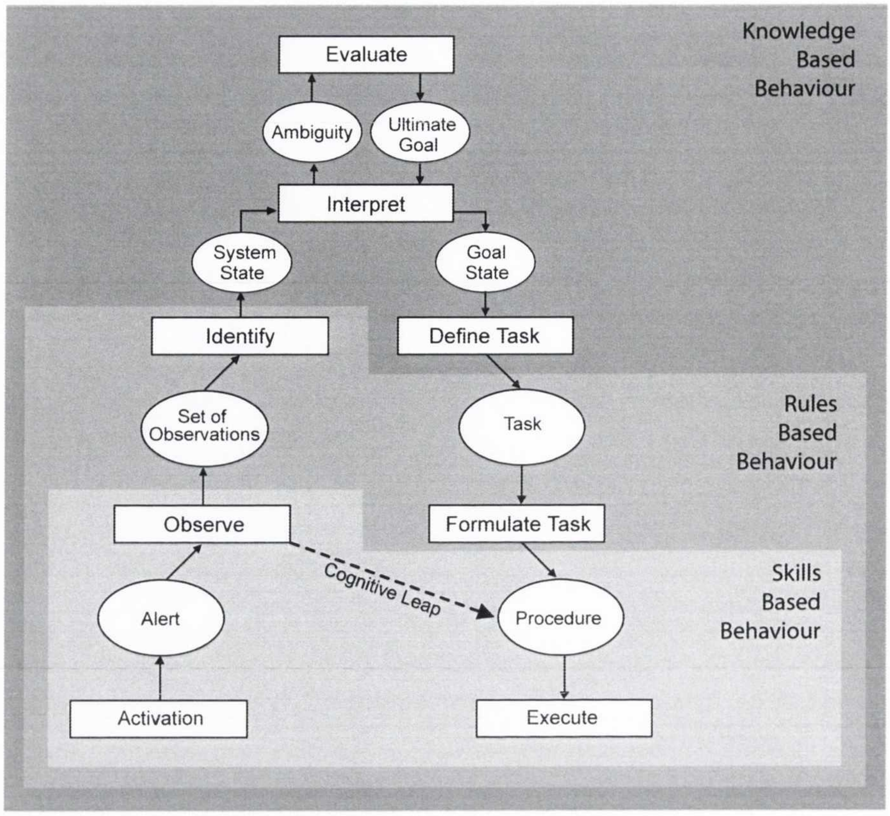


Figure 2.7: The Decision Ladder

to measure the output of equipment or sections of the process. These systems tend to focus on quality control issues rather than operations management and therefore have different invariant relationships to the overall manufacturing system. Given these different positions it is questionable as to whether they can, or should, be integrated into a single abstraction hierarchy.

Lind describes a methodological problem in relation to the abstraction hierarchy's integration of the process and control aspects of a system (Lind, 2003). This approach is justifiable in systems where human operators have direct control over a process, as in these situations the ADS provides an accessible conceptual model of functionality. However, where embedded control systems are part of a work environment, a monolithic representation that merges process and control systems can result in an inaccurate system model. HVM enterprises use embedded control at multiple levels of management so the applicability of the ADS is questionable in this case.

Another issue relating to the ADS is how to decide the manner of the system decomposition. Generally a physical decomposition of an engineered system is used, but

Rasmussen originally suggested that conceptual divisions may also be used where the focus of the system is not mechanically oriented (Rasmussen et al., 1994). As was identified in section 1.2, control of a HVM enterprise involves the separation of responsibility across different departments each of which has a different perspective on the system. As control moves to higher levels of abstraction it becomes necessary to integrate these perspectives, which will involve resolving alternative conceptual views of the system. There are no existing recommendations for how an ADS can be constructed to deal with this.

The type of decision-making used in manufacturing control also creates difficulties for CWA. As the framework was originally developed around process control, its view of decision-making is based around the diagnosis of faults in physical systems. This involves causal reasoning that traces relationships between system components. With manufacturing operations the physical process is part of a larger system of constraints. Fluctuating markets, client relationships, labour issues and process development all have an affect on the control strategies, but these cannot be revealed by an ADS that is based purely on the physical system. Much of the decision-making involved at management level is intentional and involves resolving these external constraints with internal ones. However CWA has been shown to have limited utility for intentional domains (Wong et al., 1998).

A final criticism is that while specific models are provided for work domain and control task analysis, the latter three phases are only vaguely defined and do not commit to any particular tools or analytical approach (Cummings, 2006). Strategies analysis makes reference to information flow diagrams but stresses that these will be context dependent and will vary between domains. Social organization analysis has been carried out by indicating how roles relate to work domain and decision models, but existing examples have focused on small systems and it is not clear how this can be carried out with much larger enterprises. Workers competency analysis makes reference to the SRK taxonomy and provides some high-level design principles but again their application will be context dependant making them difficult to re-use. The ambiguity about the latter stages of the framework has meant that many applications of CWA only make use of the first two phases. These may provide enough information to inform the design of control displays for small scale engineering systems, but with HVM enterprises better models of the social and external constraints are required.

2.3.2 Distributed Cognition

The Distributed Cognition (DC) framework integrates theories from cognitive science and social anthropology and has been used to analyse work systems in context (Hutchins, 1995a,b; Hollan et al., 2000). The basic premise builds on the information-processing concept of a problem space, but while a cognitivist view depicts the problem space as an internal mental construct, DC extends the boundaries of the problem space to incorporate *knowledge in the head* and *knowledge in the world*. It proposes that real-world problem solving involves the coordinated use of knowledge structures in the mind, in our environment and from other individuals.

The problem space associated with work system is described using a hierarchy of goals (figure 2.8a). Information is studied in terms of its *representation*, the manner in which it is propagated and transformed to achieve a system goal. Representations can be internal, in the form of an individuals knowledge state, or external, in the form of a cognitive artefact or information tool. For example, a worker may solve a problem using an internal memorised procedure or they may refer to an external procedural checklist or they may use a colleagues expertise to access a procedure. The concept of representation is more tangible than information and can be used to structure descriptions of the work system gained through observational study. Problem solving in DC systems involves the transformation and subsequent communication of system representations by an individual or artefact (figure 2.8b). By extending the problem space beyond the individual the framework can describe how a social organisation supports learning and can comment on the role of artefacts, both visual and physical, in complex systems.

Individual distributed cognition and socially distributed cognition are two approaches to DC research that differ from both a practical and methodological perspective (Perry, 1999). Individual distributed cognition occurs between people and artefacts in their environment. Experiments using well-defined puzzles provide quantitative evidence to show how different visual representation of data can encode more or less knowledge structures for a specific task (Zhang and Norman, 1994). Socially distributed cognition occurs be-

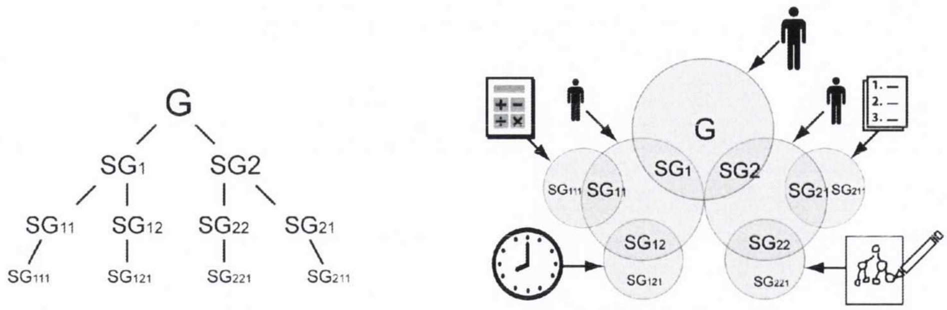


Figure 2.8: Abstract problem space and distributed cognitive system

tween people working towards a common goal. It uses qualitative, ethnographic methods of investigation to reveal how information flow and transformation occurs during cognitive work (Ackerman and Halverson, 1998). Despite the practical and analytical differences between these approaches they share a common agenda, to show how the problem space structures are distributed throughout a work system.

2.3.2.1 Application

DC began as a psychological theory but its focus on real world problem solving makes it a useful framework for studying the design of digital tools and collaborative work environments (Hollan et al., 2000). Since its inception DC analysis has strongly aligned itself with ethnographic methods of analysis. This is a holistic approach that accepts that the properties of a system cannot be studied independently in a laboratory and that analyses of work systems must be carried out "in the wild". This requires long-term, immersive fieldwork that produces rich descriptions of behaviour and practices. For example Hutchins' much cited analysis of naval navigation was carried out over several months and included examinations of the social hierarchy associated with command, the use of redundant knowledge structures to support learning and how data is transformed into information by navigation tools (Hutchins, 1995a). Descriptions of real world activity are analyzed to reveal patterns of behaviour and develop hypothesis about the cognitive system. From these analyses models of system functionality are developed. DC does not prescribe to any specific modelling techniques but with collaborative work systems it is usual to produce models of the social structures, information flows, physical environment and designed artefacts. In comparison to CWA, DC can be seen as a more bottom-up approach to revealing system functionality, where the actions carried out on the ground level are modelled to gain an understanding of higher-level goals. This has the advantage of providing a better understanding of the soft constraints such as social policies and their implications for system control.

2.3.2.2 Limitations

The application of DC has largely focused on understanding teamwork in collaborative environments and analysing the manner in which tools support work. Its approach is more descriptive and explanatory than predictive, and design guidance comes from interpreting observations in order to develop implications for design. This approach is subject to a number of limitations that may affect its application to HVM. Firstly, ethnographic study is a lengthy procedure and requires extensive access to users in their workplace. This limits the number of roles that can be researched in any level of detail. HVM enterprises are large-scale systems with hundreds of workers distributed over a wide physical area. Even if a only small sample of archetypal workers are selected, ethnographic analysis would still be difficult to conduct. Secondly, DC does not provide prescriptive modelling tools for describing system functionality. While it provides the theoretical framework for a wide range of work system studies, the specific modelling techniques vary between work domains. This lack of a stable analytical process makes it difficult for a system designer to integrate DC into a design methodology.

2.3.3 Activity Theory

Activity theory views human cognition as a complex, socially situated phenomenon. The theory originated from the Russian school of cultural-historical psychology that emerged in the early half of the 20th century and its concepts have influenced both of the previous CSE frameworks. Again the theory opposes a purely cognitivist or behaviourist view by describing human consciousness as occurring within the context of activity and maintains that analysis must focus on cognition in praxis. The basic unit of analysis for cognition is described as an *activity* that is carried out by a *subject* to transform an *object* into an *outcome* through use of a *tool* or *mediating artefact* (Leontev, 1978). The activity itself

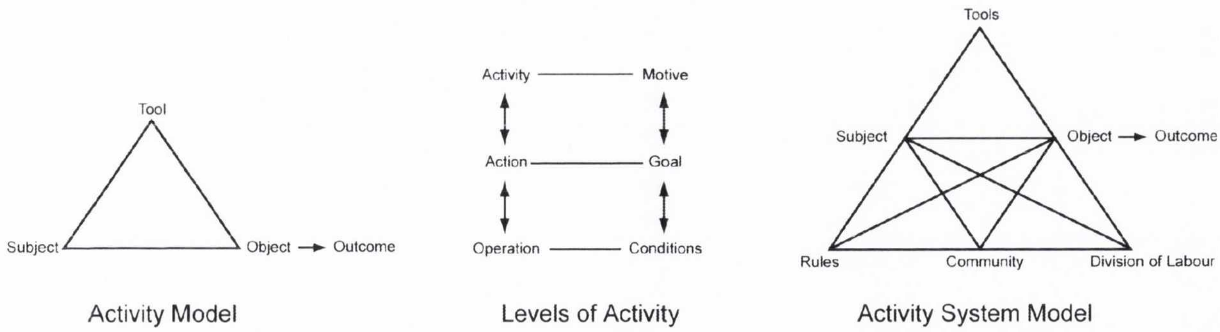


Figure 2.9: Activity model, levels and system

is driven by a motivation, which can be broken down into lower level goals. These goals are achieved through conscious actions such as planning. Actions are made up of routine operations. These are fluid, learned actions that are carried out unconsciously but which occur in the context of conditions or constraints. It is important to realise that these levels are not strictly exclusive but are dynamic. For example an operation may become an action if one of its conditions is not met and requires conscious planning (see figure 2.9). Engestrom expanded the original individual model to incorporate a community of workers, setting up new relationships in an activity system model (Engeström, 1987). In this model, tools mediate between subject and object, rules mediate between subject and community and division of labour mediates between the community and the object. While the triangular model may appear stable, it is important to remember that the object of the activity may change depending upon the subject even within the same system. These changes can lead to *contradictions*, which are seen as important to learning and development in work systems.

2.3.3.1 Application

Activity theory has received much attention in recent decades due to its focus on describing complex cognitive aspects of education and work (Nardi, 1995). While the theory has been applied to a number of interface projects, its role in the design process has been described by Nardi as "...a powerful and clarifying descriptive tool rather than a strongly predictive theory." What the analysis aims to deliver is an understanding of social consciousness and *motivation* rather than a labelling of tasks and actions. For this reason it is thought that activity theory may be more useful in the analysis of intentional decision-making than the previous two analytical frameworks (Nathanael et al., 2002). Similar to DC, activity theory uses ethnography to gather information about a work system. The activity checklist is a research tool that was developed to assist designers in using activity theory principles when examining a work system (Kaptelinin et al., 1999). It provides a list of topics relating to tool mediation gathered under the main themes of:

1. Means and ends—the extent to which the technology facilitates and constrains the attainment of users' goals and the impact of the technology on provoking or resolving conflicts between different goals.

2. Social and physical aspects of the environment—integration of target technology with requirements, tools, resources, and social rules of the environment.

3. Learning, cognition, and articulation —internal versus external components of activity and support of their mutual transformations with target technology.

4. Development—developmental transformation of the foregoing components as a whole.

The checklist also places these themes within a series of questions that a designer can use during an interview with system users. Following ethnographic studies the activity model can be used to analyse descriptions of a system. Kuuti outlines three areas where this can help in the design of interactive systems (Kuutti, 1995). Firstly, it highlights the multi-levelness of interaction. The distinction between operations, actions and activities and the relationships between these levels provides a means for understanding how low-level interactions relate to higher-level strategies. Secondly it allows interaction to be studied within a *social context*. AT's strong emphasis on tool mediation and the cultural history of tools supports a rich understanding of context. It provides structures for analysing the context in which work occurs. Thirdly it deals with dynamics and development. Usability plays an important role in HCI research, but while post-hoc evaluations can give performance metrics they tend to be short term and do not take into consideration *the development of expert skills*. Again, an emphasis on work context and a tools cultural history provides scope for integrating developmental aspects into interfaces.

2.3.3.2 Limitations

As with the previous frameworks there are a number of factors that limit the utility of activity theory from studying a HVM enterprise. Firstly, as the framework relies on ethnomethodology it suffers the same drawbacks as DC in relation to dealing with large-scale systems. It is unfeasible to carry out activity interviews with all of the workers within the HVM environment. A more fundamental issue relates to the analysis of co-operative work. Despite the extensions made by Engstrom, Kaptellinin states that the framework *is still strongly focussed on the mental models of individuals* and the factors that affect their development (Kaptelinin, 1995). However, manufacturing systems involve the co-ordination of teams of workers which is difficult to represent within the activity system model.

A final but fundamental issue is that the current theory is not operationalised enough to make it widely accessible to the design community (Rogers, 2005; Kaptelinin, 1995). The concepts of activity, objects and subjects are ambiguous terms that can be interpreted in different ways. It will be necessary to provide more clearly defined analytical tools if the framework is to be of use to the development of a design methodology.

2.4 Analytical Gaps in CSE Knowledge

Cognitive work analysis describes a system in terms of an abstraction hierarchy where physical and engineering constraints define the functional structure of the work system. Distributed cognition portrays a work system as a distributed problem space where agents and artifacts process and transform representations to achieve higher-level goals. Activity theory views the system as an activity made up of a subject and tools that transform an object into an outcome. Each provides a valid method of describing a work system, but each perspective tends to reflect the academic fields and work domains around which they were originally developed. As a result cognitive engineering exemplars are generally presented from a particular stance. Projects that focus on the control of mechanical systems move towards CWA, those concentrating on teamwork situations are more DC oriented, while those that deal with intentional decision making tend to look to activity theory. The result is a range of different approaches to cognitive engineering, each of which targets a particular type of cognitive system (Nathanael et al., 2002). This approach is unsurprising as the heterogeneity of tasks, social systems and goals across different domains will make certain perspectives more appropriate than other. However, this means that a designer must choose an analytical approach at the outset of the design process. This requires some way to map different types of cognitive systems depending on their characteristics.

2.4.1 Mapping Cognitive Systems

The dichotomy of coherence and correspondance driven domains is a useful concept for moving from a human computer interaction view towards one of human work interaction design, but in reality cognitive systems can be described using a range of different factors. Rasmussen provides a map of cognitive systems using the categories of *work domain*, *tasks* and *users* to differentiate between classes (Figure 2.10). Work domains are classed along a continuum between natural or intentional environments on the left and highly-structured or causal environments on the right. This division has qualities similar to Simons distinction between ill-defined versus well-defined problems (Simon, 1996) and Checklands soft and hard systems (Checkland and Scholes, 1990). The distinction between types of work domains is further clarified by their association with different types of users. On the left, human actors are autonomous users who are served by the system. Examples of this include general information systems where control is defined by the user's tasks. On the right the user is defined as an operator who serves the system. As control is dictated by system goals, their tasks and pace of work are all determined by the system design. In the centre, users are shown as autonomous agents who operate within constraints. Here the constraints are defined by policy rather than physical laws and the user maintains some discretion over their behaviour. Based on these definitions it appears that each of the analytical frameworks discussed can be matched to three distinct user types located along a continuum between intentional and causal systems. AT's focus on motivation makes it more suitable for intentional system and fully autonomous users. CWA's focus on means-ends relations makes it more suitable analysing operators in causal systems. DC's focus on representations makes it suitable for modelling users who must work within

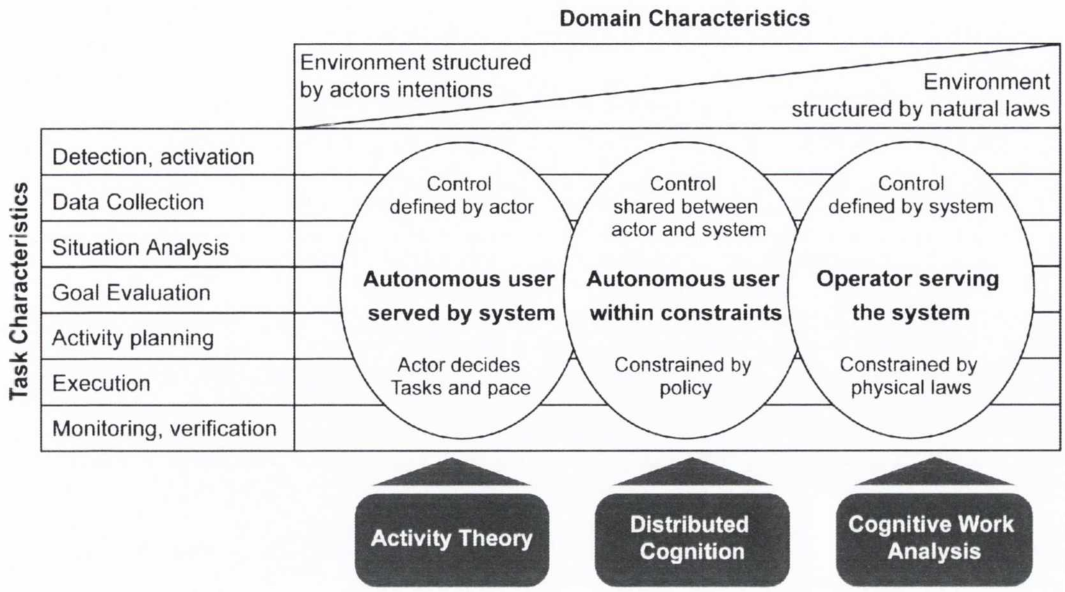


Figure 2.10: Work domain, Task and User (WTU) map of cognitive work domains aligned with analytical frameworks. After (Rasmussen et al., 1994)

policy-driven constraints.

However, despite this apparent correlation, in reality cognitive systems are not so easy to categorise. As was shown in section 1.2.1 managing a HVM enterprise involves the control of a highly specified causal system, the resolution of less defined scheduling issues and intentional decision-making in relation to operational goals. While these activities were previously associated with specific roles at different levels of management, increasing levels of automation means that an individual worker may have to align causal reasoning with intentional goals. This reveals a gap in CSE analytical knowledge as there is no overarching methodology for analysing such systems. Maramas suggests that in reality it is necessary to amalgamate approaches to cope with the complexity of real-world work systems (Marmaras and Nathanael, 2005) but it is difficult to know how to combine these alternative views. In order to combine the models generated from the range of analytical frameworks it is necessary to identify how they contribute to the overall design process.

2.4.2 Informing Visual Design

The analytical frameworks described in this chapter were developed to examine real-world work situations in-context. They aim to reveal the ecological constraints imposed on the cognitive work system and to identify valid information requirements. They take an explanatory and descriptive approach to human work and problem solving and offer some opinion on the role of information representations in cognition. Despite this, their utility for informing design practise is questionable. Rogers revealed that although many industry practitioners are aware of these theories they do not employ them in their day-to-day work. She cites one designers comment that most of these approaches are *"...difficult for designers to use and generally too theoretical to be relevant to a practical, human-focused solution, developed in the timeframe of a design project"* (Rogers, 2005). The difficulty in applying these frameworks in the practice of design can be attributed to a number of factors. Firstly theory does not do design itself, at best it can provide methods and guidelines that are interpreted by designers who then employ them to meet their design goals. Secondly, it takes time for theory to be translated into practice. Until the theories have been successfully applied to a wide range of suitable case studies, they will not be taken up by the general design community. Thirdly, the ethnographic approach taken by many current frameworks requires considerable time and effort. This must be balanced against the budgets encountered in commercial work. Finally and most importantly, there is little consensus about what the *design outputs* of these various frameworks should be. It is relatively clear that each framework reveals information requirements and places them within a system model, but these models are open to interpretation. More often than not these analyses produce "implications for design" that must be transformed into specific visual design goals by the interface designer. However none of the frameworks provide visual encoding guidelines for achieving this.

2.5 Summary

This chapter addressed the first challenge in designing visual decision support systems for achieving enterprise observabililty, that of developing a model of functionality. Three prominant cognitive systems engineering frameworks have been reviewed. While each framework takes a systems view of cognition, their perspectives are suited to particular types of work domains and different approaches to problem-solving. Control in a HVM enterprise requires a distributed cognitive system that can deal with both intentional and causal goals. The lack of a unifying framework that can model such a system signifies an analytical gap in CSE knowledge. In an effort to bridge this gap, the utility of analytical frameworks for informing design practice has been examined. Currently the relationship between system analysis and system representation is difficult to define. In order to resolve this, the next chapter examines the representational principles used in CSE.

Chapter 3

Representing Cognitive Systems

The Graphic User Interface has been described as a system image from which a user develops a mental model of system functionality. While the analytical frameworks aim to develop models of system functionality, the communication of this model to a user require an interface designer to graphically encode values and relationships into a meaningful representation. In this chapter a range of visual display principles and guidelines are reviewed. Their advantages and limitations in relation to the design of visual decision support for high volume manufacturing are identified. A map of visual design research is developed that identifies representational gaps in CSE design knowledge.

3.1 Designing a System Image

How can a designer transform their conceptual model of functionality into an effective system image? Norman proposes that *affordances* play an important role in communicating conceptual models when designing physical tools and functional objects (Norman, 2002). An affordance is a quality of an object, or an environment, that makes it possible for an individual to perform an action; for example a spoon affords grasping and scooping up food (Gibson, 1979). The term was extended to incorporate perceived affordances, where the appearance of an object suggests how it can be used. A dial for instance, suggests that it should be turned rather than pushed, pulled or slid. While the term is frequently used in relation to interface design, Norman states that *"affordances, both real and perceived, play very different roles in physical products than they do in the world of* *screen-based products. In the latter case, affordances play a relatively minor role: (while) cultural conventions are much more important"* (Norman, 2007). To get around this apparent misuse of the term, four principles for generating a conceptual model for screen design are provided:

1. Follow conventional usage, both in the choice of images and the allowable interactions.
2. Use words to describe the desired action. i.e. menus and labelled buttons
3. Use metaphor to convey interaction concepts. e.g. the desktop metaphor
4. Follow a coherent conceptual model so that, once a part of the interface is learned, the same principles apply to other parts.

The fourth guideline faces a bootstrapping issue of how to develop and establish the conceptual model in the first place. Norman's solution is, to follow the initial three steps; i.e. convention, words and metaphor (Norman, 2007). These principles are based around coherence-driven work and aim to generate a simplified conceptual model through analogy and conventions. However, this approach cannot be used for correspondance-driven domains, where the system image must provide an accurate model of system functionality in order to facilitate control (see section 2.1.3). These principles focus on the selection of images, words and metaphors but the system image for a complex work system must attempt to encode the functional constraints in a manner that is interpretable to the user. While analysis of cognitive systems can reveal constraints in the form of information requirements, the visual design challenge is to represent this information in a format that communicates the relationships between these constraints.

3.1.1 Interface as Sign System

Control interfaces provide measurements of system parameters, but while a controller monitors these parameters, their interest is not with the values per se but rather what these values mean in the context of the functional system. Work systems are designed to achieve specific goals, and these values act as signs that describe the system state in relation to these goals. In this manner the interface can be described as a *sign system* (Nadin, 1988) that mediates between a controller and a work domain. A controller must be able to interpret the meaning of these signs if they are to maintain control. From a semiotic perspective, sign interpretation involves a three-way relationship between the *object* that is represented, the *representamen* (that which represents) and the *interpretant* (Peirce, 1931). For interpretation to occur, the interpretant must have knowledge of the objects that are represented, the object must be suitable for some form of representation and the representamen must be able to express itself in a manner that is perceivable to the interpretant (figure 3.1). These relationships allow signs to be described at a number of levels.

- Syntax relates to the internal structures used by the representamen. It describes the elements of a representational format i.e. letters, numbers, colours etc. and the relationships these bear to one another. Syntax describes the rules that are used to construct verbal, visual or even haptic languages.

- Semantics describes the relationship between the representamen and the object. It relates to the power of the representation to capture properties of the object. As different syntaxes can encode more or less detail, a range of different representations may be generated.

- Pragmatics relates to how the sign is interpreted. It places the sign within a context of use and describes its power in communicating relevant information. The pragmatics of a sign can change depending on both the interpretant and the context of use.

Semantics plays a central role in communicating meaning but different signs communicate meaning in different ways. Peirce provides a further classification of signs based on the manner of semantic encoding employed (Peirce, 1931).

- Iconic signs use pictorial representations to provide a visual resemblance between the representamen and the object.

- Indexic signs use characteristics of the object rather than visual qualities of the object itself.

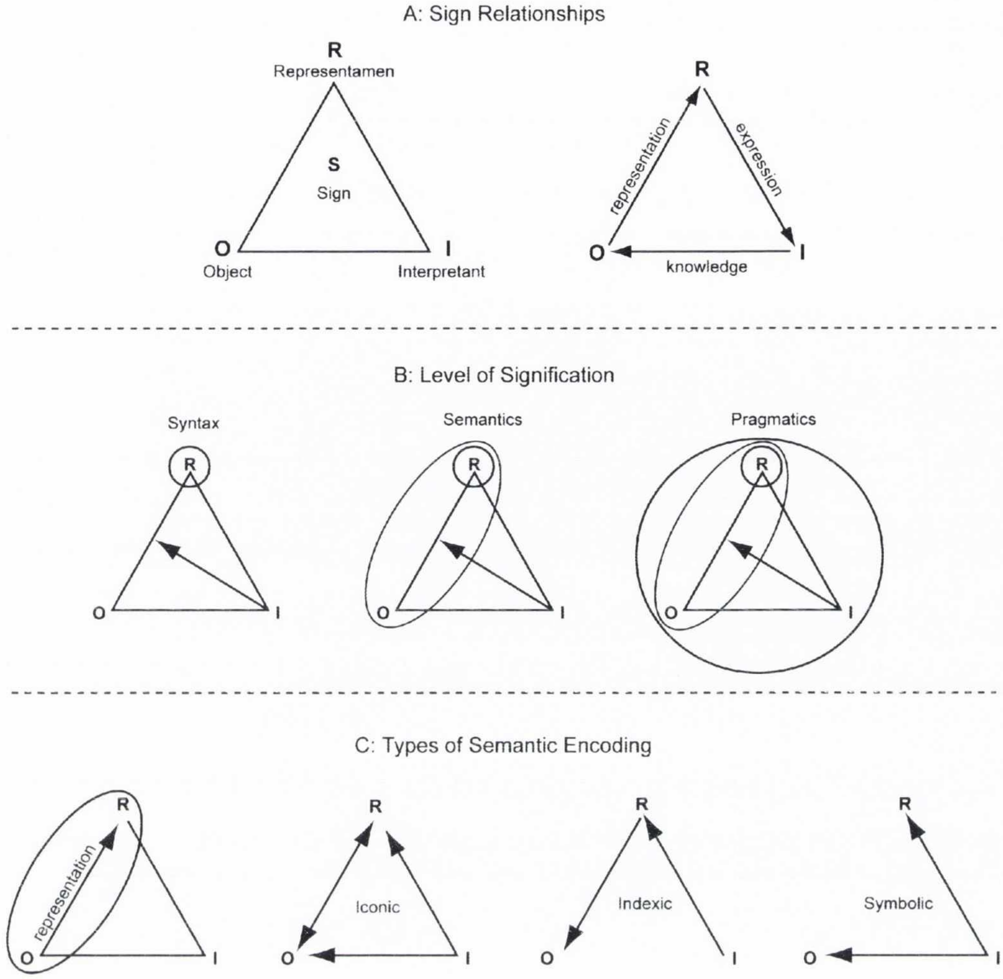


Figure 3.1: Sign relationships, level of signification and types of semantic encoding. after (Peirce, 1931)

- Symbolic signs use abstract tokens that have an established conventional meaning.

These classifications provide a means for describing alternative representational formats that can be used in control displays.

3.1.2 Representational Formats and Semantic Power

Information can be represented in many different formats but the choice of format has serious implications for control. Woods provides a striking example from the Apollo 13 space shuttle incident, where an explosion caused by a system fault almost resulted in disaster (Woods et al., 1999). The incident is traced back to a mission controller responsible for monitoring a range of values relating to the onboard cryogenics system. The control display consisted of 54 changing values presented on a single screen (fig.

3.2a). When a controller failed to recognise unusual changes in one pressure reading, the continuing rise in pressure resulted in an explosion that finally alerted mission control that a problem had occurred. While all of the necessary data was available the analyst did not respond to the information. This examples can be used to differentiate between *data availability* and *system observability* (see section 1.1.2). Woods argues that this difference can be attributed to the semantic power of the representational format and distinguishes between three different display types (Woods, 1991, 1995).

Text based or *propositional representations* can be considered to be a symbolic sign system as the visual relationship between tokens and information is arbitrary. Interpretation must occur at a number of levels. Initially the alphanumeric symbols must be read to establish the value of the sign. Subsequently the meaning of this value must be interpreted in terms of the goal that the data aims to support. In the case of the shuttle incident, the value related to a pressure reading that should have remained within predetermined control limits. While the propositional representation gives a precise reading of the current value, it does not provide any context for interpreting this value.

Iconic representations are widely used as action cues within interface design. However their interpretation requires a viewer to draw on tacit knowledge or social conventional to make an association between symbol, state and appropriate action. In the shuttle example, the appearance of a warning icon in the display could have been used to indicate the pressure issue (fig. 3.2b). While this can directly communicate a system state it occurs only after certain critical thresholds have been crossed. Again a lack of contextual information, makes it impossible to support the continuous monitoring of behaviour normally associated with control.

Analogical representations refer to charts, graphs and meters. While these are generated from the same data as the previous two formats they provide considerably more information. The graphical encoding of the data allows it to be presented within a context. Rather than depicting the state through iconic or symbolic signs, the structure and behaviour of the tokens that make up an analogical representation *correspond* to the structure and behaviour of what is represented *through some natural constraint*. This concept of constraint correspondence provides a functional and temporal context for sign interpretation that allows the state to be *directly perceived* without the need for tacit knowledge. An alternative analogical representation of the pressure parameter (figure 3.2c) clearly demonstrates this formats ability to:

- Represent data within the context of related issues. The graphic representation can include upper and lower performance limits and past performance.
- Highlight changes and events. The gradual increase in the value becomes a visual trend in the display making it easier to detect.
- Highlight contrasts. The sudden plunge in the reading is made explicit in the graphic presentation. The context of past performance makes the cause of this failure explicit.

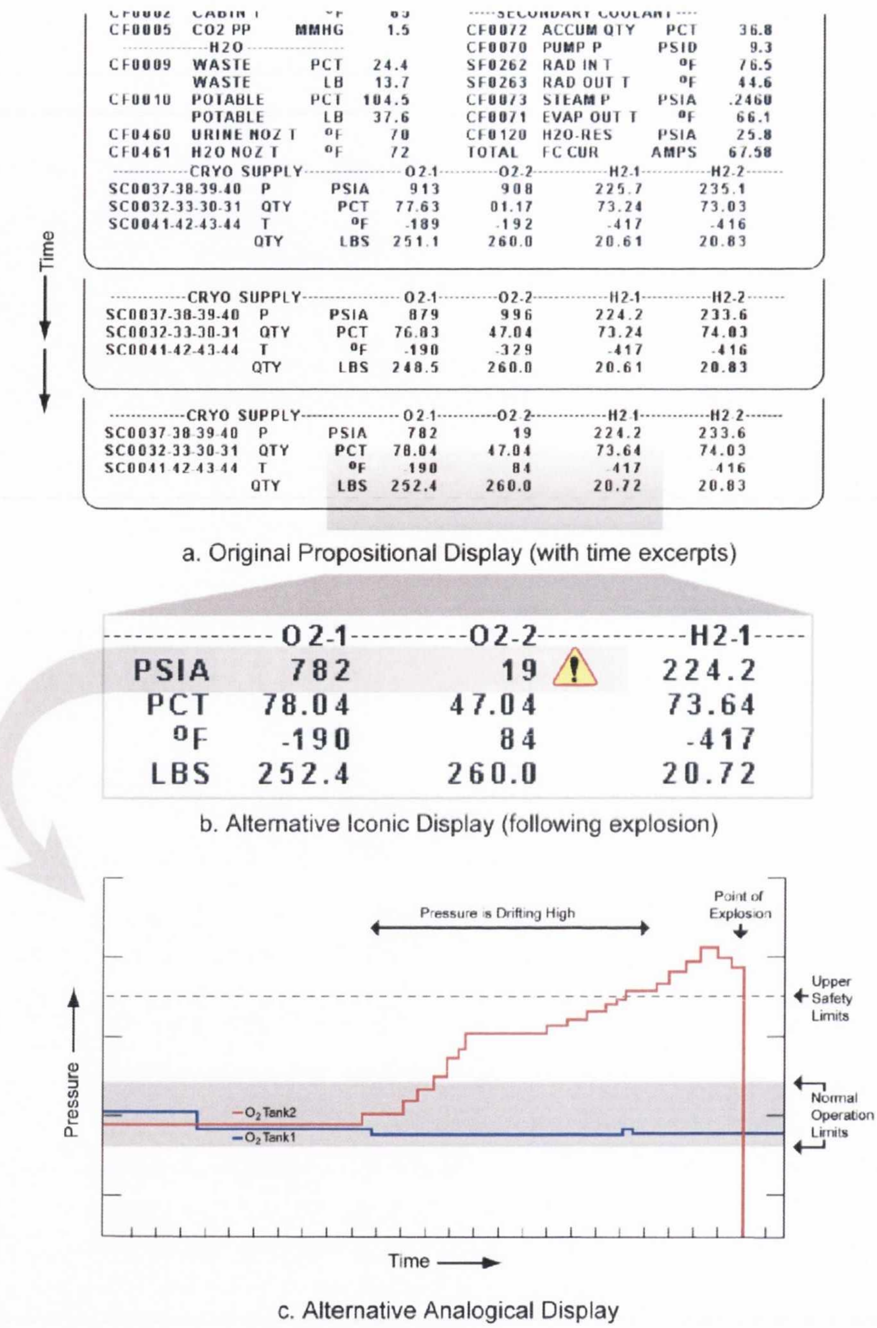
3.1.3 Representation of Complex Systems

These differences permit us to reintroduce the concept of affordance in relation to screen-based design. Different representational formats can be said to have different *cognitive affordances* (Zhang and Patel, 2006) as they provide different possibilities for interpretation. Propositional representations afford the precise reading of specific values or data signals. Iconic representations afford the communication of a particular state. Analogical representations afford the monitoring of behaviour in relation to goals.

The cognitive affordances provided by analogical representations make them the most appropriate representational format for supporting system observability. However, their application to control displays for complex systems creates a number of representational challenges.

First there is a spatial design challenge. The example above focuses on an individual variable but the mission controller was responsible for monitoring 52 different parameters. While the analogical display improved observability it occupies a much larger areas of the screen. How can a large number of analogical representations be presented within a single interface?

Second there is a visual encoding challenge. One solution to the spatial issue described above, is to use alternative analogical representations that presents the data in a more spatially efficient manner. A different graphical encoding could be applied to the data to



a. Original Propositional Display (with time excerpts)

b. Alternative Iconic Display (following explosion)

c. Alternative Analogical Display

Figure 3.2: Propositional, iconic and analogical representations. after (Woods et al., 1999)

achieve the same semantic goals. However, a multitude of graphic encodings are possible, so how can a designer know which one to select?

A final and more complex issue relates to achieving the higher-levels of semantic encoding necessary for controlling a complex system. The task above describes monitoring a single parameter, but complex systems consist of subsystems that are in turn comprised of a large number of tightly coupled components. This coupling describes the causal relationships that define how a system works and knowledge of these relationships is necessary to support higher level control tasks like fault diagnosis (see section 2.2). A system image for control of a complex sociotechnical system requires the generation of a display that shows both the state of components and their relationships to the functional system as a whole. So the final challenge is how to generate a display that makes the functional relationships within a system explicit?

3.2 Ecological Interface Design

Ecological Interface Design (EID) is a framework for designing control interfaces for complex systems (Burns and Hajdukiewicz, 2004; Vicente and Rasmussen, 1992). It was specifically developed to support workers when dealing with unanticipated events and fault diagnosis as well as handling normal operational tasks. The framework developed out of cognitive work analysis (see section 2.3.1) and uses the models arising out of work domain analysis to guide the design process. EID principles emerged from a particular understanding of how humans cope with complexity and a brief overview is provided here.

3.2.1 Coping with Complexity

To control a mechanical system it must be possible to affect and monitor variation in the processes that make up a system. Ashby's law of requisite variety proposes that the variety in a control system must be equal to or larger than the variety of the system being regulated in order to achieve control (Ashby, 1956). This implies that control systems are complex by their nature and must grow in complexity in proportion to the systems they control. With large-scale systems the potential for interaction between variables further increases this variety, resulting in a huge range of possible system configurations. This complexity makes it exceedingly difficult to develop robust automated controllers that can successfully respond to all system states. At the same time the scale and dynamic nature of these systems results in a level of complexity that would appear to prohibit human monitoring. Human controllers in highly automated systems can be directly or indirectly responsible for thousands of different parameters, presenting a highly complex control situation.

However Rasmussen and Lind suggest that the complexity of a system cannot be regarded objectively, but should be studied in terms of its representation (Rasmussen and Lind, 1981). As even very simple objects become complex when viewed under a microscope, complexity can be considered as a matter of perspective. The abstraction hierarchy outlined in section 2.3.1 provides a model for how humans cope with complexity when observing functional systems. It proposes that humans view systems at different levels of abstraction and move between levels when engaged in causal reasoning. Each level can be described as a representation that corresponds to system behaviour at that particular degree of granularity.

At lower levels of abstraction, information about individual components is provided. This level identifies quantitative values that act as simple *signals*. These report the performance of a parameter but not in relation to any higher-level system goals. When individual or multiple signals are represented relative to an expected behavior or target they become *signs* that support judgments about how well a component or subsystem is achieving its goals. At higher levels of abstraction, the configuration of multiple signs can be used as *symbols* that represent a particular system state[1]. These levels of representation can be related to the levels of cognitive control defined by the skills, rules, knowledge taxonomy (section 2.3.1.1).

Rasmussen provides an example of this in relation to the control of a valve using a flow-meter (Rasmussen, 1983)(figure 3.3). A flow meter consists of a measurement scale and a pointer indicating current flow levels. Given the task of stabilising the flow at a specific target the pointer acts as a signal. The analogical encoding affords temporal feedback that facilitates this skills-based control task. If the controller closes the valve

[1]Rasmussen's terminology for differs from that of Pierce. The transformation of information between of signal, sign and symbol all relate to the pragmatics of use rather than the semantic format of representation.

but the pointer still indicates a flow, the flow meter then acts as a sign. The discrepancy between the valve position and the flow-level breaks the rule encoded in their natural mapping i.e. that these values should match. This difference allows the controller to use rule-based behaviour to interpret it as a sign of mis-calibration between the meter and the valve. If the user recalibrates and the problem arises again the meter then acts as a symbol. It becomes an element in the overall system consisting of pipes, valves, meters etc. It signifies the presence of a leak or some other fault in the system, diagnosis of which requires knowledge of the causal relationship between components. This is an example of knowledge-based behavior. Full control in all situations requires a human operator to move between these representational levels.

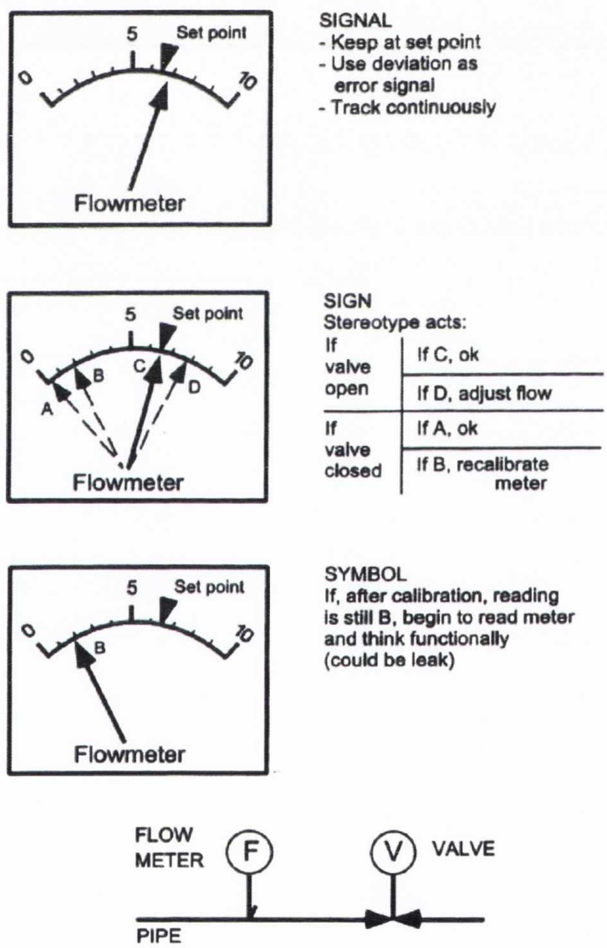


Figure 3.3: Analogical display component as signal, sign and symbol. (Rasmussen, 1983)

3.2.2 EID Visual Design Principles

The law of requisite variety makes it exceedingly difficult to specify every possible outcome that may arise within a complex system. This unlikelihood of being able to completely specify system states limits the application of fully automated control to complex systems. An alternative approach is to allow a human operator to respond to events using their understanding of the system state. Essentially this approach allows humans to use their problem solving expertise to respond to situations that were not predicted by the system designers. Ecological interface design is a design framework that attempts to support human decision making in these complex control situations. It does this by providing three design principles for achieving levels of cognitive control at the skills rules and knowledge levels. This concept of matching representations to levels of cognitive control forms the basis of the ecological interface design principles. These provide three high-level recommendations for the representation of information in control displays (Vicente and Rasmussen, 1992).

1. Skills Based Behavior – To support interaction via time-space signals, the operator should be able to act directly on the display, and the structure of the displayed information should be isomorphic to the part-whole structure of movements.

2. Rule Based Behavior – provide a consistent one-to-one mapping between the work domain constraints and the cues or signs provided by the interface.

3. Knowledge Based Behavior – represent the work domain in the form of an abstraction hierarchy to serve as an externalised mental model that will support knowledge-based problem solving.

The knowledge related principle results in powerful representations that support causal reasoning and problem solving. As the abstraction hierarchy is defined by system constraints (see section 2.3.1), representations that make these constraints explicit should support reasoning and tasks that go beyond standard operating procedures. In this manner, ecological designs can be used to cope with the inherent variety that exists in complex systems by allowing the operator to come up with strategies for closing the gap between the current and the ideal system state. These principles have been kept at a very high-level of abstraction to ensure generalisability to other domains but were originally explained in relation to the DuressII thermal hydraulic microworld that was outlined in section 2.3.1.1 (Vicente and Rasmussen, 1990; Bisantz and Vicente, 1994). The resulting ecological display is presented in figure 3.4 and will be used to clarify how these principles are implemented. The goal of the system is to generate a supply of heated water. The specific volume and temperature of the water is determined by an external demand and the human operator must control and/or monitor how this is achieved. The system consists of two redundant feed-water streams consisting of pumps (PA & PB) and valves (e.g. VA, VA1, etc.) that deliver water to two reservoirs. Two heaters (HTR 1 & 2), each associated with a reservoir, warm the water before sending it on to the external system. There is coupling between components as either or both feed-water streams can supply either or both of the reservoirs.

Skills based behavior is supported through direct manipulation of analogical representations of system components. For example, the valves have handles that can be dragged to their target points and monitored using flow meters. The focus of attention will move between regions at different stages of operation and component representations are laid out in a manner that reflects these movements through spatial clustering.

Rules based behavior is achieved by generating configural displays that reveal the system state at different levels of abstraction. A configural display combines two or more variables into a unified graphic form. For example, with each reservoir the flow of water is represented using a configural display that aligns an input (MI1) and output (MO1) flow meter with a volume meter (V1). This configuration generates a trapaziod shape that provides a direct visual cue to whether the volume in the resevior is increasing, decreasing or stable.

Knowledge based behavior is achieved through the composition of the display in a manner that reflects the ADS work domain model. For example, the functional purpose of the overall system is to generate a supply of heated water and this is indicated through the output temperature and volume variables. This purpose is achieved at the abstract function level by regulating water and heat transfers and the nature of this relationship is captured through the configural display described in figure 3.4. The factors that constrain the transfer of water and heat at the generalised function level are captured through the

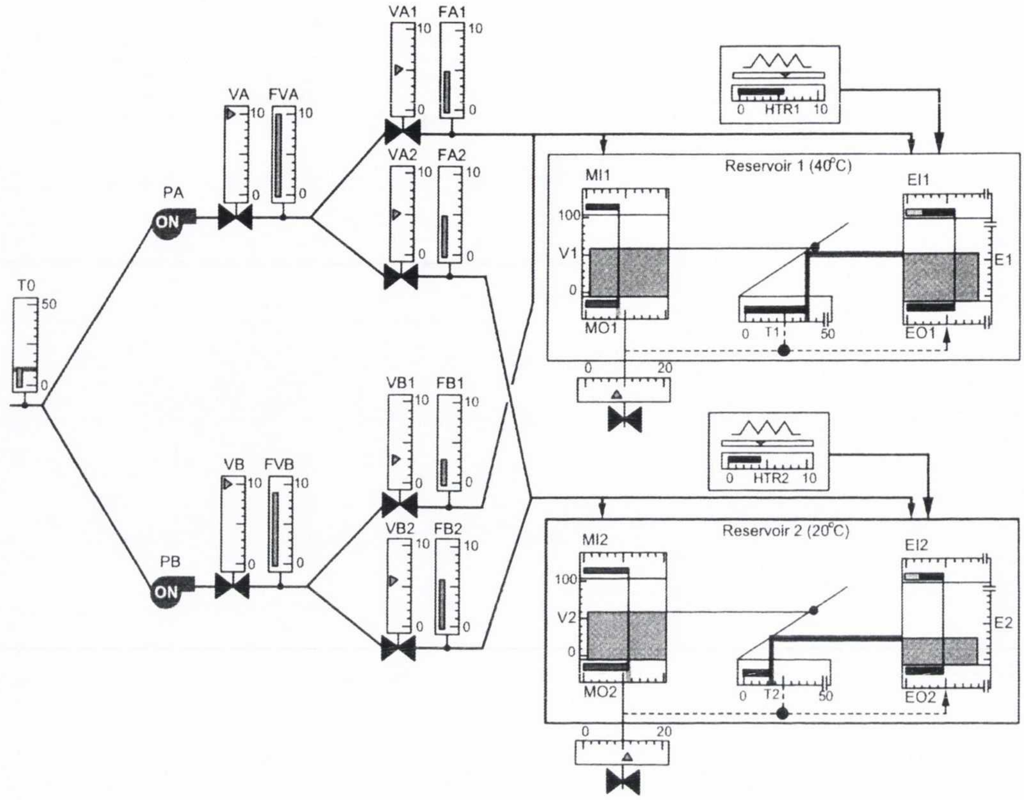


Figure 3.4: Ecological Display for DURESS II. (Vicente and Rasmussen, 1990)

input/output displays described in the previous paragraph, while the components that regulate input can be monitored and controlled through direct manipulation of their associated icons. In this manner an operator can trace the causal relationships that exist both within and between levels of functional abstraction.

The Duress system has played a central role in the justification, explanation and evaluation of EID principles. Numerous investigations of this interface supply detailed analyses of how visual representations support control tasks and provide a comprehensive rationale for the design of visual displays (Bisantz and Vicente, 1994; Vicente, 1991; Howie and Vicente, November 1998). One major advantage of the approach is its direct fit with the outputs of CWA. While most other analytical frameworks can only provide high-level implications for design, the ADS directly informs the visual presentation of system information. This connection provides an explicit means for transferring analytical knowledge to design knowledge.

Despite these advantages, the application of EID principles is not a straightforward process. One of the stated goals in developing EID's visual design principles was to ensure generalisability (Vicente and Rasmussen, 1992), but to make them broadly applicable it was necessary to define them as abstract goals that could be realized in various contexts. Consequently, although the principles deal with the generation of semantic meaning in an interface, *they do not recommend or comment on the visual syntax that should be used to achieve this.* This lack of specification means that designers must look to EID exemplars to understand how the principles can inform representational design. This can explain why EID has predominantly been applied to process control projects, where designers can use the syntax and design concepts outlined by the Duress study. While the framework has been used with other domains, including transportation and medical systems, these have presented major design challenges and require the principles to be interpreted and implemented in different ways (Burns and Hajdukiewicz, 2004).

3.2.3 Applying EID principles to HVM

From a control perspective a high volume manufacturing enterprise is obviously quite different from a representative microworld such as Duress and characteristics of the HVM domain may make it difficult to apply the EID principles. Three specific issues are discussed below.

Suitability of the System Model. As EID is closely related to the ADS modelling tool it suffers from the same limitations identified in section 2.3.1.2. Specifically this modelling tool is suitable for causal systems constrained by physical laws but is weak at modelling more intentional domains. As human controllers in manufacturing may have to consider both intentional and causal constraints during decision-making, it is questionable whether the ADS can provide all of the necessary information requirements. One study that focusses specifically on manufacturing scheduling, identified that systems influenced by social or temporal issues require a functional model based on activity as well as causal relationships (Higgins, 1998). However there are no existing guidelines for incorporating multiple system models into an ecological display.

Lack of Visual Vocabulary. EID was initially developed within the context of Supervisory Control And Data Acquisition (SCADA) systems. Traditionally SCADA development involves generating *mimic* displays that act as human machine interfaces. A mimic display uses *pre-existing, domain relevant icons* that act as graphic components in a control interface (e.g. the pump and valve icons in fig 3.4). The symbols are laid out to form a diagram that 'mimics' the physical relationships in the process being controlled. The Duress interface was developed as an enhancement of a mimic display, providing goal-relevant information at multiple levels of functional abstraction. However as a result, this exemplar does not comment on the visual encoding of components. This creates a challenge for systems that do not have existing mimic diagrams or standard component icons. Depending on the scale and the complexity of a HVM process, the system may or may not have an established set of icons describing system components. In these situations the designer must develop a visual vocabulary for the domain before EID principles can be applied.

Scale of System. A final major challenge relates to the scale of HVM systems. The principles were originally developed around a microworld, but as systems become larger and more complex these principles become more difficult to apply. The representation of the work domain model in the form of an abstraction hierarchy is a fundamental concept, but when a system has hundreds of components this becomes difficult to achieve. Even when symbolic notations for components already exist, the display limitations of the screen may make it necessary to represent them in an alternative manner. One solution to this challenge is to use information visualization techniques within an ecological design. Duez and Vicente provide an example of this in an ecological display for network management system (Duez and Vicente, 2005) (fig. 3.5). However this approach can only be used a) where a suitable visualisation technique that describes data relationships already exists and b) where this technique can be successfully incorporated within an ecological display. Where these conditions exist visualisation techniques can be applied in an opportunistic manner. Otherwise, the designer must attempt to design an ecological visualisation from first principles.

While the first challenge is a methodological one, the latter two are representational challenges that must be overcome in the generation of an ecological display for HVM. The EID principles are predominantly *compositional* in nature and require the existence of

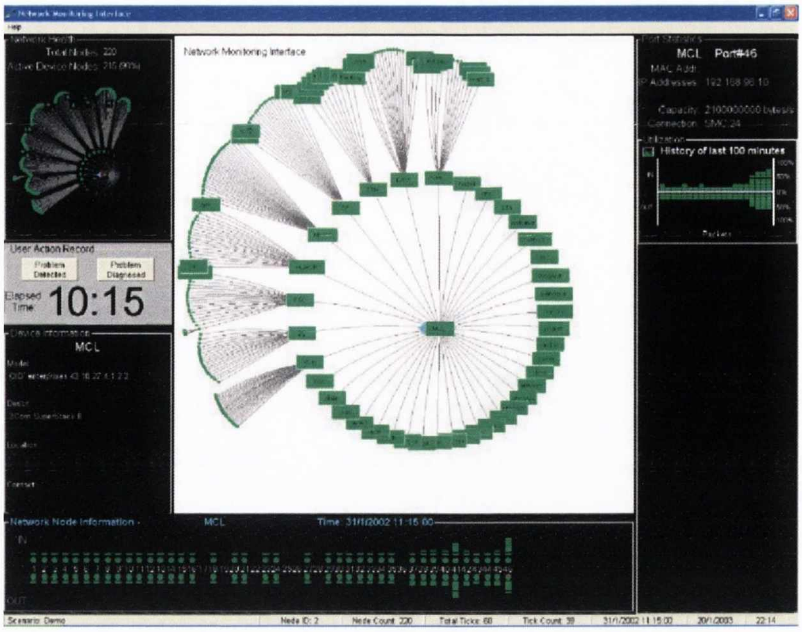


Figure 3.5: An ecological visualisation. (Duez and Vicente, 2005)

tokens that represent system components. In the absence of these, design must initially be carried out at the component level. In addition the visual form may need to be compatible with information visualisation techniques in order to cope with system scale. As EID does not deal with graphic encoding at the *syntactic level*, design guidance for the generation of these tokens must be sought elsewhere.

3.3 Alternative Representational Design Guidelines

One of the challenges facing design research is that visual representations are so diverse. Different work domains tend to use specialist representations; geographic maps, statistical displays, engineering diagrams etc. These representations are generated using domain specific visual vocabularies i.e. icons, color codes etc. Guidelines for the creation and use of specialist representations exist within their particular disciplines, however to develop a new visual vocabulary for a work system, it is necessary to understand precisely *how representations support cognition*. The difficulty is that scientific investigation into this phenomenon is relatively new and our understanding of how graphics work is still incomplete (Scaife and Rogers, 1996).

One of the earliest investigations into the role of graphics in problem solving was

Larkin & Simons study of physics diagrams (Larkin and Simon, 1987). This demonstrated that diagrams reduce the required amount of cognitive processing as their representational formats explicitly express the number of items in a problem space and the relationships between them. This decreases the amount of searching that is required to derive inferences necessary for problem solving. In addition, diagrams cluster related items together making it easier to identify and solve sub-goals in the overall problem space. Another advantage is that, unlike alphanumeric representations that must be accessed in a serial manner, *"diagrams automatically support a large number of perceptual inferences which are extremely easy for humans"*. While this study revealed some of the cognitive advantages of using graphics, the precise nature of these perceptual inferences remained unclear.

Zhang and Norman provide further insight in their theory of distributed representations (Zhang and Norman, 1994). This suggests that the use of external representations in problem solving is a form of individual distributed cognition, where the abstract problem space that describes a problem can be divided between internal and external representations that are used in co-ordination during problem solving. The theory proposes that external representations can be directly processed using perceptual operators. By externalising parts of the abstract problem space (i.e. rules, constraints etc.), the amount of cognitive processing required to solve a problem is reduced. A further observation known as the representational effect is defined as *"The phenomenon that different representations of a common abstract structure can generate dramatically different representational efficiencies, task complexities and behavioural outcomes"*. Certain visual encodings are more sucessful at representing the dimensions or structural relationships involved in a problem space and supporting perceptual operators. Their experiments showed that puzzle representations that used spatial encodings (i.e. position) resulted in better problem solving efficiency than those that used the visual encodings of colour or size.

While these studies provide strong empirical evidence that graphic representations can improve problem solving, their focus is on explaining this phenomenon rather than providing explicit design guidelines. However they share a core concept, that visual displays improve information interpretation through perceptual efficiency and there have been a number of attempts to develop generic visual design guidelines around this idea.

This research has been carried out in diverse fields and provides different perspectives on the graphical encoding of data. Here four key perspectives are reviewed in terms of their theoretical basis, design guideline and any limitations they may have in relation to the design of control displays.

3.3.1 Image Theory Perspective

One of the most comprehensive theories of visual design is Bertin's semiology of graphics (Bertin, 1983). This outlines techniques for converting data into efficient graphical representations that can be used to answer questions. It ties together the relationship between data structures and visual structures into a cohesive image theory of graphic construction.

3.3.1.1 Theoretical Basis

The semiology provides a taxonomy of the basic visual variables from which all images are constructed (figure 3.6). These consist of the two planar variables that define position and six retinal variables; size, brightness, texture, colour, orientation and shape. These variables have different perceptual lengths based on the number of perceptible differences that each can achieve. For example, the planar variables have the longest perceptual length, as the eye can detect very small differences in position, while brightness has a much smaller perceptual length, as the eye is less sensitive to differences in shade. Bertin proposed that these variables can be used to graphically encode data in a manner that supports the direct perception of information.

A core concept that underlies this work is that humans generate meaningful information out of elementary data by identifying relationships within a dataset. Stevens' theory of scales in measurement defines four basic data scales that describe the type of relationships that can exist (Stevens, 1946):

- The Nominal scale refers to data that describes a basic category such as a name or a style. It is non-proportional and only supports differentiation by type, e.g. an unordered set of names

Visual Variable	Type of Perception			
	Associative	Selective	Ordered	Quantitive
Spatial X	YES	YES	YES	YES
Spatial Y	YES	YES	YES	YES
Size		YES	YES	YES
Brightness		YES	YES	
Texture	YES	YES	YES	
Colour	YES	YES		
Orientation	YES	YES		
Shape	YES			

Figure 3.6: Visual variables

- The Ordinal scale refers to data that is associated with a scale of magnitude. It supports differentiation by both category and rank and permits ordering in a greater or less than relationship, e.g. an ordered set of grades A, B, C.
- The Interval scale refers to data that has a scale of magnitude with an equal interval of variation. It supports differentiation by category, rank and distance between variables. It also supports simple mathematical function like subtraction, e.g. dates
- The Ratio scale refers to data where variables have a direct proportional relationship to one another due to the inclusion of an absolute zero on their scale of magnitude. They support differentiation by category, rank, distance and ratio. They are generally expressed as numbers and support a full range of mathematical functions, e.g. number of errors.

These scales can be ranked from the low-power nominal scale to high-power ratio scales as each level supports addition types of questions in cumulative fashion. The visual variables can be matched to four styles of perception that relate to these data scales (see figure 3.6). The *nominal* scale is divided into *associative* and *selective* perception, the *ordinal* scale is supported by *ordered* perception and Stevens' *interval* and *ratio* scales are unified into *quantitative* perception. Image theory suggests that by matching a data scale to an equivalent visual variable it is possible to answer questions using perceptual rather than cognitive operations on the data.

Tables 3.1 & 3.2 provide an illustration of how these scale can be applied. Data tables are used to structure multivariate data into cases and variables from which information is derived by making comparisons between variables and/or across cases (table 3.1). However, the propositional encoding used in a data table results in a visually homogenous display that makes it difficult to identify patterns or relationships. For example, to identify the tool with the longest running time in table (table 3.1), it is necessary to compare 15 values against each other. Table 3.2 uses visual scale matching to highlight information in the data. In this case, the tool with the longest running time can be visually identified using quantitative perception without the need for cognitive processing. Visual scale matching has been applied to each of the variables in table 3.2 to support a range of questions. Just as correct scale matching can improve task efficiency, incorrect scale matching can make tasks more difficult. For example, if running time variable had been encoded using shape, it would be necessary for the viewer to maintain an internal mapping between this low-powered visual variable and high-powered data scale (Zhang, 1996). The concept of mapping data to visual scales provides a detailed explanation of how analogical representations can provide natural constraints in visual displays.

This mapping principle underlies the generation of most graphs and charts and can be used to support different types of cognitive tasks. As well as the elementary selection tasks mentioned above, visual variables can be combined to support synthesis of new information. The simple scatterplot combines two ordinal or quantitative variables by encoding each on a planar variable. The resulting display reveals trends in the data that cannot be observed in the data table. For example figure 3.7 reveals an unsurprising negative correlation between the tool age and the running time.

3.3.1.2 Design Guideline

Image theory provides an important design guideline:

> *For information to be represented as a single (pre-attentive) image, each of its components (data variables) must be homogenous and must correspond to an ordered concept (visual variable)*

Tool	Recipe	Running Time (Hours)	Tool Age (Months)	Service Time (Minutes)	Technician Experience
T_xyz6498	P84635	150	7	65	medium
T_xyz6023	P84686	156	36	180	expert
T_xyz0835	P84635	174	35	158	novice
T_xyz7381	P84635	174	28	154	high
T_xyz0270	P84686	177	21	122	medium
T_xyz0893	P84686	218	15	110	novice
T_xyz6329	P84635	296	11	111	medium
T_xyz0954	P84635	363	13	128	high
T_xyz0763	P84686	420	15	143	high
T_xyz0273	P84635	486	8	72	novice
T_xyz6464	P84686	593	4	50	novice
T_xyz2374	P84686	627	9	100	medium
T_xyz6373	P84635	631	9	145	medium
T_xyz9367	P84686	661	2	80	novice
T_xyz0535	P84693	678	30	60	high

Table 3.1: A data table demonstrating different data scales

3.3.1.3 Limitations

While visual scale matching provides a useful concept for developing visual forms, its application to complex control displays it subject to a few limitations.

Firstly, the theory was developed as a means to interpret meaning from large volumes of data rather than supporting reasoning at multiple levels of abstraction. Bertin's definition of the term image is restricted to graphs that provide a meaningful visual form, perceptible in the minimum instant of vision. Such images, he argues, can only represent up to a maximum of three dimensions *"formed by three homogenous and ordered variables; the two planar dimensions and an ordered retinal variable"*. As a result, image theory has had a stronger impact on the field of information visualization (Card et al., 1999; Siirtola, 1999) than on cognitive systems engineering. Despite this, the basic visual mapping concept remains valid and can be used to inform the design of analogical displays.

A further issue is that multiple visual encodings are possible for each data scale (see figure 3.6). As different combinations of these variables are also possible, a large range of valid representations can be generated for any set of data. Figure 3.8 shows a number of valid encodings for the data in table 3.1. As a result, image theory does not prescribe a single visual solution but can inform the design of a range of potential graphic encodings from which the design must chose an appropriate solution.

This is further complicated by the fact that data can be transformed between different

Tool	Recipe	Running Time (Hours)	Tool Age (Months)	Service Time (Minutes)	Technician Experience
T_xyz6498	P84635				medium
T_xyz6023	P84686				expert
T_xyz0835	P84635				novice
T_xyz7381	P84635				high
T_xyz0270	P84686				medium
T_xyz0893	P84686				novice
T_xyz6329	P84635				medium
T_xyz0954	P84635				high
T_xyz0763	P84686				high
T_xyz0273	P84635				novice
T_xyz6464	P84686				novice
T_xyz2374	P84686				medium
T_xyz6373	P84635				medium
T_xyz9367	P84686				novice
T_xyz0535	P84693				high

Table 3.2: Graphic version of table 3.1

scales as part of the problem solving activity. For example, the running time variable in table 3.1 can be transformed from a quantitative variable to an ordinal variable by dividing the number of hours into distinct ranges i.e. short, medium and long. While data transformation can still support the identification of relationships in datasets, they further extend the number of visual encoding options available to a designer.

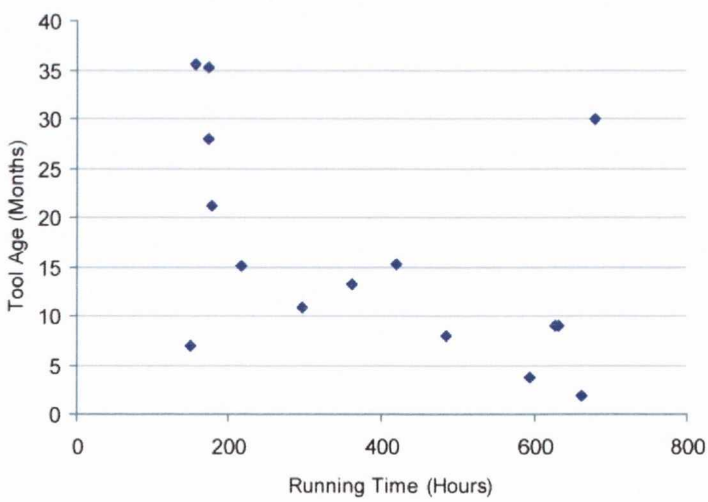


Figure 3.7: Scatterplot of tool performance data

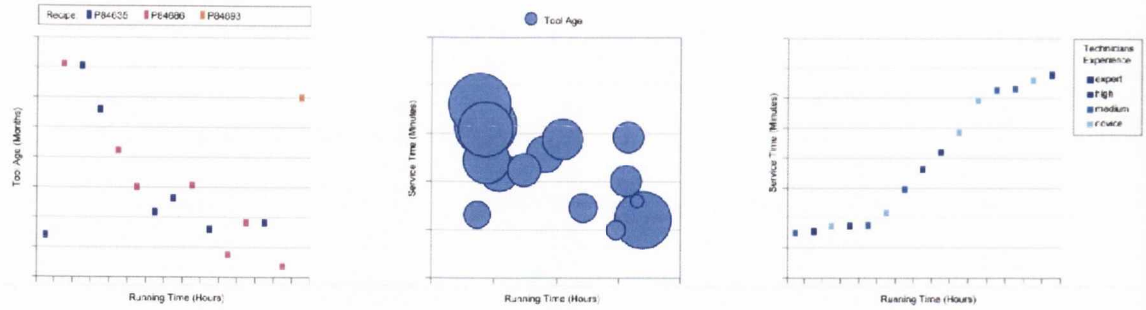


Figure 3.8: Multiple visual encodings of tool performance data

3.3.2 Psychophysical Perspective

Cleveland and his colleagues approach the visual encoding phenomenon from an alternative perspective. They investigated the psychophysical qualities of the different visual variables and their role in understanding statistical data (Cleveland, 1985; Cleveland and McGill, 1985).

3.3.2.1 Theoretical Basis

In this case visual perception is defined as "*the visual decoding of the quantitative and qualitative information encoded on graphs*". In this context visual decoding means *"instantaneous perception of the visual field that comes without apparent mental effort"* (Cleveland and McGill, 1985). A number of experiments were carried out on the visual decoding process involved in interpreting graphs. These focused specifically on the *accuracy* that could be achieved by different display formats. Ten basic perceptual judgements or specifiers, that are commonly used to decode quantitative information in graphic displays, were identified. These consisted of ; angle, area, colour hue, colour saturation, density, length, position along a common scale, position along identical non-aligned scales, slope and volume. A series of experiments investigated which of these specifiers provide the most accurate perceptual interpretation of a quantitative value. From these the ordering of elementary perceptual tasks shown in table 3.3 was derived.

Accuracy	Perceptual Feature
Most Accurate	Position along a common scale
	Position along identical non-aligned scales
	Length
	Angle-Slope
	Area
	Volume
Least Accurate	Colour hue – Colour Saturation - Density

Table 3.3: Basic tasks model ranking of specifiers

3.3.2.2 Design Guideline

This ranking is used to define the basic task model for data display that provides another visual encoding guideline;

> *Encode data on a graph so that visual decoding involves tasks as high as possible in the ordering of the basic tasks model.*

This is a remarkably simple principle to understand and apply. It implies that graphic forms that use position as the main encoding, i.e. bar charts, scatterplots etc. provide more accurate encodings than all others. This would suggest that other forms of graphical encoding such as pie-charts, area maps and colour maps are inferior graphing techniques, however the author acknowledges that the model should only be used as a rough guideline rather than a definitive rule as it is subject to a number of limitations.

3.3.2.3 Limitations

Firstly, graphs typically encode a number of variables using different specifiers but this model only relates to *one dimension* of decoding. Even the basic scatterplot display involves encoding of two dimensions in order to identify relations between variable pairs. As the interaction between specifiers is not studied here, the model has limited application.

Secondly, the model deals only with the identification of an individual *quantitative* value. Graphs are often used to observe performance across categories and to identify outliers. These types of perceptual tasks are not addressed.

Thirdly, the model deals with perceptual processing of the specifiers but does not comment on other related factors that support interpretation of graphs. The visual style of axes, scales and labels can all effect how easy a graph is to interpret (Kosslyn, 1993; Tufte, 1983) but these elements are omitted from the model.

3.3.3 Visual Integration Perspective

The guidelines presented so far have focused on visual encoding to support information extraction and comparison. *Information integration* is another cognitive operation that is critical for supporting reasoning and problems solving. Integration can be understood as a computational process that involves combining data. Larkin and Simons study shows how diagrams reduce the amount of searching needed when locating data for integration (Larkin and Simon, 1987). It is possible however, to remove the need for cognitive integration by generating graphic representations that make the product of integration directly perceivable.

3.3.3.1 Theoretical Basis

The Proximity Compatibility Principle (PCP) is used to inform the graphic encoding of multiple variables to achieve perceptual integration in control displays (Wickens and Carswell, 1995). PCP proposes that the range of elements that make up a visual control display can be thought of as having two forms of proximity; task and display.

Task proximity identifies the relationships between variables in terms of the cognitive tasks for which they are used. This relationship exists along a continuum ranging from closely coupled to completely independent. This continuum can be structured using a task taxonomy consisting of three levels; integrative processing, non-integrative processing of similar tasks and non-integrative processing of dissimilar tasks or task independence.

Display proximity relates to how variables are visually presented and is described in terms of a continuum from low proximity to high proximity (see figure 3.9). Low display proximity occurs where variables do not share any common representational properties. For example, a numeric value and a dial that are spatially separated and are not aligned have very low display proximity. Variables that share a graphical encoding such as colour hue or a common alignment will have closer display proximity as they can be visually associated with one another. Variables that are enclosed, connected or spatially clustered

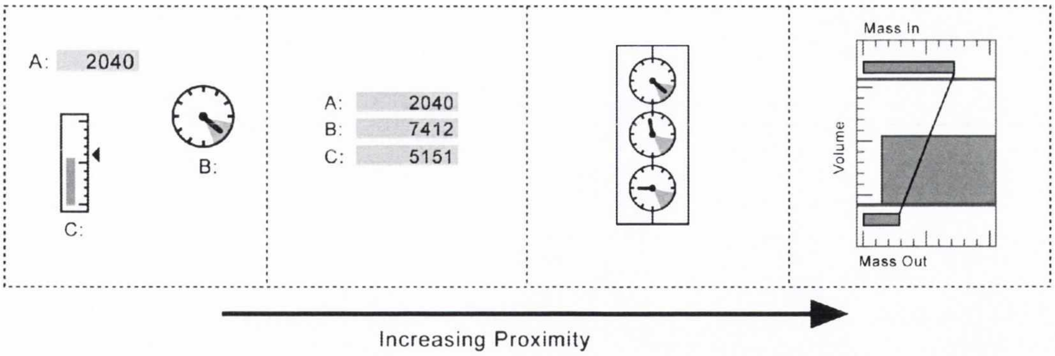


Figure 3.9: Display Proximity

together have high display proximity. The highest form of display proximity occurs when the visual variables used to encode individual data variables produce a *configural dimension* (Garner, 1974). This is a perceptual dimension where individual stimuli can be distinguished, but whose configuration produces an additional perceptual form. The trapezoid shape generated in the Duress reservoir display (see figures 3.4 and 3.9) is an example of this, where the combination of visual variables used to encode input/output flow result in a form that can also be perceived holistically as an indication of the reservoirs transitional state.

3.3.3.2 Design Guideline

The Proximity Compatibility Principle provides another design guideline:

> *"The benefit of closer display proximity is increased, or its cost decreased, as the task integration requirements are increased".*

High display proximity generates what's known as an *object display*, where the configuration of graphic specifiers can be perceived in a parallel rather than serial manner. PCP maintains that improved performance for integration tasks comes from the *objectness* of the display, however it also maintains that the salience of these objects can make it difficult to separate out their constituent variables, making focussed tasks more difficult to perform. It suggests that the generation of object displays has both benefits and costs depending on whether data must be integrated or accessed independently.

3.3.3.3 Limitations

PCP provides a rationale for generating analogical, object/configural displays but further investigations provide evidence that their advantages go beyond the information processing benefits of data integration. Buttigieg and Sanderson suggest that their main advantage relates to the *emergent features* they create rather than the objectness of the display (Buttigieg and Sanderson, 1991). Emergent features can be exploited to support global readings that correspond to system states, without damaging focussed attention tasks that are carried out through the basic visual encoding. For example, the angle of the line connecting the input and output flows in the reservoir display is an emergent feature created by the alignment of the two flow meters (see figure 3.9). This argument suggests that, it is this emergent feature created by the line, rather than the trapezoid shape, that conveys the important information. An extensive review of control displays carried out by Bennett and Flach provides further support for this (Bennett and Flach, 1992). It shows how configurable displays result in a significant improvement for integrated tasks without any significant cost for focussed tasks. These investigations concur with a view that the advantages of configural displays in control interfaces relates to the semantics associated with their emergent features. This provide a rationale for implementing EID's second principle, as an emergent feature can act as a visual cue corresponding to a system state. In addition, the cue is generated from the visual encoding of its constituent component variables providing an explicit model of their causal relationship.

While this refutes the concept of costs associated with configurable displays, PCP still provides a rationale for developing such displays. However the difficulty facing a designer is that, rather than prescribing a limited number of configurable representations, the principle simply provides an argument supporting their generation and use. As was identified in section 3.3.1, data can be accurately encoded using multiple visual variables. While PCP identifies further benefits of integrating visual variables, this can still be achieved in a multitude of ways (Bennett and Walters, 2001).

3.3.4 Aesthetic Perspective

While the previous approaches have developed guidelines based on the properties of data and users tasks, Tufte takes an alternative approach to generating design knowledge. Through a qualitative analysis of visual displays he extracts a range of general design principles aimed at achieving *graphical excellence.* Topics such as graphical integrity and data density are investigated by studying specific visual displays in terms of their composition and visual encoding. Numerous illustrated examples and design critiques make the material very accessible and the work is widely cited in information graphics literature.

3.3.4.1 Theoretical Basis

Considerable attention is placed on how supporting features of the design such as scales and grids can affect the accurate reading of data. One general prescriptive principle based on this is the *data-ink ratio.* This refers to the quantity of ink that shows information e.g. points on a graph, as opposed to the amount of ink supporting the information e.g. gridlines, labels etc. The ratio is described using the equation;

Data Ink Ratio = data ink/Total ink used.

A data ink ratio of 1 would result in a scatterplot consisting of just the points and indicates the proportion of a graphic that can be erased while still making data relationships visible. Graphs with low data-ink ratios result in a distortion of the data and decreased legibility.

3.3.4.2 Design Guideline

> *The data-ink-ratio principle recommends removing all non-data ink within reason in order to reduce the noise and interference it creates.*

Doing show should result in a graph that highlights *data variation* rather than *style variation.* A directly related concept is that of *chart junk.* This refers to all features that are used for aesthetic embellishment rather than data communication. Tufte recommends the elimination of chart-junk as it generally results in misinterpretation of data relationships.

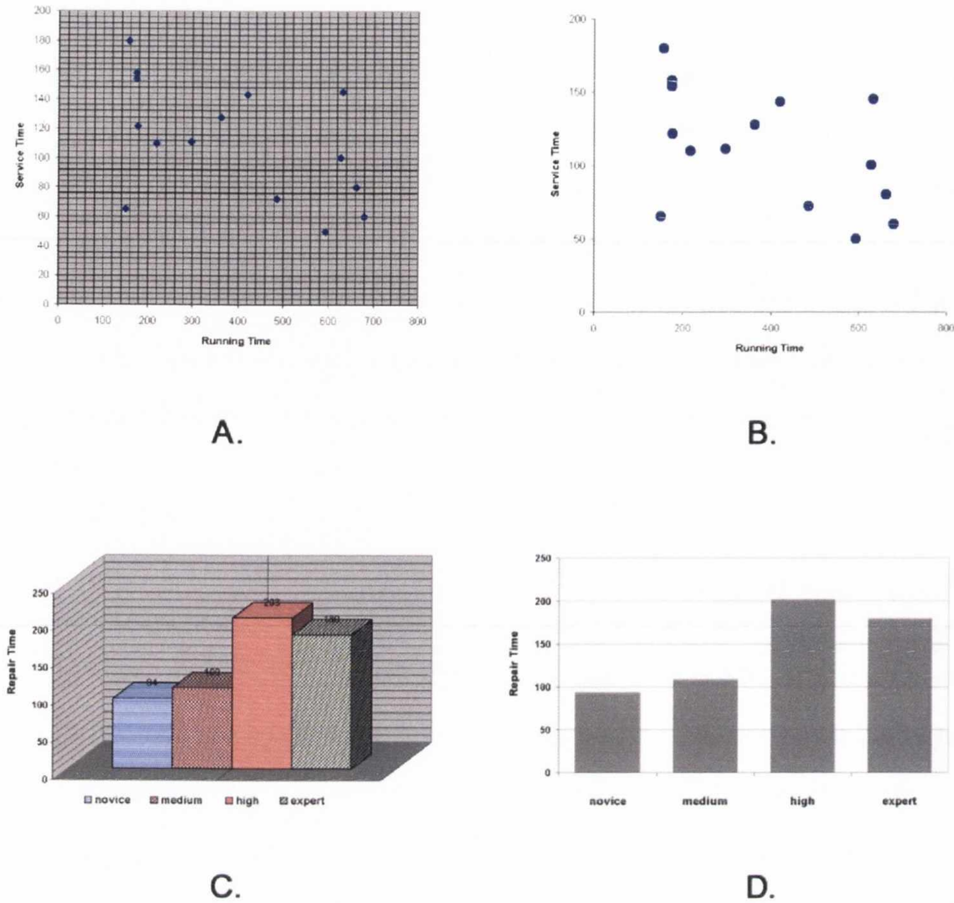


Figure 3.10: Graphs illustrating chart junk and the data ink ratio

Specific examples of chart-junk include moiré vibrations, an optical illusion of shimmering, caused by using patterns to give shading to graphs (fig. 3.10). Instead of pattern flat shades or color hues to distinguish between elements. The grid itself is also identified as a form of chart-junk as its standard rendering using dark lines often obscures the data. Tufte associates the grid with graph construction rather than data interpretation and recommends its removal or a lightening in its shade.

3.3.4.3 Limitations

The guidelines provided by Tufte are a useful resource for ensuring that graphs present information that is legible and accurate. The delivery format of the guidelines, through design exemplars, makes them very accessible to the wider design community. However, the guidelines focus on eliminating representational errors rather than informing the process of visually encoding information. Most of the case studies examine the rendering of an existing graphic form rather than the design principles that informed its initial generation. While the many examples may provide design inspiration, their re-use is dependant on having a design problem with similar data and cognitive tasks. The principles can guide the process of design but they are in no way prescriptive and the data-ink ratio principle has been shown to have less predictive power, in terms of information extraction, than the basic task model discussed earlier (Carswell, 1992). The purpose of these guidelines are to make a designer aware of the effects that aesthetic design decisions can have on legibility rather than advising on specific graphic formats.

3.3.5 Reviewing the Perspectives on Visual Design

Visual design research is a hugely diverse field and these four perspectives are not an attempt provide a comprehensive summary. However they are representative of a narrower subset of this body of work and were selected based on three characteristics. Firstly, while each stems from a particular academic background they discuss visual design in a generic manner and attempt to *generate broad principles*. Secondly, they are *popular* and are among the most widely cited research in the field. Thirdly they are specifically *design focused*. While much of the research focuses on explaining the benefits of graphics representations, these studies are aimed at directly informing design practice.

On reviewing the perspectives they also share a number of qualities in terms of the types of data and tasks they examine. All of them focus on the visual display of *quantitative* data and they tend to support an information processing view of the role of graphics in supporting problem solving. The advantages of visual encoding are discussed in relation to its ability to transform the cognitive operators associated with data extraction, comparison and integration into perceptual tasks. In this manner they share a more *cognitivist perspective* on visual displays and do not investigate the role of visual encoding in generating a system image. To further understand the utility of these guidelines for the design of visual decision support systems, it is necessary to examine where they can fit into an overall design process.

3.4 Gaps in Visual Design Knowledge

Despite the large selection of visual design research available, visual design is still considered to be more of an art than a science, with designers drawing on intuition and experience rather than guidelines and processes. However when dealing with the design of interfaces for complex work domains, designers are increasingly seeking structure and theory (Rogers, 2005). The principles and guidelines reviewed here provide a useful starting point but a number of issues must be overcome before they can be widely applied in practice.

3.4.1 Limitations of Visual Design Research

Issues with the current levels of visual design research include problems relating to scope, presentation and variability and each of these are discussed here.

Restricted scope. In order to design a system image a designer must create a visual representation that can be interpreted at multiple levels of abstraction. The difficulty is that most design principles focus on an individual level of interpretation. One way of grounding visual design research is to relate the principles and guidelines back to the representational medium of the visual interface. Woods provides a *Graphic Display Hierarchy* that describes six levels of representational objects that make up an interface; Pixels, Graphic Atoms, Graphic Fragments, Graphic Form, Views, Workspace (Woods, 1997b). Each level is built up from the objects that exist on the lower levels preceding it (table 3.4). As the system image is communicated through the configuration of these objects, it is possible to examine visual design principles and guidelines in terms of their ability to inform the construction of objects at each of these levels. The various guidelines developed from an information processing perspective explain how visual encoding can be used to replace cognitive operations with perceptual ones. These work at the level of visual syntax and relate to pixels, atoms and fragments. The EID visual design principles, on the other hand, are more structural. They explain how semantic relationships between work domains and visual objects can be achieved. This focus on relationships is more closely associated with views and workspaces. While both information processing and semantic perspectives touch on the design of graphic forms they arrive at this point from opposite directions (see figure 3.11) and neither can provide a fully comprehensive and explicit methodology that covers all levels of the graphical display hierarchy.

Presentation of design research is often idiomatic, contradictory and non-prescriptive. Many of the guidelines and principles are written in the language of cognitive science and can be difficult to interpret when faced with practical design issues. As much of the research into graphic representation has been carried out using controlled experiments to identify the cognitive benefits of particular graphic formats, the findings are often presented in terms of data integration, perceptual operators and configural dimensions, making them difficult to relate back to the designers compositional and visual encoding tasks. This approach has been necessary to produce conclusive results but it requires a controlled environment and can only investigate a limited range of tasks. This can further limit the applicability of the finding in real world design situations where the designer generally needs to construct a multifunctional, multidimensional interface. Many debates also exist between design guidelines (Lin et al., 2006) and these conflicting views make it different to select from a growing number of design frameworks. A further issue is that, outside of the basic tasks model, visual design research has tended to provide abstract principles and non-prescriptive guidelines rather than concrete rules for graphical encoding (see figure 3.11). This makes it very difficult to develop a defined process as the application of design research to design practice depends very much on how these guidelines are interpreted by the designer.

Design variability is inherent. Possibly the greatest challenge relates to the fact that multiple solutions are possible when constructing visual representations. There are a number of reasons for this. Firstly, as different visual variables can be used to encode the same data types and visual forms are constructed out of combinations of visual variables, there are a multitude of syntactically valid design solutions for each representational problem. Secondly, as the display levels involved in the construction of an interface are linked, design decisions made at one level will effect representation at another. For example, if colour is used to encode a semantic relationship between graphic forms it may not be suitable for encoding nominal relationships between graphic atoms. We can conclude from this that the design of graphic forms is not definitive but involves balancing both semantic and information processing goals.

Level	Example	Description	Decisions
Pixels	.	The smallest graphical unit, constrained by the limits of the screen	"Colour" or light emission
Graphic Atoms	A,3,+,*,#	Composed of pixels; a letter, digit, line, color block	Colour, size,shape, thickness,angle, forms of reference
Graphic Fragments	Word,2002	Composed of graphic fragments; a word, number, scale; not a complete graphic form	Position, content, organisation of fragments, forms of reference, proportion and salience
Graphic Form	A graph or indicator	Composed of graphic fragments, this level conveys meaning	Analog and digital forms, display of context, salience across graphic fragments
Views	A window or single cohesive display screen	Composed of graphic forms, this level brings related graphic forms together to describe a process or show sequence	Relations across graphic forms, salience between forms, organisation of forms
Workspace	The entire display application	Composed of views, the workspace defines the virtual action space of operator	Relations across views, navigation, overview and workspace status

Table 3.4: Graphical Display Hierarchy. after (Burns and Hajdukiewicz, 2004)

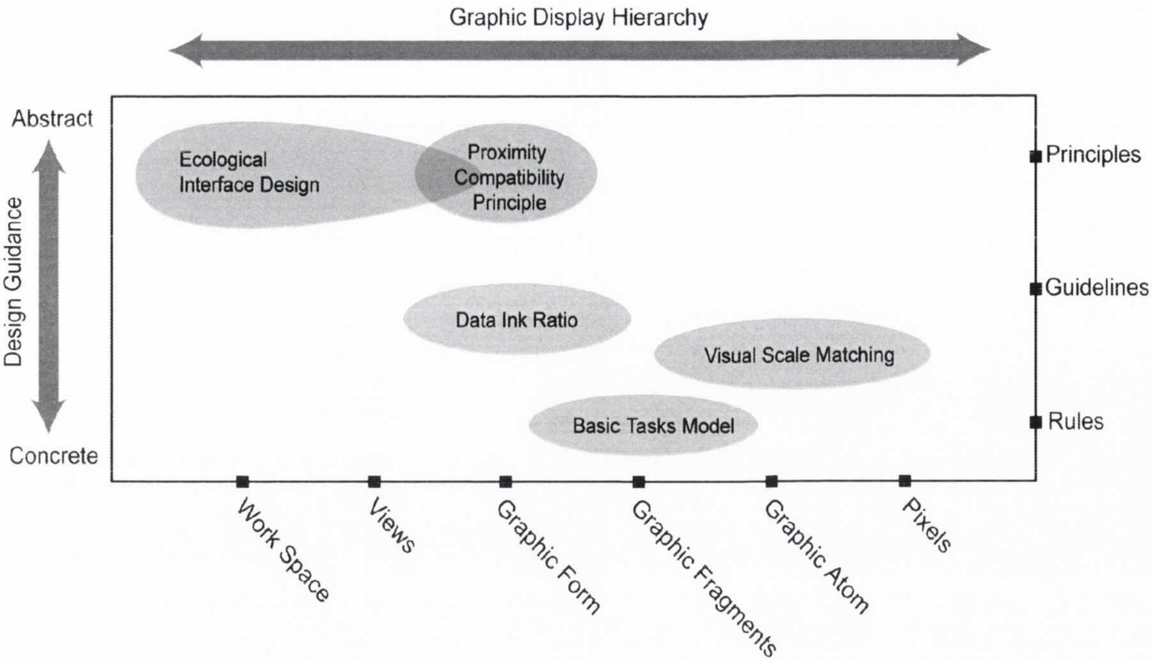


Figure 3.11: Mapping Visual Design Research

3.4.2 The Complexity of Visual Design

These limitations reveal a certain disparity between the goals of design research and those of design practice. *Design research* has predominantly involved investigating visual representations and examining their utility for certain tasks. Basic research can look at the semantic power of simple visual variables, but in order to investigate more complex representations, existing displays or generated representations must be studied. Consequently, research into visual displays has tended to involve a post-hoc analysis of the *product of design*. By defining the properties of displays and investigating their ability to support cognitive tasks the researcher attempts to reveal generalizations that can form the basis of *prescriptive* design guidelines or principles.

Design practice involves the visual encoding of information and the composition of visual components into a semantic representation. However, due to the variety that is inherent in visual design and compounded by the fact that designs can be read at multiple levels of abstraction, the *process of design* is by necessity an *exploratory* activity. The large selection of principles available combined with the need to combine semantic and perceptual goals makes the generation of simple prescriptive design guidelines impossible to achieve. Despite this, designers of visual decision support displays still a require design support. One option is to construct a generic model of the cognitive engineering design process that can capture and communicate how design happens. This is the focus of the next chapter.

3.5 Summary

This chapter has reviewed the principles and guidelines that are currently availible to support representational design in cognitive systems engineering. It explains how graphic displays provide better support for control as they provide a correspondence between system constraints and natural visual constraints in the interface. It introduces the EID principles, which highlight the importance of arranging interface elements in a manner that reflects the functional work domain model. These principles inform the design of an interface that allows a controller to trace causal relationships through a system. The HVM domain generates a number of challenges for these principles as the scale of the system and the lack of an existing visual vocabulary requires visual components to be designed from first principles. Four alternative perspectives on visual design are reviewed to identify guidelines for encoding data. In general, the utility of visual design research is limited in terms of scope, language, contradictions and inherent design variablility. While this makes it impossible to outline a simple design process for visual decision support, an alternative approach is proposed through the development of a generic model of the CSE design process.

Chapter 4

Modeling the Process of Cognitive Systems Engineering

The previous two chapters have examined the analytical and representational gaps in CSE design knowledge. A third design gap, identified in section 1.1.2, describes the difficulty in communicating the overall CSE design process. In this chapter we show that, while CSE aims to support the analysis and design of cognitive systems, current approaches to communicating design knowledge make it difficult to demonstrate how theoretical methods inform design activity. In practice these design gaps are bridged each time a project is carried out, but in order to transform contextual design practice into more generic design knowledge, a model of design is required. A generic model of the design problem space is introduced. This portrays design as an ill-defined problem that is structured into a number of phases. Sketching is identified as a universal design activity and sketches are identified as design artifacts that support progression through design phases. The concept of design artifacts is extended into analysis by describing analytical models as exploratory sketches that support conceptualisation. A meta-model of the complete CSE design process is developed around these concepts. This is a flexible model that allows alternative methods and design techniques to be combined in a pragmatic manner, while at the same time providing a stable conceptual framework for developing design theory and proposing methodologies.

4.1 The Design Gap

The design gap is a term that is generally applied, not to analytical or graphical issues independently, but to the process of moving from systems analysis to visual design. The transition from requirements to representation must occur each time an interface is developed, but it is notoriously difficult to express and has been euphemistically described as the part of every project where "a little magic happens" (Wood, 1997). As a result, interface design is often portrayed as a craft-based activity, where an interface designer combines their domain expertise with creative ability to generate visual forms. The increasing role of visual interfaces in the control of critical systems has motivated a drive to transform design from a craft to an engineering discipline and was one of the initial reasons for the foundation of CSE (Dowell and Long, 1998). However as illustrated in figure 4.1, none of the frameworks offer a complete bridge over this design gap. With Distributed Cognition and Activity Theory, interpretation of the analytical models is required in order to reveal *implications for design*. These implications are generally quite high-level and require further interpretation by the designer before they can be used to inform the design of visual design prototypes. Cognitive Work Analysis integrates with Ecological Interface Design principles to provide a more comprehensive design process. Despite this, its lack of instruction in relation to design syntax means that visual design remains fairly unspecified and designers must look to exemplars to see how these principles can be applied. Consequently, these principles have been interpreted in a number of different ways (see section 3.2) and again a designer must assess and interpret the utility of different design techniques in the context of their own design projects. A more serious issue is that the common approach taken to *reporting CWA/EID exemplars* avoids discussing the design rationale that leads to design solutions. In order to understand why this occurs, it is necessary to take a closer look at the concepts that underlie the CWA framework.

4.1.1 Validity, Verification and the Reporting of Exemplars

Rasmussen and his colleagues propose that design is an inherently variable and opportunistic process that is deeply rooted in the context of its target domain (Rasmussen

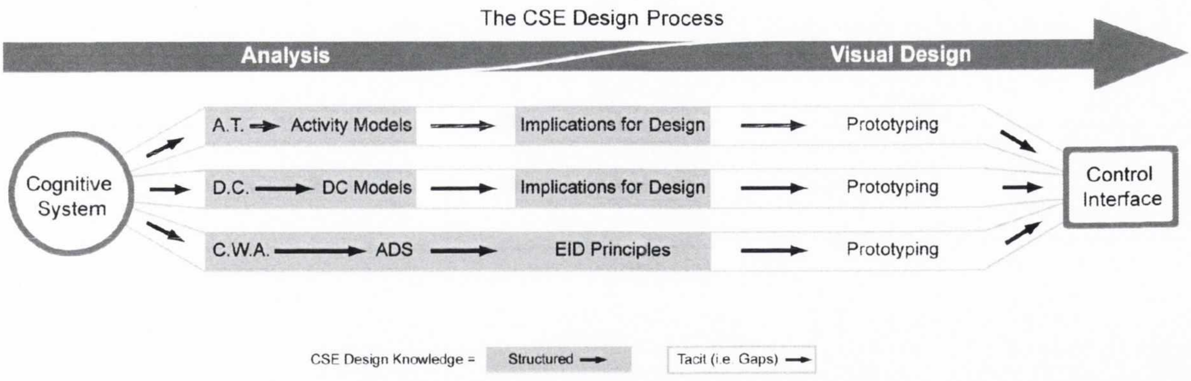


Figure 4.1: Design gaps in existing frameworks

et al., 1994). This places serious limitations on the utility of design guidelines, which are generally provided in a context free format to ensure widespread applicability. As a result designers will inevitably draw on existing visual vocabularies and their own experience when developing new designs within a work domain. However to ensure that designs are appropriate for the work being carried out it is necessary for the designer to appraise their design concepts in terms of both verification and validation.

Verification assesses whether a design satisfies the specifications that it set out to achieve. With an interface this involves examining whether the design is compatible with the perceptual and cognitive constraints of a user for a range of tasks. In short it involves checking *is the design right?*

Validation assesses whether a design supports the goals of the overall system. In terms of a control interface, this involves examining whether the design makes the underlying system interpretable to the controller. Does it provide a valid system image that explicitly expresses the functional constraints of the system and supports knowledge-based reasoning? Fundamentally, this is checking *is it the right design?*

Rather than specifying design practice, the CWA framework uses the five phases outlined in section 2.3.1 to model the context of work. This has been described as a form of *analytical evaluation* that is carried out during the cognitive engineering process to ensure design validity (see figure 4.2). The EID principles use the first of these models to provide high-level compositional guidelines. However, specific guidelines relating to visual syntax are undefined, under the assumption that domains will come with their own visual vocabularies. Consequently, the process of visual encoding is left to the designer's

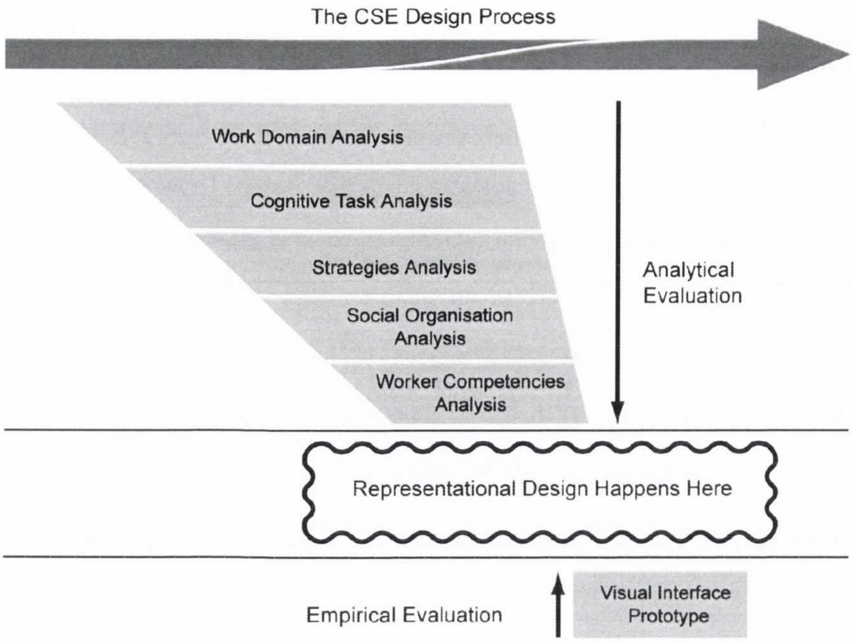


Figure 4.2: The hybrid approach to CSE evaluation

discretion. Verification of the suitability of the designer's decisions is achieved through *empirical evaluation* of the final design, by testing its performance in relation to specific tasks. This hybrid approach to evaluation, using both analytical and empirical methods has become a common format for reporting CWA/EID exemplars.

4.1.2 Extracting the Complete Design Process

There are two weaknesses with this approach. The first is that it does not encourage a detailed description of the representational design process. Analytical evaluation describes the context of the work and can be used to define information requirements, but it does not provide specific design guidance. On the other hand, empirical evaluation is carried out on the *product of design* as a post-hoc evaluation. It is used to test design assumptions rather than informing design practice. This hybrid approach of using analytical and empirical evaluation avoids the discussion of *how design happens* (figure 4.2). For some domains this is not a serious issue as the use of standardized, pre-existing visual forms has removed many of the visual encoding decisions involved in the design process. However, sociotechnical enterprises such as high volume manufacturing create representational challenges that require unique visual encodings (see section 3.2.3). In order to support designers working in these domains, it is necessary to provide exemplars that are more comprehensive in their reporting of design. Rather than testing the end-product of design the presentation of a detailed design rationale can communicate the *process of design* and diseminate design knowledge (Moran and Carroll, 1996).

The second weakness relates to the scope of this approach. While EID principles provide a direct bridge between analysis and design, they only refer to the initial work domain model generated by the CWA framework. As a result design guidance only relates to structural aspects and causal relations that describe a cognitive system. Other characteristics (i.e. social organization, cognitive strategies and intentional factors) can influence how information is accessed and used but are are not dealt with by this framework. As was shown in section 2.3.1.2 this limits the range of work domains where this approach can be applied. If the purpose of an analytical framework is to help in the design of new technological solutions, then it must be possible to tie them to the pragmatic issues that affect design practice. This would simultaneously increase their utility while providing more comprehensive design methodologies.

The five phases of CWA not only ensure design validity, they act as a vehicle for driving the design process forward (see figure 4.2). Its models are used to extract the relational structures and information requirements that are necessary for supporting cognitive work and they are used to define the content that is required in the final design. In a similar fashion, principles that define the format in which this content is represented are also required. Design may be a deeply contextual activity but it is also a process that moves from an initial starting point to a final design. To identify where analytical models and representational principles are used to progress design, a model of the CSE design process is required. This model can act as a template for reporting exemplars that will ensure that design rationales are provided and will encourage the development of comprehensive CSE methodologies.

4.2 Modeling the Design Process

The practice of design has been studied from a range of different perspective including architecture (Alexander, 1977), graphic design (Rand, 1985) and interactive media (Scaife et al., 1997). While these make important contributions in their own domains, an understanding of the design process requires analyzing the qualities of design practice that are shared across different areas of specialization. While the question of 'what is design' is too broad to be of any practical use, investigations into the generic characteristic of design as a problem solving activity can provide useful concepts for model building.

4.2.1 The Design Problem Space

The process of design can be understood as a problem solving activity where designers make decisions to reach a design solution or goal state (Archer, 1984). However, design problems have a number of unique characteristics that differentiate them from more general problem solving. If we consider problem solving as a movement from a start state to a goal state, with design problems the goal state is never available in advance and in many situations even the start state is not known before the design process begins. In terms of control structures that regulate the movement through a problem space, design problems do not have definitive controls as there is no right or wrong solution, only better or worse ones. Control structures are also difficult to define as choices are multidimensional, e.g. a design decision at the level of syntax may have an effect at the semantic level.

These characteristics mean that design can be considered an *ill-defined problem* (Goldschmidt, 1997). It also means that before problem solving can begin, the designer must initially explore the problem space to develop these structures before working towards a solution. From this we can derive that a generic design problem space must include a *problem-structuring phase.* Only after the problem has been structured can problem solving proceed. However, problem solving is not a simple serial progression. The scale and complexity of design problems and the levels of detail that need to be considered have resulted in three *distinct problem solving phases* being proposed; preliminary design, refinement and detailed design (Goel and Pirolli, 1992). These are based on observed high-level distinctions but more phases may be required for certain design tasks. A number of additional factors contribute to the need for these phases (Goel and Pirolli, 1992). Firstly, the complexity of most design problems requires them to be broken down into

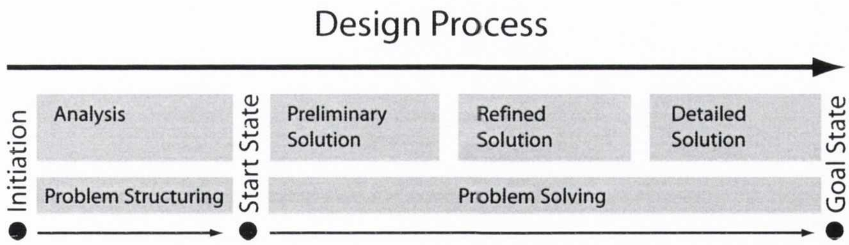


Figure 4.3: Mapping the design problem space to a process

components that can be worked on independently. Secondly, the lack of clear control rules means that the designer must decide when the result satisfies the requirements of the design. They must develop their own evaluation criteria and stopping rules. Thirdly, as components are developed independently and there are no right or wrong solutions, the artifact as a whole must be frequently assessed to confirm whether the components are working together towards a satisfactory solution. Another issue is that information can be revealed during the design process that can restructure parts of the problem space. This can require a designer to revisit a partial solution or even an entire problem-solving phase multiple times during the design process. Figure 4.3 illustrates a basic structure for a generic design process based on these observations.

4.2.2 Concept Expansion and Commitment

This structure describes generic characteristics of the design problem space that are experienced across all forms of design. Problem solving involves knowledge state transformations but relatively little domain knowledge is available at the outset of a design project. *Problem structuring* involves expanding the information available to the designer. It can be described as a *generative* activity whereby knowledge of the purpose and constraints relating to the object of design are developed. It also involves the initial conversion of this knowledge into partial solutions. This activity can take the form of brainstorming and concept generation. In contrast to this, *Problem solving* is a *reductive activity*. While concept generation is important for framing a problem, the goal is to produce a finished design. This makes it necessary to reject certain concepts in favor of others. This commitment to aspects of the design is essential for progressing the design problem solving activity. Laseau (Laseau, 2000) describes this as the elaboration and

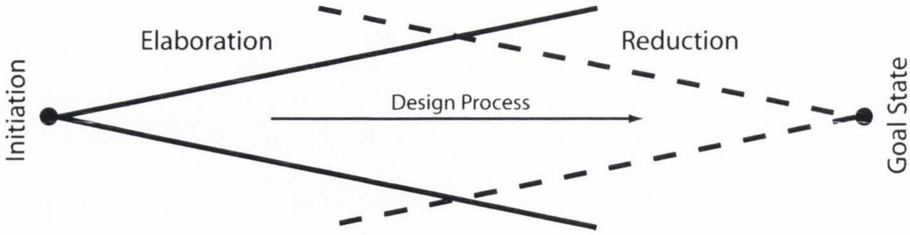


Figure 4.4: Design elaboration & reduction (Laseau, 2000)

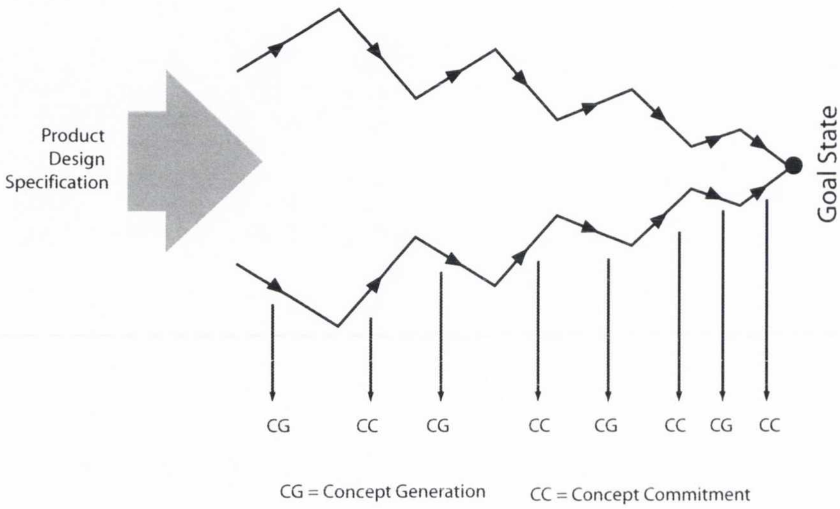


Figure 4.5: The product engineering design funnel (Buxton, 2007)after (Pugh, 1990)

reduction of concepts which, when combined, provide an abstract description of the design process (Fig.4.4). However, this simple model does not communicate *the iterative nature of design*. The process of design exploration followed by design commitment has been identified in the field of product engineering where a more elaborate model has been developed (Buxton, 2007; Pugh, 1990). This describes a design funnel consisting of waves of concept generation and controlled convergence that leads to a final design (see figure 4.5). The number of concepts generated decrease in-line with the commitments made, allowing for movement from high-level ideas to detailed rendering. These commitments can be equated to the problem-solving phases outlined in the generic problem space.

4.2.3 The Role of Sketching in Design

Evidently it is *conceptualization* that drives the design process, as commitment to a concept marks transition between design phases. This observation makes it apparent that any generic model of design will need to identify and describe the role of *transitionary* *artifacts* in the design process. The most common form of design conceptualization comes in the form of sketching. Buxton proposes that sketching is a universal activity that plays a critical role in design (Buxton, 2007). Sketches make progression possible as they provide a means for both exploring ideas and comparing options. They are by their nature quick, inexpensive and disposable while at the same time supporting the communication of ideas in a format that encourages feedback. Sketches allow concepts to be presented side by side and judged against one another. They allow commitments to be made to a partial solution that can be further rendered in subsequent phases. Sketching is what designers use to bridge the representational design gap in practice. Despite this, sketches or the concept of sketching are difficult to include in a generic model of design. While they are transitionary, exploratory design artifacts, sketches tend to use the visual syntax of the work domain where they are being applied. This makes it difficult to separate them from the contextual nature of design practice. If sketching and conceptualization are to be used within a design model, a means for converting observed practice into defined process is required. The activity theory framework described in section 2.3.3 provides a means for achieving this.

4.2.4 Using Design Artifacts to Extract Process from Practice

Activity theory places practice within a cultural-historical context. The practitioner is a subject who uses tools to transform an object into an outcome (figure 4.6a). Tools are *mediating artifacts* that allow this transformation to occur. A simple example is modeled in figure 4.6b showing how a carpenter uses a hammer as a mediating artifact to transform nails and pieces of wood into a box. This is a relatively simple activity model of practice, but Wartofsky proposes that tools can be further differentiated into primary and secondary artifacts (Wartofsky, 1973).

> "Primary artifacts are those that are used in the *direct production*; secondary artifacts are those used in the preservation and transmission of the acquired skills or *modes of action or praxis* by which the production is carried out. Secondary artefacts are therefore representations of such modes of action, and in this sense are mimetic, not simply of the objects of an environment

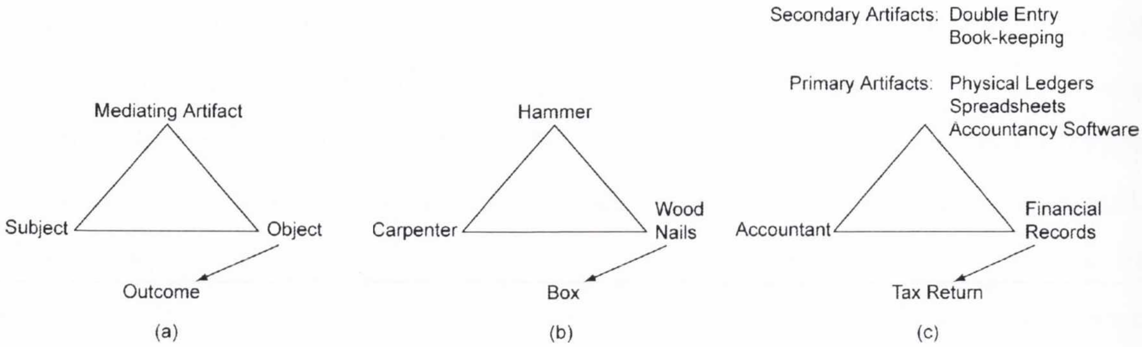


Figure 4.6: Design Artifacts

which are of interest or use in this production but of these objects as they are acted upon, or the mode of operation or action involving such objects."

In relation to the carpentry example, the hammer used in the construction of the box is a primary artefact, but the process of joining wood through the practice of hammering is a secondary artifact that is used by the carpenter across many projects. As well as supporting specific activities, artifacts define modes of practice.

This differentiation becomes more apparent when we look at cognitive work. For example, an accountant transforms a client's financial records into a valid tax return (Figure 4.6c), but there are a number of ways in which this can be achieved. They can manually transcribe a clients receipts into a paper ledger and carry out calculations, alternatively they may used a spreadsheet application to process the data, or their client may use accountancy software that keeps records up to date. Each of these tools is a primary artifact that describes the activity that will be carried out in each case. On the other hand, double entry book-keeping is a secondary artifact describing a universally applied method of accounting that must be used irrespective of the format of the primary tools. This demonstrates how primary artifacts describe practice while secondary artifacts describe process.

This distinction can be related back to design, whereby primary design artifacts are the transitionary concepts generated during a design project, while secondary artifacts describe the modes of activity involved in a design process (Bertelsen, 2001). In this manner sketches become primary artifacts. However, a requirement for sketching does not provide a practical concept for structuring the design process. Sketching is a catch-all term describing a free-form contextual activity. What is required are principles that can

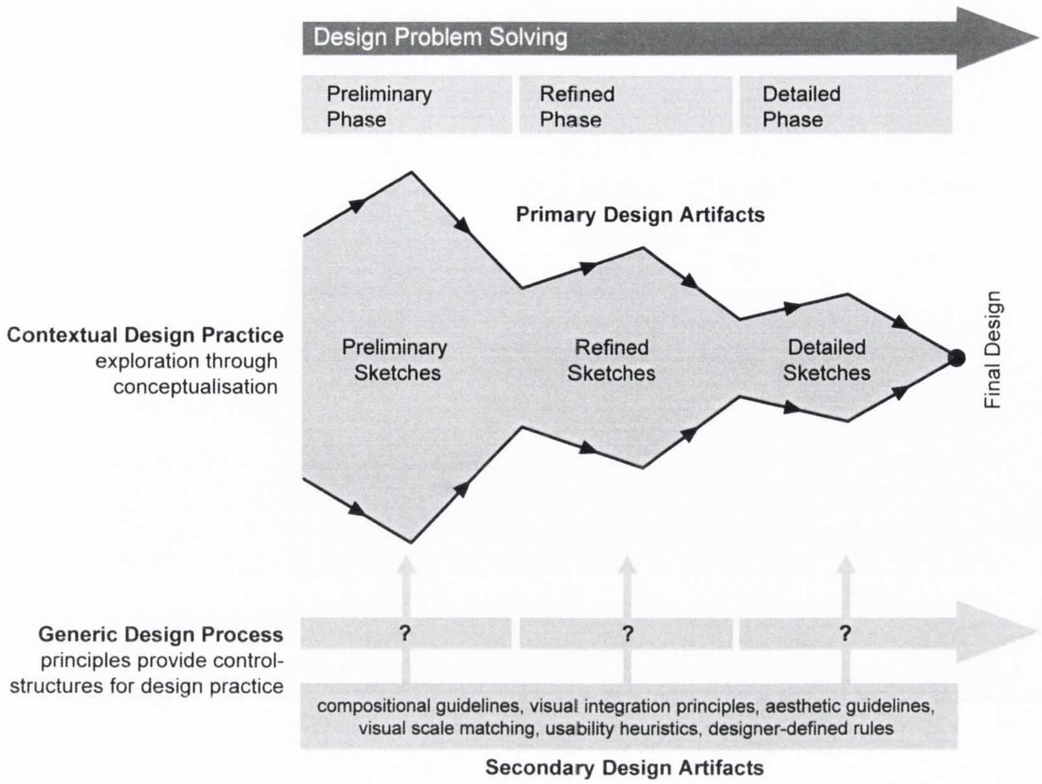


Figure 4.7: A generic model of design

be used to control the sketching activity. These principles act as stopping rules that allowing a designer to stop design exploration, to commit to a particular concept and move to the next phase of the design process. These principles may be any of the design guidelines described in chapter 3 or they may be heuristics developed by the designer. In either case, the process of design can only be revealed through the identification of these secondary design artifacts. Primary and secondary design artifacts provide us with a conceptual framework for analysing contextual design practice and extracting more generic design methods and principles (see figure 4.7). However to provide a full meta-model of the CSE design process, it is necessary to incorporate the analytical methods used during design problem structuring.

4.3 A Flexible Model of Cognitive Systems Engineering

The model of product engineering (figure 4.5) assumes that a product design specification is available to the designer. This significantly reduces the problem-structuring phase in the overall design process. With simple physical products specifications are generally quite straightforward to develop, but the scale and complexity of the information products produced by cognitive engineering makes problem structuring significantly more difficult. While the previous chapters present analysis and design as two distinct activities, in practice their purpose and even their conceptual approach is quite similar. By understanding analytical models as particular types of sketches it becomes possible to extend the generic design model shown in figure 4.7 into the problem-structuring phase.

4.3.1 Analytical Models as Sketches

Each of the analytical frameworks aims to produce models of the cognitive system. However, the analytical models presented in exemplars are generally restricted to the final concepts that the analyst commits to. These are the product of extensive research where many incomplete or partial models are produced as temporary design artifacts. Problem structuring is a generative process where information about the system is increased. Defining a functional model is a reductive process, where an analyst commits to a particular understanding of functionality and structures the information into a format that reflects this view. In this manner, analytical models can be considered to be sketches or conceptualizations of system functionality. The models produced during a specific project are primary design artifacts in the design problem-structuring phase. One the other hand, the principles proposed by the analytical frameworks are secondary design artifacts that allow an analyst to judge when a specific model is complete. For example, a model of the management structure within an organization is a primary artifact that can be used to understand workflows. At the same time, modeling social structures is a defined method for analyzing distributed cognition in a work system and acts as a secondary artifact in a cognitive engineering process.

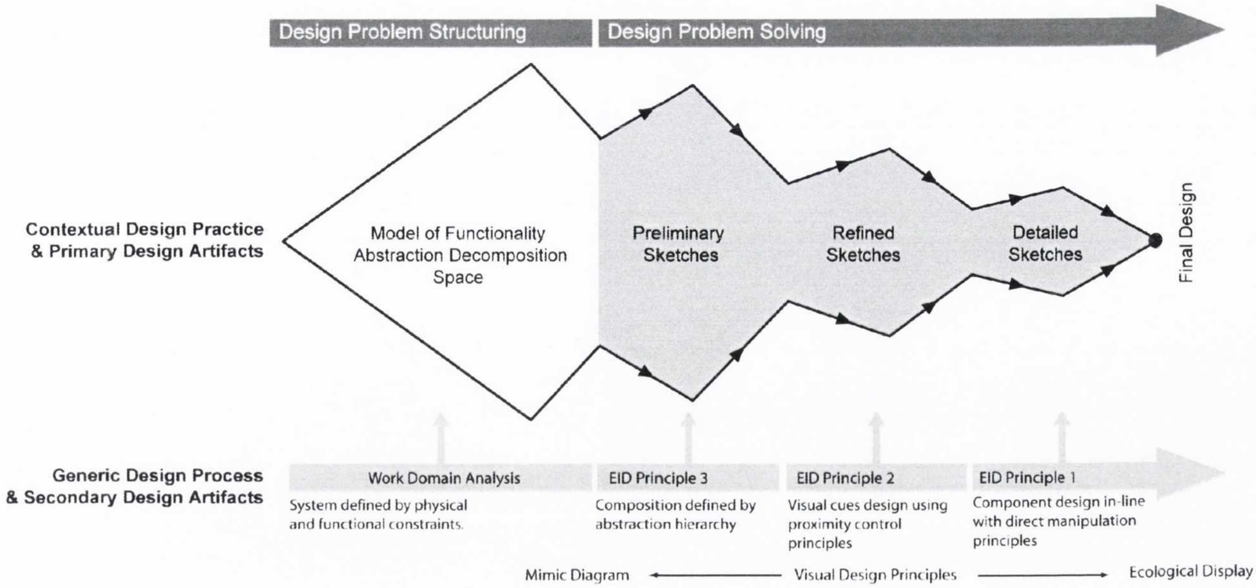


Figure 4.8: The CSE process meta-model applied to EID

4.3.2 A Meta-model of the Cognitive Engineering Process

In this manner problem structuring can be described using the same concept of information elaboration and reduction that describes design problem solving. This allows problem structuring to be incorporated into the design model to provide a meta-model of the entire cognitive engineering design process. Figure 4.8 uses the standard EID process to demonstrate how this meta-model can be applied. The problem-structuring phase uses work domain modelling as a secondary artefact and produces an ADS as primary artifact that models system functionality. The three visual design principles can be considered as secondary artefacts that inform the preliminary, refined and detailed phases of design problem solving respectively. Preliminary design is guided by the principle of presenting the information requirements in the form of an abstraction hierarchy to serve as an externalised system model. Refined design relates to the generation of configural displays that satisfy the second principle of mapping system constraints at different levels of abstraction to visual cues. Detailed design relates to the principle that requires direct manipulation and a representation that supports the part-whole structure of movements.

This view on the role of analytical models in the cognitive engineering process allows us to reconcile the various approaches to cognitive systems research outlined in chapter

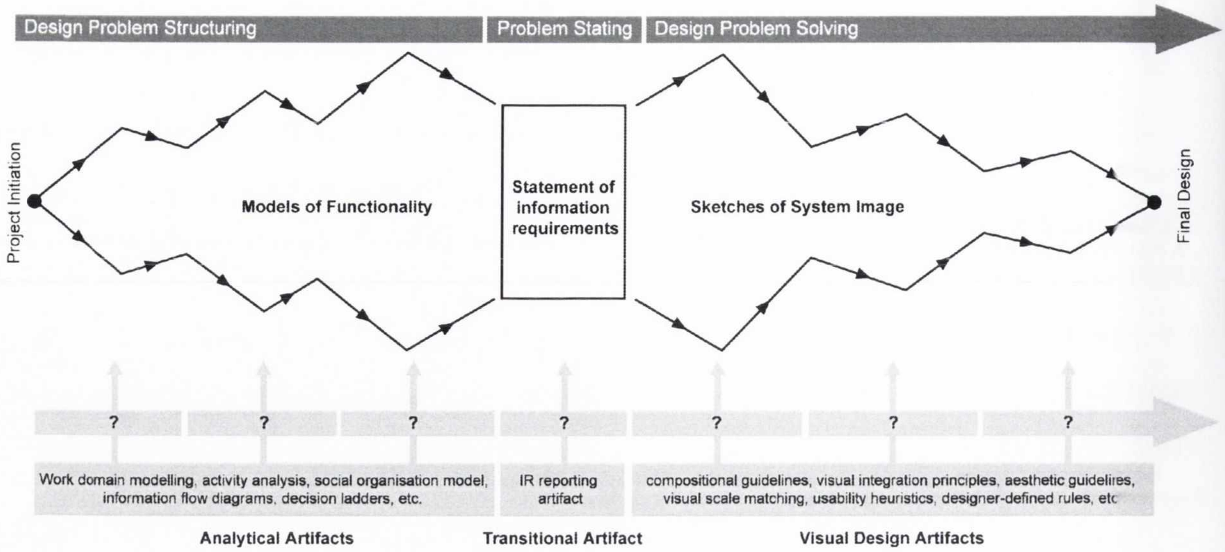


Figure 4.9: Multiple phases of analysis and design

2. The advantages and weaknesses of each approach relates to the academic backgrounds and the illustrative exemplars around which they were originally developed. While these concepts have been used to develop complete frameworks, it is important to realise that none can produce a fully comprehensive model of a cognitive system. Their purpose is to provide *a means for understanding the complexity of real-world functionality* not to replicate reality. Once we accept that models can only provide partial views, the various models provided by these frameworks can be used in a pragmatic manner, as *components in an analytical toolkit* for studying system functionality. Each new model brings an alternative but complementary perspective on how the system operates and models can be mixed to provide a more robust system model. In the same way that problem-solving goes through different phases, problem structuring may also progress through multiple phases as different views of a system are developed (figure 4.9).

The meta-model depicted in figure 4.9 provides a means for structuring design practice in a generic manner. However it does not answer a range of important questions. How can an analyst know which analytical models to use, when should they be applied in a design process and how can they be integrated in a manner that informs the practice of visual design?

Before these questions are discussed further, it is necessary to reflect on the issues covered so far. Chapter 1 identified a need to develop and communicate cognitive systems engineering approaches for the design of visual decision support systems in a sociotechnical enterprise. Chapter 2 outlined the current analytical approaches to studying cognitive systems and demonstrated the limitations of each for modelling such a system. Chapter 3 reviewed current visual design guidelines and concluded that design is a contextual and exploratory activity that cannot be formalized into a straightforward linear process. In this section we have argued, that although design is a highly contextual activity, through the observation of practice it is possible to identify high-level principles that inform design decisions. By identifying these principles it becomes possible to show how theories can be combined to inform design practice within the context of a particular work system. This particular configuration of theoretical principles can be seen as design knowledge that can be applicable to other work systems that exhibit similar characteristics. The meta-model provides a *conceptual tool* for conducting this practice-led research.

4.3.3 Using the Model to Build CSE Design Theory

Practice-led research is concerned with the nature of practice that leads to new knowledge that has operational significance for that practice. The requirement for such an approach is best described by Archer (Archer, 1995) who states,

> There are circumstances where the best or only way to shed light on a proposition, a principle, a material, a process or a function is to attempt to construct something, or to enact something, calculated to explore, embody or test it.

Practice-led research takes its epistemological stance from the concept of *double-loop learning* (Argyris et al., 1985). This concept proposes that learning and *theory development both defines and is defined by action.* Any organization or process is subject to governing principles that define the action strategies that are used to achieve goals, and all action results in changes and consequences (see figure 4.10). If the consequence of action does not match these goals, alternative known strategies may be applied until a suitable outcome is reached. This is described as single-loop learning. However, where existing strategies fail to achieve the required goals, new strategies must be developed. If these are successful, the governing variables themselves must be questioned and adjusted, or new ones proposed. This is described as double-loop learning, as action bears

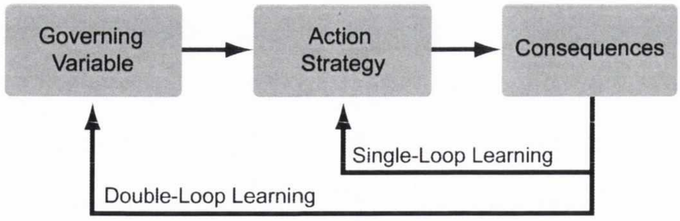


Figure 4.10: Double-loop learning

the responsibility of both affecting changes and developing principles. This approach to generating practice-relevant knowledge has become widespread in the art, design and architecture disciplines (Rust et al., 2007) and has also been applied to research in the medical (Potter et al., 2006) & services (Weinberg, 1971) sectors.

While practice-led research has not been explicitly applied to cognitive systems engineering, the strong design focus taken by the discipline means that it is well placed to benefit from this approach. CSE analyses the interaction between human and technical agents with the aim of designing better work systems. However making changes to technological systems in organisations has been described as "a kind of experimental intervention into ongoing fields of activity" (Flores et al., 1988). Woods describes how CSE researchers develop designs that embody *hypotheses* about how technology shapes cognition and collaboration (Woods, 1998). These designs are prototypes rather than final design solutions as their purpose is to test these hypotheses about how a cognitive systems works. Should a design prototype fail to capture system functionality adeqeuately, then the strategies used to generate it must be examined and new strategies proposed. In this way cognitive systems engineering can be essentially considered as a practice-led research process.

Action research (Lewin, 1948; Trist, 1976; Checkland and Scholes, 1990) is an iterative inquiry process developed around the concept of double-loop learning and is the most well-known practice-led research method. It was originally applied to organizational learning and the approach generally requires extensive, long-term experimentation making it difficult to apply to shorter, design focussed studies. *Case studies* provide another form of practice-oriented research, where theory is interpreted though its application to practice. As their purpose is to clarify by example, case studies can be applied to individual projects and can therefore be relatively succinct. However where existing theories prove inadequate for a design problem, case studies are insufficient. *Action-Case research* (Braa and Vidgen, 1995) is practice-led research method that seeks to combine the advantages of both action research and case studies. The action-case method was initially developed to overcome the timescale issue when applying action research to information system design but it has also been used to support theory building in relation to representational design (Yen et al., 2002). By applying double-loop learning within the context of a design project, design knowledge can be generated.

In relation to cognitive systems engineering, the action-case approach can be applied to develop new design knowledge in the form of comprehensive design methodologies. CSE aims to inform design by providing governing principles in the form of models and guidelines. The CSE meta-model provides a generic structure of the design process and can be used to identify where these principles succeed or fail to inform design practice. By reflecting on the manner in which a designer bridges design gaps it becomes possible to identify new principles and to incorporate these into more comprehensive design methodologies.

4.3.4 Bridging the Design Gaps

While a number of gaps in CSE design knowledge have been identified, the reality of the matter is that successful design does occur, in spite of incomplete knowledge. This suggests that studying how design problems are solved in practice provides a reasonable starting point for closing these gaps.

Visual design involves the development of concepts that change the way in which the designer sees the problem. In this way, each concept can be considered an action strategy. Designers continue to generate concepts until they are satisfied that a goal state has been reached. As such, conceptualization is a form of single-loop learning. However, extracting the governing principles that control conceptualization though examination and reflection on practice involves double-loop learning. In this manner design theory can be developed and methodologies can be proposed. This will help to reduce the representational design gap.

Cognitive system analysis involves the construction of models that describe system functionality. While a number of approaches have been proposed for achieving this, they each take an alternative view on how functionality occurs. By using the analytical models in a pragmatic way, to describe causal and intentional aspects of the system as required, these alternative views can be mixed. By describing how this occurs in practice and by identifying how these models inform design decisions, procedural methodologies for analyzing particular types of cognitive systems can be revealed. This should bridge the analytical gaps described in chapter 2.

A CSE design process involves both analysis and design. As was discussed at the start of this chapter, the relationship between analytical models and visual design is notoriously difficult to describe. The utility of models as design artifacts has been discussed in relation to causal systems (Wong, 1999; Potter et al., 2002) but these are subject to certain limitations. There is even less information available on how models of intentional systems can be used to directly inform visual design. Through the reporting of the complete CSE design process and subsequent reflection on the design decisions made along the way, transitional artifacts between models and sketches can be identified. The specification of these transitional artifacts should reduce the design process gap.

The meta-model depicted in figure 4.9 provides a tool for analysing the practice of cognitive systems engineering. In the following chapters this approach is applied to two separate projects with the aim of extracting and defining re-useable design knowledge. While the initial goal was to generate CSE methodologies, other forms of re-usable design knowledge have also been produced. Problem structuring artefacts include models of system functionality that can inform future design projects. An example of this is provided in this work, where a structural model of the enterprise developed in the first project is re-used in the second design project. Problem solving artefacts include multiple sketches of task-focussed graphical representations. These sketches can inform representational design for other projects involving similar tasks. The methodologies themselves provide a context for selecting and re-using secondary design artifacts. Although the projects reported here involve different cognitive systems and have different scopes, the two extracted methodologies have similarities in terms of the steps involved and the sequence in which they are applied. The second methodology is essentially an extension of the first that models a wider range of constraints. This raises interesting questions about the characteristics and categories of cognitive systems that are further discussed in chapter 8.

4.4 Summary

This chapter has examined the gaps in CSE design knowledge and highlights the fundamental difficulty in describing how design happens. A meta-model of the CSE design process has been developed that presents design as an ill-defined but structured form of problem solving. By distinguishing between primary and secondary design artifacts it is proposed that this meta-model can be used to extract generic design methodologies from contextual design practice. The action-case method is proposed as a suitable form of practice-led research for generating design knowledge. This provides the basic approach for conducting two studies that apply the CSE meta-model to design projects in a sociotechnical enterprise.

Chapter 5

Semiconductor Manufacturing Overview

Semiconductor manufacturing is carried out in a large fabrication environment and involves intricate process flows, high levels of automation and a sizable social organisation. These characteristics make it an appropriate example of sociotechnical enterprise. The following chapters report on two design projects carried out in this domain. In this chapter an overview of the industry is presented to introduce the purpose and context of the design work. Although this information is presented here before the design projects the process of gathering this information should be seen as part of design research. Hoffman describes familiarisation with the domain as a bootstrapping approach (Hoffman, 2005), whereby the designer can develop a basic knowledge of specialist vocabularies and general principles. In this case eight one-hour web-based training courses were completed covering; an introduction to semiconductor manufacturing, the production process, material handling, safety, manufacturing execution systems, engineering reporting tools, quality control and Moore's law. In addition to this a course in cleanroom gowning procedures was completed and a factory window tour was carried out.

5.1 The Production Process

Semiconductor manufacturing involves the production of Integrated Circuits (IC's) or computer chips. IC's are made up of millions of transistors built on top of a silicon base.

The semiconducting properties of silicon make it ideal for building transistors and this has resulted in the term *semiconductor manufacturing.*

The manufacturing process can be divided into four main phases; front-end and back-end processing, testing and packaging. Front-end processing involves the generation of transistors on the surface of a silicon wafer. Back-end processing involves the building of the circuits that link the various transistors. Testing allows for faulty product to be removed from production and occurs both throughout the process and at the end of the line. Packaging involves cutting the wafer into individual chips or 'dice', attaching the pins that allow a die to communicate with the end device and sealing the dice to protect their structures. The whole process involves over 800 process steps.

The number of transistors that can be placed in an IC directly increases the processing power of the chip. Consequently, there is a continuous drive to reduce the scale at which individual transistor components are built. Current techniques can construct components measuring just a few nanometres in diameter, allowing millions of transistors to be built within just a few square millimetres. Working at these scales requires a unique production process involving chemical, physical and photographic processes.

5.1.1 Processing Techniques

As transistor components are too small to permit physical assembly, they are built up on the surface of a silicon wafer layer by layer using additive and subtractive techniques. Production begins with a silicon wafer. This is a thin disk of pure silicon on the surface of which millions of transistors are generated by selectively modifying the electrical properties of the silicon. This is carried out by; patterning the surface of the wafer, changing the molecular structure, altering the physical structure and repeating these stages until the transistors are formed. This marks the completion of front-end processing.

Once the transistors have been formed, the wafer is coated with successive layers of conductive and insulating materials that are each patterned and shaped to form the circuitry. The number of layers built will depend in the complexity of the product. The completion of this activity marks the end of back-end processing. While different materials are used on different layers and in different ends, processing steps can be broadly categorised into deposition, patterning, removal and modification activities (see figure 5.1).

Deposition involves laying down conducting or insulating material onto the surface of the wafer. Different techniques such as physical vapour deposition and chemical vapour deposition are used. These layers are the building materials for the circuitry.

Patterning is a lithographic process where the wafer is coated with a light sensitive substance called photoresist, placed under a lens and has a pattern projected onto its surface. Exposure to light modifies the photoresist, hardening the exposed regions in a process very similar to photographic printing. The non-exposed regions are then washed away using a developing solution.

Removal involves stripping away material from the wafers surface and this can occur in a number of different ways. Following patterning, an etching process erodes into the exposed areas of the wafer removing regions of the underlying material. After this a different type of removal process known as planing is used to strip away remaining photoresist and polish down the underlying material to an even surface.

Modification is a process whereby the electrical properties of a material are changed. For example silicon can be converted from an insulator into a semiconductor by introducing impurities into its crystalline structure. This is referred to as doping and this activity is what allows transistors to be built out of the silicon wafer. Diffusion and Ion implantation are the two main forms of modification used in semiconductor manufacturing.

These categories broadly define methods for building IC's and explain how components are constructed layer by layer. However, each category has a number of specific techniques associated with it. Each technique requires a specialised process tool and these can use various combinations of materials depending on the component being built.

5.1.2 The Physical Product

The process above describes how IC's are constructed. The basic elements involved are *transistors*, the *metal layers* that make up the circuit design and metal *interconnects* that join the layers to the transistors. Figure 5.2 illustrates the relationship between these elements and the product or WIP (Work In Progress) that moves through the line. Each

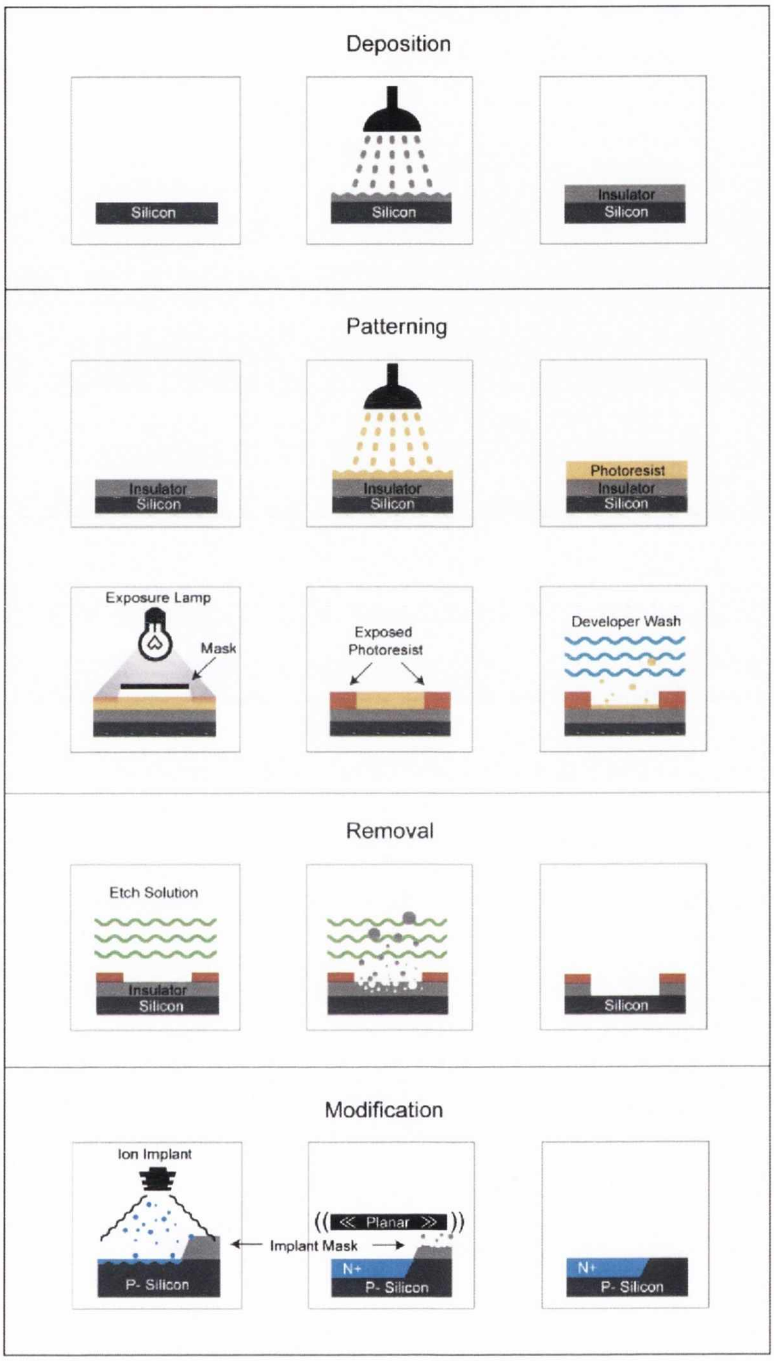


Figure 5.1: Processing Techniques

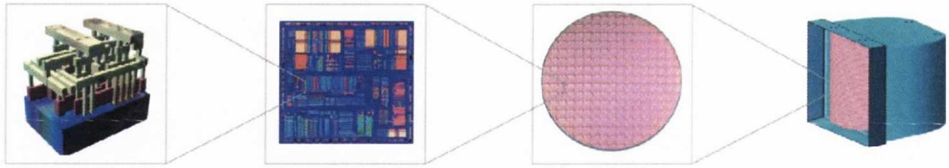


Figure 5.2: The product at different resolutions

IC consists of a number of *cells* made up of transistors, layers and interconnects. As complete IC's are still very small, thousands are built on the surface of each silicon wafer. The surface of a wafer is a grid of dice, where each *die* is a complete IC that will form the final-end product, the computer chip. *Wafers* are thin discs of silicon and are shaped this way to accommodate processing techniques and to provide a more robust form as silicon is relatively brittle. Wafers are transported in *lot boxes* or *FOUPs* (Front Opening Unified Pods). These containers contain 25 wafers and are used to protect the wafers as they move through production. A *lot* describes the basic unit of WIP that runs through the process. Wafers in a lot will generally have the same specifications.

5.1.3 The Production Line

The four *phases* of front-end and back-end processing, testing and packaging mark the major divisions in the production line and the movement of WIP between these phases is progressive. The line can also be subdivided into a number of manufacturing *regions* each representing approximately one week's progression through the line. Movement of WIP between the regions is also progressive. Within the phases and regions, *process steps* indicate points where the wafers enter a tool and have *operations* carried out on them. Although movement between process steps is conceptually progressive, the actual movement of the WIP through process tools can be iterative. The layered method of manufacturing means that deposition, patterning and removal operations occur repeatedly as each *layer* is created. *This means that the same operation can be carried out on different layers by the same process tool* (see figure 5.3). There are a number of reasons for tool re-use. Firstly, process tools are extremely expensive and have very high depreciation rates. It is necessary to buy the minimum number of tools and to use them as much as

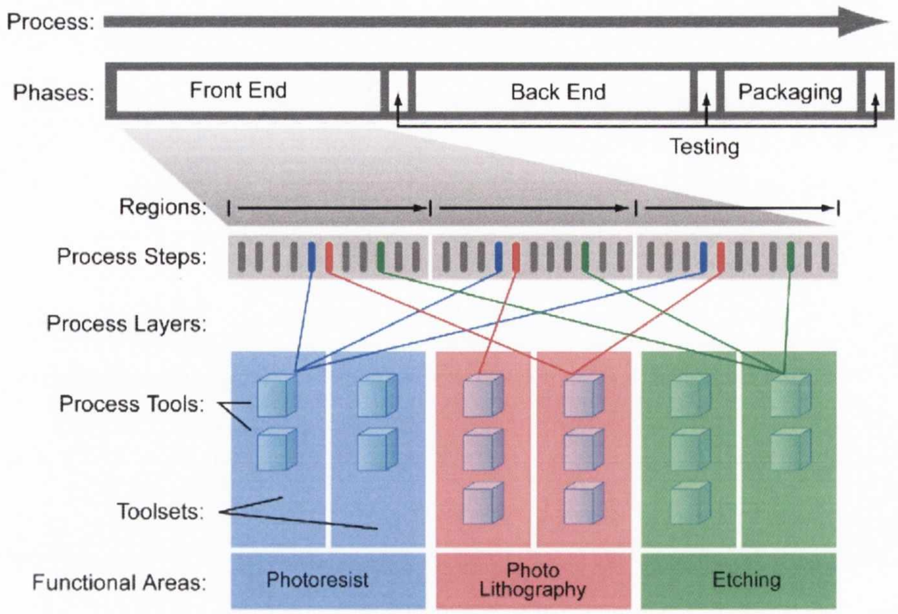


Figure 5.3: Processing and WIP movement

possible in order to recoup investment. Allowing multiple layers to run on a tool reduces the chances of it standing idle. Secondly, individual tools can have particular quirks e.g. a tiny offset in a lithographic tool. As the process operates on a nanoscale, small variation can have major impacts. Re-using a tool can minimise layer-to-layer variation and reduce the impact of these quirks. However, this highly re-entrant process flow creates unique challenges for WIP tracking and scheduling.

5.2 The Manufacturing Enterprise

The previous section describes the production process but this is only one aspect of the overall enterprise. The work environment and social organisation also play a major role in supporting enterprise functionality.

5.2.1 The Fabrication Environment

The entire production process takes place in a Semiconductor Fabrication Plant referred to as a *Fab*. The physical layout of the fab is complex and fab designs change with technological developments, but the general structure is based around a number of conceptual divisions of processing equipment. Operations are carried out by specialised *process tools* and a number of tools that carry out the same operation form a *toolset*. Toolsets that use the same model of tool form a *module* and modules that carry out the same general functional activity such as etching or lithography form a *functional area* (see figure 5.4). Traditionally fab layout was designed to minimize cycle time, the time it takes to move wafers between process tools, so the layout was based on clusters of toolsets. More recently, the development of Automated Material Handling Systems (AMHS) reduces transportation constraints and the industry is moving towards a more farmed layout where the fab is organised by functional areas (Yeaman and Stachura, 2002).

As a *nanotechnology*, semiconductor manufacturing is extremely sensitive to environmental factors. Tiny particles introduced by workers or material impurities can damage an IC rendering it non-functional. Consequently, the majority of processing takes places in a *cleanroom* environment and manufacturing technicians must change into specially designed gowns when entering the cleanrooms to avoid introducing particles into the environment. The fab is divided up into a number of different cleanrooms, which are each divided into two zones; the *bay* and the *chase*. In the bays manufacturing technicians move between the loading ports of process tools loading wafers for processing. Most of the tool is located in the chase. This is a more highly controlled cleanroom environment where processing occurs and where the equipment technicians carry out maintenance on the tools.

5.2.2 The Social Organisation

Several departments take responsibility for different parts of the overall system functionality. Collaboration between departments occurs at different levels to ensure that the facility operates and develops in a stable manner. While certain departments are common across all industries (i.e. IT, R&D, HR etc.), three are particular to production enterprises; namely manufacturing, engineering and quality/yield. These departments view the manufacturing facility in distinctive ways and have different goals (see figure 5.5).

Manufacturing: The manufacturing department deals with the production of orders for clients. It must ensure that WIP moves through the line in a timely fashion so that

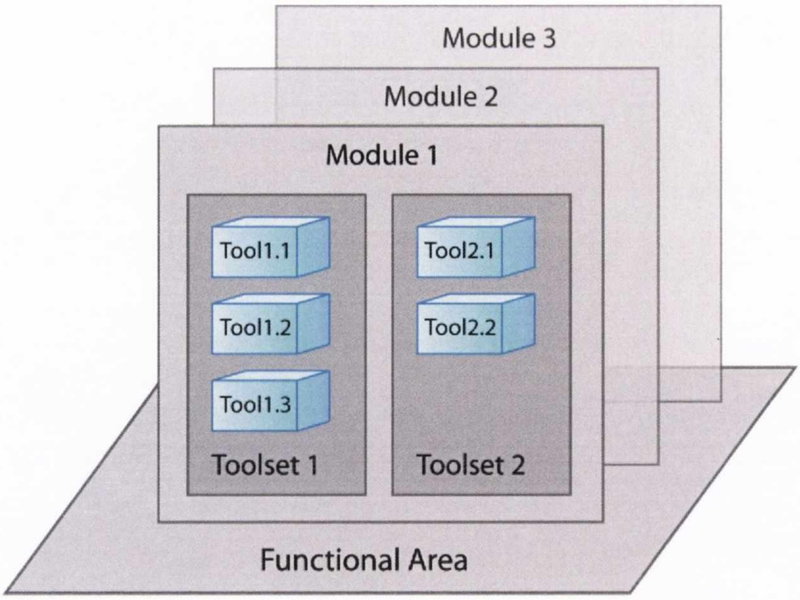


Figure 5.4: Categories of processing equipment

delivery dates are met. Its primary activities relates to scheduling and WIP management. A manufacturing view of the fab is predominantly process focused, as activities include identifying WIP build up in the line, locating orders and monitoring progress.

Engineering: The engineering department deals with the maintenance of equipment in the fab. It must ensure that process tools are in good working order and are available for manufacturing. Activities include carrying out preventative maintenance and diagnosing and repairing mechanical faults. As tools are highly specialized, an engineering view of the fab is based around the structural divisions (functional areas, toolsets etc.) shown in figure 5.4

Quality Control: The QC department examines performance of the fab in relation to yields. Two forms of yield are used to gauge performance. Line yield refers to the number of good wafers produced without being scrapped and can be used to indicate the effectiveness of material handling, process control, and labour. Die yield refers to the number of good die that pass metrology tests at any point in the process. This is used to identify and remove damaged WIP from the process and measures the effectiveness of process control, design margins and environmental cleanliness. The information is derived from metrology and process control data so QC views are based on variation in tool parameters, between tools and across functional areas.

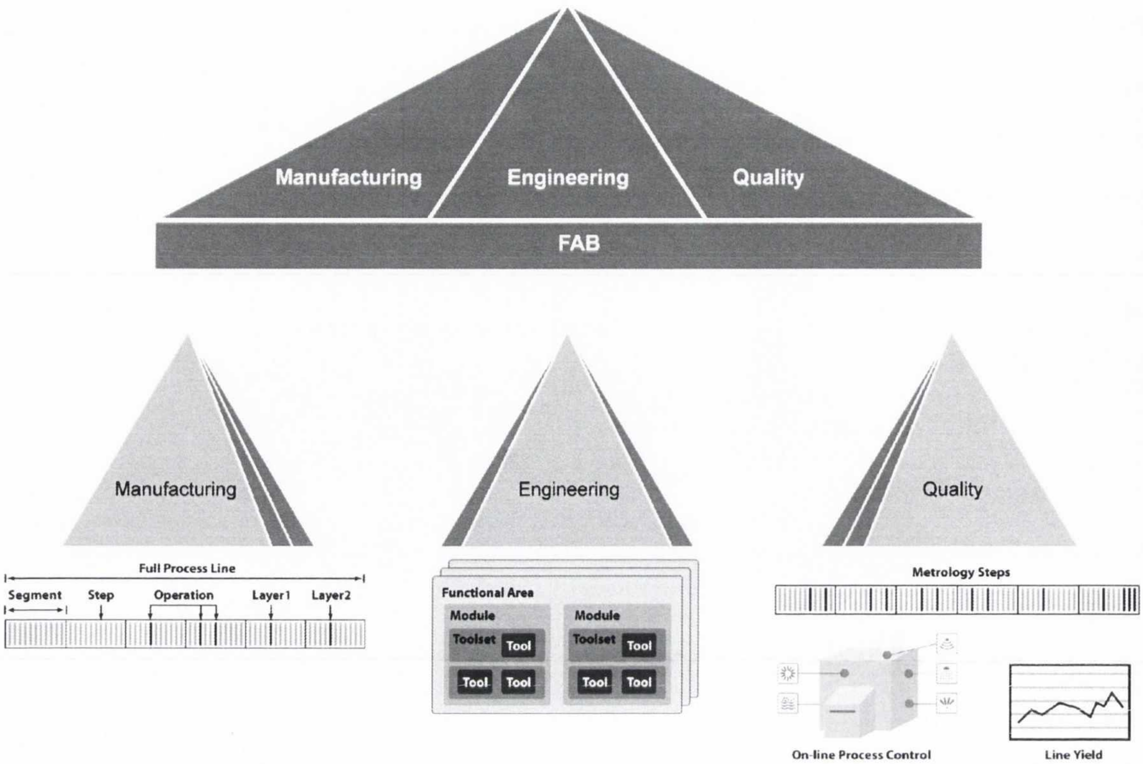


Figure 5.5: Manufacturing, Engineering and Quality Views of the Fab

The social organisation of the fab is divided among these various departments and involves multiple levels of management. Communication plays a key role in ensuring that the fab runs smoothly and frontline workers can trigger activities in any of these departments through observations or requests for intervention.

5.3 An Evolving Industry

Moore's Law is based on a prediction made by Gordon Moore, one of the founders of Intel Corporation, in 1965 (Moore, 1965). It proposes that the number of transistors that can be placed on an integrated circuit is increasing exponentially, doubling approximately every 2 years (fig.5.6). This prediction was initially based around the ability to reduce component size and has been consistently met for the last three decades. This "Law" has become a self-fulfilling prophecy defining a target that industry continually strives to achieve. From a manufacturing perspective, this drive to match Moore's law has required more complex chip designs and production processes.

The standard approach for increasing transistor count has been to reduce the dimen-

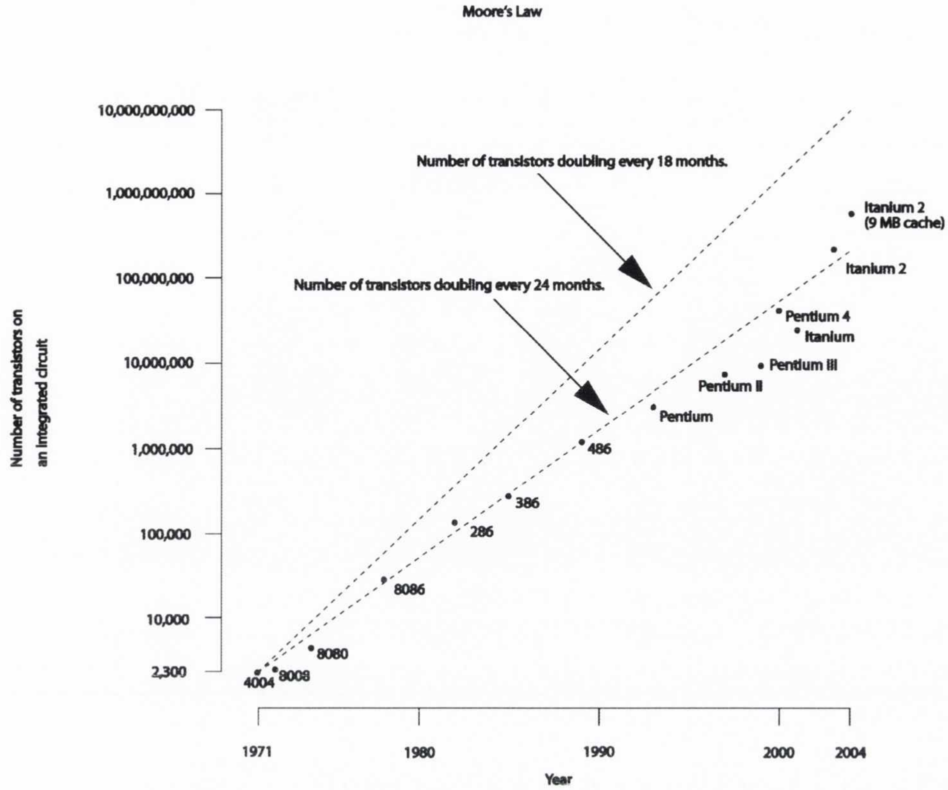


Figure 5.6: Moores Law (Moore, 1965)

sions at which components are built. However this generates new and complex manufacturing challenges. Amongst the more serious of these are process variation and contamination. Contamination control involves the management of particles, metals, organics, and any other undesirable contaminants that result from processing. Contaminants damage the dice during processing and have a direct effect on die yield. The difficulty with smaller component sizes is that the maximum critical particle diameter, also known as "killer defect" size, also decreases (Report, 2000). Managing particles at this minute scale requires new contamination control techniques. As current cleanroom environments are insufficient for controlling particles at this scale, hermetically sealed mini-environments are now pervasive. Automated handling within tools and transportation in airtight FOUPs ensures that wafers are never exposed to the cleanroom environment itself. This has dramatically reduced human and cleanroom contamination but attention now turns to contamination from equipment and the process itself (fig 5.7).

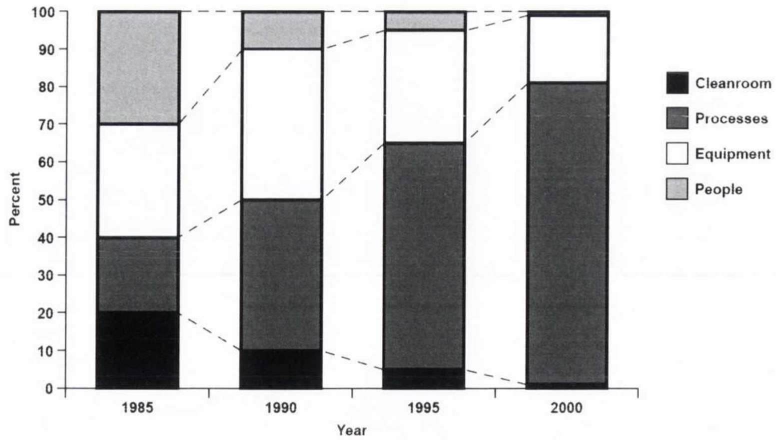


Figure 5.7: Sources of defects. After (Report, 2000)

5.3.1 Advanced Process Control

Improved process control provides the only means for handling this type of contamination. As discussed in the previous section, metrology steps are dispersed throughout the process to measure the performance of processing activities. While this can ensure that the majority of defects are detected, metrology occurs only after the processing has occurred. This means that it detects faults that will either require rework or cause a wafer to be scrapped. It also requires WIP to be placed on-hold while tests are conducted and this slows down the overall production rate. This has motivated a desire to move from off-line or post-processing metrology to a more on-line approach. By generating a model of normative processing behaviour, tools can be monitored in real time during the processing activity. If behaviour becomes erratic and parameters move outside of normal limits the tool can be taken off-line to prevent damage. This preventative approach can reduce the need for rework and the production of scrap. Advanced Process Control Systems (APCS) are currently used to achieve this. When tools begin to stray from normal targets the APCS can issue a warning, allowing a human controller to inspect the problem. By automatically monitoring multiple parameters within process tools, these systems can both minimize process variation and make defects easier to detect.

APCS's require careful management. Developing normative models requires process engineers to carry out multivariate analysis of tool performance. Changes in processing techniques and the introduction of new products means that this is an on-going task. In addition, certain types of variation may trigger a warning but may not indicate a true fault. Understanding the difference will depend on the level of experience held by the human controller. Finally, the automatic system itself is not foolproof. The sensors used to measure parameters can become damaged and controllers may set-up parameter targets incorrectly. These limitations make it necessary for human controllers to monitor and control the APCS. The scale and complexity of the system, coupled with a diverse set of users makes the development of an APCS supervisory display a challenging design problem. The project in the next chapter deals with the design of visual decision support system for APCS health monitoring.

5.3.2 Remote Operations Control

The second project deals with changes to system functionality on a much larger scale. The higher-levels of precision described above, have a direct effect on production costs. These rising costs are being met by increasing the capacity of high volume manufacturing. This approach seeks to reduce unit costs by escalating line yield. One of the proven strategies for achieving this is to increase the size of the silicon wafer on which the semiconductors are built. Larger wafers result in more end-of-line dice for the same amount of processing. Over the decades wafers have increased in size from 3 inches to the current 300mm standard and are set to increase further. However, the levels of precision involved in manufacturing dictate that a change in wafer size requires the development of new processing tools. The latest move from 200mm to 300mm has proven to be the most complex yet as it requires an entirely new set of design parameters for the factory (Planta, 1997).

An initial challenge relates to ergonomic issues. Technicians could physically load 200mm FOUPs into process tools, but the size and weight of 300mm FOUPs exceed human manual-handling constraints. This has required the development of a fully pervasive Automated Material Handling System (AMHS). As this system is responsible for the transportation, loading and storage of WIP, an advanced Manufacturing Execution System (MES) was also required to manage new scheduling challenges resulting from these changes. As was discussed in chapter 1, these developments are part of a wider automation roadmap that is defining the future of semiconductor manufacturing (Srinivasan, 2001). The mechanical automation is accompanied by data automation systems including equipment control, scheduling and manufacturing systems. The tight integration of these systems is essential to achieve continuous, uninterrupted processing.

Together this combination of mechanical and intelligent automation has changed the functionality of the manufacturing enterprise. Pervasive mechanical automation means that machine operators no longer need to be co-located with processing tools. As a result, operations control is moving to a more centralised, remote operations model. However, this move changes the cognitive system associated with manufacturing control in a number of ways. The second design project relates to the development of a new visual decision support system for remote operations control.

5.4 Summary

This chapter provides a brief overview of the characteristics, functionality and evolution of the semiconductor manufacturing industry. Manufacturing operates at a nanoscale and requires high levels of process and environmental control. The production process itself is very intricate and management of the overall systems is split between a number of departments including manufacturing, engineering and quality control. A continuing drive to reduce feature size has required increasing levels of process control and the development of new on-line automated PCS systems. The higher costs associated with increased precision are being offset with higher volumes of production through more pervasive automation. These developments provide the background for two design projects that are reported in the following two chapters.

Chapter 6

Designing Visual Decision Support for PCS Health Monitoring

6.1 Introduction

In the previous chapter the role of Advanced Process Control Systems (APCS) in semiconductor manufacturing was introduced and a number of challenges relating to their successful management were identified. This chapter reports on the redesign of a visual decision support system used to monitor PCS health in a semiconductor fabrication facility (fab). This requires the three design gaps in CSE knowledge to be bridged by answering the questions:

1. What analytical approach should be taken to identify the information requirements?
2. What representational guidelines are appropriate for designing a system image?
3. What design process should be followed in the development of this interface?

The CSE meta-model is initially used to structure the reporting of this project into problem-structuring, problem-stating and problem-solving phases and through this process a visual prototype is designed. This prototype is evaluated in terms of both validity and verification. The CSE meta-model is subsequently used to extract a more generic design methodology from this design process by identifying the secondary design artifact used and the order in which they were applied.

6.2 Process Control in Semiconductor Manufacturing

A process control system (PCS) combines statistical and engineering techniques to control the output of a specific process. Within manufacturing industries, PCS's play a critical role in ensuring that products are manufactured to the same quality and standard. Sensors measure different parameters (temperature, pressure etc.) across multiple machines to ensure that they are complying with pre-set targets. Advanced PCS's (APCS) automatically monitor these readings to ensure that out-of-control machines are taken off-line as quickly as possible and that engineers are alerted to the problem. As APCS plays such a critical role in system control it is essential that it functions correctly. However, targets in a PCS are not static. They can move over time due to process developments, product changes and other factors. Another problem is that sensors themselves can fail or become damaged leading to erroneous data being reported. Consequently PCS systems must be carefully managed to maximise detection and minimise the risk of false alerts. The scale and complexity of the semiconductor production process requires PCS management to be distributed across teams in the process engineering department.

6.2.1 PCS Health Monitoring

The process engineering department is responsible for designing, developing and monitoring production processes in the fab. The department is structured along the same lines as the engineering hierarchy outlined in section 5.2.1 allowing process engineers to develop specialist knowledge in particular areas. PCS management involves inspecting tool performance data, identifying anomalies and diagnosing causes. In this way the health of the overall PCS system can be monitored and controlled. As on-line process control becomes more pervasive across the enterprise the reporting structures associated with PCS management is changing (figure 6.1).

The original model of PCS health monitoring involves identifying tool performance through a combination of metrology and on-line line sensor data and communicating the overall PCS health through management reporting structures (figure 6.1a). Junior

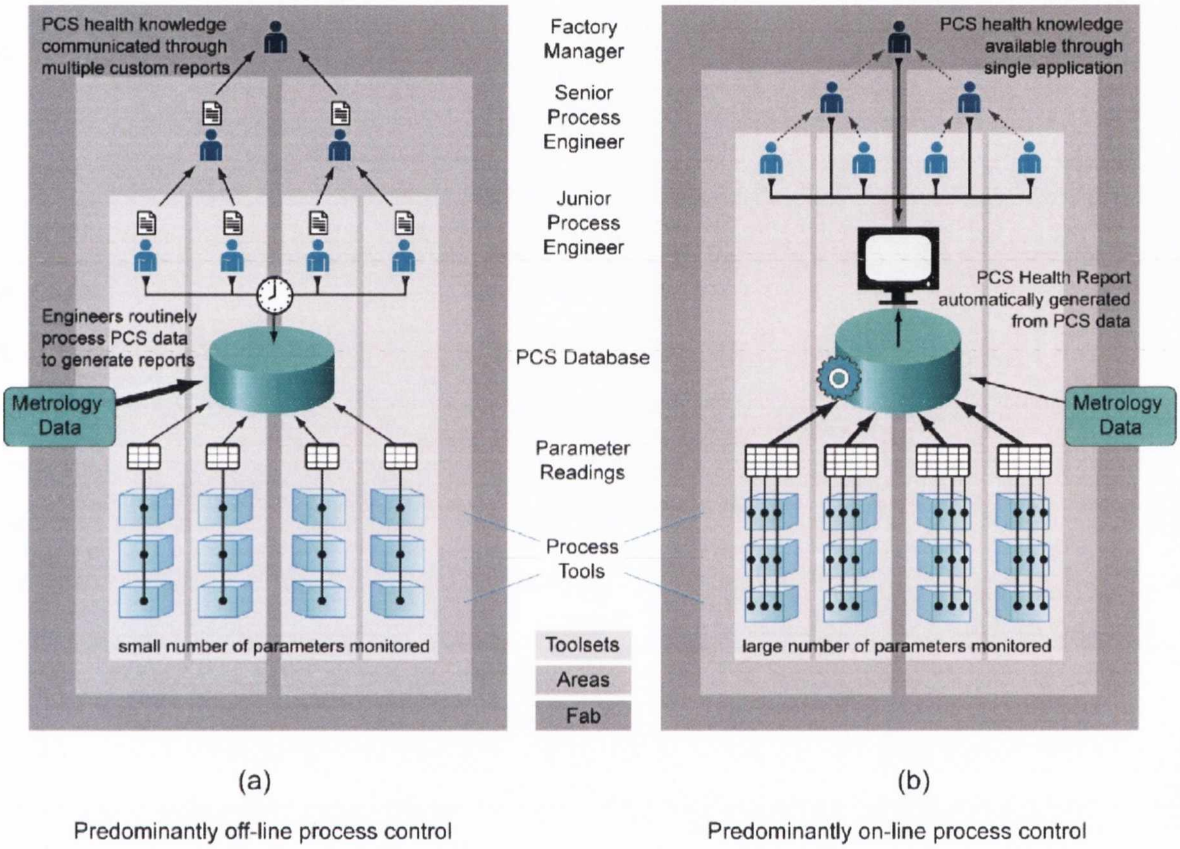


Figure 6.1: PCS managment a) original configuration b) using PCS health report

process engineers are responsible for the performance of a toolset or a number of toolsets. They routinely inspect the parameter data provided by their tools, check that tool sensors are operational and carry out adjustments to performance models when required. They work under a senior process engineer to whom they communicate PCS performance both verbally and by generating reports. Senior process engineers are responsible for PCS management across an entire functional area. They work with junior process engineers on developing performance models and generate area-level, PCS performance reports. Senior process engineers answer to the factory manager who is responsible for fab-wide engineering issues. The factory manager uses the functional areas PCS reports to gauge the overall health of the PCS across the fab.

As on-line process control grows in importance, more data is being collected from a larger number of parameters. Increasing data volumes make data inspection more difficult to complete, so more efficient approaches to PCS management are required. At the same time improvements in data analytics means that performance models are becoming more accurate, allowing PCS data inspection to become increasingly automated. This allows the human aspect of control to move towards a management by exception approach (Dekker and Woods, 1999), where the system identifies anomalies and humans respond to resolve issues. This is transforming PCS health monitoring from a human-driven activity to one that is handled by a joint cognitive system. However this pervasive use of automated control requires a higher degree of system observability (see section 1.1.1). A PCS health report is an application that automatically processes PCS health data to provide performance reports for the entire fab. This allows managers and process engineers to inspect PCS performance at different levels of abstraction through a unified interface (figure 6.1b). This has the advantage of removing the repetitious task of custom report generation while at the same time ensuring a consistent reporting style and navigation structure across the fab.

6.2.2 The PCS Health Report

A simplified example of an existing PCS Health Reporting application is presented in figure 6.2. Screen 1 shows a drill down used to access a particular fab and process.

Screen 2 is the health report overview showing different tool modules within the facility, their corresponding health readings and the indicators that are used to generate the health readings. Screen 3 is an indicator chart showing the various sensor readings in a module, the tools these readings relate to and the parameters they are measuring. This particular chart is an On Target Indicator (OTI) showing the sensors standard deviation from the target for a set duration. This is one of three control indicators used to calculate the health metric. Screen 4 is a trend chart for the performance of a single sensor over a time period. The screens shown here represent only one drill down through the system. Different paths may be taken to resolve different issues. This system was developed in-house to provide the information required for PCS health monitoring, however it has not gained widespread acceptance. While the data itself is relevant, the presentational format does not appear to support the full range of tasks associated with monitoring and optimising the PCS. At the outset of this project a number of issues and requirements were identified.

1. The spreadsheet style presentation format makes it difficult to see the relationships between health values and the various indicators that generated them.

2. The current system provides performance trends only at the sensor level but management would like to be able to view health performance trends at fab, functional area and module levels.

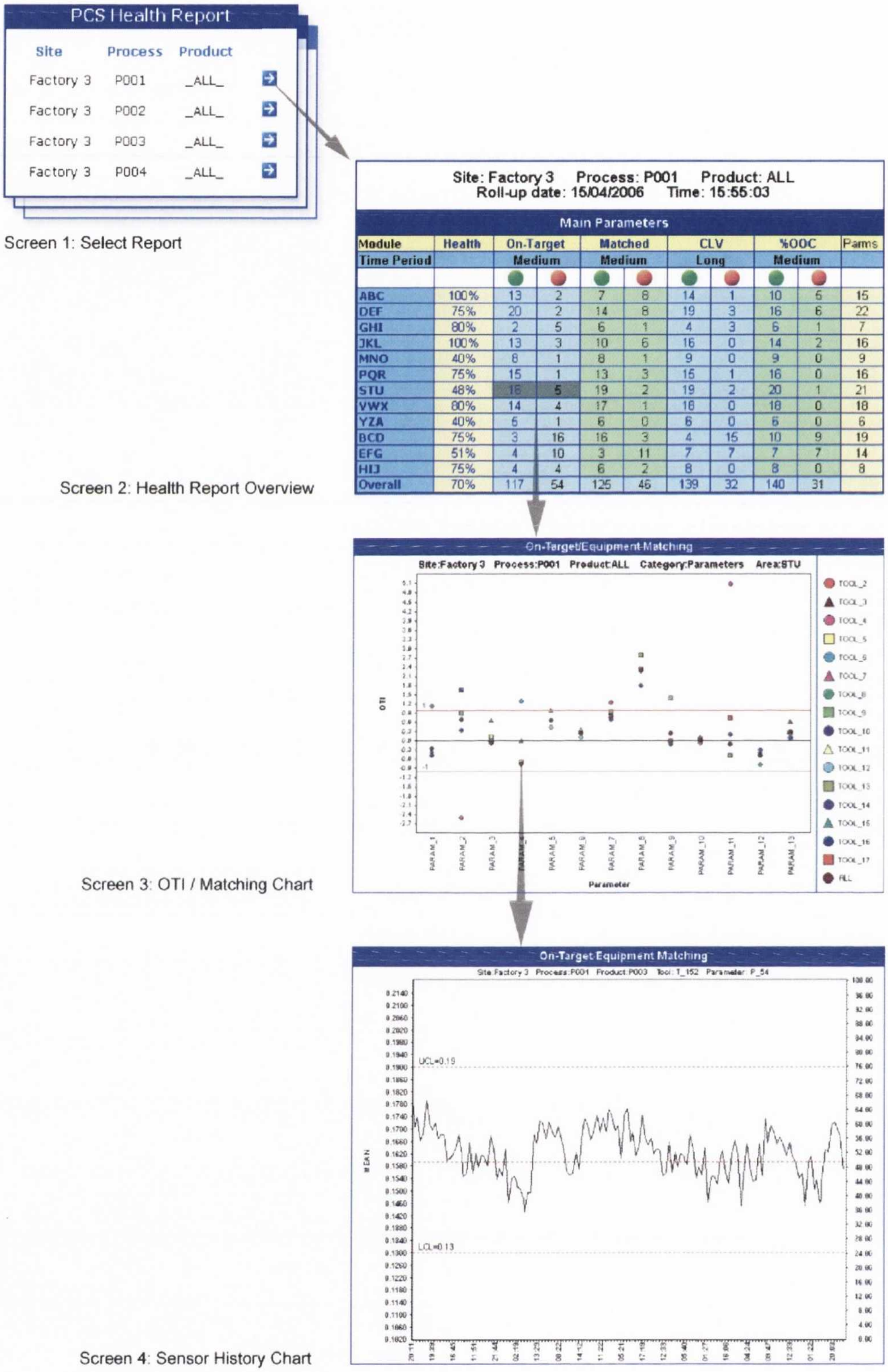
3. A number of issues exist with the indicator charts including:

    (a) Their presentation in pop-up windows makes it difficult to relate information with the overview

    (b) Their format makes horizontal scrolling necessary for screens with a large number of parameters

    (c) The format also makes it difficult to locate and select specific tool sensors.

It was proposed that a new visual decision support system for process control health reporting was required in order to resolve these issues.



Module	Health	On-Target		Matched		CLV		%OOC		Parms
Time Period		Medium		Medium		Long		Medium		
ABC	100%	13	2	7	8	14	1	10	6	15
DEF	75%	20	2	14	8	19	3	16	6	22
GHI	80%	2	5	6	1	4	3	6	1	7
JKL	100%	13	3	10	6	16	0	14	2	16
MNO	40%	8	1	8	1	9	0	9	0	9
PQR	75%	15	1	13	3	15	1	16	0	16
STU	48%	16	5	19	2	19	2	20	1	21
VWX	80%	14	4	17	1	18	0	18	0	18
YZA	40%	6	1	6	0	6	0	6	0	6
BCD	75%	3	16	16	3	4	15	10	9	19
EFG	51%	4	10	3	11	7	7	7	7	14
HIJ	75%	4	4	6	2	8	0	8	0	8
Overall	70%	117	54	125	46	139	32	140	31	

Figure 6.2: A drill down through a PCS health report

6.2.3 Preliminary Project Review

As this project requires an interface for a large complex system involving temporal data, system monitoring and fault diagnosis, Ecological Interface Design (EID) would appear to provide a suitable design framework. However, while EID has been frequently applied to process control systems, PCS health reporting involves monitoring the performance of an *embedded control system* rather than the process itself. As was identified in chapter 2, this creates a range of challenges for applying the framework.

From an analytical perspective, many of the existing EID exemplars are developed around material process flows where the relationship between physical and functional constraints is fixed. The functional structure of these processes can be described through natural laws defining mass or energy transfer and their physical structure can be described through their transportation mechanisms. The fab does involve a material process flow however the PCS is designed around engineering rather than manufacturing concerns. While complexity in material processes comes from coupling and causal relationships, complexity in the PCS health report stems from the huge numbers of components involved. Despite this, the design problem still involves generating a system image that is meaningful to end-users.

From a representational perspective, the EID principles may not provide sufficient support to inform the design process. As the principles relate directly to the work domain model the analysis issue above must be resolved before principles can be applied. Even after this, the system does not have an existing visual vocabulary to draw on, so visual design must be carried out right down to the syntactic level. There is an additional problem of scale. While process control tasks usually involve balancing a small number of variables to achieve a goal, a single PCS health indicator chart can provide hundreds of individual sensor measurements whose configuration indicates a particular system state. The combination of these factors results in a complex visual design challenge.

While EID provides a useful starting point for analysing complex sociotechnical systems, it may not be sufficient to describe all of the characteristics of this particular domain. In the following sections the design process applied to this project is described in terms of the CSE meta-model covering problem structuring, problem stating and problem solving phases.

6.3 Design Problem Structuring

The aim of this phase is to examine the cognitive system associated with PCS health monitoring in order to generate a model of functionality. A number of high-level constraints related to semiconductor manufacturing were discovered during the bootstrapping activity described in the previous chapter. In addition to this a documentation analysis was carried out on the original PCS Health Report User Guide. Based on the constraints described by these activities a work domain analysis is used as the initial analytical approach.

6.3.1 Work Domain Analysis

In its original format an Abstraction Decomposition Space (ADS) models a work domain by revealing the relationship between its functional abstraction and its physical decomposition. To construct an ADS Vicente suggests defining the high-level functional purpose of the system, then the low-level physical form of its components and then populating the intermediate levels (Vicente, 1999). However as was identified earlier, this approach faces a methodological issue when it comes to embedded control systems (see section 2.3.1.2). While the fab has an obvious physical manufacturing process it can also be described from alternative perspectives. PCS health monitoring is more closely related to the engineering structures, that conceptually divide up the fab, than to its physical manufacturing process. At the same time, the relationships between the indicators in the PCS health report cannot be described through physical coupling. The various health indicators are generated using statistical models of normal behaviour rather than causal relationships described by natural laws. An initial analysis of means-ends relationships examines the engineering structure and the statistical control mechanism independently.

6.3.1.1 Structural Decomposition

As the purpose of the health report is to ensure that the machines in the facility are conforming to normal behavior, its structural decomposition should correlate to the engineering view. This hierarchy consists of the fab, several functional areas, a collection of modules, a number of toolsets and individual tools (see figure 5.4). Each tool has a number of sensors that record different parameters to provide the basic data used by this system. This model of the fab is based on industrial engineering specifications for semiconductor manufacturing that were identified during the bootstrapping phase and is shown in figure 6.3.a.

6.3.1.2 Control Hierarchy

An abstraction hierarchy is developed from the statistical control mechanisms in the current health monitoring application. The functionality of this system was derived from a detailed analysis of the user manual and two one-hour interviews with the original system developers. During these interviews and application walk-through protocol was used during these interviews to elicit knowledge about the system functionality. The functional purpose of the health report is to maintain the accuracy and stability of the process control system. At the *functional purpose* level, this is presented as a PCS *health value* for each module. At the physical form level, data is gathered through parameter sensors located in individual tools.

The health value is calculated from two sources, a control indicator and a validation indicator. These values are not displayed in the original interface. The *validation indicator* measures how many of the sensors in an area are functional i.e. recording data. The *control indicator* measures the stability of the PCS in terms of parametric variation. These sources are placed at the next level of *abstract function* in our hierarchy.

The control indicator is derived from a number of *sub-indicators* that use statistical methods to measure different types of variation in the sensor data. These include the *On-Target/matched Indicator (OTI)*, *Control Limit Variation (CLV)* and *percentage Out Of Control (%OOC)* readings shown as columns in the health report overview (fig 6.2 screen 2). The values displayed are the number of parameters in a module that pass or fail a specific test. For example, the first module in the health report overview has 13 successful and 2 failing parameters for the on target indicator (OTI). The sub-indicators form the *generalised function* level of the abstraction hierarchy. The individual sub-indicator charts show the parameter readings along with the individual sensor readings

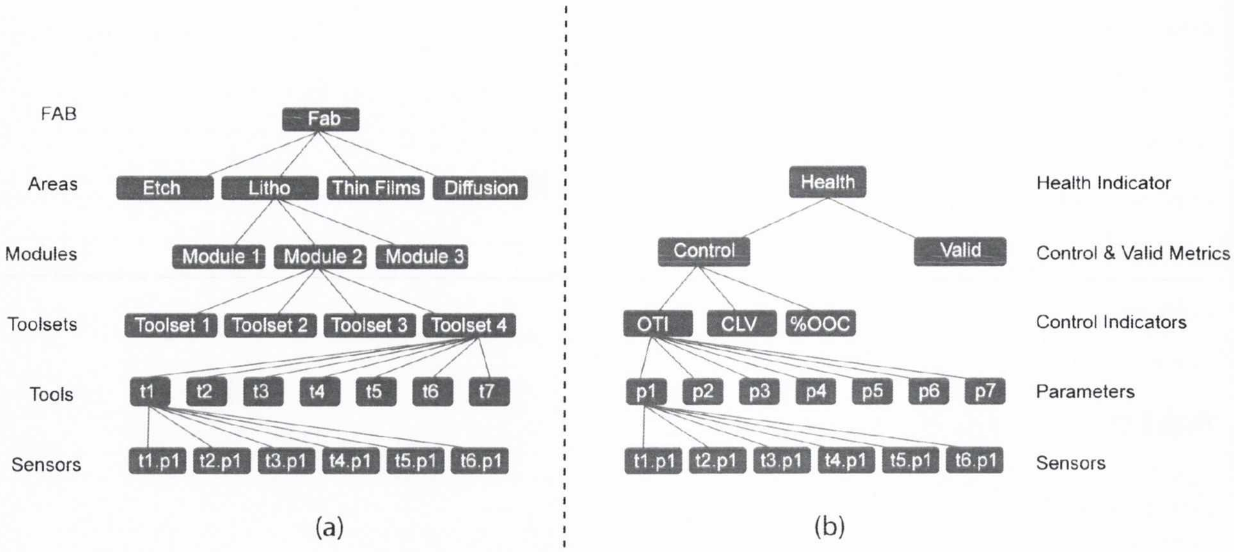


Figure 6.3: The two hierarchies associated with the PCS health report

that are used to calculate them. Figure 6.2 screen 3 presents the On Target Indicator chart, a control chart showing deviation between the tool sensors and the target for each parameter. The mean deviation across tools gives a parameter reading (labelled ALL in the chart key). As this *parameter reading* is used to generate the sub-indicator value, it is placed at the level of *physical function*. The indicator chart also presents the sensor reading for each tool. A sensor is described by the parameter it measures and the tool in which it is located. This *topological information* can be described as its *physical form* and is placed at the lowest level of the hierarchy. The control hierarchy is illustrated in figure 6.3.b.

6.3.1.3 Abstraction Decomposition Space

While these structures are presented independently in figure 6.3, they are related at a number of levels. The PCS health report needs to reconcile these structures by explicitly displaying their relationships in the interface design. To generate an Abstraction Decomposition Space (ADS) for the system, the control hierarchy is spread across the structural decomposition (figure 6.4). This defines different levels of abstraction at which health monitoring in the new system should occur. Figure 6.4 shows both the levels supported by the current system and the extended functionality that is required. Overall fab health is calculated from the mean health of its functional areas, which are in turn derived from

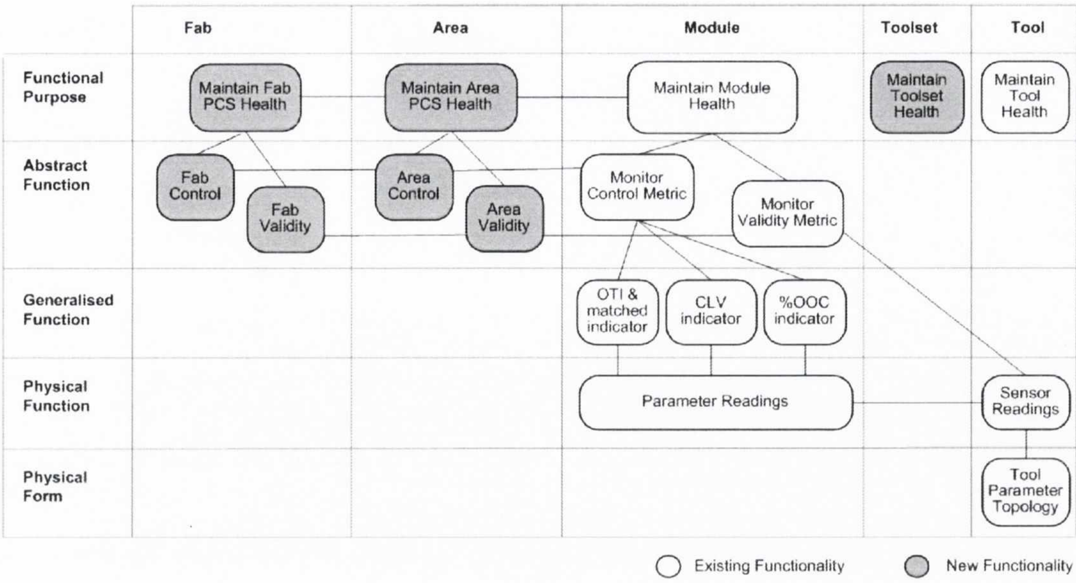


Figure 6.4: Abstraction Decomposition Space (ADS) for PCS health monitor

their modules. The health value for a module is calculated by combining its validity and control indicator values. This relationship makes it possible to provide higher-level metrics for validity and control at fab and area levels. The control indicator is based on its sub-indicators, which are, in turn based on parameter readings across a module. These parameter readings are generated from sensors located in individual tools. The response (or lack thereof) from individual sensors is used to generate the validity indicator at the abstract function level. The tool/parameter topology can be described as the physical form level of the work domain model.

6.3.1.4 Work Domain Model Validation

The structures depicted in figures 6.3 and 6.4 are initial sketches of proposed system functionality. They act as primary design artefact that can be presented to users. While there is evidence that users can operate using inaccurate or incomplete mental models, by describing the system in terms of constraints, structures and relationships, the work domain model provides an objective system model that can help to identify and address any misconceptions. One factor that becomes evident from the model is that *information is not currently provided at toolset level.* As this is part of the engineering structure, this omission may be responsible for its low acceptance by users. The ADS was reviewed with one senior and two junior process engineers in three separate one-hour sessions. The engineers were asked to trace through the relationships described by in the model in order to validate its accuracy.

Senior process engineers are responsible for processes across an entire functional area and tend to use the report to manage parameter targets. Their initial response to the ADS was that it provided an accurate description of the health report in relation to the physical model of the fab. However, junior process engineers identified toolsets as important regions that mark the boundary of responsibility for their role. They tend to manage a small number of toolsets (1-3) and only in very rare occasions would they be responsible for an entire module. As the level of toolset is not present in the existing health report they were manually generating their own graphs from raw data to support tasks relating to their own toolsets.

The ADS identifies how the system state is currently reported at various levels of abstraction. It provides information requirements in the form of quantitative information (e.g. health values) and qualitative information (e.g. hierarchical relationships). However the comments made by the junior process engineers indicated that the model does not describe the full range of tasks involved in monitoring the control system. To examine these in more detail a task analysis is required.

6.3.2 Task Analyses

The original health report came with a detailed user guide outlining procedures for interpreting and interacting with the different views and charts. This was used as the basis for an initial task analysis and was supported by interviews. Hierarchical Task Analysis (HTA) has been used in the past to supplement EID (Jamieson et al., 2007) and is also applied here.

6.3.2.1 Hierarchical Task Analysis

The health report user guide is presented in list format with accompanying illustrations of the interface and charts. A series of non-normative states are described and assigned appropriate response flow checklists. The main challenge in generating a HTA was to decouple the task descriptions from the original visual design. The descriptions referenced the current visual representations of the data thus fixing the interaction sequences and low-level actions. In order for the constraints to be understood independent of the representation, the descriptions were abstracted into a Goal-Task-Subtask format that removed all reference to interface elements. A portion of this HTA expanded for the on-target indicator chart is presented in figure 6.5.

The analysis begins at the health report overview (fig 6.2 screen 2) where the goal is to ensure that a module is running effectively. Plan 0 has three steps; locate the module, check its health values and where necessary, review the indicator summary values. At the next level, plan 3 shows that if any of the indicators are below a set target their corresponding charts should be accessed. The On-Target Indicator (OTI) chart (fig 6.2 screen 3) allows engineers to monitor parameters and ensure that they remain within control limits. The HTA outlines three major tasks that the engineer must carry out with the OTI chart (see table 6.1). Firstly, locate any sensor that lies outside of the control limits. This indicates abnormal process behaviour within a tool, and requires the parameter to be returned to an in-control state to avoid producing scrap (e.g. Param_11 in the OTI chart fig 6.2 screen 3). Secondly, detect unmatched parameters. If the sensor values for a specific parameter are widely spread across the tools, they are said to be unmatched. This indicates between-tool variability that causes major problems for multi-layer operations and can have an adverse effect on line yield. (e.g. Param_4 in the OTI chart). Thirdly, find off-target parameters that are matched. An entire set of tools may be off-target for a parameter. There are two probable causes for this. An incorrect parameter target may have been set or a change in the product may have a knock-on effect on the processing requirements. In either case, the target for the process parameter needs to be checked and adjusted (e.g. Param_8 in the OTI chart). Generating the HTA from the user guide reveals a number of event-based information requirements that were not captured by the work domain model including the control limits and targets.

6.3.2.2 Task Model Validation

The hierarchical task model provides an additional sketch of functionality and was again validated, this time with only two junior process engineers in two separate hour long sessions. Each participant stepped through the tasks and methods to check the accuracy

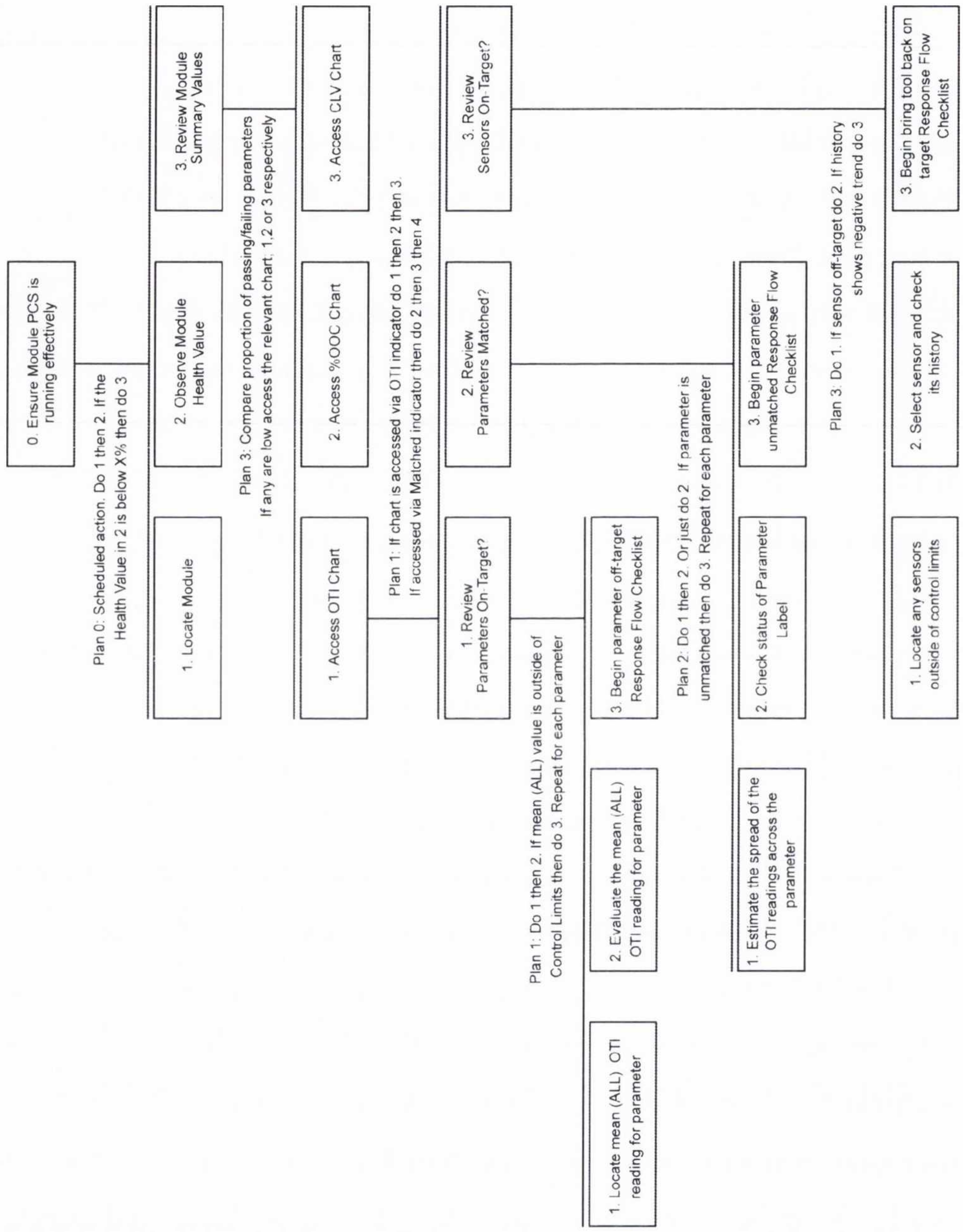


Figure 6.5: Hierarchical Task Analysis of PCS Monitoring

Task Code	Task	Action	Response
A.1	Find off-target sensor	Locate any sensor that lies outside of the control limits.	Examine sensor history Identify cause Request maintenance Return tool to an in-control state
A.2	Detect unmatched parameters	Identify parameters whose tools sensors are widely spread.	Return tools to an in-control state Starting with off-target tools (As above)
A.3	Find off-target parameters that are matched.	Identify parameters whose sensors are tightly packed but whose mean value is outside of control limits.	Examine parameter target setting Correct if necessary Examine sensor histories Identify possible cause for shift Request maintenance

Table 6.1: PCS Health Monitoring Core Tasks

of the model. Their initial response was that the model captured all of the activities required to carry out PCS health monitoring. This was surprising as it made no mention of toolsets or the validity metrics that were featured in the work domain model. During interviews, it became apparent that users meant that the HTA accurately described all of the activities that were effectively supported by the *current* health report. Activities such as the custom generation of toolset graphs were not possible with the current system and were therefore not associated with the core PCS health monitoring tasks. In fact these activities can be described as *workarounds* (Koopman and Hoffman, 2003) that were developed by the engineers to cope with missing functionality. This provides evidence that the user's mental models of their work have been formed to some degree by the information systems they use.

During this reviewing process a number of additional activities that form part of the process engineer's workload were specified (Table 6.2). The first activity relates to inspecting specific sensors. A process engineer may wish to observe the performance of a specific sensor based on information sources outside of the report e.g. phone call

Task Code	Task	Action	Response
B.1	Locate a specific sensor	Locate a sensor by its tool & parameter reference	Examine it's reading as in A.1
B.2	Identify erratic performance on a tool	Identify and compare each parameter reading in a tool	Identify patterns in sensor performance. These can act as a fault signature.
B.3	Check whether sensor is working	Check which sensors are missing	Request maintenance
B.4	Check toolset health	Observe sensors by toolset and parameter	Examine readings as in A.1

Table 6.2: PCS Health Monitoring

from equipment engineer, tacit knowledge of past history etc. The second relates to fault diagnosis. Being able to compare multiple parameters on an erratic tool can reveal patterns, known as fault signatures, which can aid diagnosis. The third deals with the validity indicator. A general indicator value is provided at the module level, but it is important to be able to identify which specific sensors are non-operational. Finally, a junior process engineer may wish to understand whether a problem with a particular parameter is caused by their toolset. Currently this is achieved by producing custom, toolset-specific charts.

6.3.2.3 Control Task Analysis

What all of these tasks have in common is that they take a structural, engineering-focussed view on the system. They look for measures associated with sensors, tools and toolsets, all of which are related to the structural hierarchy of the system. This is very different from the original system where transitions between levels were made through indicators and parameters, elements of the control hierarchy. In order to see how these tasks are supported by the current design a Control Task Analysis (CTA) was carried out (Vicente, 1999). The decision ladder provides a model that reveals the level of information abstraction required to support specific tasks. CTA provides different information from the hierarchical task analysis as it is based on control tasks carried out to achieve the functional purpose of the system rather than procedures used by individual operators.

This can be used to identify situations where cognitive leaps may be made between a state of awareness and an action. This occurs where an expert operator recognises a pattern in the system state and can respond directly without referring to higher level system goals.

Figure 6.6 shows the decision ladders relating to tasks involving parameter inspection (A.2 and A.3) and tool inspection (B.2). In both cases a set of observations had to be integrated by the operator to answer the question posed by the task. This cognitive activity is required as the current design (Fig 6.2 screen 3) does not provide perceptual cues that support these tasks. Rather than providing direct indicators of performance at toolset or tool level, the operator must make multiple observations, and mentally integrate the data to reveal the system state. If the values were arranged to express the structural relationship as well as the control hierarchy these questions could be answered at the *observe* stage rather than at the *identify* stage. These decision ladders provide an additional set of models of system functionality. These decision ladders provide an additional set of models of system functionality. A half hour review session was carried out with one of the junior process engineers who traced through the information processing and system state steps. During this review a number of additional problems with the existing designs were revealed.

1. *Locating Parameter Value.* Users had difficulty locating the parameter reading (the "ALL" icon in fig 6.2 screen 3) as it was encoded in the same manner as the sensors and it generally lay at the centre of a cluster of icons.

2. *Selecting sensor icons.* Users found it difficult to click on the small sensor icons when accessing performance history.

3. *Occlusion of icons.* Icons with the same or similar OTI values tend to overlap making it difficult to observe and click on any icon other than the foremost. This can result in incorrect selections.

4. *Ability to locate a specific sensor.* The current design had been developed to highlight the sensor values; however, the visual encoding makes it difficult to locate sensors based on tool I.D.

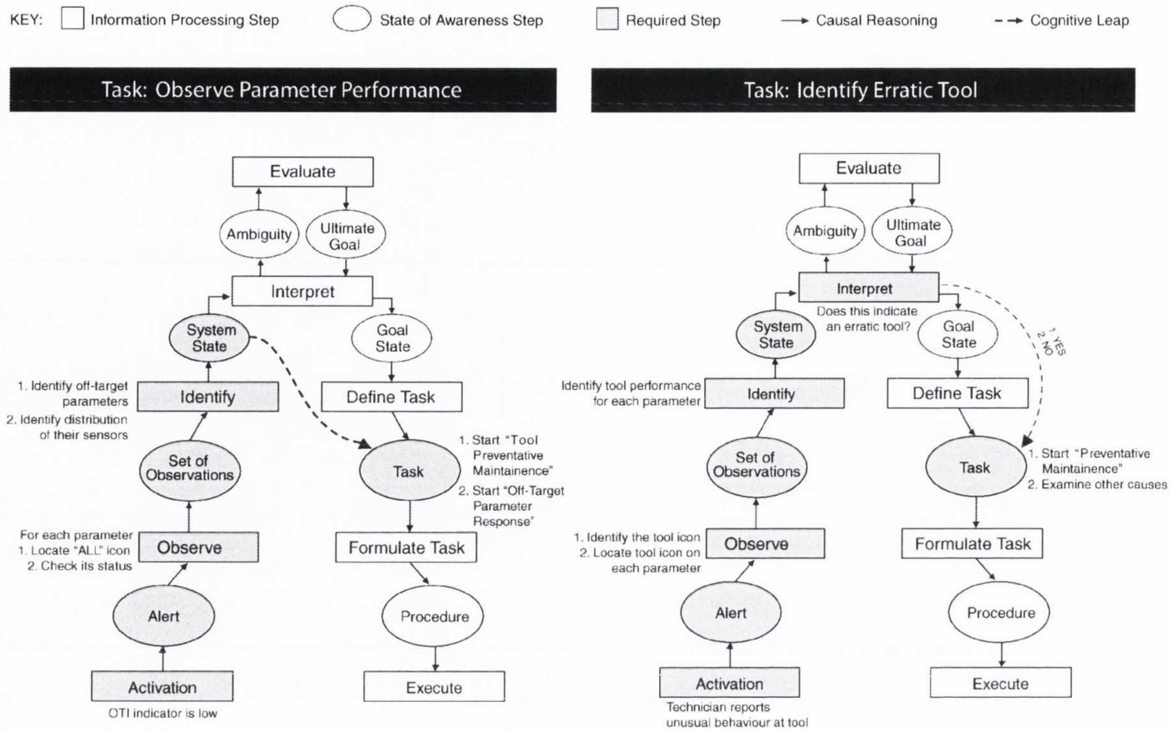


Figure 6.6: Decision Ladders for two monitoring tasks

5. *Ability to view all parameters at once.* In the original chart, the parameters are widely spaced in order to be read as units. In situations involving a large number of parameters horizontal scrolling is required to view the state of a module.

6.3.3 Design Problem-Structuring Review

A number of design artefacts were produced during the problem-structuring phase. Each of these can be described as analytical models or concepts describing the functionality of the PCS health report.

The work domain analysis revealed how this project requires an adjusted ADS, where abstraction moves between structural abstraction at higher levels and functional abstraction at lower-levels. This adjustment was necessary as the structural hierarchy provides a familiar model of the physical relationships in the work domain, while the functional abstraction provides an external model of the less familiar control system relationships. While the resulting model identifies a number of information requirements it does not reveal how operators use this information when carrying out control tasks.

A hierarchical task analysis produced a task model that revealed the core tasks supported by the current health report. The validation of this model demonstrated the advantage of analysing activity *after* the work domain, as the ADS identified structural levels that were not expressed in the current system. This motivated a more detailed analysis of activity.

A control task analysis identified specific issues with the current design that impeded performance for certain tasks. The manner of displaying information requires the operator to carry out a number of cognitive operations to identify the system state. By integrating both the structural and control hierarchies in the visual display, an interface could provide perceptual cues that support the full range of tasks.

6.4 Design Problem Stating

Different techniques were used to analyse the health reporting system and these produced a number of different models. An Information Requirements (IR) matrix can be used to identify how different analytical approaches can contribute to requirements gathering. Previously this has been used to show how different analytical techniques can compliment one another (Jamieson et al., 2007). Here an extended IR Matrix is generated in order to compile the various analytical outputs into a single transitional artifact. This allows the various qualities that need to be expressed in the interface to be communicated.

6.4.1 Extended IR Matrix

The information requirements are described using four main categories across 10 columns (table 6.3).

6.4.1.1 Abstraction Hierarchies

The first three columns are used to indicate the position of each requirement in the context of the work domain and the goals of health monitoring. The divergence between the abstraction of the control system and engineering model of the fab makes it difficult to generate a simple hierarchical relationship. The five levels of Rasmussen's original abstraction hierarchy are presented in the first column. In the second column the different structures of the control hierarchy are aligned to these. In the third column the

Abstraction Hierarchies			Information Requirement			Interaction Requirement			Notes
Functional Hierarchy	Control Hierarchy	Structural Hierarchy	Display Element	Description	Data scale	Action	Navigation	No. of elements	Notes
Functional Purpose	Health	Fab	**Fab Health**	0 - 100%	Q	Link to fab trend	Expand structure	1	Observe Performance
			Fab Health History	Trend Chart	Qt		Expand structure	1	Identify Performance Change
		Area	**Area Health**	0 - 100%	Q	Link to area trend	Expand structure	4	Observe Performance
			Area Health History	Trend Chart	Qt		Expand structure	4	Identify Performance Change
		Module	**Module Health**	0 - 100%	Q	Link to module trend	Hi-lite Module AF	<10	Observe Performance
			Module Health History	Trend Chart	Qt		Hi-lite Module AF	<10	Identify Performance Change
Abstract Function	Valid indicator	Fab	**Fab Valid**	0 - 100%	Q	Link to trend	Expand structure	1	Identify Manual Faults
			Fab Valid History	Trend Chart	Qt		Expand structure	1	Identify Performance Change
		Area	**Area Valid**	0 - 100%	Q	Link to trend	Expand structure	4	Identify Manual Faults
			Area Valid History	Trend Chart	Qt		Expand structure	4	Identify Performance Change
		Module	**Module Valid**	0 - 100%	Q	Link to trend	none	<10	Identify Manual Faults
			Module Valid History	Trend Chart	Qt	Link to OTI Chart	Open OTI	<10	Identify Performance Change
	Control Indictor	Fab	**Fab Control**	0 - 100%	Q	Link to trend	Expand structure	1	Identify control issue
			Fab Control History	Trend Chart	Qt		Expand structure	1	Identify Performance Change
		Area	**Area Control**	0 - 100%	Q	Link to trend	Expand structure	4	Identify control issue
			Area Control History	Trend Chart	Qt		Expand structure	4	Identify Performance Change
		Module	**Module Control**	0 - 100%	Q	Link to trend	Hi-lite Module GF	<10	Identify control issue
			Module Control History	Trend Chart	Qt		Hi-lite Module GF	<10	Identify Performance Change
Generalised Function	OTI	Module	**Module OTI**	No. Pass/No. Fail	N & Q	Link to OTI Chart	Open OTI	1	Specify control problem
	Matched		**Module Matched**	No. Pass/No. Fail	N & Q	Link to OTI Chart	Open OTI	1	Specify control problem
	OOC		**Module OOC**	No. Pass/No. Fail	N & Q	Link to OOC Chart	Open OOC	1	Specify control problem
	CLV		**Module CLV**	No. Pass/No. Fail	N & Q	Link to CLV Chart	Open CLV	1	Specify control problem
Physical Function	Parameter	Module	**Parameter Readings**	"+/- 5 Std Dev"	Q	Link to trend	Open Trend	<30	Compared against limits. Must be salient
			Parameter Label	e.g. P_3	N	none	none	<30	All labels should be visible without scrolling
			Parameter Matched	Yes/No	N	none	none	<30	Highlight unmatched parameters
			Control Limits	"+/- 3 Std Dev"	Q	none	none	2	benchmark for sensor values
			Target	0	Q	none	none	1	benchmark for sensor values
		Toolset	**Toolset label**	e.g. ts_12	N	highlight TS sensors	none	<10	to associate related tools
Physical Form	Sensor	Tool	**Sensor Readings**	"+/- 5 Std Dev"	Q	Link to trend	none	params*tools	Compared against limits. Must be salient
			Tool Label	e.g. T_4	N	none	none	<40	Associate sensor with tool
			Sensor Reading History	Trend Chart	Qt	none	open Trend	1	should be easy to access
			Structural Hierarchy	Hierarchy of labels	N & O	Structural Navigation	n/a		Must support Navigation
			Control Hierarchy	Hierarchy of labels	N & O	Functional Navigation	n/a		Must support Diagnosis

Table 6.3: The IR matrix for the PCS health report

relationships between these control values and the structural hierarchy that describes the fab are identified.

6.4.1.2 Information Requirement

The next three columns describe the information requirement itself. Column 4 specifies the name of the display element, column 5 provides a description of the information involved and column 6 assigns the information to a data scale. As discussed in chapter 4, many ecological displays rely on mimic display components for representing information requirements. However, this PCS information relates to data values within a control system and does not benefit from having existing display components. The data scale is provided to inform the visual design of a system image during the design problem-solving phase.

6.4.1.3 Interaction Requirement

The next three columns describe the interaction requirements of the display elements. These are necessary to cope with the additional representational constraints imposed by the scale of the system. The application aims to provide a fab-wide synopsis of PCS performance. This involves a large number of modules, hundreds of toolsets and thousands of parameters. The structural hierarchy should support navigation through the system allowing information to be provided on demand. This requires certain information requirements to play the dual role of representation and navigation. Column 7 describes the action supported by each display element while column 8 describes navigation. Another unique factor of this system is that many of the display elements are not individual graphic objects but are classes of objects that make up the visual display. For example 'area health' describes a category that will appear four times in this example providing information about the etching, lithography, thin films and diffusion functional areas. Similarly, different modules will contain different number of parameters. Column 9 provides information about the number of instances that occur for each display element.

6.4.1.4 Notes

The detailed task analyses identified specific cognitive tasks that are carried out with the information requirements. The task model review also identified a number of situations where these tasks are difficult to complete. Column 10 provides notes that summarise these issues. These will be used during design problem solving to support the visual design process.

6.5 Design Problem Solving

Now that the information requirements have been identified and structured the cognitive engineering process moves into the design problem-solving phase. An ADS was generated during analysis and this allows the EID principles to be applied. However a number of unique characteristics associated with this system create representational challenges that must be overcome.

6.5.1 Challenges to EID Visual Design Principles

EID provides three visual design principles that were equated to three phases of the design problem solving process (section 4.2.4). In terms of the *preliminary concept* the associated principle advises representing the work domain in the form of an abstraction hierarchy to serve as an externalised system model. The Duress exemplar presented in section 3.2 used the proximity of mimic symbols as the primary method of encoding different levels of the abstraction hierarchy (Bisantz and Vicente, 1994). The health report system cannot use this approach as its sub-systems (i.e. indicators, parameters, sensors etc.) are not available in a mimic display. This lack of an existing graphic structure means that the abstraction hierarchy must be visually designed from first principles.

In terms of the *refined concept*, the associated principle recommends providing a consistent one-to-one mapping between system constraints and the cues or signs provided by the interface. With physical/material process flows, the work domain model reveals constraints that define and dictate operator behaviour. The difficulty with the health monitor system is that the work domain model cannot identify the full range of system constraints. As demonstrated in the problem-structuring phase, additional task analyses were required to understand the bahaviour associated with health monitoring. The goal structures and tasks identified by this, place additional constraints on the work and these must also be supported by the display.

In terms of *detailed concept,* the associated principle states that the representation should be isomorphic to the part-whole structure of movements. With physical engineering systems, this principle is achieved through the arrangement of mimic symbols in a manner that reflects the structural relationships of their real world counterparts. However, the 'components' of our system do not come with pre-existing symbols. They are control values from which graphic representations must be generated. The design of these elements is constrained by both the sheer volume of values involved and the actions that are carried out with them.

In the absence of existing mimic displays the visual design principles have limited utility. More explicit guidelines are required to support the design of visual representations of system information. The system involves both complex relationships and enormous volumes of data. These characteristics point towards information visualization as a potential source of design guidance.

6.5.2 The Visualisation Reference Model

Information Visualization (IV) has been defined as *the communication of abstract data through the use of interactive visual interfaces* (Keim et al., 2006). It is a research domain that combines themes and methods from scientific visualization, information graphics and exploratory data analysis. While the majority of research in IV focuses on the technical aspects of generating visual representations of large or complex data sets, some work has been carried out on identifying information visualization design methodologies. Card, Mackinlay and Shneiderman define the visualisation design process as generating *"adjustable mappings from data, to visual form, to the human perceiver"* and provide the visualisation reference model to illustrate how this occurs (Card et al., 1999). It shows that raw data can be compiled into data tables before being converted into visual abstraction and presented as views on a dataset. Data scale transformations can be applied

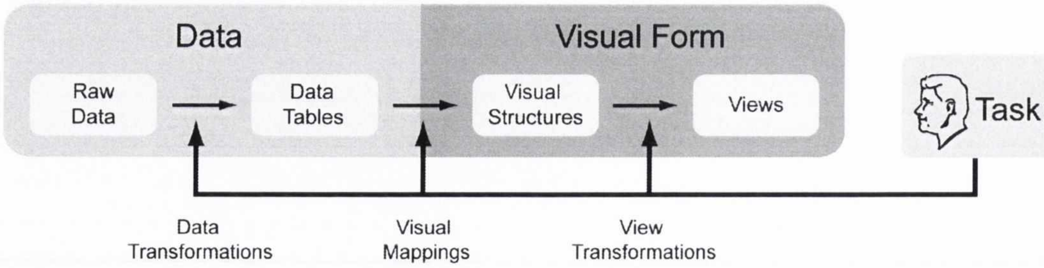


Figure 6.7: The Visualization Reference Model. after (Card et al., 1999)

to match data to specific tasks, alternative visual mappings can be used to match data to the appropriate visual variables and view transformations can be applied to modify these visual abstractions into different views. A user can control these modifications in order to gain a better understanding of the data (figure 6.7).

The reference model relies on three important concepts. Firstly, data can be categorised into perceptual scales, in accordance with the theory of scales of measurement (Stevens, 1946). This concept was discussed earlier in chapter 3. Secondly, visual scale matching as referred to in Bertin's "rules of graphic systems" (Bertin, 1983). This relates to the conversion of data values into graphic forms and was also discussed in chapter 3. The third concept involves the transformation of data scales to match a specific task. Data can be transformed from a higher perceptual scale to a lower perceptual scale, either mathematically or through its visual encoding, in order to suit specific cognitive tasks. For example, quantitative comparison of two datum can be achieved by encoding with position. This supports accurate judgment of magnitude i.e. A is twice B. Alternatively, if the task involves ordinal judgement, the same data encoded using brightness will simply show A as greater than B. If the comparison is a cue for action (e.g. if A is greater than B then abort the procedure), then the relationship can be encoded using a single nominal visual variable. Colour hue is often used for this i.e. green to go or red to stop. While the progression from quantitative comparison to nominal comparison is reductive in terms of data availability, it is more focused in terms of task support (Petersen and May, 2006). This ability to guide the design of visual forms based on cognitive tasks makes the reference model useful for the design of control interfaces.

The analysis phase of the EID framework outputs a set of information requirements, which can be identified as data sources and relationships. The visualisation reference model uses raw data as its starting point and provides a set of explicit guidelines for converting data into visual form. As these are essentially two sides of the design process gap, this suggest that the two approaches can be combined to reduce this gap and provide a more concrete design process. In the following sections the three phases of design problem solving are described, indicating how the visualization reference model can inform and control conceptualization in each phase.

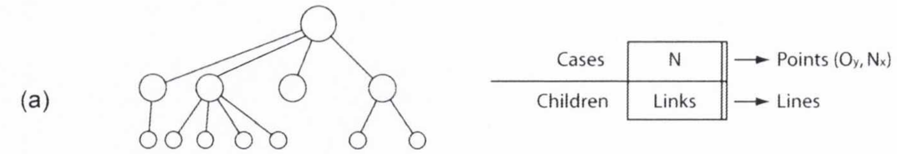
6.5.3 Preliminary Concept: Visual Hierarchies

The preliminary design concepts are developed around the design goal of *representing the system in the form of an abstraction hierarchy*. The three visualization activities of data scale analysis, visual scale matching, and scale transformation are applied to generate design concepts.

6.5.3.1 Data Scales

A hierarchy can be defined as a series of ordered groupings of elements within a system. Based on this definition, the data scales involved in any hierarchy are, an ordinal relationship between different levels, a nominal relationship within levels and a nominal relationship between parent and child elements (see figure 6.8). Tree structures are commonly used as conceptual models of hierarchies. A tree structure is composed of nodes, connections and leaves. Nodes are organizational structures that can contain other nodes or leaves. Connections indicate the relationship between nodes. Leaves are low-level data that cannot be subdivided. Information visualization has predominantly used two alternative graphical representations of hierarchies, connection and enclosure (figure 6.8)(Card et al., 1999).

Connection uses the most literal visual representation of the tree structure. Nodes are represented by shapes, the ordinal relationship between levels is encoded using position on one spatial axis, the nominal relationship within levels by position on the perpendicular axis and the nominal parent-child relationship using connecting lines.



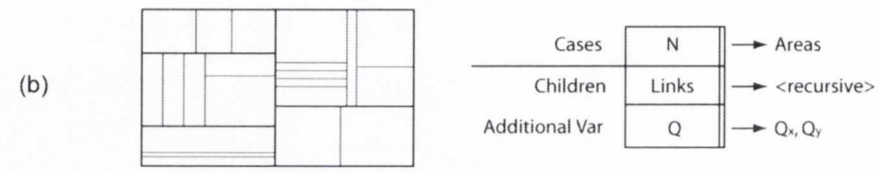


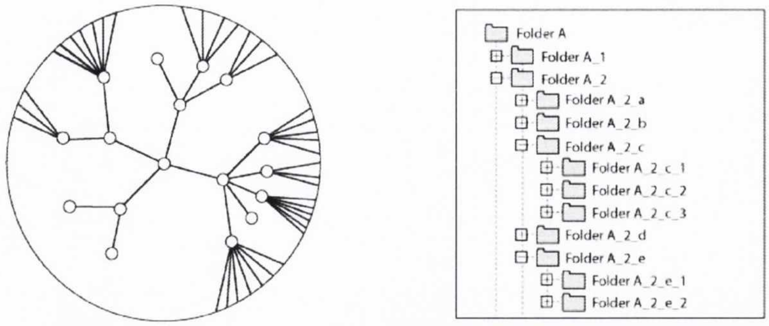
Figure 6.8: Visual hierarchies (a) connection and (b) enclosure. after (Card et al., 1999)

Enclosure uses an alternative encoding where nodes are represented using areas. The ordinal relationship between levels, the nominal association between parent and child and the nominal relationship between nodes on the same level are all encoded by using area to indicate enclosure.

While these two approaches describe the most common forms of hierarchical representation, it is possible to generate alternative formats using representations that combine ordinal and nominal visual variables (Waloszek, 2004). The key challenge in this case is to develop a representation that can communicate the relationship between the structural and control hierarchies that describe system functionality.

6.5.3.2 Visual Scale Matching

Connection and enclosure provide visual representations of hierarchies but they have different advantages and limitations. The connection format is excellent for revealing the structure of very large hierarchies. It has been modified a number of times to act as a navigational component. Systems like the hyperbolic browser (Lamping et al., 1995) and Windows Explorer^{TM} combine connection with interaction to allow user to move around large hierarchies accessing detailed information on demand (figure 6.9). One disadvantage of this format is that the visualization is restricted to displaying structure. While information on specific nodes can be accessed, it is difficult to make comparisons across the hierarchy. The enclosure format is better at supporting this as it can embed values



a. b.

Figure 6.9: Sketches of (a) hyperbolic browser (b) windows explorer^{TM}

associated with nodes directly into the image. Treemaps use the enclosure method to communicate quantitative relationships between values. Applications like SmartMoney's "Map of the Market" (Wattenberg, 1999) have extended the encoding power of treemaps to embed up to 3 additional variables associated with nodes (fig. 6.10). This representation uses the market share values of companies to generate a snapshot of the stockmarket and encodes price fluctuations to show current performance. Despite their advantages for comparitive tasks, the use of recursive areas in treemaps limits their application to relatively shallow hierarchies as deeper nodes become progressively smaller until they are no longer visible.

A number of conceptual sketches were developed based on these graphical encodings of hierarchical structures. At its higher levels *the structural/engineering hierarchy* is used to observe health performance across the fab and to navigate to problem areas. This would suggest that a treemap approach would be useful here. By using the inverse health value, areas and modules with low health would become highly salient features in the display (fig. 6.11sketch 1). While this would highlight poorly performing areas, the arbitrary shapes generated by the treemap's space-filling recursive algorithm, coupled by the small display sizes at the lowest levels of granularity would make navigation difficult. An expanding tree similar to that used by windows explorer provides an intuitive means for navigating through a hierarchy. By presenting the health value alongside the structural labels, both navigation and comparison of health values are supported.

The *functional/control hierarchy* is somewhat different. The work domain analysis showed that this primarily extends from module health down through a number of

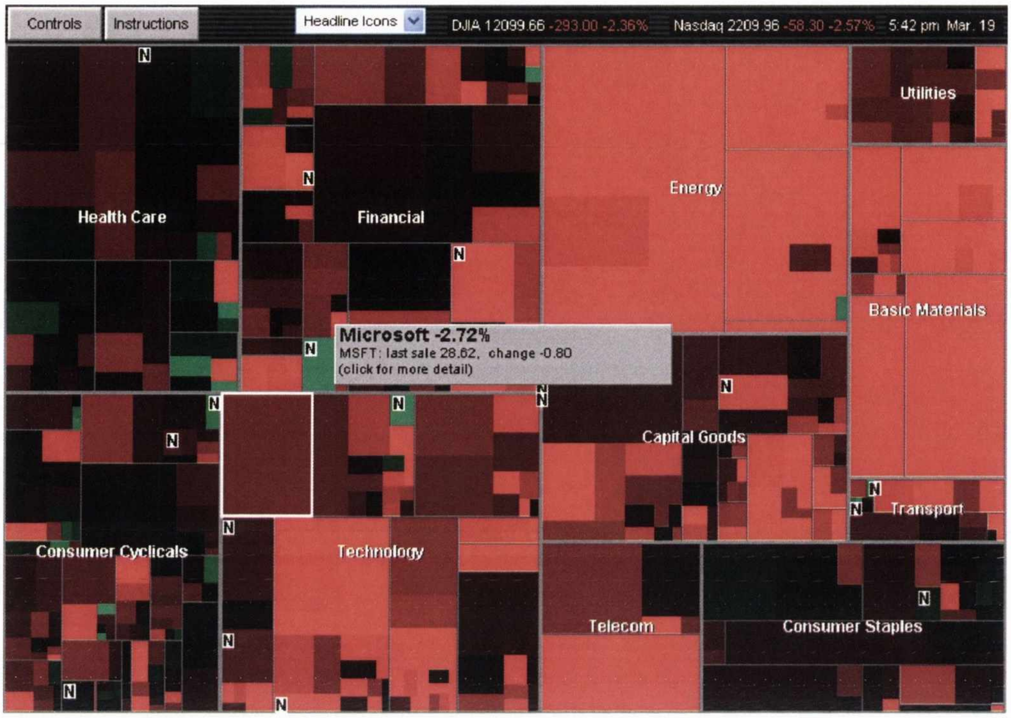


Figure 6.10: Map of the market. (Wattenberg, 1999)

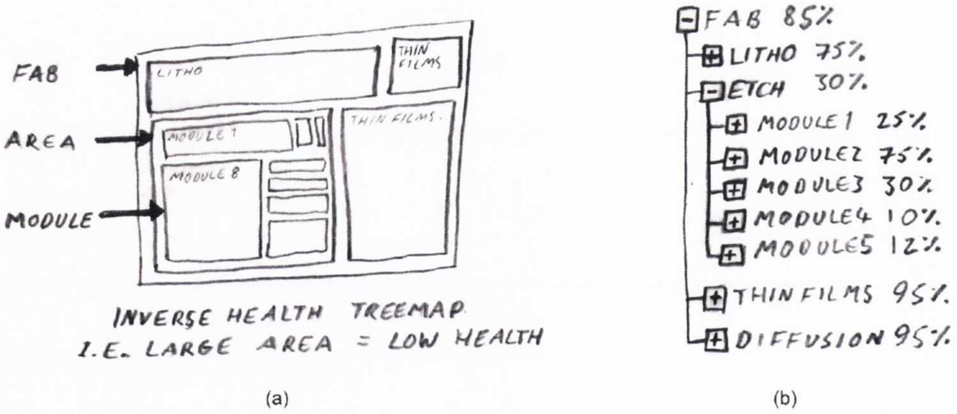
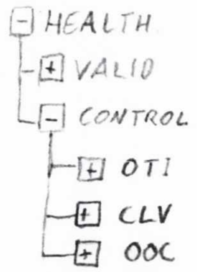
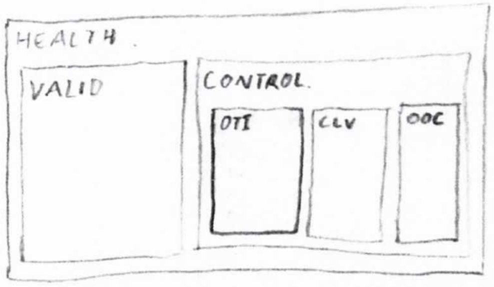


Figure 6.11: Sketches of the structural/engineering hierarchy



(a)



(b)

Figure 6.12: Sketches of the functional/control hierarchy

functional levels down that of individual sensor readings (see fig. 6.3.b). The original application does not explicitly represent these levels. While the health reading is stated, there is no expression of how it is derived from the control indicators. In order to make the system more observable it is necessary to show these relationships. The sketch in figure 6.12 demonstrates how the either the connection or the enclosure technique can be used to make the relationship between lower-level data and higher-level information explicit.

The difficulty is that both the structural and control hierarchies need to be combined in order to generate a visual form that matches the work domain model.

6.5.3.3 Scale Transformation

In figure 6.12b the ordinal relationship between levels is encoded using enclosure. If there was some way of flattening this structure it could be integrated with that of the structural hierarchy. The sketch in figure 6.13 shows how this can be achieved. By encoding the ordinal levels using the tonal visual variable the values can be presented alongside one another. The nominal parent-child relationship is encoded using hue that is activated on roll-over. This flattened hierarchy can be integrated with the structural hierarchy to form an expanding matrix that can be read vertically or horizontally.

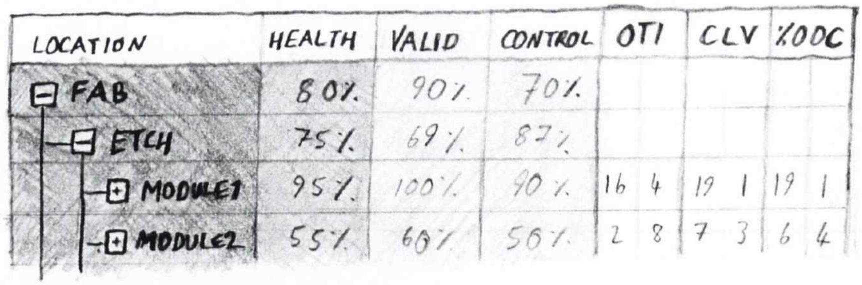


Figure 6.13: An expanding matrix using numeric encoding.

6.5.3.4 Graphic Form

The resulting display has several advantages over the original overview. It supports the vertical comparison of high-level health values while at the same time it reveals the relationship between health values and their associated indicators on the horizontal axis. The display is spatially efficient as the structural drill down from fab level provides details only when requested. This provides sufficient space for displaying charts in the lower half of the screen. This area can be used to present either the indicator charts or trend charts of the higher-level metrics. The format also supports interaction whereby clicking on a value changes the information presented in the chart area to correspond to the selected value i.e. clicking on current module health will provide a view of module health history or clicking on OTI reading will provide the OTI chart. Despite these advantages the values are still represented numerically rather than visually and this makes it difficult to get a quick overview of performance. This limitation will be dealt with in the refined design phase.

6.5.4 Refined Concept: Mapping Constraints and Cues

The refined design concepts are developed around the design principle of mapping constraints to visual cues. As was mentioned earlier, the goals and tasks associated with controlling the system will be used to inform the visual encoding process.

6.5.4.1 Data Scales

The health value is presented as a percentage and is derived from a combination of validity and control indicators. The validity indicator represents how many sensors are functional i.e. returning information, while the control indicator shows how well the PCS is performing in relation to a number of statistical control models. While these values are not shown in the original overview they have been included in the information requirements of the new system. Both of these values are available as percentages. The individual control indicators are presented in the original report using two columns that provide counts of how many parameters are passing or failing a particular test (see fig. 6.2 screen 2).

6.5.4.2 Visual Scale Matching

As the health, valid and control indicators are all available as percentages these values can be assigned to a visual variable that supports quantitative perception. By encoding each of these using the length of a bar, the comparison of values across structural levels is transformed from a cognitive task to a perceptual one. For example a user can compare the health readings of modules by vertically scanning the health column. This approach works at the higher levels of abstraction, however information for the three low-level indicators (OTI, CLV & OOC) is not reported as a single percentage but as a count of the parameters that pass or fail each control test. Different modules have different numbers of parameters and this presentation format reflects this. An initial sketch was developed that used individual strokes to represent each parameter in the system (fig. 6.14). This encoding makes a visual estimation of indicator performance feasible. However, diagnosis of problems involves locating a poorly performing module and then locating the test that causes the poor control reading. In order to make the relationship between the low-level and high-level control indicators more explicit, a consistent visual encoding should be used.

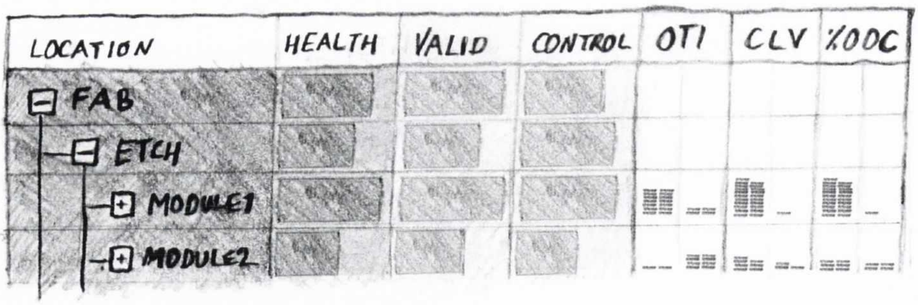


Figure 6.14: Expanding matrix using visual encoding

6.5.4.3 Data Transformation

On reviewing the hierarchical task analysis it was shown that an engineer prioritises which module and chart to investigate based on *the number of failing parameters relative to the overall number of parameters* (fig. 6.5 level 2 plan 3). As this is a proportional comparison, it is possible to normalize these figures by transforming them into percentages. Now that all of the required values are represented on a common quantitative scale, it is possible to display each value using a vertical bar in the same manner as the health reading as shown in figure 6.15. Detailed numeric information is available on rollover.

6.5.4.4 Graphic Form

The resulting graphic form has translated the cognitive tasks associated with health monitoring into perceptual tasks. The matrix presentation combined with a common visual encoding allows a user to read associations on both vertical and horizontal axes, in accordance with the proximity compatibility principle. For example during top down diagnosis, a user can compare the performance of the functional areas by vertically scanning their health values. If one is low, clicking on its label will expand the matrix to show all of its constituent modules (see figure 6.15). This process can be repeated to find the module that is causing the low reading. A horizontal scan across different indicators for a module can inform the engineer on the nature of the problem and which sub-indicator chart to access to carry the diagnosis down to the next level. The relationship between indicator levels is provided through contextual highlighting while the relationship between indicators and charts is available through contextual linking. However, the design so far has focused on the higher-level indicators. In the detailed design phase we deal with the representation of the lower level components.

6.5.5 Detailed Design: Representing Part-Whole Structure of Movements

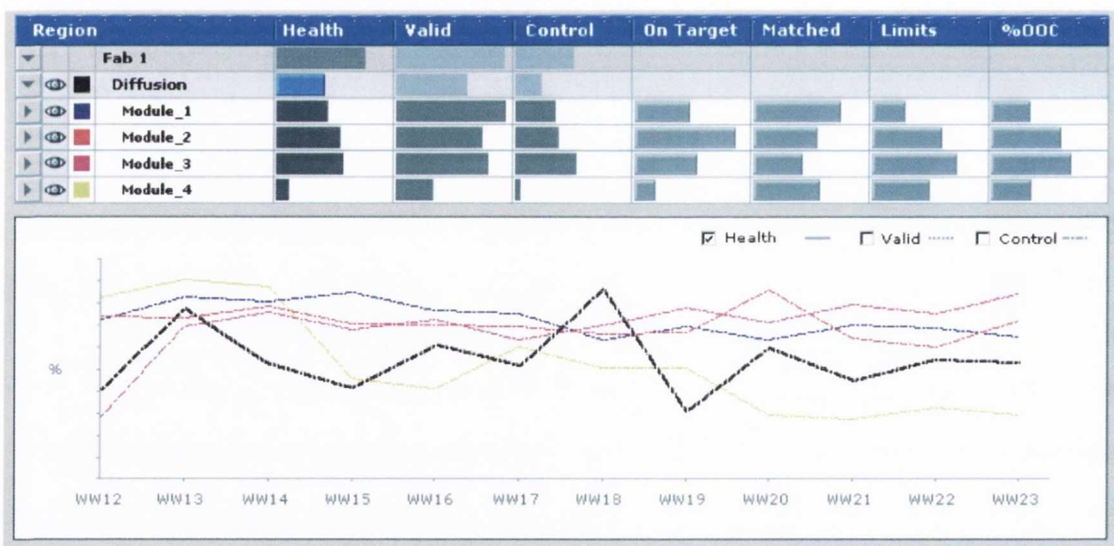
The preliminary design concepts are developed around the design goal of representing the system and an abstraction hierarchy but the expanding matrix described above only displays information down to the generalised function level associated with modules. Below this level indicator charts have been used to communicate information at the levels of physical function and physical form. A number of interaction problems with the on-target indicator chart were highlighted in the analysis phase (see table 6.2). Here, the visualisation reference model is applied again to generate a new graphic form that improve interaction strategies for a range of different tasks.

6.5.5.1 Data Scale Analysis

The information requirements matrix identifies all of the variables associated with the physical function and physical form levels (table 6.3). These variables exist on both the nominal scales (parameter/toolset/tool labels and matched status) and the quantitative scale (parameter and sensor values, target and control limits). The original OTI chart actually displays all of this data already with the notable exception of the toolsets (fig. 6.2 screen 3). In fact, the current graphic form actually uses valid visual scale matching for the data scales revealed in the IR matrix. However the current design is only one of a number of possible solutions. Peebles and Cheng note that multiple visual representations can be generated of the same data set depending on how matching is carried out (Peebles and Cheng, 2003). In order to generate the most appropriate solution their *graphic reasoning theory* recommends that "designers should (a) consider how different quantities are encoded within any chosen representational format, (b) consider the full range of alternative varieties of a given task, and (c) balance the cost of familiarization with the computational advantages of less familiar representations. In order to achieve this the data scales are matched to different visual variables to generate *a design space* of possible

Perceptual Task: Vertical scan of area health values shows diffusion to be low

Physical Action: Click on diffusion label for more information



Perceptual Task: Horizontal scan of indicators shows control to be low. The OTI indicator for module 4 appears to be cause

Physical Action: Click on module 4 OTI indicator to view OTI Chart

Figure 6.15: The refined design concept

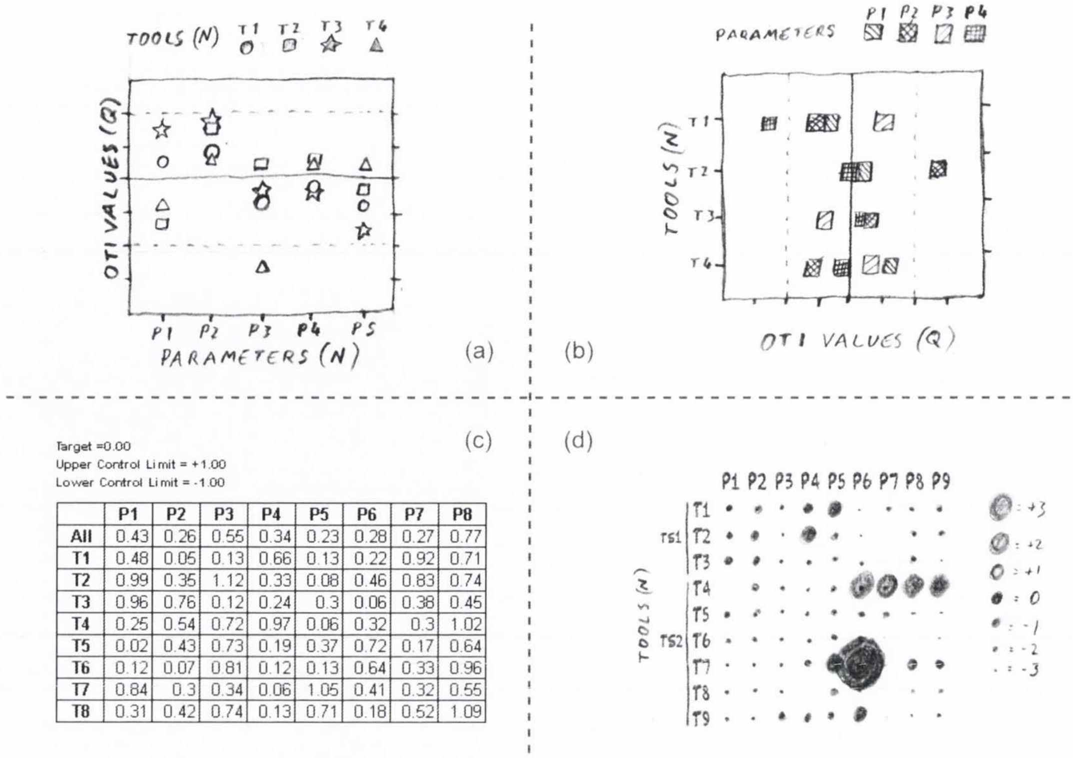
alternative solutions.

6.5.5.2 Visual Scale Matching

Visual scale matching is used to generate a number of conceptual sketches that make up the design space. Figure 6.16a is a sketch of the original encoding shown in figure 6.2 screen 3. The quantitative readings, control limits and target are encoded on the vertical axis. The sensor/tool/parameter topology is encoded by way of an icon. This icon sits at the junction of three dimensions; the *quantitative reading* encoded through position on the vertical axis, the *nominative parameter* that the sensor is recording is encoded through position on the horizontal axis and the *nominative tool* on which the sensor exists is encoded through shape. This design uses direct scale matching, but as the parameter values are encoded on the perceptually powerful spatial axis, the result is a parameter–centric view of the system. It is easy to identify that parameter 3 (P3) has one off-target tool while parameter 4 (P4) has all of its tools on-target. However, it is more difficult to derive whether tool 3 (T3) is on target for all parameters. While shape is effective in encoding a nominal variable, it does not afford selective perception (see section 3.3.1) and therefore the readings of parameter values across tools is not well supported.

An alternative sketch is shown in figure 6.16b. Here the quantitative values have been displayed on the horizontal axis, nominative tools have been encoded on the vertical axis and the nominative parameters are encoded through texture. While it is now possible to see how well an individual tool is performing for all its sensors, it is difficult to focus on parameter performance as this display reverses the perspective encountered in the previous sketch

An alphanumeric/spreadsheet style presentation may seem counter intuitive as a visualisation but in fact it makes the relationship between the nominal dimensions very explicit by creating a tool/parameter matrix. The major limitation of this display is that the data is difficult to read. Each figure must be read and independently calculated to see whether it lies within the control limits. Figure 6.16c shows a reduced example of the indicator chart data with four points lying outside of control limits, clearly demonstrating that such points are difficult to locate.



	P1	P2	P3	P4	P5	P6	P7	P8
All	0.43	0.26	0.55	0.34	0.23	0.28	0.27	0.77
T1	0.48	0.05	0.13	0.66	0.13	0.22	0.92	0.71
T2	0.99	0.35	1.12	0.33	0.08	0.46	0.83	0.74
T3	0.96	0.76	0.12	0.24	0.3	0.06	0.38	0.45
T4	0.25	0.54	0.72	0.97	0.06	0.32	0.3	1.02
T5	0.02	0.43	0.73	0.19	0.37	0.72	0.17	0.64
T6	0.12	0.07	0.81	0.12	0.13	0.64	0.33	0.96
T7	0.84	0.3	0.34	0.06	1.05	0.41	0.32	0.55
T8	0.31	0.42	0.74	0.13	0.71	0.18	0.52	1.09

Figure 6.16: Design rationale for OTI chart redesign

In figure 6.16d the same matrix is used but the values have been encoded through scale, the only remaining visual variable that accurately encodes quantitative perception. This encoding has the potential to overload the vertical axis to display the toolsets as well as tools and makes it possible to see how tools within a specific toolset are performing. However, some new problems arise with this presentation. It is now possible to see the data from either perspective, but very small readings become increasingly difficult to see while very large readings occlude other readings in the display. It also presents a challenge in encoding the target and the control limits in the display.

6.5.5.3 Data Transformation

Figure 6.16d provides the best support for sensor identification but obscures the important quantitative sensor values. While this straightforward scale matching does not highlight the important data, it may be possible to transform this quantitative data to a lower scale that still supports the tasks.

Examination of the hierarchical task analysis and interview notes reveals that *only*

Information Requirement	Data Scale Source	Data Scale Transformation	Visual Scale
Parameter and Sensor Variables	Q (-10<X<+10)	O (>0,>1,>2,>3) N (+,-)	● ● ● ● ● ●
Control Limits	Q	N (within, outside)	● ●
Parameters	N	N	X → ●●
Tools	N	N	Y ↑ ●●
Toolsets	N	N	Y ↑ ●●

Figure 6.17: Data Scale Transformations

ordinal and nominal cognitive operations are carried out on this quantitative data. The task of checking whether a sensor is off-target involves checking whether the sensor is greater than or less than the control limits. This is an ordinal estimation leading to a nominal state (on target or off-target). From the monitoring perspective, the user is less concerned with the precise quantitative readings and more interested in discrete classes of distance from target (e.g. on-target, on-target but close to control limit, outside of control limit). Once this level of information is presented within the chart, detailed information can be accessed on demand.

With this knowledge, a series of data transformations can be carried out (figure 6.17). The parameter and sensor values can be converted to a discrete ordinal range showing distance from the target. The direction of the distance is a nominal variable with two categories, above or below. The crossing of the control limits is another nominal variable with two categories, within or outside.

6.5.5.4 Graphic Form

With these data transformations complete, a new scale matching exercise can also be carried out (figure 6.17). Distance from target is split into six discrete ordinal regions, three within limits and three outside of limits and have been encoded using six different sizes of graphic point. The nominal "direction of distance" variable is encoded with the pre-established control limit colours; blue for above and red for below the target. These hues have been modified so that their luminance is balanced to ensure equivalent salience. The nominal "within or outside control limit" variable is encoded using tone. Icons outside of the control limits change from low colour saturation to high colour satu-

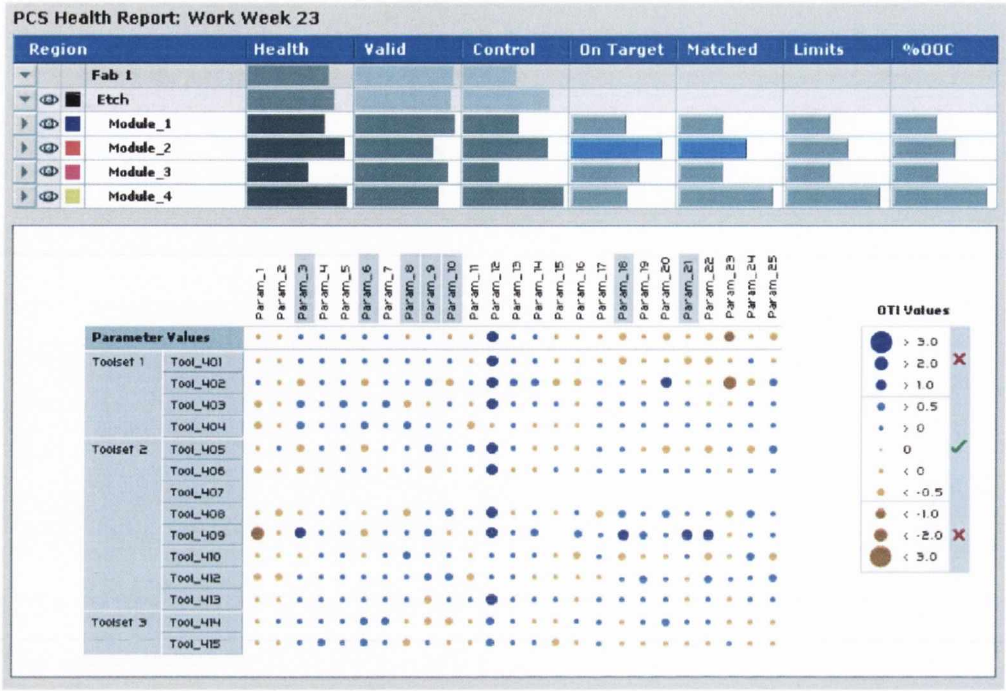


Figure 6.18: Redesigned PCS health report with OTI chart redesign

ration. Technically tone is better suited to ordinal variables, but by using wide variation between the tones, the two categories are easily distinguishable. The nominal variables of parameters and tools are encoded on the horizontal and vertical axes respectively. This enables the tools to be organised into toolsets, which are labelled on the same axis. The redesign was initially reviewed against the 5 issues identified in section 6.2.2 and a number of improvements were noted.

- Issue 1: The mean parameter value is now easily identified in the first row in the display
- Issue 2: Off-target sensors are larger and therefore more salient and easier to select
- Issues 3 & 4: The matrix presentation supports locating specific sensors and eliminates occlusion
- Issue 5: The overall design is more spatially efficient eliminating the need for horizontal scrolling

6.5.6 Final Design: An Ecological Visualisation

The problem-structuring phase revealed a large complex control system involving structural and functional hierarchies that were aligned using a modified ADS . The graphic form in figure 6.18 makes the relationships revealed in this work domain model explicit. The use of an expanding matrix allows lower-level data to be accessed through higher-level metrics. For managers and senior engineers this supports top-down diagnosis of system issues allowing them to quickly identify problematic areas and modules. For junior engineers this technique can be used to navigate directly to their modules and move efficiently between the different indicator charts. The expanding matrix presentation also frees up a lot of screen space allowing charts to be displayed alongside the work domain model. At higher levels of abstraction, this chart area can be used to display health values over time (see figure 6.15). At lower levels it can be used to show the indicator charts at the physical function level within the context of the overall work domain model.

The chart display is closely coupled to the structural/navigational display. At higher levels, selection of an individual health metric will display its health history as a line chart. As the user moves down through the abstraction hierarchy the information in the chart area changes to reflect this. On selection of a control indicator, the relevant indicator chart is displayed. The formatting of the chart is aligned with the work domain component with both the physical decomposition and functional abstraction carrying through to the chart itself. Looking down the left hand side of the screen shows the fab, functional area and module in the work domain component, while the toolsets and tools of the selected module are shown in the chart area. Similarly, the functional abstraction is encoded on the horizontal axis. Looking across the work domain component we can see the relationship between the health indicator and the valid/control indicators, followed by the relationship between the control indicators and its four sub indicators. Finally, the value of the sub-indicator is derived from the parameter values located in the first horizontal row of the on-target indicator chart. This formatting reveals the relationship between low-level data and high-level information in a manner that supports managers and engineers alike.

6.6 Analytical Validation

As an initial evaluation of the differences between the old and new design, the control task analysis that was carried out during the activity analysis (section 6.3.2) is repeated with the new interface. Figure 6.19 shows the decision ladders for two of the main tasks, observing parameter performance and detecting erratic tools. In both cases, we can see that the original design requires more information processing steps to complete the tasks. As the original graphical encoding (shown on the left) did not support reading the data from both structural and functional perspectives, additional information processing steps are required to identify the system state. In the new design (shown on the right), the graphical encoding allows the user to observe the system state and move directly to the appropriate response. This provides a final analytical validation that the redesign should give a better performance than the original design. In order to verify this, an empirical usability evaluation is carried out.

6.7 Empirical Verification

Chapter 4 discusses the limitations of empirical methods for evaluating and communication a design process. Despite their restrictions when dealing with ecological display designs, empirical experiments can provide human performance metrics for specific, critical tasks and allow a designer to investigate any assumption made about perceptual efficiency. An experiment was conducted to verify that the new interface results in better or at least equal performance to the original display (Upton and Doherty, 2007).

6.7.1 Method

6.7.1.1 Scope of the Study

It has been noted that evaluation of ecological designs can be problematic (Vicente, 1999). The variability of real world scenarios is difficult to simulate in a laboratory environment and in many cases the EID approach can radically change usage models, hindering comparative analysis techniques. In this case, an evaluation has been carried out on a portion of the redesign, namely the On-Target Indicator (OTI) chart. There

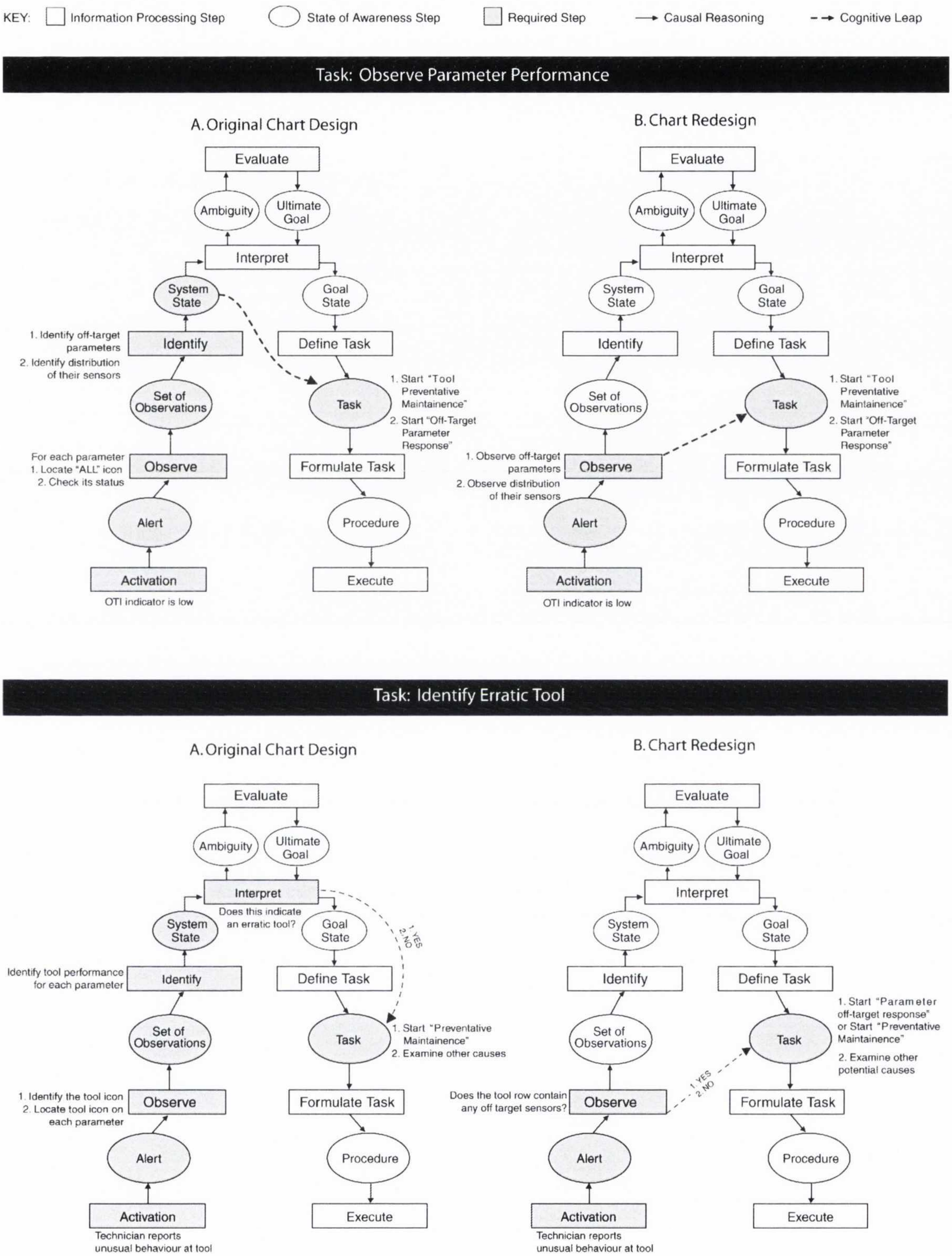


Figure 6.19: Control Task Analysis comparing the two designs

were a number of reasons for this. Firstly, the proposed methodology guides the design of individual interface components that make up the overall ecological display. As such, it is appropriate to test the usability of these outputs. Secondly, the original OTI chart and the new design use the same underlying data and share well-defined, measurable tasks which allow a comparative study to be carried out. Finally, a detailed study of a single graphic form permits us to explore the differences that can be attributed to visual presentation. This level of exploration cannot be carried out with an integrated multidisplay interface as it becomes difficult to differentiate between the effects of the different graphic forms

6.7.1.2 The Displays

The original OTI chart (fig. 6.20) takes the form of a modified control chart, with parameters on the horizontal axis, values on the vertical axis and tools (machines) encoded by way of icons. Control charts are widely used in industrial settings and play an important role in statistical process control. During a task analysis, a number of problems were noted with the original design (see chapter 6). One of the main issues was that the display allowed the key users (process engineers) to identify problems with particular parameters, but did not provide adequate support for diagnosis of these problems. It also did not support the identification of specific tool performance, another desirable feature. This chart was originally selected from a range of templates provided by a charting application. One of the key ideas behind EID is to embed a model of the work domain within the visual design of the display. This externalised system model supports the user when dealing with unanticipated events. A work domain analysis of the on-target indicator chart revealed that the information it displayed related to two perspectives of the work domain; the functionality of the monitoring system and the physical organisation of the equipment. While the original design highlighted the former quality, the visual encoding of the tools made specific equipment issues difficult to discern. The redesigned OTI chart (fig. 6.21) is an ecological display that captures both perspectives, providing equal support for off-target parameter and tool detection and diagnosis of equipment issues. Through the proposed visual design methodology, data transformations were carried out. This reduced the quantitative data associated with the sensor readings to a set of ordinal ranges. Following this visual scale matching was carried out to generate a design space of potential solutions from which the redesign was chosen. This experiment studies whether the data transformations and subsequent scale matching have had an impact on the usability of the chart.

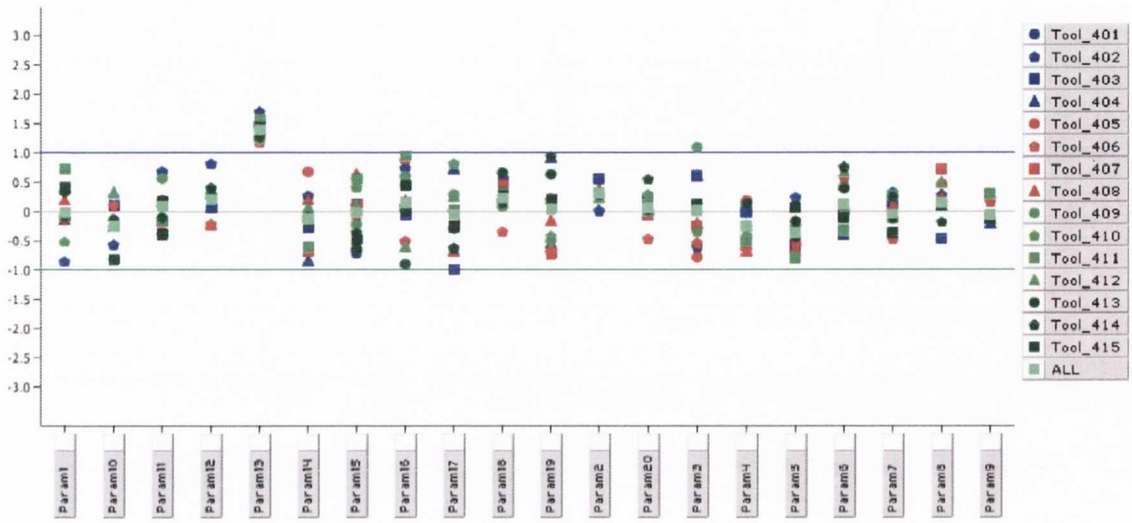


Figure 6.20: The original OTI chart (Chart A)

6.7.1.3 Participants

A total of 20 participants, 14 males and 6 females, took part in the study. Their ages ranged from 22 to 40 years of age. 10 were postgraduate students from the computer science department of Trinity College Dublin and 10 were industry employees. None were considered to be domain experts as they had no knowledge of the process control health monitoring or the displays involved, however all were experienced computer users. Despite lacking domain expertise this group was considered suitable due to the perceptual nature of the experiment. The participants carried out the study during regular working hours but were not compensated in any other way for their time. Access to expert users was difficult, however the experiment was repeated on a much smaller group of four process engineers providing anecdotal evidence presented in the discussion section of this paper.

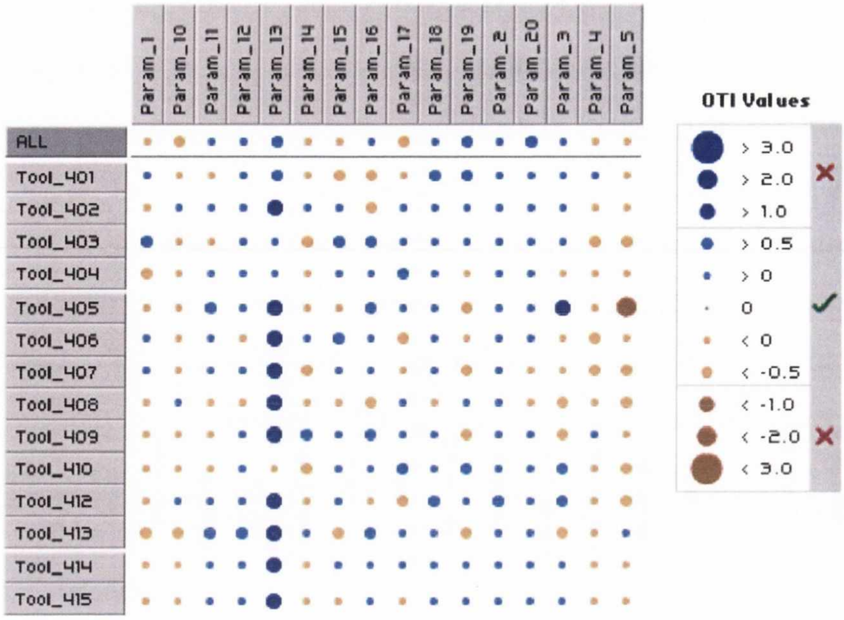


Figure 6.21: The redesigned OTI chart (Chart B)

6.7.1.4 Experimental Platform

System Data During the interface design process a number of OTI charts, using real system data, were studied to identify key features indicating abnormal behavior. Twenty mock datasets were generated with specific features encoded in each. Each dataset involved 300 sensors, consisting of 20 parameters on 15 tools. Process engineers validated these data sets as being representative of the scale and complexity involved in real-world monitoring.

Interfaces Two displays were studied in the experiment. The original OTI chart takes the form of a modified control chart. This was labeled chart A in the experiment. The redesigned OTI chart is a more ecological display incorporating the sensor values and their equal relationship to the physical system (tools) and the functionality of the monitoring system (parameters). This was labeled chart B in the experiment.

Materials A custom web application was developed using Macromedia Flash software and a MySQL database to carry out the experiment. This application both presented the information to the participants and logged their performance. The study was carried out on desktop computers running Windows XP. The graphics were presented on 17" LCD

Monitors with a 32bit colour setting.

6.7.1.5 Tasks

Four primary tasks were selected from the range of activities associated with the OTI charts. These tasks are described below and are ranked in accordance with their increasing levels of difficulty and complexity

1. *Select off-target sensors.* This involves identifying individual sensor readings that lie outside of the control limits. These need to be brought back into control to keep the process stable.

2. *Select off-target tools.* This involves identifying tools that contain sensors that lie outside of the control limits. A task analysis showed that users often need to see the performance of a specific tool based on information from outside sources (e.g. machine technicians). While the action here reverses this process it provides a good indication of whether the relationship between tools and sensors is made explicit in the display.

3. *Select parameters that are off-target but matched.* This involves (a) identifying parameter sensors (labeled ALL) that lie outside of control limits and then (b) identifying whether this parameter is matched. Matched parameters exhibit tight clustering of their tool readings. Unmatched parameters have a highlighted label. An off-target but matched parameter indicates that its control-limit were set incorrectly and need to be adjusted.

4. *Select tools with three or more off-target sensors.* This state indicates a "dog" tool, one that exhibits erratic behavior. This involves identifying individual sensor readings on the same tool that lie outside of the control limits. This tool must be taken down for maintenance.

In each case, the participant was required to identify features relating to their task by selecting the appropriate interface elements i.e. sensor icons, tool labels, parameter labels. A chart can contain from 0 up to 3 features. Once all features are selected a submit button must be pressed to mark completion of the task.

6.7.1.6 Design

This is a within-subject design. The four tasks were presented in a random order and the chart type order was alternated for each user and for each task. The tasks were repeated four times for each chart (8 in total) to capture the four different number of features (0-3). The order of the number of features was also randomized for each user. An increased number of features is thought to increase the complexity of the task. As a result, some interaction between the independent factors was expected. Separate models were used for measuring efficiency, accuracy and satisfaction and the analyses were carried out separately for each task.

6.7.1.7 Performance Measures

Efficiency relates to the amount of time taken to complete a task. This is measured as the time between the initial presentation of a chart and the selection of the submit button once the task is complete. Accuracy relates to the number of errors incurred. An error is the incorrect selection of an interface element or failing to select an element that corresponds to a feature. Satisfaction is a subjective judgment of the displays. Once the participant had completed the task with both displays they were required to select which one provided better support or if they were equal. All of these performance measures were recorded by the application during the experiment.

6.7.1.8 Training & Supplementary Materials

Each participant was presented with a short animation giving an overview of the work domain, the tasks and the chart types, including interaction techniques for each chart. Following this, they registered their name and were presented with the tasks in a random order. Each task was preceded by a description accompanied by two animated demonstrations of how to complete the task with either chart. At this stage the participants were asked to explain the task and their interaction strategies. If correct they were allowed to proceed, if not they were asked to re-read the instructions and were tested again to see whether they fully understood the task. The original design was labeled Chart A and the redesigned ecological display Chart B.

6.7.1.9 Hypotheses

Task 1, select off-target sensors, involves detecting ordinal differences between objects i.e. is a sensor greater or less than the control limit. Based on the basic tasks model of graphic efficacy (Cleveland and McGill, 1985) chart A, the original design, should give better performance results as it encodes the sensor values and control limits using position along a common scale. This encoding is shown to be the best for quantitative perceptual tasks.

Task 2, select off-target tools, involves detecting ordinal differences between objects, then identifying nominal relationships between objects. While chart B may prove slower for the initial ordinal task, its matrix layout provides better support for the nominative association between icon and label. This layout also removes the risk of data occlusion, where icons of similar value lie on top of each other. Together these should result in faster completion times and less errors for chart B.

Task 3, select parameters that are off-target but matched, involves identifying nominal relationships between labels and icons (i.e. finding the "ALL" reading), detecting ordinal differences between objects (position of "ALL" reading), then identifying a nominal state (matched status). The layout of Chart B separates the parameter reading from the sensor readings. It also presents the parameter reading beside the label where the matched status is encoded. Based on the proximity control principle (Wickens and Carswell, 1995) this should result in better performance for chart B.

Task 4, select tools with three or more off-target sensors, involves identifying nominal relationships between objects, then detecting ordinal differences between objects. The task constitutes a global question and involves understanding the data from the quantitative and two nominal variables. As chart B follows Bertin's rules for graphic construction (Bertin, 1983), its visual form should make the target area pop out of the graphic form and result in better performance.

6.7.2 Results

The analyses were carried out separately for each task. For the efficiency (log of time) and accuracy (number of errors) measurements, generalised linear models were employed incorporating the repeated measures aspect of the design. As the satisfaction measurement was taken at the end of each task block, it had a smaller number of observations making a significance test unsuitable. Instead a confidence interval for the proportions is reported. The results are charted in figure 6.22.

Task 1: Select off-target sensors

Efficiency An Analysis of Variance (ANOVA) shows effects for chart type $F(1, 57) = 9.9918$, $p < 0.01$ and number of features $F(3, 57) = 16.954$, $p < 0.001$ but also a chart type by number of features interaction $(3, 57) = 8.5018$, $p < 0.001$. A Fisher LSD post hoc test on this interaction shows no significant difference between the charts ($p=0.594$) where no features exist, but mean performance time improvements for chart B were significant with 1 & 2 features ($p<0.0001$ & $p<0.0005$ respectively) and present but not significant ($p>0.056$) with 3 features.

Accuracy An ANOVA shows strong interaction between chart type and number of features. A post hoc test was carried out with the following results. Chart A results in more errors than chart B in all cases where a feature exists. This difference is significant for 1 and 2 features ($p < 0.001$ and $p = 0.016$ respectively) but not significant for 3 features.

Satisfaction 14 out of 20 participants chose the redesigned chart compared to 3 out of 20 each for both the original chart and no preference. A 95% confidence interval for preference of Chart B over the other two options ranges between 55% and 91%. [1]

Task 2: Select off-target tools

Efficiency An ANOVA shows effects for chart type $F(1, 57) = 32.9$, $p < 0.001$ and number of features $F(3, 57) = 35.327$, $p < 0.001$ but again a chart type by number of features interaction $F(3, 57)= 7.5698$, $p < 0.001$. The mean performance time was better for chart B in all cases where a feature existed. A fisher LSD post hoc test on the

[1] generated using wilsons standard error

interaction shows this difference to be significant for 1 and 2 features (both $p<0.001$) and for 3 features ($p<0.01$).

Accuracy An ANOVA again shows a strong interaction between the two factors. As no errors were incurred when no features were present, this level was not included in the analysis. A post-hoc test was carried out on the other results and showed that chart A resulted in more errors than chart B in all cases and that this difference is significant for 1 feature ($p < 0.001$) and 2 features ($p = 0.010$) but not significant for 3 features.

Satisfaction 16 out of 20 users chose the redesigned chart compared to 3 out of 20 for the original chart and 1 out of 20 expressing no preference. This time the 95% confidence interval for chart B over the other two options ranges between 67% and 97 % [1]

Task 3: Select parameters that are off-target but matched

Efficiency An ANOVA shows no-interaction between chart types and number of features $F(3, 57) = 1.0498$, $p < 0.3777$. However, a strongly significant main effect is reported for chart type $F(1, 57) = 12.1$, $p < 0.005$ with chart B giving significantly faster performance times than chart A, and a weaker effect for number of features $F(3, 57) = 3.1829$, $p < 0.05$.

Accuracy An ANOVA showed a weak interaction between factors. The post hoc test showed a significant difference ($p < 0.001$) in favor of chart B where no feature exists. Although the number of errors was greater for chart A than chart B for 1 & 2 features no significant difference between chart types was shown. For 3 features the number of errors incurred was matched.

Satisfaction 16 out of 20 users chose the redesigned chart compared to 3 out of 20 for the original chart and 1 out of 20 expressing no preference. This time the 95 % confidence interval for chart B over the other two options ranges between 67% and 97 % [1].

Task 4: Select tools with three or more off-target sensors

Efficiency An ANOVA shows effects for chart type $F(1, 57) = 24.2$, $p < 0.001$ and number of features $F(3, 57) = 29.8$, $p < 0.001$ and again a chart type by number of features interaction $F(3, 57) = 5.5$, $p < 0.005$. Mean performance time was faster for chart B in all occasions and a fisher LSD post hoc test on the interaction shows this difference to be significant for no features ($p<0.001$), one feature ($p<0.05$) and two features ($p<0.001$) but not significant for 3 features ($p=0.53$).

Accuracy An ANOVA showed no interaction between number of features and chart type. This task demonstrates a main effect for chart type with chart B having significantly fewer errors than A ($p< 0.001$) and a feature effect with 2 ($p<0.05$) and 3 features ($p<0.01$) having significantly more errors than 1 feature. In this analysis 0 features was omitted.

Satisfaction 16 out of 20 users chose the redesigned chart compared to 3 out of 20 for the original chart and 1 out of 20 expressing no preference. This time the 95 % confidence interval for chart B over the other two options ranges between 67% and 97 % [1].

6.7.3 Discussion of Results

For most tasks both number of features and chart type have an effect on user performance. An interaction between these two factors is also present making it difficult to report main effects. We provide a general discussion of the results below.

Task 1: Select off-target sensors Chart B gave faster performance times in all cases except where no feature was present; in this case chart A was faster. In general, chart B resulted in fewer errors than chart A and gave a higher rating for satisfaction. It was originally expected that chart A would outperform chart B for this task. The results show that this is the case only when no features are present i.e. when the system is in

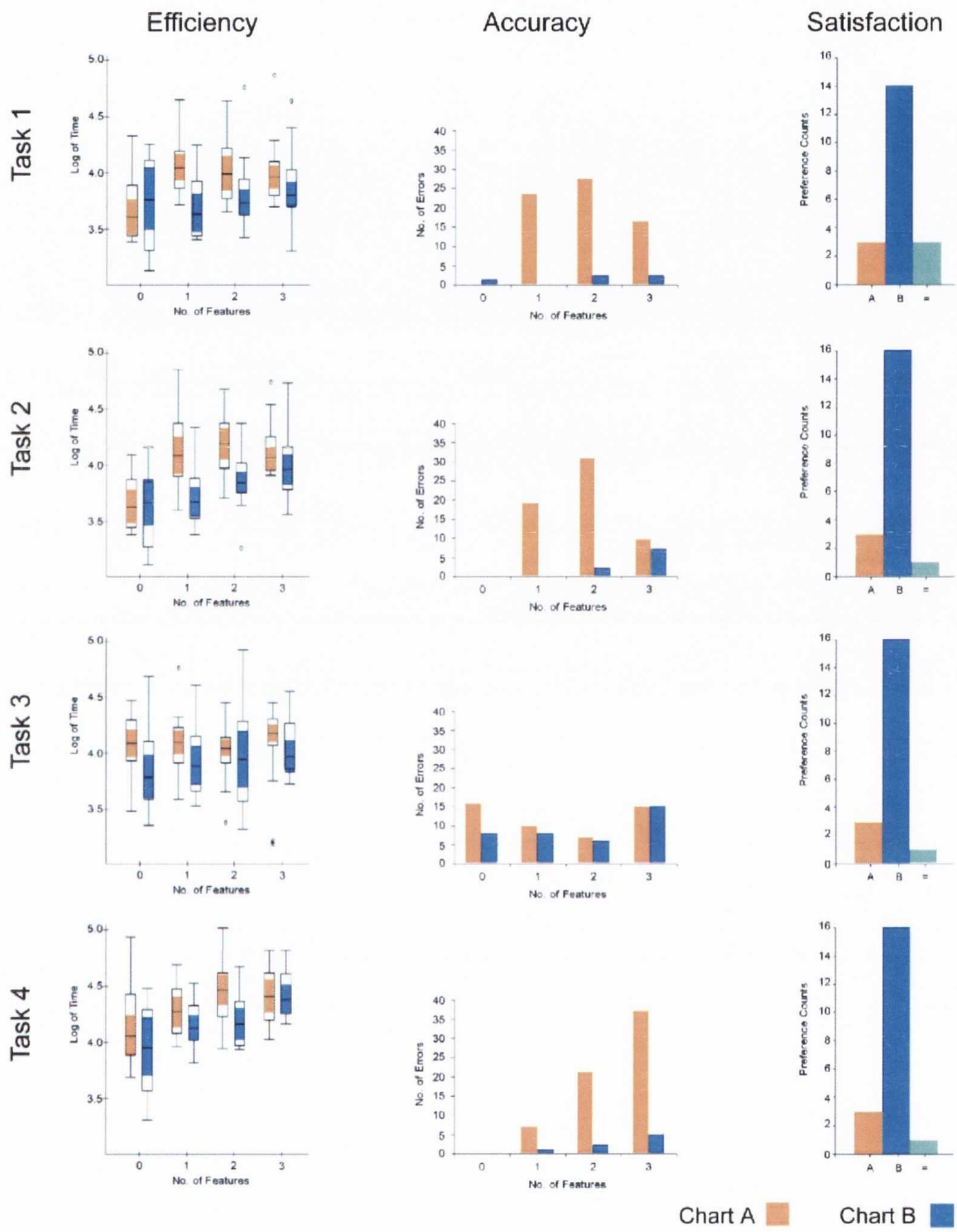


Figure 6.22: Results for each of four tasks and three performance measures

control. While chart A's use of position on a shared scale should improve detection of a feature, the large number of icons may create visual noise that reduces performance. Chart B's encoding method causes off-target sensors to increase in scale and saturation. This improves the salience of these features. Their presentation within a matrix display eliminates the potential for data occlusion which may have resulted in the improvements in accuracy.

Task 2: Select off-target tools Again chart B was faster in all cases where a feature existed but this time the differences are greater. Chart B resulted in fewer errors than chart A and again gave a higher rating for satisfaction. This was the expected result and is attributed to the matrix presentation. This layout makes it easier to relate the sensors to their tools as they are located on a shared spatial axis.

Task 3: Select parameters that are off-target but matched As predicted the results show a significant improvement in efficiency for chart B and better accuracy in all cases except where three features exist. In this case equal numbers of errors are committed. This was the most complex task as is evident from the high number of errors committed with both charts. We attribute the improvements in chart B to the graphic encoding that makes it easier to detect the "ALL" (parameter mean) icon and to integrate it with the matched parameter status.

Task 4: Select tools with three or more off-target sensors It was predicted that chart B would give a better performance due to the spatial encoding of the tools. This eliminates the need to temporarily store values in short term memory and allows the user to assess a tool by scanning the chart vertically. The results show that this is the case with a strong chart effect for accuracy and general improvements for efficiency.

The Number of Features Effect At the outset of the experiment an effect was expected for number of features. The strong interaction between the two main factors was not expected as it was assumed that an increase in features would increase difficulty incrementally for both charts. The results clearly show that this is not the case. If we look at number of errors we can see that this assumption only holds for task 4. For chart A with tasks 1 and 2 the number of errors increased from one to two features, but dropped off with three features. This is an interesting result requiring further exploration. It is possible that with three features present, the additional noise in the display causes the user to change their task performance strategy. While the current study can only identify different responses, future investigations of the displays using methods such as eye-tracking may provide useful information on viewing and task perforamce strategies.

Supplementary Study While it was difficult to access a reasonable number of expert users, four process engineers agreed to carry out the experiment. The small study was carried out as a validation exercise to tesk the acceptability of the new design to the target users. We expected a certain amount of bias towards chart A due to their familiarity with the display. In fact, when presented with the new design (chart B) one engineer stated, "I don't like it and I don't think it will work". While the numbers were not sufficient to generate a statistical model, we observed some interesting results. There was a similar pattern of behavior between this test group and the main group for efficiency. In all tasks chart B gave faster mean response times than chart A where a feature existed. There was too much variation in the errors figures to draw significant conclusions, but the satisfaction measurement showed chart B was preferred for tasks 1 and 4, chart A and B were considered equal for task 2 and chart A was preferred for task 3. This is an encouraging result considering the engineers were more familiar with chart A.

6.7.4 Review of Experimental Outcome

Many psychophysical theories e.g. (Wickens and Carswell, 1995; Cleveland and McGill, 1985) give general guidelines for representing data based on specific cognitive tasks. The original OTI chart was constructed in-line with these guidelines using position to support quantitative judgments between datum. However, the results suggest that the new design is at least equal, and in many cases better, for carrying out the required tasks. This raises the question whether traditional approaches to cognitive graphics processing are too narrow for interactive displays? Many of these approaches rank visual variables in terms of their ability to support a specific task, but cognitive tasks rarely occur in isolation when working with dynamic charts and often a range of tasks can occur in quick succession.

Also, while earlier theories have tended to focus on quantitative relationships, ordinal and nominal relationships play an important role in understanding complex systems. The proposed methodology suggests that the visual encoding of information requirements should be defined by both their position within a work domain model and the tasks for which they are used and the results are supportive of this.

Evaluation Issues in EID The same characteristics that make it difficult to apply simple graphics guidelines also make it difficult to evaluate visual displays for complex systems. While carrying out this experiment a number of specific evaluation challenges were identified. Firstly there is a difficulty in accurately representing work scenarios. While this experiment measures performance for a range of tasks associated with the OTI chart, this is just part of a larger health monitoring system that is used by process engineers. The engineers have access to a much wider set of resources including tacit knowledge and information from co-workers. These factors are beyond the scope of this experiment which can only show what an individual can understand through the displays. A similar issue relates to data. Original data is often unavailable for use in experiments for confidentiality reasons. Even when it is accessible the format is often unusable. In our case users had to identify stable and unstable system states. However, the frequency and severity of problems is unpredictable so it would be unreasonable to expect participants to monitor real world data. As a result mock datasets had to be generated. Secondly there is a trade-off between representing the real-world and the practical limitations associated with experimental evaluation. The number of features factor was introduced to make the study more representative of a real-world monitoring scenario, but the interaction between number of features and chart type makes it difficult to generate statistically significant results for the main effects. A smaller range in the number of features factor would make it easier to obtain significant results but would reduce the validity of the case study. In light of this it is better to think of the experiment in terms of exploration and validation of potential design solutions rather than purely and evaluation study. Finally, there is an issue as to whether the metrics of efficiency, accuracy and satisfaction provide the best means for evaluating an ecological interface. While these metrics tend to be pervasive in usability testing, the results can only inform us in general terms about the differences between displays. Knowing that one design performs better than another is obviously very helpful when choosing a system to implement, however usability metrics do not reveal the actual strategies that users employ when working with graphics. In section 6.7.3 we attribute possible causes for the performance differences between displays. Alternative measurement techniques such as eye-tracking could help to accurately identify these causes and increase our understanding of how graphic forms are used during decision making.

6.7.5 Conclusions of Verification

This aim of this experiment was to study whether a redesign of a chart following the proposed design methodology would affect its usability. The results suggest that the new design provides better support in terms of efficiency, accuracy and satisfaction for a range of key tasks. While the experimental design resulted in strong interactions, post-hoc analyses suggest that chart type is responsible for the improvements in the performance metrics, providing evidence that the design methodology can result in a more usable design.

6.8 Discussion

In chapter 4 ecological interface design was identified as one of the most comprehensive CSE design frameworks, as it provides a direct link between systems analysis and visual design. However this framework was developed around the control of physical processes and is somewhat limited when it comes more abstract information-focused work systems. The PCS health report provides an interface for monitoring information relating to an embedded control system in a large sociotechnical enterprise. The design process presented here began with the EID approach but required a number of additional techniques to cope with the characteristics of the work domain. While this design process produced a successful interface solution, in order to generate re-useable design knowledge it is necessary to go beyond the context of this specific project. In this section the CSE meta-model is used to trace conceptualization and identify the primary design artefacts produced during design. By specifying the higher-level principles, used to control con-

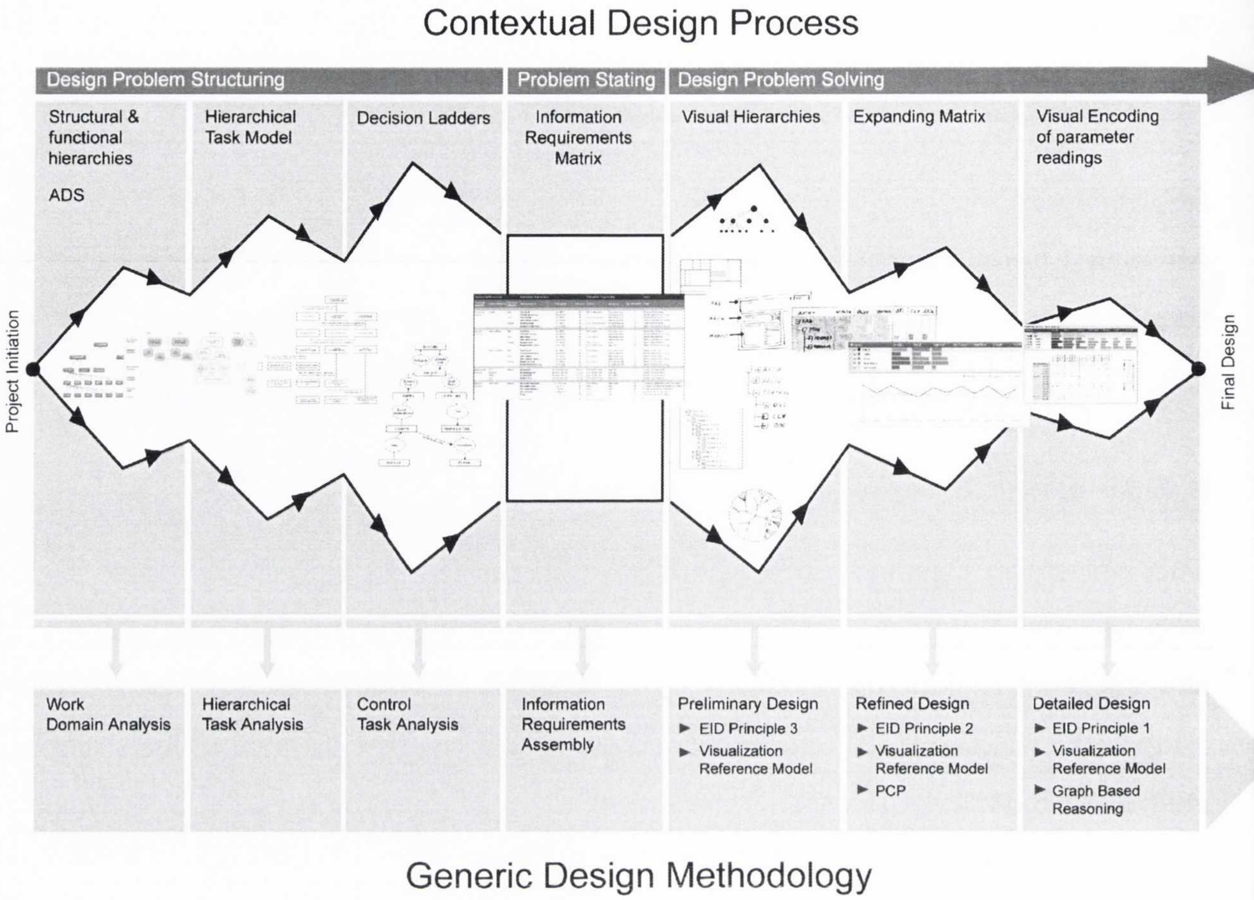


Figure 6.23: A model of design artifacts used in this process

ceptualization, the secondary design artefacts involved in the process are extracted. This provides a more generic design methodology that can be applied to work domains with similar characteristics.

6.8.1 Design Process and Design Artifacts

6.8.1.1 Work Domain Analysis

- *Primary Design Artifacts.* The two hierarchies depicted in figure 6.3 are conceptual sketches of the systems physical and functional constraints. While these hierarchies are not directly aligned, it was possible to integrate them into a unified ADS (fig. 6.4).

- *Secondary Design Artifacts.* As the project started out using the EID framework the initial secondary design artifact is Work Domain Analysis. This is based on the principle that functionality can be described in terms of system constraints.

- *Benefits and Limitations.* The ADS outlines the structural relationships between system components at different levels of abstraction and provides an explicit model of the work domain. However it cannot identify all of the information requirements, as the monitoring activity is not entirely defined by engineering constraints. While the model describes the system under observation, it cannot fully describe the monitoring activity, which involves detecting and interpreting system data in a number of different ways. In order to understand how this occurs, it is necessary to model monitoring practice.

6.8.1.2 Hierarchical Task Analysis

- *Primary Design Artifacts.* The hierarchical task model depicted in figure 6.5 outlines the various tasks associated with monitoring using the original health report system. It describes events and systems states that require responses and outlines normal operating procedures.

- *Secondary Design Artifacts.* Hierarchical task analysis provides the next secondary design artifact used in this process. HTA is based on a principle that system functionality can be described based on the users mental model of their work and actions.

- *Benefits and Limitations.* The HTA can successfully identify tasks that describe user interaction with a system. However as this project demonstrates, a users model of system functionality can be shaped by the limitations of their work tools. In this case, users only realised omissions in their mental models when presented with the ADS. In addition the HTA reports the tasks in a procedural manner and does not take into account how a user perceives the system state. This requires a more detailed analysis of cognitive tasks.

6.8.1.3 Control Task Analysis

- *Primary Design Artifacts.* The decision ladders shown in figure 6.6 identified the information processing involved in interpreting system data. They provide a detailed analysis of how users move between different levels of system state awareness.

- *Secondary Design Artifact.* Control Task Analysis provides the final secondary design artifact in the problem-structuring phase. CTA is based on the principle that users reason about a system at different levels of abstraction depending on the type of decision-making involved.

- *Benefits and Limitations.* In this project CTA is used to relate the various monitoring control tasks back to the structures revealed in the work domain model. By doing this it is possible to take advantage of EID's visual design principles while also benefiting from the models of behaviour generated through activity analysis. The difficulty in applying multiple analytical models is that the information requirements they reveal need to be compiled into an integrated format.

6.8.1.4 Information Requirements Assembly

As multiple analytical methods are applied during design problem structuring, the information requirements matrix has been developed as a transitional artifact for compiling the various analytical outputs.

- *Primary Design Artifacts.* The primary artifact in this case is the IR matrix in table 6.3. Some aspects of this artifact are particular to this project. For example, the multiple hierarchies associated with the health report have required us to specify three columns to describe the position of individual information requirements in terms of the overall system functionality.

- *Secondary Design Artifacts.* As the four main categories; level of abstraction, information requirement details, interaction requirement and notes are applicable to any work domain, they provide a generic structure for modeling information requirements. This can act as a secondary design artifact that supports the integration of multiple system models.

- *Benefits and Limitations.* As design is an iterative process, it is necessary for the designer to continuously revisit the information requirements to ensure design validity. While the IR matrix loses some of the expressive power of the individual system models, it provides a central reference point for inspecting requirements. However the IR matrix should not be considered to be a static document. As with the various models this artifact is a sketch of proposed requirements, which can be edited as more information about the system arises. This became evident during the design phase when the data scales associated with the OTI chart underwent transformations to accommodate cognitive tasks.

6.8.1.5 Preliminary Design

- *Primary Design Artifacts.* The preliminary sketches in figures 6.11, 6.12 and 6.13 are used to explore how high-level structural relationships can be visually encoded. While many alternative representations are possible, this conceptualization process is controlled by a number of visual design principles.

- *Secondary Design Artifacts.* The first principle relates to the EID requirement to visually represent the abstraction hierarchy. However, as this principle does not provide more detailed guidelines for graphically encoding the hierarchy, the visualization reference model is applied to generate multiple sketches of hierarchical structures. The need to integrate the control and structural hierarchies into a unified form, in line with EID, led to a commitment to a preliminary design concept.

6.8.1.6 Refined Design

- *Primary Design Artifacts.* The refined sketches in figures 6.14 and 6.15 were generated to examine how the health indicators could be presented to the end user. They experiment with different representational formats to identify which ones provide better perceptual cues.

- *Secondary Design Artifacts.* As with the previous phase, the sketching process is controlled by the EID principle, while concept generation is informed by the visualization reference model. The EID principle relating to matching cues and constraints provides a distinct design goal. The process of data scale transformation is used to generate visual encodings that provide better support for control tasks. PCP informed the use of consistent visual encoding to support perceptual comparison across both the structural and control aspects of the display matrix.

6.8.1.7 Detailed Design

- *Primary Design Artifacts.* The detailed sketches deal with the representation of system components. The sketches in figure 6.16 demonstrate that a wide selection of alternative encodings can be generated, each of which provides a different level of task support.

- *Secondary Design Artifacts.* The sketches are informed by the final EID principle that requires the presentation of visual components in a manner reflective of the part-whole structure of movements. The visualization reference model is applied to generate a range of alternative solutions. Commitment to one of these concepts is based on graphic reasoning theory, which highlighted the representation that supported the widest range of tasks. Further refinements were made through data transformations to provide the final design solution.

6.8.1.8 Benefits and Limitations of Visual Sketches

One of the fundamental difficulties in designing ecological visual representations is that they can be read at multiple levels of abstraction. As images can be perceived in a holistic manner, design decisions made at one level will influence the design choices at another level. The three phases of design problem solving provide a structured approach for dealing with this issue. By starting at a high-level of abstraction, the designer can ensure that the overall workspace provides a valid representation that acts as an external system model. Subsequent phases narrow the focus of the design until individual components must be visually encoded. This progressive reduction of the design choices allows the final design decisions to be constrained by earlier ones, reducing the complexity of the visual design task. Within each design phase the different processes described by the visualization reference model provide explicit techniques for concept generation. Design commitment occurs within the context of earlier design decisions and is controlled by the EID principle associated with the particular problem-solving phase.

6.8.2 A Design Methodology

The series of secondary design artifacts used in this process provide a design methodology that answers the three research questions posed at the start of the chapter. In doing so the approach extends the EID framework in a number of ways.

Firstly, in terms of identifying which analytical models to apply, this approach supplements work domain analysis with hierarchical task analysis and control task analysis. This makes the approach applicable to a wider set of work systems than those catered to by EID. While work domain modelling is important for building a structural model of a system, the hierarchical and control task analyses allow a designer to identify specific information processing tasks and strategies. As a result, this methodology can be applied to domains beyond those that are constrained by an engineered system or physical process.

Secondly, in terms of choosing appropriate visual design guidelines, the methodology highlights the importance of conceptualization and shows how sketches are both generated and constrained by visual principles. The techniques associated with the visualization reference model provide a more concrete approach for creating concepts. The role of EID principles are clarified and shown to provide goals relating to different phases of the design process. Essentially they act as constraints that guide commitment to particular concepts.

Finally, in terms of the overall design process, the methodology provides better support for bridging design gaps at various levels of abstraction. The utility of the work domain model for guiding the overall visual composition of the display workspace has been demonstrated. A more important contribution is the explanation of how task analyses not only reveal a wider set of information requirements but can also guide the visual encoding of system data. Through the identification of information processing tasks and strategies, system data can be transformed and matched to efficient visual encodings. This allows the methodology to guide the design process right down to the syntactic level.

The issue of reusability is obviously very important when proposing a design methodology. In chapter 4 it was proposed that methodologies generated using the CSE meta-

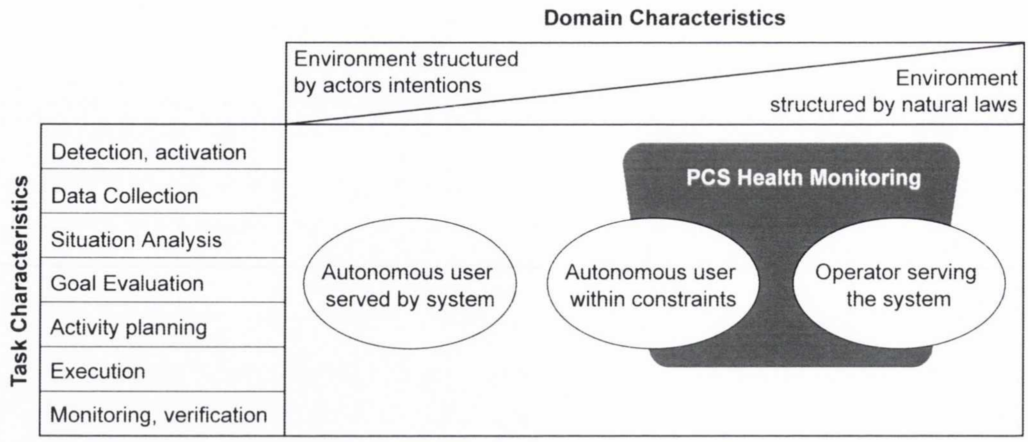


Figure 6.24: Locating the PCS system on the WTU map

model should be applicable to work systems with similar characteristics. Using the Work, Task, User (WTU) map, introduced in section 2.4.1, the attributes of the cognitive system associated with PCS health monitoring can be described. In terms of work domain, PCS health monitoring is associated with a physical, engineered work environment. Sensors measure processing activity and engineering structures describe different levels of system abstraction. In terms of tasks, the cognitive system must detect anomalies in sensor performance, analyse the situation and evaluate it in terms of system goals. The resolution of issues are planned by this cognitive system but their execution is handled by a separate equipment engineering department. In terms of users, the categories provided by the WTU map are not easily applied to this system. As users must respond to problems detected by the health monitor they can be identified as operators serving the system. However, the targets that are used by the system are not imposed by natural constraints but by operational policies. Process engineers may change targets to meet new operational goals, so in this sense they can be considered to be autonomous users operating within constraints. Figure 6.24 shows how the characteristics of this system map to the WTU map. Other systems that occupy the same region will exhibit similar characteristics and can make use of the secondary artifacts used in this methodology. However this system exhibits a range of additional attributes that are not dealt with by this mapping approach, including multiple users and the application of automated control. This suggests that to make design methodologies more re-usable it is necessary to provide a better classification of cognitive systems. This issue is further discussed in chapter 8 following the second case study.

Chapter 7

Designing Visual Decision Support for Remote Operations Control

7.1 Introduction

In this chapter a second design project is reported. The project deals with the design of a visual decision support system for remote operations control in a semiconductor manufacturing enterprise. The cognitive system involved in this project is different from the previous one in a number of ways. Firstly, the move to the Remote Operations Control Centre (ROCC) is a current activity and aspects of the new work system are still under development. While the previous project involved redesigning an interface for a stable work system, this project analyses *an evolving work system* and identifies the effects of changes on system functionality. Secondly, while the previous project focussed on engineering structures within the fab, operations control involves balancing manufacturing and engineering goals across the entire enterprise. This involves a more extensive set of constraints and requires close collaboration between workers at different levels of management. As a result this project has *a much larger scope* than the previous one[1]. Despite

[1]As this project involved the analysis of a bona fide cognitive system in an industrial facility, aspects of the system that were deemed commercially sensitive have been omitted from this report. In addition all low-level system data presented here has been deliberately altered to ensure confidentiality. The goal of this research is to report on the design process for generating visual decision support systems in a sociotechnical enterprise. The target domain is representative of such an enterprise and these restrictions do not affect its utility for identifying the goals, constraints and relationships that define the cognitive system being examined.

these differences the project still involves a complex design problem and the same three questions relating to design knowledge gaps must be answered:

1. What analytical approach should be taken to identify information requirements?
2. What representational guidelines are appropriate for designing a system image?
3. What design process should be followed in the development of this interface?

These questions are resolved during the problem-structuring, problem-stating and problem-solving phases used to report the design process. The increased scope of this project is used to test the extensibility of the CSE meta-model and a second design methodology is extracted from this design process.

7.2 Project Overview

Manufacturing Operations Control (OC) relates to the core manufacturing goal of moving Work In Progress (WIP) through a production process to meet product demand. For most of the 20th century OC was tied to a mass-production model that used High Volume Manufacturing (HVM) to reduce unit costs and secure competitive advantage. However, during the latter half of that century this model was predominantly replaced by lean manufacturing, a more efficient and dynamic approach to production. Three main characteristics of lean manufacturing have been identified (Womack, 1991). The *lean philosophy* that identifies customer satisfaction and minimisation of waste as primary goals, new *lean principles* in relation to operations control, development and co-ordination and specific *lean production approaches* for achieving these principles.

In terms of operations control, overproduction is considered a very serious waste resulting in excess inventory, storage costs and product depreciation. *Just-in-time production* is a key lean manufacturing principle where processing of WIP begins only after orders have been received. This requires a more flexible approach to manufacturing than the tightly-coupled, linear assembly line associated with mass production. A *Flexible Manufacturing Systems* (FMS) is a production scheme that can deal with frequent and continuous changes to production goals. An FMS consists of a network of processing cells and storage buffers, allowing product to take multiple alternative paths through a production process. However, the versatility that is achieved by FMS means that manufacturing scheduling is a problem that must be solved repeatedly and rapidly Ammons et al. (1988). While automated systems can support simple scheduling tasks, human schedulers bring the expertise and flexibility that is required to resolve operational conflicts.

Semiconductor manufacturing combines high volume manufacturing with a lean manufacturing philosophy. Operations Control in this enterprise involves the monitoring and control of a large network of interconnected flexible manufacturing systems. Toolsets form the basic FMS processing cells but the re-entrant nature of the process flow introduces a further level of complexity. As a result, low-level decision-making must be co-ordinated with high-level production goals. The cognitive system that supports fab operations control involves physical, social and technological factors but recent developments in fab design are changing the way in which these factors are configured.

7.2.1 Developments in Fab Operations Control

In section 5.3.2 the reasons for the move from 200mm to 300mm wafer production were outlined. Operations Control (OC) is primarily the responsibility of the manufacturing department who distribute the workload across a management hierarchy. One of the major changes associated with the move to larger wafer sizes is a restructuring of this social organisation.

In the 200mm facility, the OC management hierarchy consists of a *line manager* who controls and plans high-level production strategies, *manufacturing supervisors* who monitor and resolve issues associated with functional areas and *manufacturing technicians* who manage a limited number of process tools (Fig. 7.1a). Technicians collect WIP from an *Automated Material Handling System* (AMHS), manually load it into process tools and are able to observe behaviour at the tools. All of these workers are located in the fab and communication about production goals, targets and engineering issues occurs predominantly through the management hierarchy. Verbal reporting plays a critical role in communicating system state information, although an on-line Manufacturing Execution System (MES) is also used for accessing system data.

The 300mm facility has seen some radical changes in this work configuration. Mechanical automation, introduced to overcome ergonomic constraints, has removed the need for manual loading of process tools. There has been a corresponding increase in data-processing and automated control, with the result that operations control is moving to a more centralised model. The Remote Operations Control Centre (ROCC) is a workplace located outside of the cleanrooms, where a small number of specially trained technicians are co-located with a line manger (Fig. 7.1b). In the ROCC, technicians control a large fleet of tools and the line manager can continuously monitor performance across the entire system. The remote operations model has a number of proposed benefits for manufacturing:

- It eliminates the lag-time incurred while technicians move between tools and manually load WIP for processing.
- The technicians' attention can be focused on building efficient batches of WIP that optimise tool utilisation
- The centralised location improves communication of production strategies between line managers and technicians

These factors should contribute to better performance in terms of processing speed; however the pervasive use of automation also results in systemic changes to how OC occurs. The ROCC model involves a radically different social organisation, increased responsibility for individual workers and a much stronger dependence on information systems for monitoring, interpreting and responding to the system state. These changes alter the cognitive work involved in operations control.

The move to the ROCC is being carried out in a phased approach and so far the *technical* and *organisational* changes have preceded major changes to the information systems. While the original information systems provided adequate support under the original work configuration, the demands of the remote operations model require much higher levels of *system observability*. The aim of this project is to design a visual decision support system that allows fab operations to be observed, interpreted and communicated in an efficient and effective manner. Initially this requires a model of system functionality to be developed, but the characteristics of OC make this difficult to achieve.

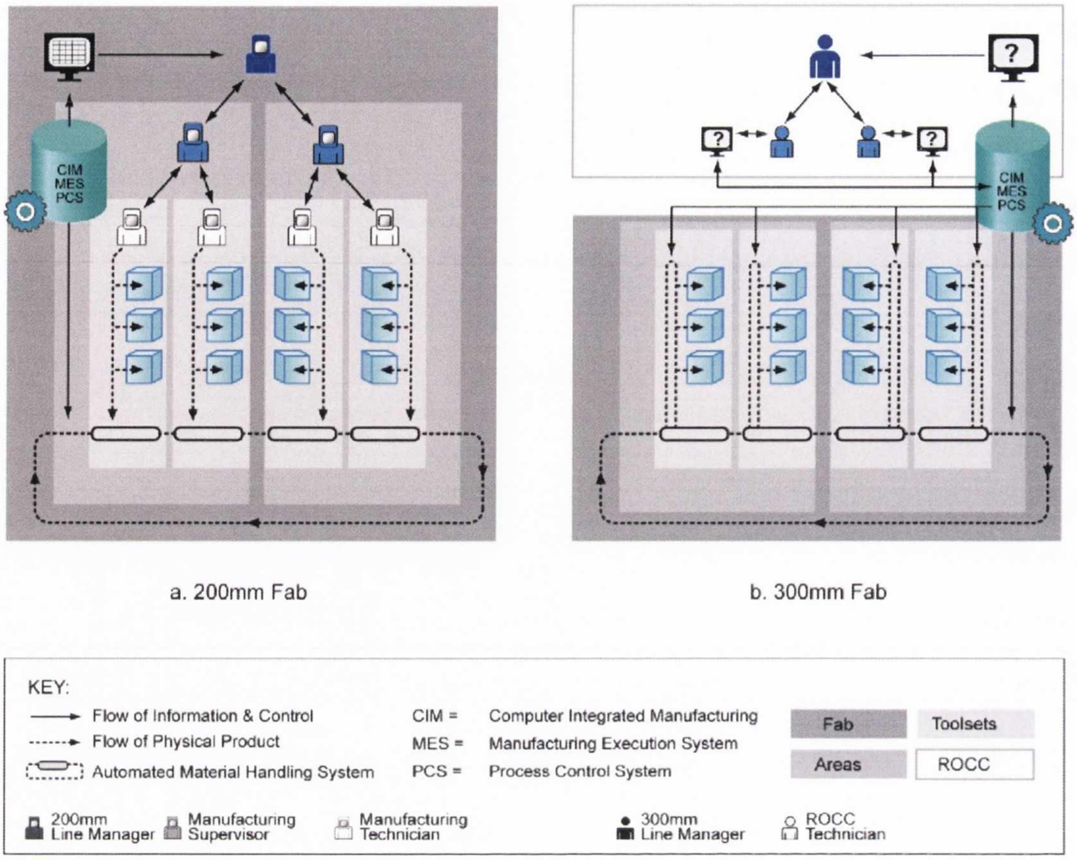


Figure 7.1: Operations Control in (a) 200mm Fab and (b) 300mm Fab

7.2.2 Operations Control as a Cognitive System

Chapter 2 identified how the different analytical frameworks were developed around different types of work environments and different perspectives on cognitive work. The difficulty with fab operations control is that the associated work system can be described using any of these perspectives.

7.2.2.1 Operations Control as Distributed Cognition

OC in the fab involves the co-ordination of human workers, information systems and automation. As the functional goals are divided amongst these agents and artifacts, the work can be described as a distributed cognitive system. A Distibuted Cognition (DC) analysis of the system requires an ethnomethodological approach, where observation and interviews are used to generate rich descriptions of work practice. Analysis of these descriptions can identify representations at different levels of abstraction and reveal the information flows involved in operations control. However, this bottom-up approach to modelling system functionality suffers from the limitations outlined in section 2.3.2.2. These are particularly relevant in the case of fab operations control, where the large number of workers makes it exceedingly difficult to carry out in-depth observational analysis. In addition the different vocabularies and constraints associated with the separate functional areas further increases the difficulty of constructing a generic system model. The ethnographic approach is also not well positioned to describe changing work environments. Ethnography is primarily a descriptive activity and this makes it difficult to apply to evolving systems.

7.2.2.2 Operations Control as defined by Constraints

Alternatively OC can be represented as a consequence of the work domains constraints. Unlike the previous project, OC relates directly to the movement of physical product through an engineered process. This process provides a stable set of constraints that define how processing can occur and can be used to develop a work domain model. This model can identify the structures and vocabulary for describing OC in a generic manner. Despite this, the development of an ADS model of the fab faces a number of challenges. Firstly the complexity of the process, with its iterative flow, in-built flexibility and large scale makes the identification, analysis and representation of causal relationships very difficult. Secondly, temporal constraints play an important role in Fab OC as the product must meet delivery dates. However, temporal constraints are event-related and are not described by this modelling approach (see section 2.3.1.2).

7.2.2.3 Operations Control as Intentional Activity

Despite the fact that OC relates to a physical process, the flexibility designed into this process means that the system is constrained by physical, functional and also intentional constraints. WIP passes through the same processing tools, but all WIP is not the same. Under the Just in Time (JIT) principle WIP is only started after orders have been placed but changes in the system state during production can affect the progress of these orders. Consequently, an important part of OC involves prioritising specific lots in processing queues so that delivery dates for orders are met. Due to the re-entrant nature of the process, prioritisation requires a manager to estimate the impact of these scheduling decisions on the overall line. As this occurs within the dynamics of an open system, the outcome cannot be precisely predicted using formal methods and managers rely on heuristics, distributed expertise and tacit knowledge to support decision-making. These factors suggest that an activity analysis is required to fully understand OC. However this requires an ethnomethodological approach, the limitations of which in relation to this system have already been outlined.

7.2.3 Preliminary Project Review

While OC can be described from multiple perspectives, none of the analytical frameworks can generate a fully comprehensive model of system functionality. In the following section the application of multiple analyses is presented as part of a design problem-structuring process. By mixing different analytical methods a range of models are generated that provide alternative views of OC. Comparison of these models reveals relationships that give a better insight to system functionality. Beyond revealing functionality, it is necessary to identify the impact that changes to the work configuration will have on work practice. In design problem-stating phase, changes to social structures, information flows and cognitive strategies are identified. As part of this phase new system metrics are generated and information requirements are specified. Finally, the transformation of information requirements into a visual decision support interface is presented in the design problem-solving phase. As with the previous chapter the progression of visual design through sketching is described using a detailed design rationale.

7.3 Design Problem Structuring

The collaboration and intentional aspects of this work system mean that ethnography will play an important role in modelling its functionality. However the scale of the fab presents a serious challenge for conducting observational work. In order to get the best value from field studies it is necessary to carry them out in a targeted manner. The top-down approach used by cognitive work analysis allows a work domain model to be developed based on its functional goals and physical or process limitations. The constraints were initially defined based on the knowledge gained during the bootstrapping stage described in chapter 5. This *formative* model should identify aspects of the system that will remain fixed irrespective of changes to work practice. These invariants provide a stable structure around which other aspects of the system are developed and therefore present a useful starting point for problem structuring.

7.3.1 Work Domain Analysis

As an initial step in generating an abstraction decomposition space, it is necessary to state the functional purpose of the work system. However, manufacturing systems present a modelling challenge even at this early stage as they involve two conflicting high-level functional purposes; namely to *manufacture efficiently* and to *maintain system stability* (Upton and Doherty, 2005). Achieving an efficient overall production rate requires WIP to be evenly distributed and for processing rates to be consistent across the line. Continuous processing, where factories operate 24 hours a day, 365 days a year is also used. While these are effective strategies for efficiency, process tools are subject to wear and tear and cannot function indefinitely. Tools require maintenance in order to achieve high levels of precision and to avoid mechanical faults. Regular Preventative Maintenance (PM) is a strategy for ensuring system stability. The difficulty is that maintenance requires a tool to be taken off-line and this reduces the processing capability of its associated operation. The timing of PM's are planned to maximise a tool's availability for processing, but as an open, dynamic system, manufacturing is subject to unexpected events that can require PM rescheduling. Monitoring efficiency and stability in the fab involves the alternative manufacturing and engineering perspectives respectively (see chapter 5) and each of these use very different physical and functional decompositions of the fab.

7.3.1.1 The Manufacturing View

The manufacturing view decomposes the production system according to position in the process-flow and provides structures at different levels of abstraction (see figure 5.3). The *process line* is divided into four *phases* of front-end, back-end, packaging and testing. At a lower level the line is divided into several *regions* each corresponding to a week's

Decomposition →

Abstraction ↓

	Process	Phase	Region	Operation	Layer
Functional Purpose	Manufacture Efficiently Production Rate				
Abstract Function			Maximise Speed Region Rate		
Generalised Function				Maximise Speed Operation Rate	
Physical Function					Maximise Speed Layer Rate
Physical Form					Layer ID, Orders, Tools

Figure 7.2: The Manufacturing ADS

progress through the line. *Operations* relate to specific processing activities such as etching. Operations may be repeated multiple times during processing. Each repetition is described as a *layer* and occurs at an individual point in the line known as a *process step*. A number of metrics are associated with these structures providing a functional abstraction of the high-level goal of efficient manufacturing. The overall efficiency of the process line is given as a *Production Rate*, which is the number of completed operations in a shift divided by the total inventory in the line. While the four phases provide a structural level, they do not have an associated efficiency metric. A *Regional Rate* is used to describe production efficiency within regions. Operations have an *Operation Rate* that is based on the output of their associated toolset, while a *Layer Rate* describes the rate of production at an individual layer or process step. These physical structures and functional metrics can be aligned to provide an ADS (figure 7.2).

7.3.1.2 The Engineering View

The engineering hierarchy was described in the previous chapter. Physically, it divides the system into areas, modules, toolsets and tools associated with these areas. In the last project this physical decomposition was used to structure PCS health values; however the same structures are also used to support the high-level goal of maintaining system stability. An important measure of stability is the *% availability* metric. This is reported at the level of fab, functional areas and toolsets and is derived from the status of individual tools which are either *up to production* or *offline*. Again these physical structures and functional metrics are aligned to provide a second ADS (figure 7.3).

Decomposition →

Abstraction ↓

	Fab	Func. Areas	Module	Toolset	Tool
Functional Purpose	Maintain Stability % Availability				
Abstract Function		Maintain Stability % Availability			
Generalised Function				Maintain Stability % Availability	
Physical Function					Up to production / Offline
Physical Form					Tool ID, Orders, Layers

Figure 7.3: The Engineering ADS

7.3.1.3 The Abstraction Lattice

Unlike the previous project where the issue was to integrate a control hierarchy and a physical decomposition, both of these views provide valid decompositions of the work domain but taken from alternative perspectives. While the two ADS models of the fab shown in figures 7.2 and 7.3 are very different in terms of physical and functional abstraction, they share the same properties at the level of physical form where orders are processed by tools at specific layers. Figure 7.4 provides a schematic of this relationship. This commonality can act as a bridging point between the two views. While a single physical decomposition causes means-ends relationships to be represented as an *abstraction hierarchy*, using two conceptual decompositions allows these relationships to be represented as an *abstraction lattice* (fig. 7.5). An abstraction lattice represents situations where means-ends relationships can be traced down through levels of abstraction in one view and then up through levels of abstraction in an alternative view of the same system (Upton and Doherty, 2005). This type of reasoning is necessary to resolve the conflicting goals of high efficiency and high stability. This approach allows the levels of functional abstraction to be reflected across the level of physical form joining up the two ADS representations. This new ADS (fig.2) captures system variables from both views at multiple levels of abstraction. Each cell in this model represents the system state at a specific level of abstraction and adjacent cells have a causal relationship with one another. In this manner a low production rate can be traced down to the specific operations and layers with slow processing speeds. Subsequently the availability of tools and toolsets can be investigated and the causes of inefficiency identified. These relationships are based on

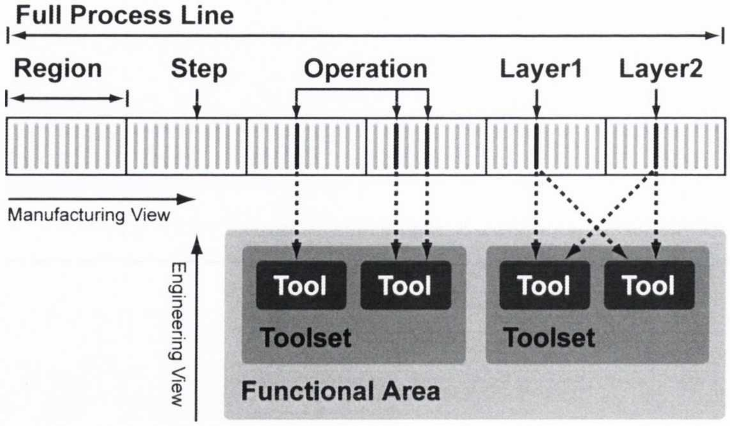


Figure 7.4: Alternative views associated with Manufacturing OC

physical and functional divisions of the fab and as such they are event-independent. They identify how and why the system is designed in the way it is, but they do not describe when certain activities need to be carried out or by whom. This artefact was presented to an operations manager who confirmed that it provided a reasonable model of operations production constraints.

7.3.2 Models addressing DC Themes

The ADS describes the fab in terms of its structural invariants but these relate to abstract goals and do not communicate the actual practices involved in operations control. As this project aims to support changes to work practice it is necessary to describe how OC happens. While the ADS describes core functionality in the fab, the social structures, cognitive artifacts and physical layout are work system resources that support functional activity. These present themes for investigating distributed cognition in the fab. By examining the configuration of these resources within the 200mm fab a clearer understanding of work practice can be attained. This provides a better position for understanding the impact of changes on the functionality of 300mm wafer production. While DC analysis usually involves a detailed observational study, many large enterprises have existing resources such as organisation charts, site plans and training manuals that can reduce the initial overhead associated with ethnographic research. Examples of each of these artefacts were sourced and a documentation analysis (Hoffman, 2005) was initially used to generate models of DC in the fab.

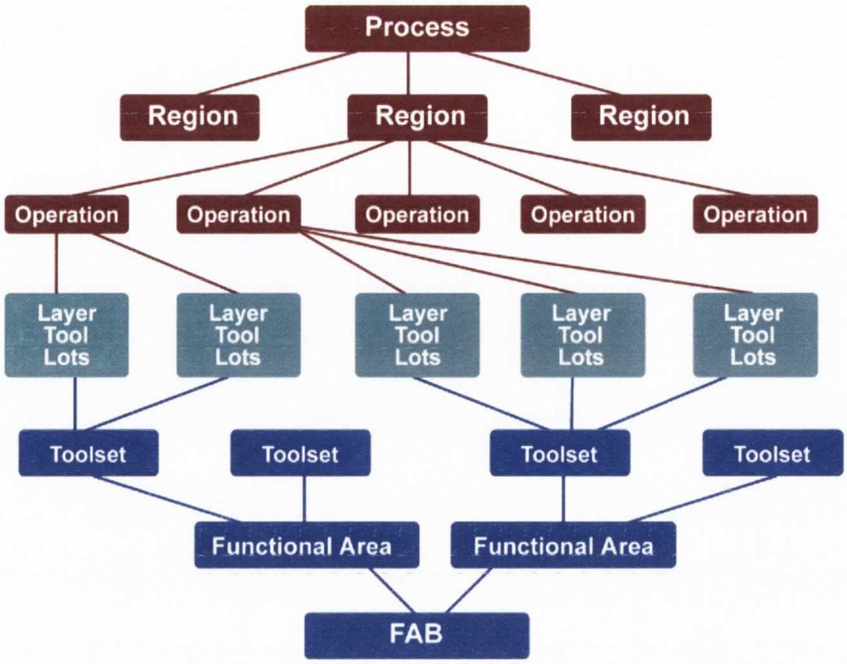


Figure 7.5: An Abstraction Lattice

	Process	Region	Operation	Layer
Functional Purpose	Manufacture Efficiently Production Rate			
Abstract Function		Maximise Speed Region Rate		
Generalised Function			Maximise Speed Operation Rate	
Physical Function				Maximise Speed Layer Rate
Physical Form				Lot ID, Tool ID Layer ID
Physical Function				Up to production / Offline
Generalised Function			Maintain Stability % Availability	
Abstract Function		Maintain Stability % Availability		
Functional Purpose	Maintain Stability % Availability			
	Fab	Functional Area	Toolset	Tool

Figure 7.6: ADS of Fab Operations Control Constraints

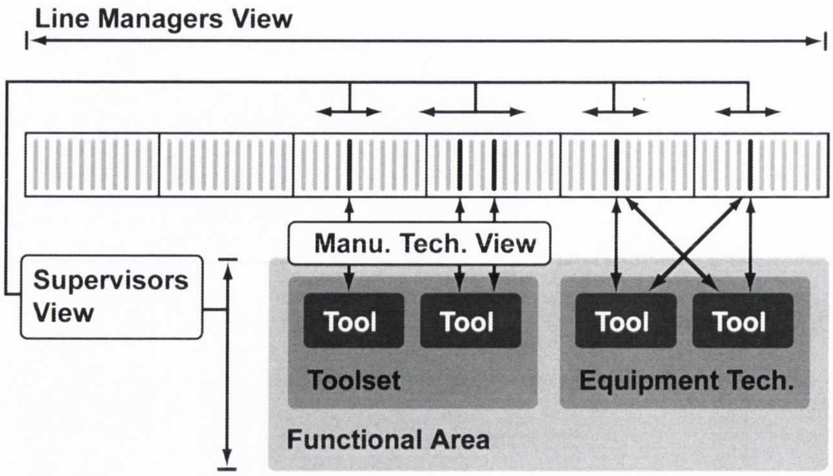


Figure 7.7: 200mm Social Organisation and Views

7.3.2.1 Social Organisation

Inspection of organisation charts reveal that the social structures in the 200mm fab are largely defined by the two views outlined in figure 7.4. Within the manufacturing department workers at three different management levels are responsible for different aspects of control. A line manager monitors the entire production line and takes overall responsibility for operations control. They plan strategies for successfully meeting delivery dates and respond to problems when they arise. Manufacturing supervisors are responsible for functional areas and sections of the line surrounding their associated toolsets. They monitor the upstream and downstream inventory and assess WIP levels in terms of the ability of their toolsets to meet processing demands. Finally, manufacturing technicians are responsible for loading and unloading WIP into process tools. Their view of the line is restricted to the layers associated with their tools. They also identify problems at their tools and can alert their supervisor to these issues. From the engineering perspective, the observations made by technicians may warrant further inspection or maintenance. Specialised equipment technicians are responsible for carrying out maintenance on specific toolsets. As well as dealing with the PCS health issue covered in the last chapter, senior process engineers monitor availability across functional areas and can request maintenance on tools. Figure 7.7 shows the structure of the social organisation and the views that the different roles have of the system.

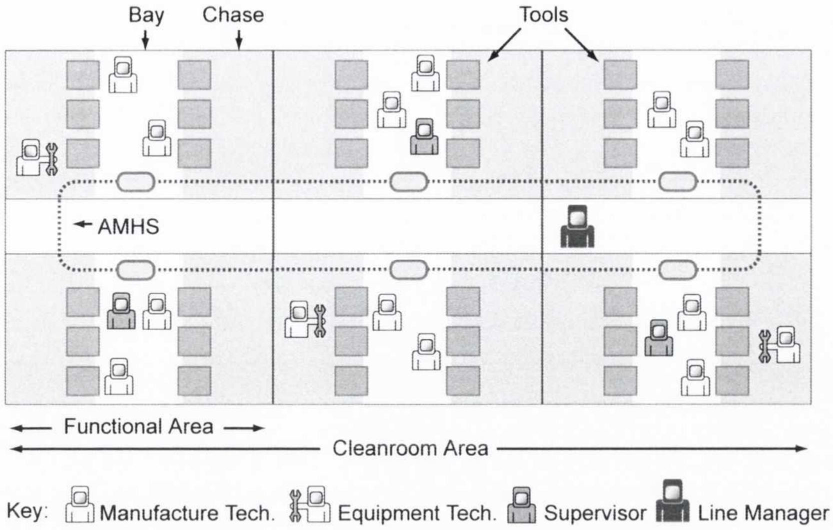


Figure 7.8: Schematic of 200mm Physical Layout

7.3.2.2 Designed Artifacts

In the 200mm fab the information systems used to support manufacturing OC have also been developed around the manufacturing and engineering views. The main application for production management has two sections; WIP and equipment. Line managers and supervisors monitor performance indicators associated with the low-level information units (i.e. operations or toolsets) in either section. The data is represented using a propositional display format by way of structured data tables. As this data is continuously updated, this is the primary information source for understanding the system state. However, the process features hundreds of individual process steps and toolsets and multiple performance values are provided for each. As a result of this, the data tables are very long, they contain thousands of data values and locating and integrating information is very cumbersome. Some filtering functionality has been provided to allow users to focus on particular aspects of the system but line managers rely strongly on tacit knowledge and subjective cognitive strategies for transforming this low-level data into higher-level information.

7.3.2.3 Physical Layout

As was discussed in chapter 5, semiconductor manufacturing is extremely sensitive to environmental factors and much of the activity takes place in a controlled cleanroom environment. The fab is divided up by functional areas into a number of different cleanrooms. Manufacturing technicians are located in bays, a part of the cleanroom where they move between their tools loading wafers for processing. Most of the tool is located in the chase, a more controlled section of the cleanroom where the equipment technicians carry out maintenance. The wafers are transported between the functional areas using an interbay Automated Material Handling System (AMHS). Supervisors are located in their functional areas where they move between technicians instructing them on operational goals and receiving information about tool performance. The line manager moves between areas getting updates on the system state and responding to production issues. A diagram of this layout is provided in figure 7.8.

7.3.2.4 DC Configuration

The structural invariants of the process, identified in the work domain model, have influenced the way in which the social organisation, designed artifacts and physical layout have been developed. Studying these themes shows how the problem space of semiconductor manufacturing has been distributed around the work system. These structures support the functionality outlined in the work domain model, but these same structures must also support the intentional goals of manufacturing, such as managing orders, maintaining stable process flow and minimising scrap. While the analysis of DC themes places the information levels derived from the work domain models into a practical context, it does not identify how the intentional goals associated with the system are achieved. In order to investigate these further it is necessary to investigate the management principles that guide decision-making in operations control.

7.3.3 Intentional Goal Model

The Theory of Constraints (TOC) (Goldratt and Cox, 1986) is a management theory that is commonly used in relation to production line management. The theory states that the volume and rate of any process is limited at some point by a constraint and that effective constraint management is imperative for process improvement. A constraint in this sense refers to a situation where the workload exceeds the capacity of the resources available to process it. In the following sections the term *constraint* is replaced with *bottleneck* in order to differentiate it from the functional constraints described in the work domain model. Five key steps are proposed when implementing a TOC approach.

1. Bottleneck identification

2. Bottleneck exploitation. (i.e. achieving stability and maximising utilisation)

3. The limiting of all other processes to the capacity of the bottleneck

4. Elevating the bottleneck to a higher capacity

5. Repeating the process to identify the next bottleneck

This approach underlies the intentional goals relating to manufacturing, but the complexity of the fab introduces a number of additional problems. TOC relates to a linear process, but the process re-entry and tool re-use featured in semiconductor manufacturing means that low availability with a specific toolset could result in multiple bottlenecks along the line (see fig 7.4). Another issue is that individual lots can have different levels of priority depending on the orders they belong to. This results in bottlenecks having different levels of severity. The constraints in the fab process line are also highly dynamic. Toolset availability can increase or decrease a number of times over each shift making bottleneck identification more difficult to achieve than with more stable processes. Finally the focus of TOC is to maximise the rate of a process. While this is an ongoing challenge for manufacturing engineering in the fab, it is not the guiding principle for line management where predictable delivery time is more important. The theory of constraints is only one of the strategies involved in line management but bottleneck management provides a useful illustration of how a goal model can be developed that describes intentional behaviour in a sociotechnical system.

7.3.3.1 Constructing the Intentional Goal Model

The conflicting manufacturing and engineering objectives associated with line management, mean that decisions at various levels do not have prescribed outcomes but involve balancing sub-goals in order to *satisfice*[2] higher-level goals. Much of the material to this point had been gathered from reviewing system documentation and completing training courses over a period of several weeks. However, analysis of line management strategies required access to real users involved in their work. The previous stages identify the technicians, supervisors and line managers as the main roles associated with operational control and these become the focus of an ethnographic study. One-hour contextual enquiry interviews (Beyer and Holtzblatt, 1998) were initially carried out with one subject in each role. Following this, 4 hours of observational study was conducted on manufacturing technicians (30mins), supervisors (30 mins) and line manager (3hrs) over 6 separate sessions. The observations involve application walkthroughs with their current tools using a think-aloud protocol. The interviews and observations were recorded and screen grabs were taken during the observation sessions as video recording was not permitted in the ROCC. These recordings were reviewed and analysed to identify goals and dependencies. The analysis results were compiled into an intentional goal model shown in figure 7.9 and are described below.

7.3.3.2 First Level Goals

Interviews and observation of the line managers in the ROCC revealed three major influences on management practice; the need to meet delivery dates, the availability of tools to process WIP and the distribution of WIP across the line. Meeting delivery dates is a hard constraint as late deliveries are unacceptable. The availability of tools and distribution of WIP are soft constraints as they influence each other and can be manipulated to control the rate that certain products move through the line. The TOC step 3 indicates that the ideal amount of WIP to run in the line is defined by the toolset with the smallest capacity; this is a known bottleneck. Outside of this, unexpected events (faults etc.) generate dynamic bottlenecks which slow inventory movement. In order to plan deliveries it is essential to know how long it takes to complete manufacturing. A set average speed must be maintained if production planning is to be successful. Based on these observations the top level goals are defined as; *satisfying customers, maintaining the speed of the line and*

[2]Satisficing is a decision-making strategy that aims to achieve an adequate rather than an optimal solution. The word was originally coined by Herbert Simon to describe a human approach to solving complex problems Simon (1955).

maintaining the spread of the inventory. These abstract high level goals were expanded in subsequent interviews to reveal sub-goals that are more closely related to identifiable components in the system.

7.3.3.3 Second & Third Level Sub-Goals

The next two levels in the goal model relate to measurements of success and strategies for achieving them. Maintaining the overall speed of the WIP involves two sub-goals. The first is to maintain a *consistent pace* across the line. The second is to minimise the number of dynamic bottlenecks in the line. At a given time the pace of processing may be high, but a large number of bottlenecks in the line could cause this to drop suddenly. *Minimising bottlenecks* can be achieved by insuring a *high level of tool availability* and *consistent levels of inventory* at the toolsets. These are the responsibilities of the manufacturing supervisor. These sub-goals are further investigated to specify how they are achieved at the next level down.

7.3.3.4 Fourth level Sub-Goals

The lowest level describes sub-goals that define the actions carried out at the tools. For example achieving high tool availability involves *regular tool maintenance*. Similarly, maintaining consistent inventory requires achieving *high usage of the tools*. We can think of these sub-goals as *operational rules*. Sometimes goals at this level may be conflicting, for instance, having a high tool usage needs to be balanced against the need to carry out tool maintenance. The operational rules for a functional area are sufficient for informing the manufacturing technician's actions under normal conditions, but some conflicts will require resolution by the supervisor based on higher level manufacturing goals.

7.3.3.5 Goals & Roles

The intentional goal model covers all the levels of operational control from high-level line management down to the rules that define when lots are loaded into tools. During the interviews the distribution of this goal structure across the various workers and their use of the designed artifacts to achieve these goals were revealed (see figure 7.9). The line manager monitors inventory across the line to build up an understanding of inventory

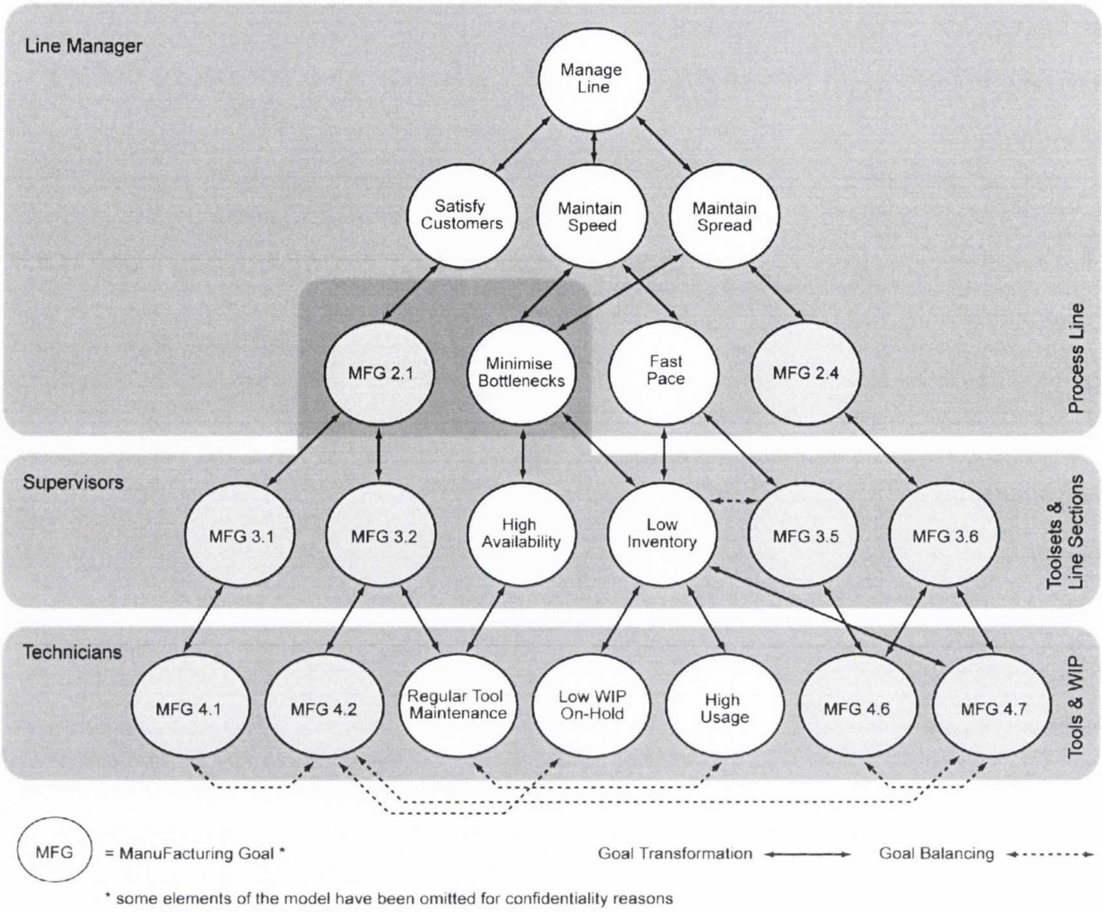


Figure 7.9: The Intentional Goal Model

position and WIP speed in order to define strategies. Manufacturing supervisors have a more specialised role and view the inventory as it relates to their functional areas. They filter the manufacturing and engineering information and combine these data sources to gain an understanding of the effect of toolset availability on the production goals. They play a critical role in bottleneck prediction and often work with the line manager devising strategies for dealing with these issues. They are also responsible for communicating the resulting operational strategies to their technicians to ensure that the final actions correlate with the overall line management strategy.

7.3.4 Modelling Cognitive Strategies

While the intentional goal model showed *why* workers carried out tasks it did not express *how* they carried them out. Control task analysis can be used to trace causal reasoning about a systems performance. The premise of this approach is that users observe and integrate data from a set of components to gain an understanding of the current state of a particular subsystem (e.g. toolset). If ambiguity exists, the current state is evaluated against the higher-level goals of the parent subsystem (e.g. functional area). This movement between states of knowledge is generally shown to correspond with the relationship between the different levels of functional abstraction revealed in the work domain model. This technique allows an analyst to identify information requirements for causal reasoning when dealing with physical systems.

The difficulty with applying this approach to operations control is that the structures described in the work domain model do not fully define the decision-making associated with line management. The intentional goal model provides an alternative abstraction hierarchy that incorporates information from the work domain model and can be used as the basis of the analysis. However a further difficulty is that the intentional goals associated with OC are distributed across a large management structure. For example, the line manager's goal of minimising bottlenecks involves achieving high availability and low inventory in each toolset; but toolset control under the original 200mm fab configuration is the responsibility of the various supervisors. Rather than developing an individual decision ladder for the whole control process, decision ladders are generated at both supervisor and line manager level. This allows *information processing tasks* to be identified for each role. It also reveals where *information transfers* occur and what *state of knowledge* is required at these points.

Supervisors maintain a high level of awareness about both the tools and the inventory in their functional areas making it possible for them to identify potential bottlenecks. Figure 7.10 shows a decision ladder for the goal of bottleneck identification. System data from technicians, equipment engineers and maintenance plans are used together to reveal the current and future status of individual tools. The supervisor integrates this data to understand the capacity of a toolset. When a low capacity is identified this acts as an alert. Following this, the supervisor observes WIP in the line through the same information system as the line manager (described in section 7.3.2.2). However, as they are already focused on a particular toolset, they only need to observe inventory levels at a few specific points allowing them to identify a potential bottleneck at an operation. Despite this knowledge the supervisor cannot decide the correct response as they do not

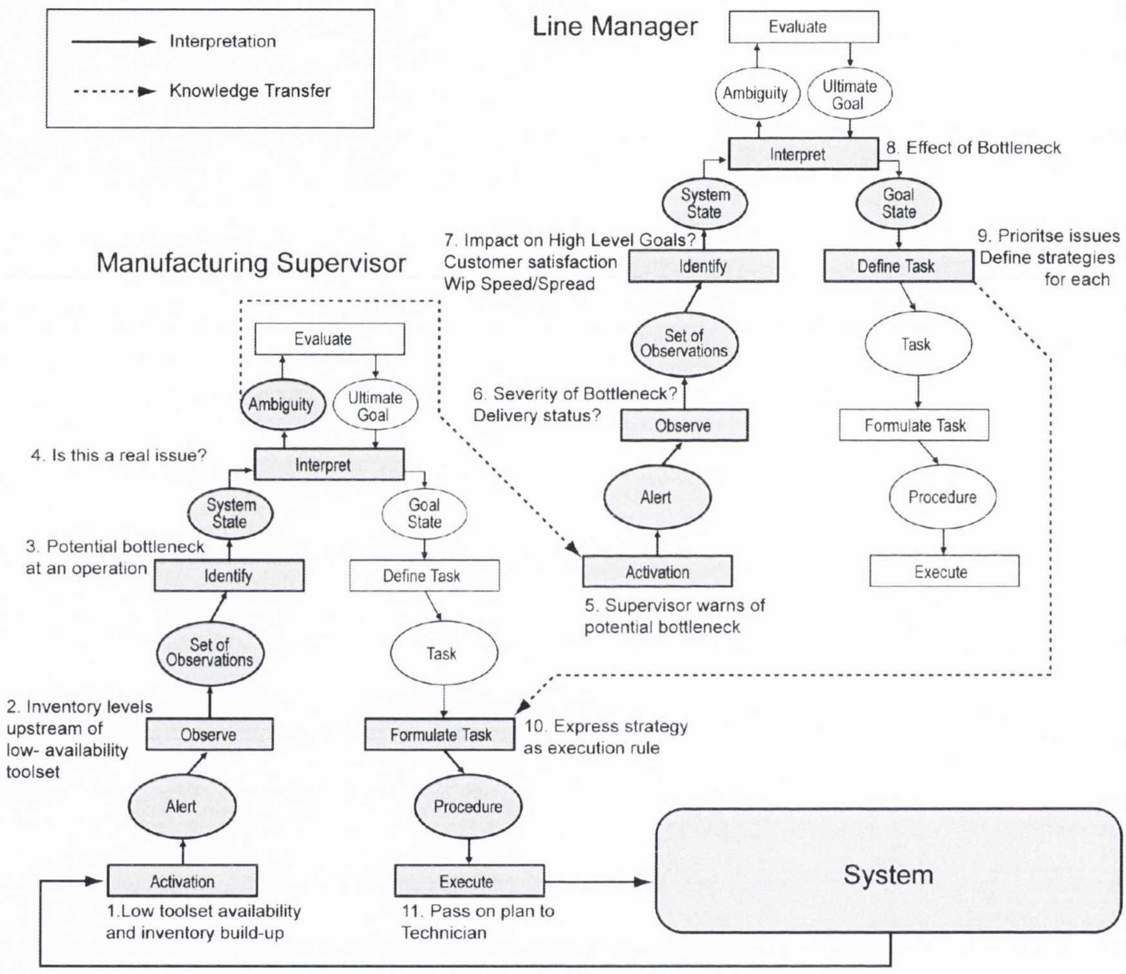


Figure 7.10: Decision Ladders for 200mm bottleneck management strategy

attend to performance across the entire line (as shown in figure 7.4). The evaluation of what impact this potential bottleneck will have on the manufacturing goals is passed up to the line manager.

Line Managers have their own strategies for bottleneck identification. They monitor the line for any changes in inventory that may impact their delivery plans. However this exception management approach can only deal with problems after they occur, i.e. when inventory has already built up, as such they rely strongly on the warnings passed on by the supervisors. The line manager's decision ladder for bottleneck management shows that the level of alert corresponds to the notification of an issue by the supervisor. The manager uses this notification to observe the operation and the inventory levels surrounding it. Other bottlenecks in the line are checked along with their relative positions in order to identify possible conflicts between WIP management decisions. The relationship between these potential bottlenecks and delivery commitments will also be considered. This allows the manager to identify the critical bottlenecks, to rank these in order of importance and to minimise their impact on the overall schedule. This analysis reveals how the highest level of the supervisor's decision ladder feeds into the lowest level of the managers.

7.4 Design Problem Stating

The design problem structuring activity has generated a range of models that describe different aspects of OC in the 200mm fab. As with the previous project, specific information requirements need to be extracted from these models and compiled into a unified artifact before a decision support application can be designed. However as this project deals with an evolving system, it is first necessary to identify how the changes affect work practices and what implications this has in terms of decision-making. In this section models of the new 300mm fab work configuration are developed. These are used to identify new approaches to OC and to inform the generation of new system variables that support higher-levels of control. These are then integrated with information requirements derived from the other models to complete the problem-stating phase.

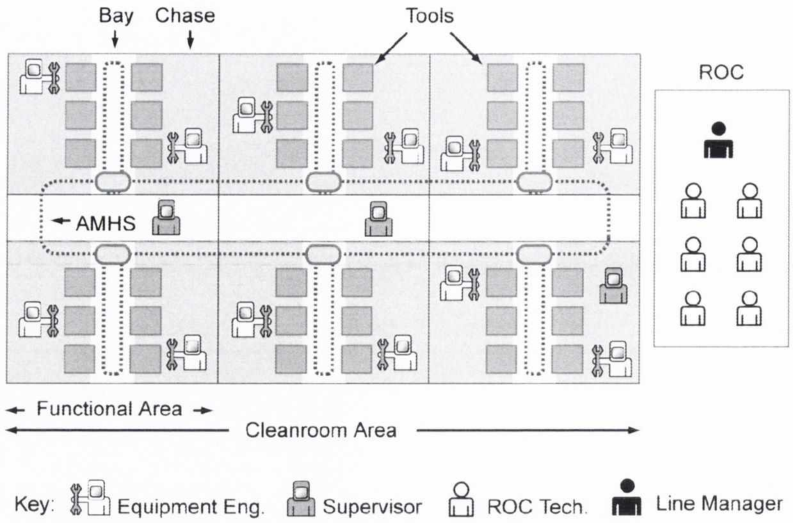


Figure 7.11: 300mm Fab Physical Layout

7.4.1 Identifying the Impact of Change

Both the ADS and the Intentional Goal Model can be described as *formative* models of system functionality. While both describe the system in terms of goals and means-ends relationships, their goal structures are independent of any particular work configuration. On the other hand the models of DC themes demonstrate how the social, physical and information systems of a 200mm fab are configured to support its functionality. While the move to 300mm manufacturing has resulted in a number of changes, the underlying process remains the same and the abstract goal structures remain valid. However the introduction of a more pervasive AMHS and the move to a remote operations model changes the way in which manufacturing is controlled. Here new models of the DC themes are generated around the new work configuration.

7.4.1.1 Physical Layout

One of the most obvious changes has been to the physical design of the new fab. The automated delivery of wafers to tools has removed the need for teams of technicians to load WIP into the tools. This has resulted in two major changes, firstly the physical dimensions of the bays have been reduced and secondly a new Remote Operations Control Centre (ROCC) now exists outside of the cleanroom environment (fig. 7.11).

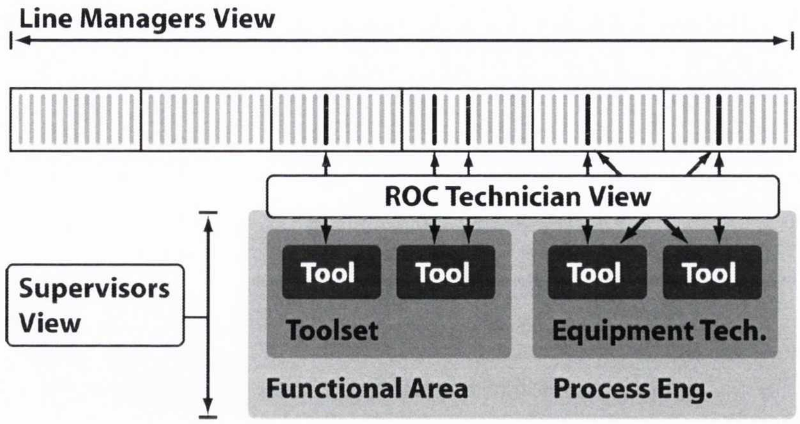


Figure 7.12: 300mm Social Organisation and System Views

7.4.1.2 Social Structures

The social structures that support fab management have also changed. While manufacturing technicians are no longer needed in the bays, the higher levels of automation increase the engineering challenges in the fab. As a result many technicians have moved into equipment maintenance roles. The role of manufacturing supervisors has also changed. As they must manage these larger maintenance teams in the fab, their view of the fab moves away from a manufacturing and closer to an engineering perspective. Their continuing location in the fab further reduces their role in operations control. The ROCC is manned by a line manager and a small team of specialised ROCC technicians. These technicians can now manage multiple tools concurrently from their desks. In fact, many technicians now control the tools for an entire functional area. These changes alter the social hierarchy and in turn the structure of the distributed cognitive system described in our intentional goal model (fig. 7.12). The OC sub-goals that were previously carried out by the supervisors in relation to manufacturing now need to be completed by either the line manager or the technicians (fig. 7.13).

7.4.1.3 Designed Artifacts

The move to 300mm manufacturing has required changes to the manufacturing execution system used in the fab (Mouli and Srinivasan, 2004), but while the information systems architecture is radically different, the information content and interface for operations control remains roughly the same. Despite the physical and social changes that have

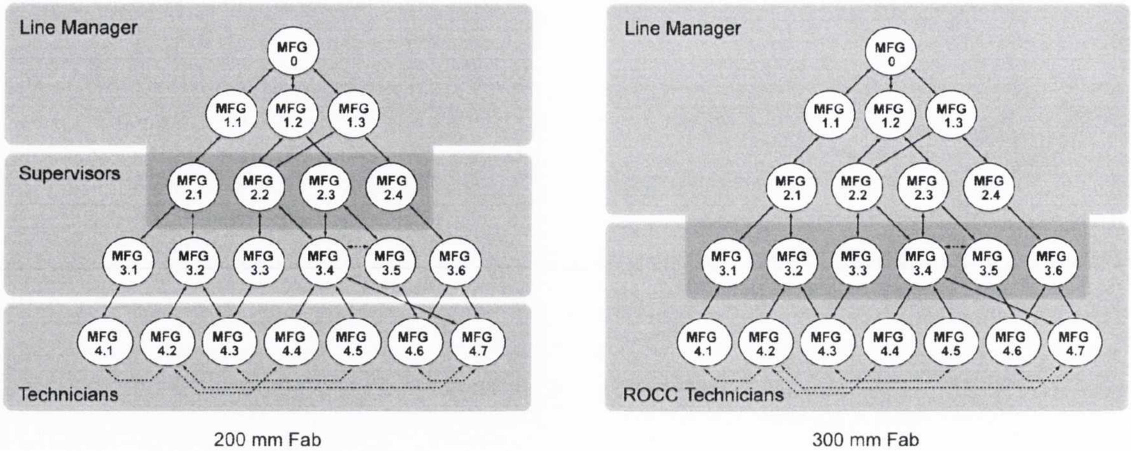


Figure 7.13: Changes in Goal Distribution

occurred, the underlying manufacturing process involves the same structures; WIP, tools, operations etc. and these continue to accurately describe the system state. As a result the tabular, propositional format in which system data is presented is unaltered. However, given the radical changes in terms of information flows and work rate it is questionable whether this representational format of the data is adequate.

While supervisors no longer play a central role in operations control, the strategies model identified their importance for bottleneck management in the original configuration. With their expert knowledge of functional areas they acted as a bridge between engineering and manufacturing views of the line. While they maintained an awareness of the availability of their toolsets they are also aware of the WIP in terms of the impact that tool availability will have on the production goals. In the absence of supervisors, the line manager must access larger volumes of data at lower levels of abstraction and carry out the same low-level calculations to identify constraints.

7.4.1.4 Effects of Change on System Functionality

In the original work system the need for technicians to load tools placed a physical constraint on the manufacturing process. The automation of this activity has increased the production rate of work but it has also changed the work configuration involved in operations control. This change places greater cognitive demands on the remaining system controllers. While the supervisors were responsible for dozens of tools, the line manger must now maintain an awareness of hundreds of them. Furthermore, the manager must switch between a linear, ordinal view of the process flow and a discrete, categorical model of tools, toolsets and functional areas. The supervisor's focused view and expert knowledge of their functional areas allowed them to use tacit knowledge and mental arithmetic to identify bottlenecks. However these strategies are unsuitable when the full line must be considered. These changes in the fab mark a transition in the nature of manufacturing constraints. The manual loading of tools created a physical constraint that limited production rates. The AMHS overcomes this but the ability to identify and respond to issues as they arise places a cognitive constraint on system control and performance. To further improve fab performance it is necessary to develop cognitive artifacts that support system observability and remove the need for low-level information processing.

7.4.2 Generating Higher-Level Variables

The design of the current information system can be described as a "design for availability" approach (Woods, 1995). Low-level system data is presented in a tabular format in order to support the widest range of tasks. There are a number of reasons why this has been predominant. Firstly, engineers tend to be expert users of spreadsheet applications and find it relatively easy to navigate and extract information from tabular representations. Secondly, from a software engineering perspective tabular representations are much simpler to develop than interactive visual interfaces. Thirdly, in terms of data flexibility the raw data allows users to generate custom reports on demand. These characteristics were advantageous when line management was supported by a large team; however in the absence of supervisors the line manager must now scan the large tables of alphanumeric data in order to understand the system state. As a next step in the problem-stating phase, the information processing activities are re-examined and new variables that can indicate system performance at higher levels of abstraction are generated. These should reduce the need for low-level cognitive operations such as data selection and integration.

7.4.2.1 The Bottleneck Variables

While the process of line management involves a large number of goals, this design process will continue to focus on the goal of minimising bottlenecks. The intentional goal model (fig. 7.9) revealed that the associated sub-goals are to keep inventory low and to keep availability high. A bottleneck occurs where the inventory is greater than the ability of the tools to process it (their capacity) resulting in inventory building up. Tools have specific processing rates that indicate how many operations they can complete over a shift, so toolset capacity can be calculated by multiplying the number of available tools by their processing rate. The difficulty discovered during work domain modelling is that many operations consist of multiple layers and WIP at these layers can be assigned to tools arbitrarily (see fig. 5.3 in chapter 5). This leads to the problem of being able to see inventory at a process step but not being able to work out the capacity at this step due to the arbitrary number of tools working on that layer at a given time. Traditionally, the manufacturing supervisor maintained awareness of their toolsets performance throughout the shift. Their knowledge of the current tool states as well as past performance enabled them to recognise potential bottlenecks. In the absence of supervisors a single measurable value needs to be generated that can inform the line manager about potential bottlenecks in the line, essentially providing the alert system state in their decision ladder (fig. 7.10).

Based on the available data it is possible to generate two different types of bottleneck values. A *toolset bottleneck* occurs where the capacity of a toolset is less than the sum of the inventory across its associated layers. This is a serious issue as it means that at least one layer will have to run at sub-optimal performance. A process step or *layer bottleneck* occurs where the total capacity of a toolset is divided evenly between its associated layers and the inventory at any layer exceeds its proportion of the capacity. This is not a true bottleneck as the total toolset capacity can deal with the inventory; however it does require tool reassignment and careful management. This analysis has generated a range of new variables shown in table 7.1. *Toolset inventory* extends information from the manufacturing view into the engineering view while *layer capacity* extends information from the engineering view into the manufacturing view. These can be used to generate a *residual capacity* figure (Capacity – Inventory) for both toolsets and layers. These values

Toolsets	Layers
ToolsetCapacity = No.ToolsxProcessingRate	LayerCapacity = ToolsetCapacity/No.Layers
ToolsetInventory = ∑Inventory@Layers	LayerInventory = Inventory@Layers
ResidualCapacity(toolset) = ToolsetCapacity−ToolsetInventory	ResidualCapacity(layer) = LayerCapacity−LayerInventory
ToolsetBottleneck = ToolsetCapacity<ToolsetInventory	LayerBottleneck= LayerCapacity<LayerInventory

Table 7.1: Bottleneck Variables

allow potential bottlenecks to be identified without the need for mental arithmetic or information foraging.

7.4.3 The Information Requirements Matrix

Now that the higher-level variables have been generated, the various information requirements revealed during the problem-structuring phase are compiled into an information requirements matrix. The same four general categories used in the previous project are applied; abstraction hierarchies, information requirement, interaction requirement and notes. Again, multiple abstraction hierarchies have been generated that can each describe the system from a different perspective. The first column presents the original CWA levels of functional abstraction, the second describes the levels of the intentional goal hierarchy while the third and fourth columns show levels associated with the two structural hierarchies. The initial challenge to constructing the matrix, is deciding how the various abstraction hierarchies can be combined to generate a system image. The intentional hierarchy is crucial in achieving this as it identifies the system values that are used to support decision-making. These values come from both manufacturing and engineering perspectives so the intentional goals provide a primary hierarchy that the other two can be aligned to. The information requirement category is subdivided into display element, description and data scale. Certain requirements relate to views on the data rather than individual values so an additional 'display level' column is used to structure these requirements in terms of Woods' graphic display hierarchy (see table 3.4 on page 70). The last two categories of interaction requirement and notes are used in a similar manner as the previous project.

Table 7.2 presents an extract from the IR matrix for the proposed ROCC visual decision support system. A number of fields lack specific information and instead use the terms confidential or undefined. There are two reasons for this. With regard to confidentiality, some of the values used in this system relate to aspects of fab design or work processes that are commercially sensitive and have been labelled 'confidential'. Despite this, the information requirements in table 7.2 should provide enough context to demonstrate how design problem stating and problem solving has been carried out. In relation to the term 'undefined'; this has been used in relation to the interaction requirements of some of the lower-level information. As the IR matrix is generated before visual design occurs, some of the details relating to interaction have not yet been developed. These will be described in more detail during the problem solving phase. A brief description of the information requirements in terms of their abstraction levels is provided below.

Functional Purpose: During observational studies, production rate was constantly referenced as a high-level indicator of overall fab performance. As observation of production rate provides feedback on line management decision, this is placed at the highest level of abstraction and is constantly present within the application workspace.

Abstract Function: To gain a better understanding of the system state that produces a particular production rate, it is necessary to describe the system from a number of perspectives. Four views are defined in relation to the mid-level goals associated with line management. These include minimising bottlenecks and keeping pace consistent. As these goals involve balancing sub-goals that relate to multiple toolsets and layers, they cannot be expressed using summary values. However they may be used to define views of system data including a bottleneck view and an inventory view.

Generalised Function: The requirements at this level relates to information associated with these abstract function views. Table 7.2 places the two bottleneck values associated with layers and toolsets at this level.

Physical Function: This information is the data from system components level. This is used to generate requirements at the generalised function level and supports diagnosis of issues. Inventory at layer, inventory at toolset, availability etc. provide a better understanding of bottleneck severity and causes.

Physical Form: This describes the structural relationships derived from the work domain model. These are used to provide a context for understanding all of the values presented above.

7.5 Design Problem Solving

The IR Matrix reveals a complex multilayered set of information requirements. The visual design process will again move through preliminary, refined and detailed phases of conceptualisation in order to approach a final design solution. The EID principles continue to provide high-level design guidance but will again be supplemented with additional visual encoding techniques. The Visual design phase was carried out over a three month period. During this time a number of different design methods were used. An iterative design approach was taken and concepts were generated and reworked throughout the process. Eight three- hour participatory design sessions were conducted with the operations manager. During these sessions concepts were reviewed and new sketches were proposed. Access to the line manager was quite restricted as their attention had to be focussed on current activity in the factory. Despite this further access to the ROCC was granted for four two-hour visits. Concepts were presented to the line manager during lulls in activity. Feedback was provided and adjustments were made to the sketches on the spot. Finally two major design reviews were conducted with the line manager, factory manager and operations manager in attendance. Each of these individuals can be considered subject matter experts with over 10 years experience in the semiconductor manufacturing domain. The design reviews occurred at the end of the refined and detailed design phases and both were conducted as one-hour participatory design sessions where individuals were encouraged to annotate concepts and generate new sketches.

7.5.1 Preliminary Concept

A preliminary concept is developed in relation to the EID principle of presenting the information requirements in the form of an abstraction hierarchy to support knowledge-based behaviour. Although multiple abstraction hierarchies were identified during the

Abstraction Hierarchies				Information Requirement				Interaction Requirement			Notes
Functional Hierarchy	Intentional Goal Hierarchy	Manufacturing Hierarchy	Engineering Hierarchy	Display Element	Description	Display Level	Data scale	Action	Navigation	No. of elements	Notes
Functional Purpose	MFG 0, 1.1-1.3	Process Line	Fab	**Production Rate**	Ratio of inventory to processing	Workspace	Q	Click on	Open Large Chart	1	Track high-level performance
Abstract Function	MFG 2.1-2.4	N/A	N/A	**Bottlenecks**	Related system data	View	N/A	Click on tab	Swap view	1	View Bottlenecks in production system
				Inventory	Related system data	View	N/A	Click on tab	Swap view	1	View Inventory in production system
				*** confidential**	Related system data	View	N/A	Click on tab	Swap view	1	* confidential
				*** confidential**	Related system data	View	N/A	Click on tab	Swap view	1	* confidential
Generalised Function	* for confidentiality reasons the information requirements below relate to the bottleneck view only										
	N/A	Operations	Toolset	**Toolset Residual Capacity**	Toolset bottlenecks in engineering view	Graphic Form	Q & N	Undefined	Undefined	* confidential	Used to view bottlenecks in context of equipment
	N/A	Layers	Tools	**Layer Residual Capacity**	Layer bottlenecks in line	Graphic Form	Q & O	Undefined	Undefined	* confidential	Used to view bottlenecks in context of WIP
Physical Function	MFG 3.1-3.6	Operations	Toolset	**Toolset Capacity**	No Tools x ProcessingRate	Graphic Fragment	Q	Undefined	Undefined	* confidential	Used to judge severity of bottleneck
		Operations	Toolset	**Toolset Inventory**	Sum of Inventory@Layers	Graphic Fragment	Q	Undefined	Undefined	* confidential	Used to judge severity of bottleneck
		Operations	Toolset	**No. Tools (total)**	Count	Graphic Fragment	Q	Undefined	Undefined	* confidential	Used to identify cause of bottleneck
		Operations	Toolset	**No. Tools (available)**	Count	Graphic Fragment	Q	Undefined	Undefined	* confidential	Used to identify cause of bottleneck
		Operations	Toolset	**Processing Rate**	Wafers processed per shift	Graphic Fragment	Q	Undefined	Undefined	* confidential	Used to judge performance
		Layers	Tools	**Layer Capacity**	ToolsetCapacity / No.Layers	Graphic Fragment	Q	Undefined	Undefined	* confidential	Used to manage bottleneck
		Layers	Tools	**Layer Inventory**	Inventory@Layer	Graphic Fragment	Q	Undefined	Undefined	* confidential	Used to manage bottleneck
		Layers	Tools	**No. Layers**	Count	Graphic Fragment	Q	Undefined	Undefined	* confidential	Used to identify cause of bottleneck
Physical Form	MFG 4.1-4.7	All structural relationships	N/A	**Manufacturing View**	Layers within context	Region/ Graphic Form	O & N	Undefined	Undefined	* confidential	Used to provide system image
		N/A	All structural relationships	**Engineering View**	Tools within context	Region/ Graphic Form	O & N	Undefined	Undefined	* confidential	Used to provide system image
		Layers	Tools	**Relationship between views**	Associate tools with layers	Undefined	N	Undefined	Undefined	* confidential	Used to provide system image

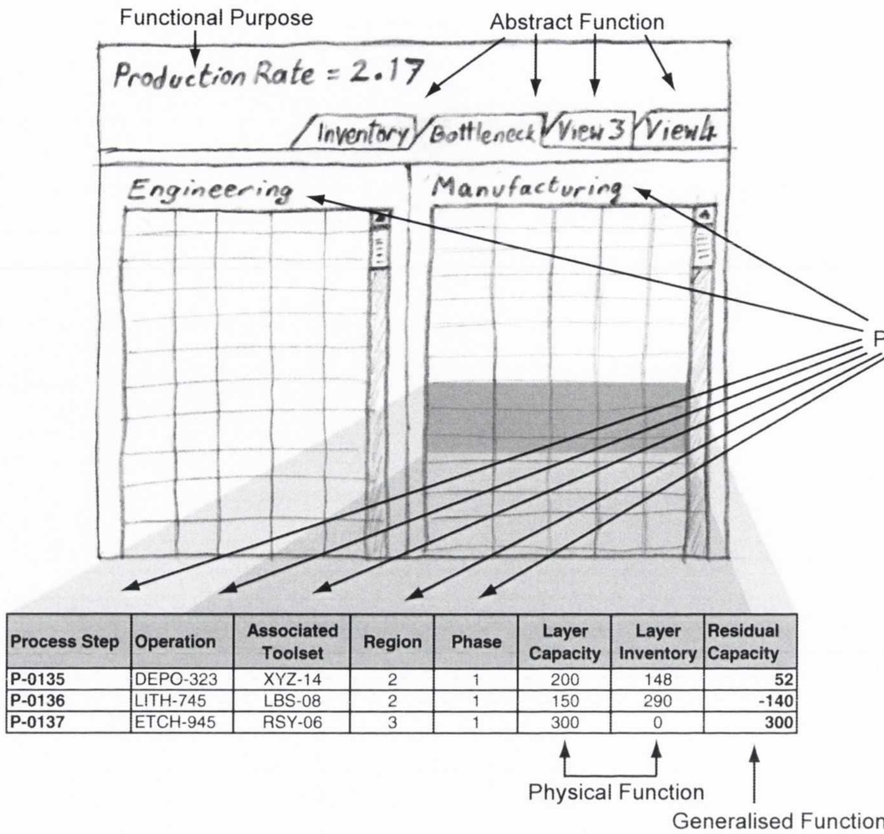
Table 7.2: The IR Matrix (bottleneck view data only)

problem-structuring phase, the intentional goal model was used as the predominant model for problem stating and provides a general structure for the composition of the interface.

An initial sketch (fig. 7.14) demonstrates how the various levels of abstraction can be represented in the interface. The application workspace supports the overall purpose of line management. As production rate provides a high-level measure of fab performance, it is presented in its own region at the top of the screen. Four alternative views of system information have been defined that support decisions at the abstract function level. These include the inventory and bottleneck views. Each view presents the relevant data for its associated goal from both engineering and manufacturing perspectives. The data relates to decisions at the generalised function and their associated physical function levels. For example, in the bottleneck view, the residual capacity at a layer is presented alongside lower level data such as layer inventory, layer capacity and number of tools. This allows a user to identify a bottleneck at a layer and diagnose its possible cause.

The lowest level of abstraction, physical form, presents a serious challenge for this display. As with the previous project, the physical configuration of fab components is not available as existing mimic diagrams. Furthermore, the re-entrant nature of the process would make such diagrams extremely difficult to interpret. The manufacturing and engineering perspectives were identified during work domain analysis as alternative structures associated with operations control. These provide conceptual models of the fab that define system functionality. The initial sketch continues to use the two data tables from the original application to present the manufacturing and engineering perspectives. As the four views separate out the relevant data for different higher-level goals, the number of columns in each of these tables has been reduced. This allows both tables to be presented alongside one another, making it easier to check the relationship between layer and toolset performance. For example a user may scroll through the manufacturing table in the bottleneck view examining layer bottlenecks. Once one has been identified, they can look across the row to find the associated toolset. This toolset can then be located in the engineering view and the severity of the problem can be assessed. In the initial sketch, the hierarchical relationships that describe the engineering and manufacturing structures are represented as variables within rows of the data tables.

The sketch in figure 7.14 shows how the abstraction hierarchy from the intentional goal



Process Step	Operation	Associated Toolset	Region	Phase	Layer Capacity	Layer Inventory	Residual Capacity
P-0135	DEPO-323	XYZ-14	2	1	200	148	52
P-0136	LITH-745	LBS-08	2	1	150	290	-140
P-0137	ETCH-945	RSY-06	3	1	300	0	300

Figure 7.14: Preliminary Sketch

model has been combined with hierarchies associated with the fab's structural models. By combining these models a system image has been generated whose representation provides better support for causal reasoning about system performance at different levels of abstraction. However, the majority of the system data remains in a propositional format. This means that the values must still be read in a serial manner making it difficult to compare and contrast information. Furthermore, the format does not provide an overview of the system state. The spatial inefficiency of alphanumeric characters means that a user must continue to scroll through hundreds of rows of data to inspect the entire line. These challenges will be tackled in the next phase of design conceptualisation.

7.5.2 Refined Concept

The refined concept uses the EID principle of mapping system constraints to visual cues to support rules-based behaviour. This principle has been applied to system data at various levels of abstraction.

7.5.2.1 Production Rate Trend

Production rate was identified as a high-level indicator of factory performance. While the effects of line management decisions are not instantly visible, they will have an impact on this figure within a number of hours so it can be used to provide high-level feedback. By representing production rate on an analogical trend chart rather than a propositional value, a line manager can monitor this figure over a shift and understand the overall impact of their strategies on the line (see fig 7.17). This is in-line with Wood's guidelines of control displays (Woods, 1995).

7.5.2.2 Encoding Bottleneck Values

As was mentioned above, the propositional format creates a number of problems for interpreting system information. The purpose of the bottleneck view is to identify, diagnose and respond to bottlenecks in the process line. To develop a graphic form that supports this, the bottleneck values need to be visually encoded into a structural representation of the fab. This process involves data scale analysis, visual scale matching and scale transformations.

Data Scale Analysis: Bottleneck detection involves both event-based information provided by the control task analyses (fig. 7.10) and the structural invariants provided by the work domain model (fig. 7.6). More specifically it involves the layers, residual capacity at layers, toolsets, residual capacity at toolsets, the manufacturing hierarchy, the engineering hierarchy and the relationship between these hierarchies (see table 7.3). Each of these data sources can be categorised as having a nominative, ordinal or quantitative scale. In some cases data may exist on two scales; for instance the process steps are nominative in that they each have a unique ID, but they are also ordinal in that they exist in a specific sequence. Table 7.3 shows all of the information requirements necessary for supporting bottleneck detection in the line as specified in the control task analysis (fig. 7.10) and the IR matrix (table 7.2).

Visual Scale Matching: To successfully match the data to a visual variable it is necessary to reveal the basic cognitive tasks being carried out with the data. These have been identified by the control task analysis. The line manager needs to detect bottlenecks and interpret the effect that they will have on the line. The tabular representation of manufacturing data in figure 7.14 provides a linear view of the process. Any graphic encoding will need to maintain this relationship between process steps, so the ordinal variable of spatial position provides one axis in a graphic form. Layer bottlenecks are identified based on residual capacity at the process steps. This is a quantitative variable and can be encoded through scale. Combining the position and scale variables results in a standard bar chart representation with positive values indicating excess capacity and negative values indicating layer bottlenecks (fig. 7.15a).

Data Transformation: This sketch was reviewed with line managers who confirmed that the representation successfully provided an overview of bottlenecks in the line, however two potential problems with this display were also identified. Firstly, values with low or zero excess are close to the bottleneck state and may also require attention, but this display format makes them difficult to detect. Secondly, while the display highlights layer bottlenecks, it does not show true toolset bottlenecks. As discussed in section 7.4.2 a layer bottleneck may appear even when a toolset has adequate capacity for its layers. To combat both these factors a new variable simply called 'bottleneck' is generated. The bottleneck variable exists on the ordinal scale and has three levels; no risk, risk and bottleneck. No risk indicates where both the process step and its associated toolset have excess capacity. Risk indicates where the toolset has excess capacity but the process step has less than 5% capacity. Bottleneck is an indication of true toolset or operation bottleneck. This ordinal variable is encoded using tonal values and figure 7.15b shows how this can be integrated into the representation.

7.5.2.3 Encoding Physical Structures

One major advantage of this format is that it provides an overview of performance across the line. However, it can only represent one variable out of the multiple columns of data in the original manufacturing data table. Consequently all of the information relating to the manufacturing structural hierarchy is lost. While the display provides an overview of bottlenecks it is difficult to identify the context of these problems in terms of regions and phases. Here the visualisation reference model is again applied to generate more graphical representations.

Information Requirement	Data Range	Data Scale
Process Steps	Step ID/ Line position	Nominative / Ordinal
Residual Capacity at Layer	+- 500 approx	Quantitative
Manufacturing Hierarchy	Line/Segment/Step	Nominative & Ordinal
Toolsets	Toolset ID	Nominative
Residual Capacity at Operation	+-1500 approx	Quantitative
Engineering/Manufacturing Relationship	Process steps -> Toolsets	Nominative
Bottleneck	No Risk >Risk >Bottleneck	Ordinal

Table 7.3: Information Requirements for Bottleneck Management

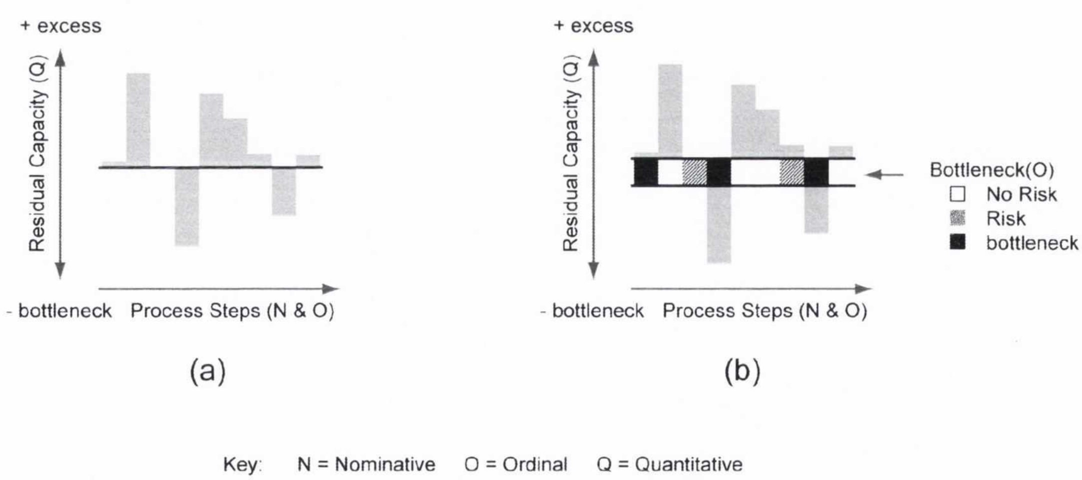


Figure 7.15: Design Rationale for Layer Bottleneck View

The structure of the manufacturing hierarchy is based on the division and subdivisions of the process flow into conceptual units namely process line, phases, regions and process steps. This is a part-whole hierarchy where child nodes have an ordinal association to their parental nodes. As the process flow is a linear view of the processing activity, the nodes within each level have an ordinal as well as a nominal relationship to each other. The cognitive tasks involve identifying and comparing quantitative values, such as inventory or bottlenecks, at and between process steps. With only three levels of depth the process hierarchy is relatively shallow and may be easily flattened into an expanding list representation similar to the windows explorer^{TM} interface (Waloszek, 2004). The ordinal relationship between nodes at all levels further supports this representation as the use of a spatial axis supports the interpretation of ordinal data relationships (fig. 7.16a). An alternative approach is to use a treemap (fig. 7.16b). Here higher-level information such as regional inventory becomes visible as an emergent feature derived from the enclosure of process step inventory encoded using scale. However, this format loses the ordinal relationship between the process steps that is necessary for comparison and for understanding the sequence of the process flow. By combining the enclosure technique with a list representation, the resulting display allows for comparisons at a particular level and between levels of abstraction (fig. 7.16c & d). A concept review with the line manager revealed that, although the values at the process steps provided key information, the values at region and phase levels were less useful for bottleneck identification. Rather than providing values at these levels, the visual structures are aligned with the process step bar chart. The result is a display that highlights functional constraints within a familiar structural model of the system (fig. 7.17).

Toolset bottlenecks are represented in the manufacturing view in relation to process steps, however while this representation indicates the existence of a toolset bottleneck, it does not fully convey the severity of a problem. As toolset bottlenecks have a greater impact on production goals a manager may use the engineering view to inspect the line. Again the length of the data table makes it difficult to get an overview of toolset bottlenecks. Even if a graphical encoding of residual capacity was provided the large number of toolsets makes a global comparison of values very difficult. During a formal one-hour design review with the line, operations and factory managers the line manager

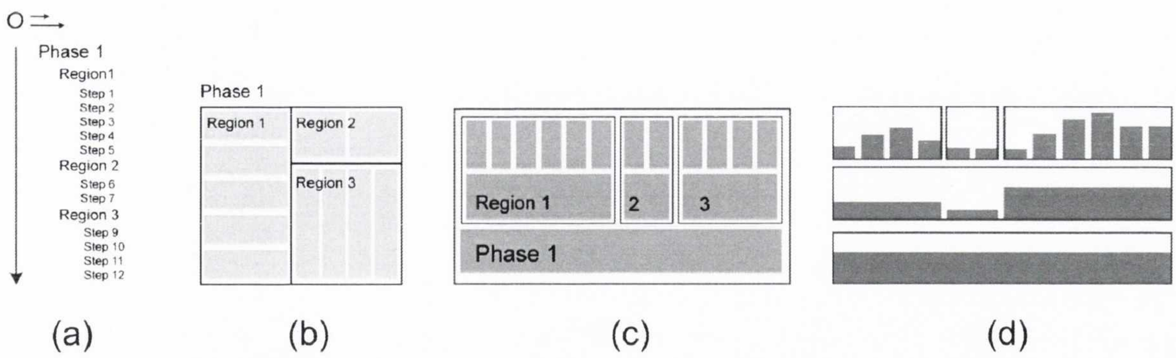


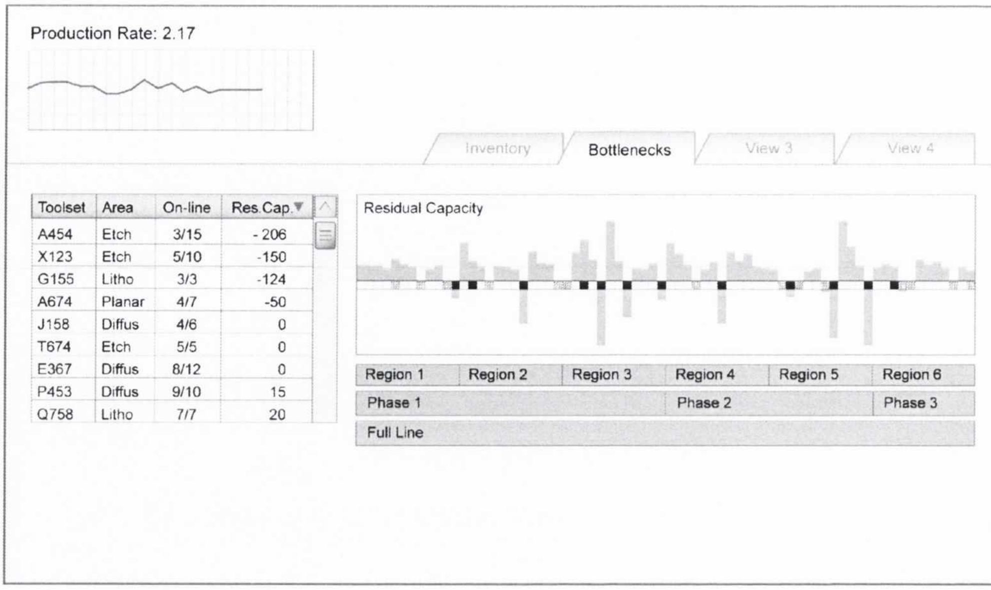
Figure 7.16: Design Rationale for Manufacturing Region

suggested that a sorted list of numeric values would provide a better means of identifying and resolving bottlenecks. This was generally agreed upon and incorporated into the design.

The refined design concept in figure 7.17 makes bottlenecks instantly visible from both the engineering and manufacturing perspectives. Despite this, the display is still subject to a number of limitations. Firstly, while it supports the identification of bottlenecks from either perspective it does not reveal the relationship between these views. As was demonstrated during the strategies analysis, one aspect of bottleneck management requires a user to identify the impact of engineering issues on manufacturing goals. This means that the relationship between the two views must be made explicit. Secondly, each view provides fairly limited information. The visual display used in the manufacturing region of the bottleneck view only communicates two variables while the original tabular display could display several. While this makes bottleneck identification easier, it makes the diagnosis of such issues more difficult.

7.5.3 Detailed Design

The detailed concept is developed in relation to the EID principle of supporting temporal control of the system through direct manipulation and making the representation isomorphic to the part-whole structure of movements. While the previous principles guided compositional aspects of the interface design, the detailed design relates to specific attributes and qualities of individual interface elements.



Toolset	Area	On-line	Res.Cap.
A454	Etch	3/15	- 206
X123	Etch	5/10	-150
G155	Litho	3/3	-124
A674	Planar	4/7	-50
J158	Diffus	4/6	0
T674	Etch	5/5	0
E367	Diffus	8/12	0
P453	Diffus	9/10	15
Q758	Litho	7/7	20

Figure 7.17: Refined Sketch of Bottleneck View

7.5.3.1 Direct Manipulation & Interaction

As was indicated earlier it can take a number of hours before the results of operational decisions become visible in the line and this time lag reduces the relevance of temporal control through direct manipulation. Despite this, interaction can still be used to overcome problems with the refined concept.

Understanding the relationship between the two structural displays plays an important role in managing the line. While these views are linked at the level of toolsets and process steps, their representations are visually separated in the display. Contextual highlighting is an interactive technique that can be used to overcome this problem. This technique highlights interface elements when another related element is rolled over or selected. By simultaneously highlighting related elements their perceptual proximity is increased and their functional association is made explicit. The application of this approach to the ROCC interface was inspired by the proximity compatibility principle. When a user rolls over a toolset in the engineering region of the display, its associated process steps are highlighted in the manufacturing region. Similarly rolling over a process step should highlight associated steps at different layers as well as the associated toolset. The sketch in figure 7.18 illustrates this, where toolset A454 is a three-layer tool that repeats an operation at three steps in the process. This technique allows a line manager to instantly

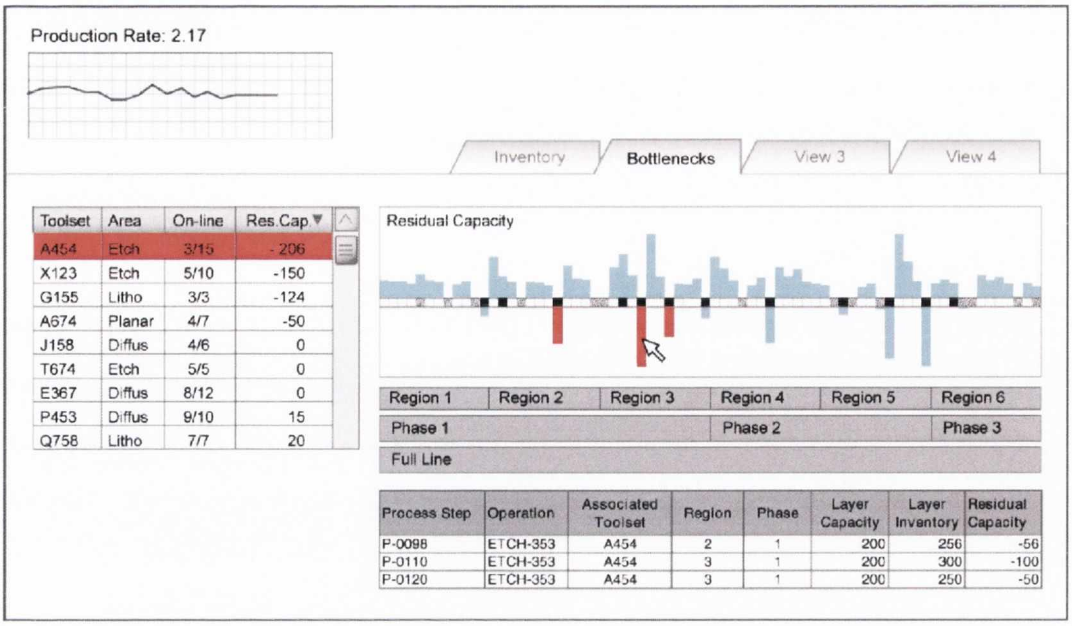


Figure 7.18: Contextual Highlighting & Process Step Details

observe the problematic steps in the line and to inspect processing performance around these steps. In doing so it removes the need for information searching, identification and integration. Interaction with visual components is also used to provide lower level details in relation to particular process steps. For example, while rolling over a process step triggers contextual highlighting, clicking on a process step or toolset provides all of the detailed information relating to the associated operation (see fig. 7.18). This overcomes the issue with layer-bottleneck diagnosis that was identified above.

7.5.3.2 Part-Whole Structure of Movements

The second aspect of this design principle relates to ensuring that the interface elements are positioned in a manner that matches user's movements during interaction. The representation shown in figure 7.18 allows a manager to understand process step and toolset issues within the context of the overall system and makes the relationship between engineering and manufacturing constraints explicit. However, when a graph using actual system data was developed, a potential problem with this display was identified. While the initial sketch was developed using a relatively small number of process steps, the actual process featured many hundreds of them. As a result the visual elements representing process steps became very narrow, potentially making interaction and selection quite difficult. Another design review of these representations was carried out to discuss this issue. This review was conducted in the ROCC with the line manager. During the review the manager demonstrated their current interaction technique for accessing the information. This suggested the supplemental interaction technique for overcoming the problem describe below.

Each shift begins with a line management handover where attention is drawn to potential problems and difficult toolsets. Following this, the line manager carries out a review of the full line. This allows them to build up awareness of the overall system state at the start of the shift. This awareness is updated in relation to the various events and management decisions made during the day; however a full line review is generally not repeated during the day, as it takes considerable time to complete. Once the line review is complete the manager focuses on front-end and back-end processing independently. Priority is given to the phase with the most issues. When attempting to resolve a specific issue in the line, for example a bottleneck at a process step, the manager focuses only on the region surrounding the bottleneck, inspecting the upstream and downstream performance.

This suggests that the structure of movements associated with line management involves a high-level overview followed by successive focusing on particular sections of the line. This is reflective of Schniederman's information visualisation mantra that suggests that all visualisations should support "overview, zoom and details on demand"(Shneiderman, 1996). This is achievable with the current design by transforming the representation of the process hierarchy into a navigation toolbar. By clicking on a particular region of the line, the display can zoom in to the corresponding level of detail. This use of graphical zooming is reflective of the part-whole structure of movement used during line management. As well as focusing the user's attention and making process steps more selectable, the increase in spatial dimensions allows further information to be graphically encoded. The utility of this is better explained in relation to the inventory view. In full-line mode the inventory view shows only the volume of WIP located at each step (fig. 7.19). However, this view is related to the abstract function of achieving a fast pace and some of the low-level goals associated with include minimising the volume of WIP on-hold in order and maximising utilisation (see fig. 7.9). If a process tool is suspected of erratic

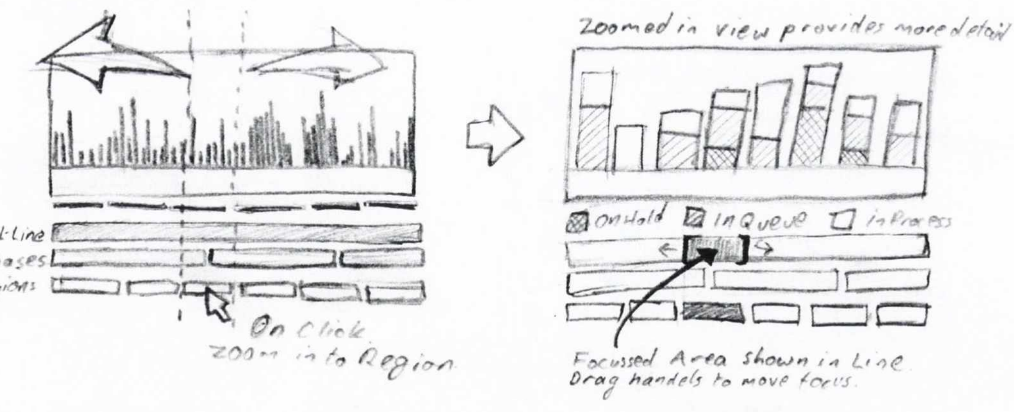


Figure 7.19: Sketch of graphical zooming from full-line to regional inventory

performance, WIP can be put on-hold while metrology tests are carried out. This stops this tools ability to process WIP. As a result, lower-level data relating to the status of WIP is available, namely inventory can be in-process, in queue or on hold. While this information can be presented as contextual information on rollover, its representation within the inventory display can be achieved through colour coding and by sub-dividing the inventory bar into three regions. This encoding makes it possible to visually diagnose one of the causes of inventory build-up that creates layer bottlenecks (see fig. 7.19).

As is evident from this design rationale, user-centred design was applied throughout the visual design process. Design reviews were carried out after each visual design phase. During these reviews line managers provided feedback on sketches and discussed the concepts in terms of their management goals. The final visual concept (see fig. 7.20) was reviewed with the line, operations and factory manager during a 2 hour design review. During this review the participants were walked through a detailed use-case involving the identification of constraints in the line and estimating the impact of these constraints on particular orders in the line. A high-fidelity prototype was developed for this activity. 24 separate screens were designed using adobe photoshop and these screens were organised in an interactive powerpoint presentation. The managers confirmed that the visualisation provided a number of potential advantages:

- The unified display removes the need for information foraging across large data tables

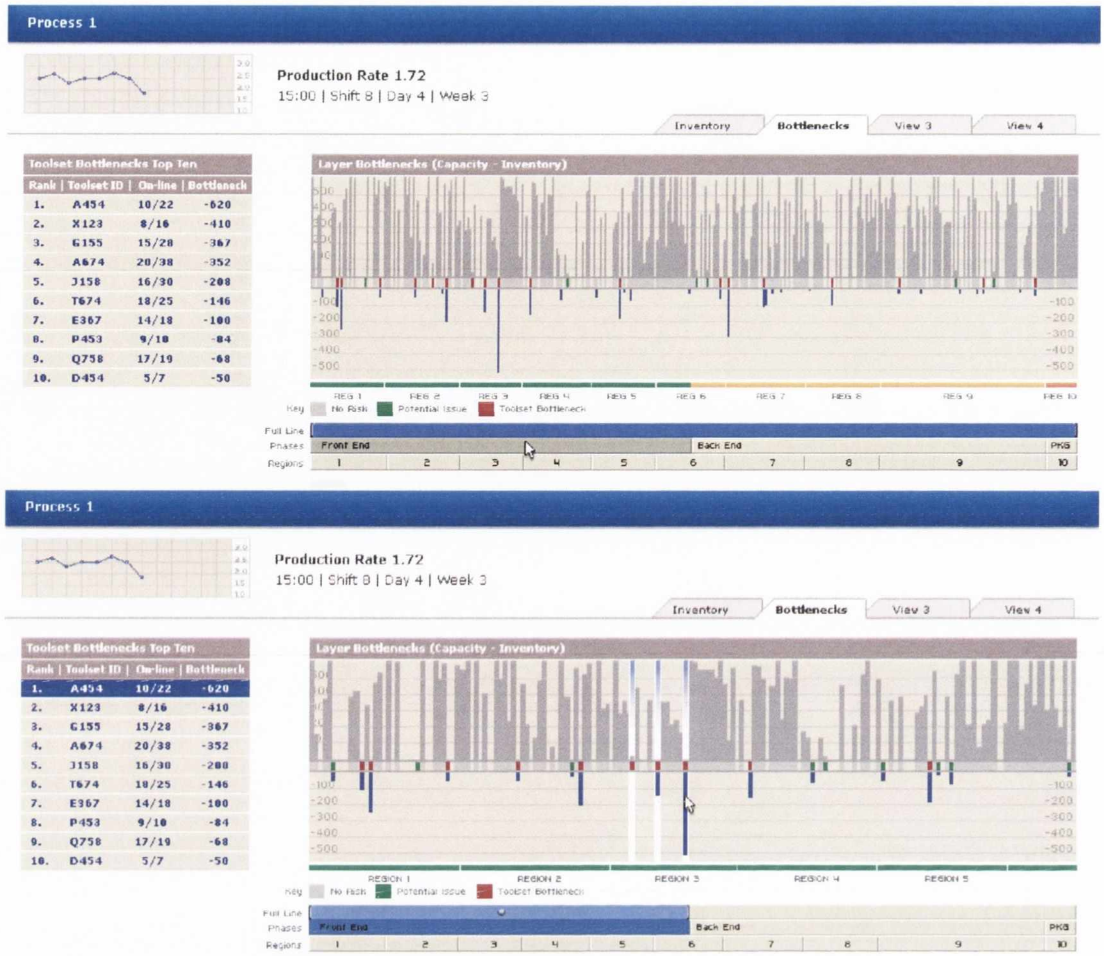


Figure 7.20: Two screens from the final visual design

- The visualisation removes the need for mental arithmetic to understand the system state

- The graphical overview supports preventative rather than reactive line management strategies

- The display supports the communication of manufacturing goals to technicians, enabling them to see the effects of their actions on the overall system

- The graphical representation improves everyone's understanding of a complex process and can align engineering and manufacturing goals to ensure minimum impact operations control

7.6 Analytical Evaluation

Control task analysis is again used to evaluate the validity of the new design in terms of communicating the system state and supporting operations control. The task of bottleneck identification and management that was used during the strategies analysis (section 7.3.4) is repeated here and a decision ladder tracks the cognitive process involved (fig. 7.21). Figure 7.22 presents the interface at four different stages during bottleneck management to provide a visual storyboard of user interaction. Specific information processing actions are numbered in the decision ladder and their corresponding graphical objects are numbered in the storyboard. Control task analysis generally traces a particular decision making path from activation through to execution, however as the lower-level execution tasks in this system are handled by ROCC technicians, this analysis stops at the formulate task level.

- *Activation:* The new interface makes it much easier for a line manager to conduct a review of the entire line. This allows a manager to carry out line reviews numerous times during each shift. In this scenario a line manager notices a drop in production rate during one of these reviews. As bottlenecks have a major impact on production rate, the manager scans the bottleneck view to locate any issues.

- *Alert:* While a number of process steps with negative residual capacity are evident throughout the line, the amount and severity of these are greater in the front-end phase of the line. The manager clicks on the front-end button in the navigation toolbar to zoom in on this phase.

- *Observe:* One process step in particular has very low residual capacity so the manager rolls over this point to get more information. The contextual highlighting triggered by the rollover allows the manager to observe that this is the third layer of a three layer operation. In the toolset panel the contextual highlighting shows that this operation is associated with toolset 'A545' which is the highest level toolset bottleneck in the line. The manager clicks on either the toolset of process step which selects the associated graphical objects. In order to get a better understanding of the impact of these bottlenecks the manager clicks on the inventory tab. The interface switches to the inventory view but the selected operation remains highlighted. The manager observes the distribution of WIP in the line and notes that the WIP levels are dropping off sharply after the third layer of the selected operation.

- *Set of Observations:* These observations inform the manager that there is a toolset bottleneck in the front end of the line that is affecting the distribution of WIP in the process. The three associated process steps are all located in region three, so the line manager clicks on the region 3 button in the navigation toolbar to zoom in on this region.

- *Identify:* When zoomed in to this level, more information becomes available in the inventory view. This allows the user to identify the potential causes and impact of bottlenecks. The manufacturing panel provides an additional breakdown of the WIP status. From this it becomes evident that there is a large amount of WIP on-hold in the latter two process steps. The process tools associated with this on-hold WIP are awaiting feedback from metrology. The tools cannot continue processing until this information is returned, so this is one of the causes of the bottleneck. The line inventory also shows that WIP is building up between these two steps while process steps later in the line are being starved of WIP. If this build-up continues, it will take longer for the bottleneck to be resolved when tool availability is restored. The toolset panel allows the manager to identify the primary cause of the bottleneck. The toolset is working with only 10 out of 22 tools on-line, which almost halves its normal capacity.

- *System State:* This information allows the manager to recognise that there is a critical toolset bottleneck in region 3.

- *Interpret:* Based on the information derived to this point, it is apparent that if this bottleneck is not resolved WIP will continue to build up in the region. This will have a negative effect on all three of the performance criteria associated with operations control namely; customer satisfaction, consistent speed and consistent spread of WIP.

- *Goal State:* Based on this interpretation of the system state, the goal is to resolve the bottleneck in region 3.

- *Define Task:* Three different tasks can be defined to achieve this goal. Firstly, availability needs to be increased by getting more of the process tools back online. Secondly the holds need to be removed from the WIP to allow processing to proceed. Thirdly, the capacity at the third layer needs to be increased in order to reduce the WIP build up.

- *Task:* Each task can be broken down into associated sub-tasks. In the case of increasing capacity at layer three, this involves setting a temporary operation rule for the ROCC technicians. Formulate Task: This rule needs to communicate the current issue and the proposed solution. In this case it will prioritise processing at layer three and request technicians to assign all tools to this layer. The procedure and execution of this task is handled by the ROCC technicians as part of their scheduling activities.

This decision ladder shows how the new interface can support line management in the new remote operations environment. Unlike the previous interface, which relied on the supervisor's continuous state awareness, the new interface presents system information in a format that directly supports OC goals. While the original work configuration relied on socially distributed cognition to generate representations of the system state, by identifying the components of the OC problem space and presenting them in a graphical format, the problem can now be solved through the interface using individually distributed cognition.

7.7 Discussion

As with the previous chapter, this project deals with the design of a visual decision support system for an evolving sociotechnical enterprise. However the design challenges in this case were even greater as operations control requires multiple perspectives on the manufacturing system and involves intentional as well as causal reasoning. The design process mixed a number of analytical models in order to develop a more comprehensive model of system functionality. These models provided both a context for understanding

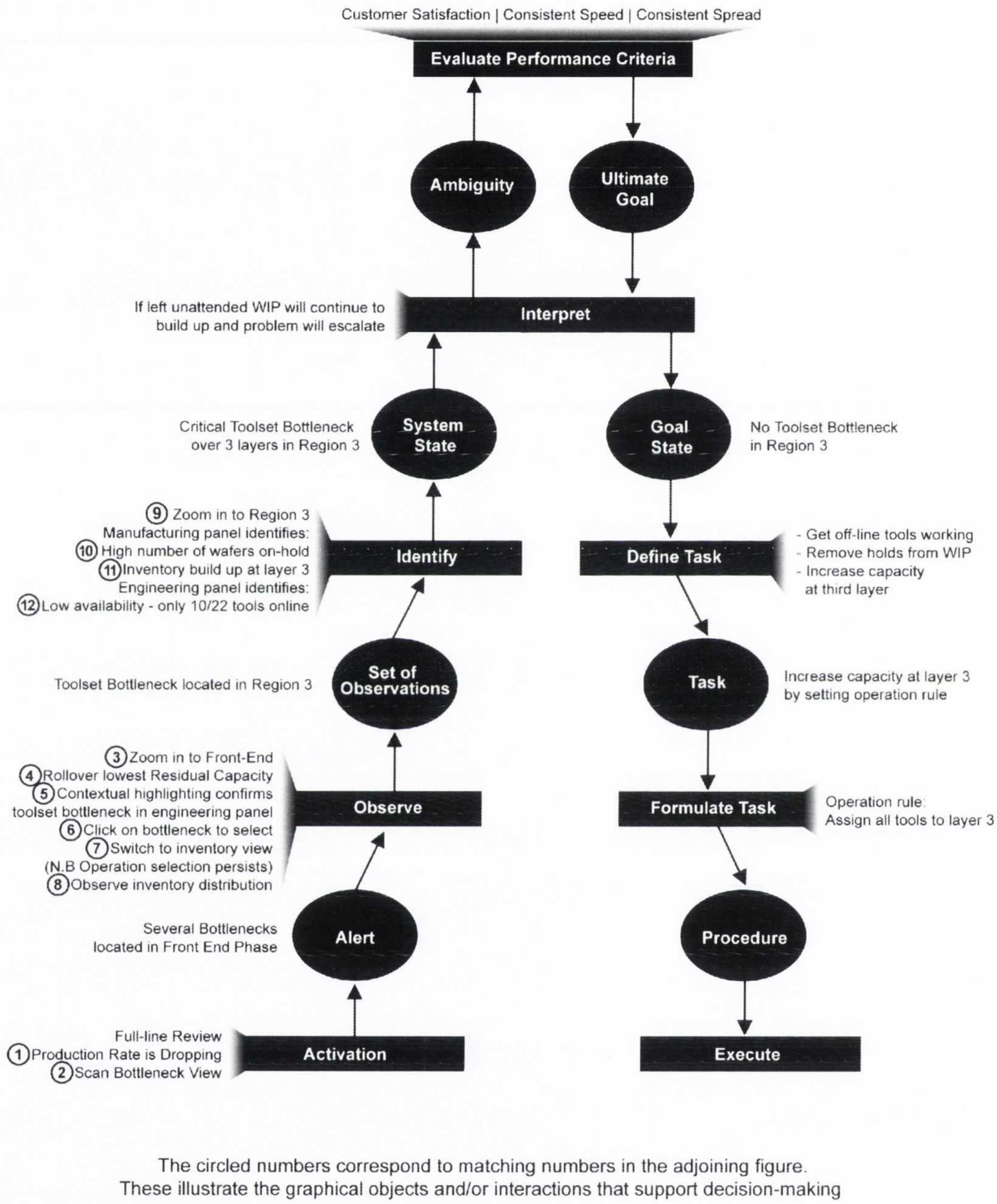


Figure 7.21: Decision Ladder representing Bottleneck Management Control Task

Figure 7.22: Storyboard of Bottleneck Management Interaction

changes that are occurring in the fab and structures that inform the design of visual representations. This design process demonstrates the advantages of applying a mixed model approach when dealing with sociotechnical enterprises.

7.7.1 Design Artifacts

A wide range of sketches and models were developed during this design process. In this section these primary design artifacts are reviewed in order to identify the secondary artifacts that describe a more generic design methodology. The CSE design meta-model is again used to describe different phases of concept generation and commitment. In each phase the primary and secondary artifacts are identified and their role in progressing the design process are discussed.

7.7.1.1 Work Domain Analysis

- *Primary Design Artifacts.* The development of a work domain model of the fab required the system to be examined from a number of different perspectives. The manufacturing and engineering perspectives provide alternative but equally accurate views on system functionality which resulted in the development of two work domain models (figures 7.2 & 7.3). The sketch in figure 7.4 identified the relationship between these views and lead to the development of a work domain model based around the concept of an abstraction lattice. This provides a work domain model for a system with high-level conflicting goals (fig. 7.6).

- *Secondary Design Artifacts.* Work domain analysis is a secondary design artifact that provides a useful starting point for examining system functionality. Its use of predefined and psychologically relevant structures supports a top-down analysis of the work system. This approach is useful for dealing with the scale and complexity of the enterprise, as it can model functionality without getting caught up in the intricacies of work practice. With this model, the analyst can begin to expand their knowledge of the work domain, generate a lexicon of related terms and understand system goals.

7.7.1.2 Analysis of Distributed Cognition

- *Primary Design Artifacts.* The practice of OC involves a distributed cognitive system spread across different aspects of the work environment. The models of social structures, information systems and physical layout developed in section 7.3.2 are primary design artifacts that how these factors support system functionality.

- *Secondary Design Artifacts.* The process of DC analysis through studying work practice is a secondary design artifact that extends the knowledge generated through work domain analysis. While the work domain model describes functional goals in terms of system constraints, the DC analysis takes these goals and describes how the work environment is configured to achieve them.

7.7.1.3 Intentional Goal Modelling

- *Primary Design Artifacts.* The intentional goal model in figure 7.9 is a primary design artifact that describes system goals relating to policy rather than physical constraints. The model extends the analysts knowledge how intentional decision-making occurs in relation to OC.

- *Secondary Design Artifacts.* The process of intentional goal modelling is a secondary design artifact whereby the policies that guide intentional decision-making are analysed and broken down into lower levels goals. These goals are further analysed until a complete model of relationships between system goals and work practices is revealed. This modelling activity is different from hierarchical task analysis in that it describes goals in terms of the functional system rather than in terms of interaction sequences

7.7.1.4 Control Task Analysis

- *Primary Design Artifacts.* The decision ladders depicted in figure 7.10 are primary design artifacts that model the cognitive strategies involves in bottleneck identification in the line. This model summarises the information processing activities of both supervisors and managers. It shows how information processing of system data allows the workers to identify the system state at higher levels of abstraction. Furthermore, the model in figure 7.10 identifies how supervisors can transfer knowledge about system performance at a particular level of abstraction and how the line manager incorporates this knowledge into their decision-making process.

- *Secondary Design Artifacts.* Control Task Analysis provides a secondary design artifact for extending knowledge about system functionality down to low-levels of cognitive processing. In doing so, it ties together levels of abstraction revealed in the work domain and intentional goal models and shows how knowledge-based decision-making it related to intentional goals.

7.7.1.5 Change Analysis

- *Primary Design Artifacts.* In the previous PCS project modelling the work systems functionality provided enough information to inform the design. However, in this project the work system is evolving, with the result that work practices and information requirements are changing. In this phase of the design process new models of DC themes under the ROCC configuration are generated. The analyst contrasts the original work configuration with the new one in relation to the goals and constraints revealed by the formative models. From this a number of information resource challenges are identified. The higher-level values outlined in table 7.1 aim to replace the information processing support that the line manager previously received under the original work configuration.

- *Secondary Design Artifacts.* Modelling changes to work practice acts as a secondary design artifact derived from this design process. As stated above this requires both formative models of system functionality and descriptive models of work practices under the different configurations. The advantage of this approach is that the metrics derived by the analysis allows decision-making to move to a higher level of abstraction. By providing higher-level metrics the cognitive workload of human controllers is reduced, as less information processing is required. However a remaining challenge is how these metrics can be presented in a manner that is interpretable to users. While the lower-level data was presented within familiar models of the fab structures, these new values relate to intentional goals rather than individual system components.

7.7.1.6 Information Requirements Integration

- *Primary Design Artifacts.* The information requirements matrix presented in table 7.2 is a primary design artifact for compiling the information from the various analytical models into a single transitionary artifact.

- *Secondary Design Artifacts.* The IR Matrix provides the generic categories of abstraction hierarchies, information requirements, interaction requirements and notes to support the assembly of system information. While the presence of three alternative abstraction hierarchies makes it more difficult to compile system data, the intentional goal model integrates data from the other two perspectives and provides a structure for integrating the new higher-level metrics.

7.7.1.7 Preliminary Design

- *Primary Design Artifacts.* The primary design artifact for this phase consists of the preliminary sketch shown in figure 7.14. This provides a concept for the general composition of the interface and defines views of the system data.

- *Secondary Design Artifacts.* EID's third principle, of presenting system information in the form of an abstraction hierarchy, provides a secondary design artifact for controlling this sketching activity. The presence of multiple abstraction hierarchies poses a challenge. The intentional goal model was selected as the primary hierarchy as it uses information from the two other constraint-based hierarchies. Information requirements associated with the different levels of abstraction in the intentional goal model were visually structured using Wood's graphic display hierarchy. This resulted in an interface that provides multiple views on the system information that each correspond to higher-level system goals. The physical form level is encoded by dividing these views into two regions, corresponding to a manufacturing and engineering view respectively.

7.7.1.8 Refined Design

- *Primary Design Artifacts.* The sketches provided in figures 7.15, 7.16 and 7.17 are primary design artifacts used to refine the initial concepts. The aim is to transform the predominantly propositional displays into graphical displays that improve data observability.

- *Secondary Design Artifacts.* EID's second principle of matching system constraints to visual cues provides the basic secondary design artifact that motivates this sketching activity. Wood's guidelines for the development of analogical displays are used to transform the production rate figure into a trend chart that provides better support for line management. The visual reference model is applied to guide the visual encoding of bottleneck information. This transforms the manufacturing views data table into a visual chart, where the system state of the entire line is instantly observable. While this chart allows the user to user perceptual operations rather than cognitive operations to recognise bottlenecks, the graphic encoding loses the structural information that describes the context of the bottlenecks. However the visualisation reference model can also be applied to the structural hierarchies to generate a graphical system image.

7.7.1.9 Detailed Design

- *Primary Design Artifacts.* The final sketches in figures 7.18, 7.19 and 7.20 are the primary design artifacts in the detailed design phase. Here the previous concepts are modified to clarify relationships between alternative views, and to improve the overall usability of the interface.

- *Secondary Design Artifacts.* The EID principle of using direct manipulation and making the design isomorphic the part-whole structure of movements provide design goals for this phase. While temporal control is less important for this display the use of interaction improve a users understanding of relationships in the system. The proximity compatibility principle provides a secondary design artifact that informs the use of contextual highlighting. This technique makes the relationship between engineering and manufacturing views explicit and plays an important role in the identification and diagnosis of performance issues. Finally, Schneidermann's visualisation mantra of 'overview, zoom and details and demand', informs a final level of interaction functionality whereby the structural display in the manufacturing region can act as a navigational element that supports interactive zooming. This interaction mimics the information access strategies used by the line manager and makes it possible to follow causal relationships in the system from high-level metrics down to low-level system data.

7.7.2 Design Methodology

The CSE design meta-model has been used to extract a range of secondary design artifacts that form another methodology for designing visual decision support in a socio-technical enterprise. This time a wider range of concepts and principles are required to deal with the complexity, scale and dynamism of the target cognitive system. Here the basis of this methodology is outlined in relation to the three research questions posed at the start of this chapter.

In terms of *which analytical approach* should be used, the design problem structuring phase involves a range of models that describe different aspects of system functionality. With this cognitive system work domain analysis provides a formative model of structural constraints but can not give a complete picture of system functionality. There are two main reasons for this, the distribution of control activities and the intentional decision-making associated with operations control.

Enterprise systems involve large scales and require control to be distributed across teams of workers who manage smaller parts of the system and collaborate to achieve system goals. Therefore in order to fully understand system functionality it is necessary to describe work practices. The DC analysis is comparable to the activity analyses in the previous project, but this time the focus is on the behaviour of a work system rather than the actions of an individual user.

This system also involve intentional decision-making as a result of in-built flexibility in the manufacturing process. Flexibility is necessary in many large complex processes to reduce the brittleness associated with tightly coupled components. However flexibility

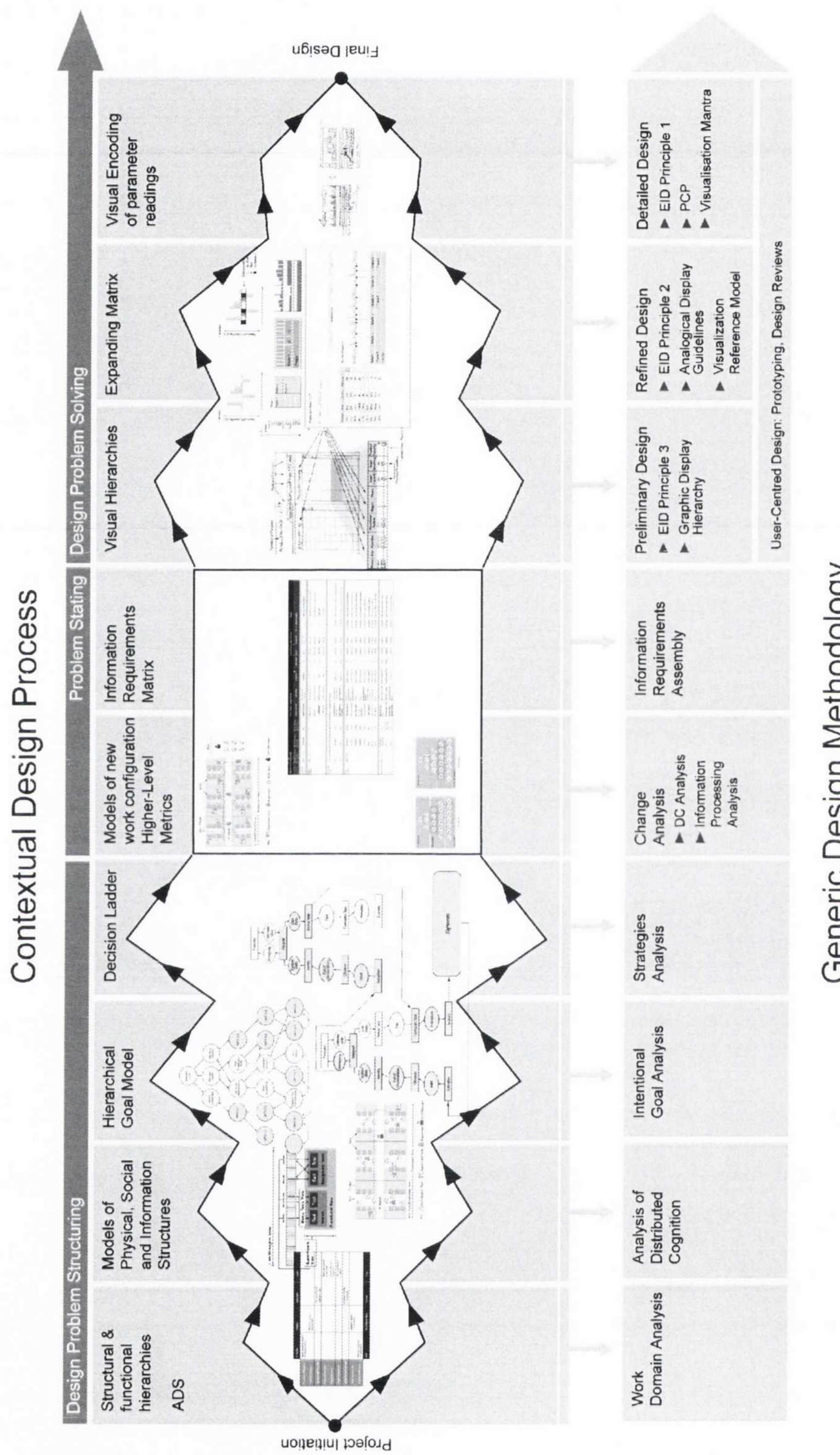


Figure 7.23: A Model of the Design Artifacts used in this Process

means that control decisions are no longer defined by the system capabilities alone but are also based on business goals and management policies. Intentional goal modelling allows an analyst to understand how these external factors influence system behaviour. This analysis produces an additional formative model but this time of non-physical factors that influence system functionality.

As the models of DC themes outline structures associated with work practice, they can inform a targeted ethnographic analysis of how intentional goals are achieved. Once the relevant worker roles have been identified, control task analysis can be used to reveal the cognitive strategies and information processing used to support system control.

In relation to *the question of representational guidelines*, the development of an ADS during system analysis means that the EID design principles can be used to structure the basic design process. The underlying theory of supporting skills, rules and knowledge based behaviour are generic principles that are applicable to any work domain. However, supplemental visual design guidelines are again required to aid with design decisions.

During preliminary design the goal of representing the system as an abstraction hierarchy is hindered by the presence of multiple hierarchical system models. In this case the functional abstraction hierarchy associated with the intentional goal model was made explicit by matching it to Wood's graphic display hierarchy while structural hierarchies can be made explicit through their graphic encoding using the techniques of the visualisation reference model.

EID's second principle requires developing visual cues to match system constraints. The techniques of data analysis, data transformation and visual scale matching associated with the visualisation reference model provide a structured approach for achieving this. The manner in which these techniques are applied are based not only on the available data but also on the context in which this data is used. Knowledge of information processing gained through descriptive analysis can inform data transformations and visual encoding methods. Similarly knowledge of constraints and relationships in the work domain can inform the structural composition of data into semantic forms. As these factors apply to the same graphic forms they must be balanced against one another in the development visual decision support displays.

In the detailed design phase the principle of using direct manipulation to support temporal control and laying out elements in a manner isomorphic to their structure of movements is provided. While this principle was originally developed at a time when direct manipulation was in its infancy, the merits of this approach are now universally accepted. In the meantime the range of interaction techniques has vastly increased. It would seem more reasonable to extend this principle to recommend applying interaction techniques that ensure high-levels of system usability. Approaches such as contextual highlighting, zooming and gestural interfaces are some examples of new interaction techniques that can be applied to support temporal control and improve skills based behaviour. The application of these techniques should be considered in relation to the characteristics of the domain.

In terms of the question of *what design process* to follow, the methodology presented here uses the same phases to describe the design process as were applied in the previous chapter. However the need to understand and support changes in system behaviour requires a more iterative approach to problem structuring and stating. The two formative models relating to work domain constraints and intentional goals describe stable aspects of the system that remain valid irrespective of technological or social developments. On the other hand, the models of DC themes describe work practices and demonstrate how workers actions are defined by the work system configuration. When designing decision support for evolving enterprises, a comparison between existing and proposed work configurations can identify the impact that changes will have in terms of cognitive workloads. This can direct the design and development of visual artifacts that provide better cognitive support. The methodology also shows how the different problem structuring artifacts are used to inform visual design. The formative models outline stable relationships in the system around which the application workspace can be structured. The visual encoding of these relationships through the composition of the interface design allows for the development of a valid system image. The models of cognitive strategies identify information processing tasks that can inform the visual encoding of system data and the application of interaction techniques that aid usability. This correspondence between analytical models and visual structures bridges the design process gap.

Again the issue of *re-usability* must be addressed. Although this project was carried out in the same enterprise, a different design methodology was generated. While

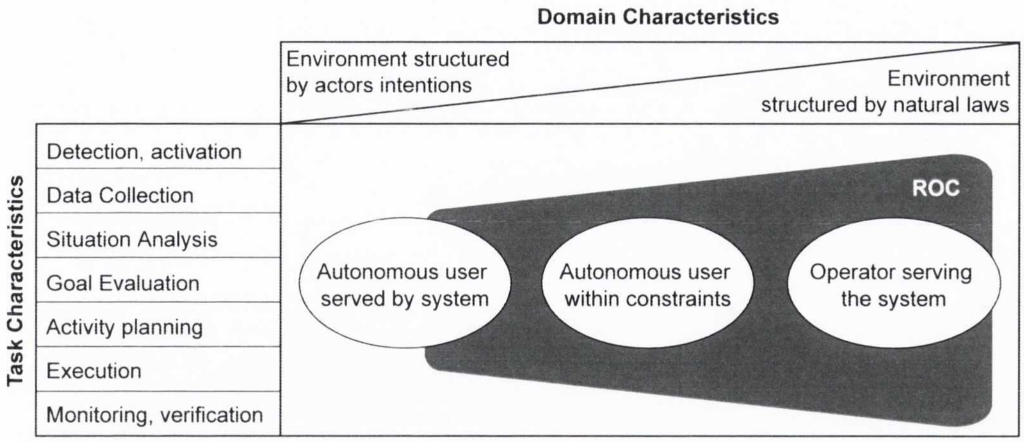


Figure 7.24: Locating the ROC system on the WTU map

many of the same design artifacts are involved (i.e. work domain analysis, control task analysis, EID principles) a number of additional design artefacts were required. In order to understand why a different methodology was needed it is necessary to identify the difference between the two cognitive systems. Again the characteristic of the cognitive system are charted using Rasmussen's Work Domain, Tasks and User (WTU) map (see figure 7.24). The system supports operational control in a causal work system, where production is constrained by the limits of physical engineering. However additional constraints must also be considered such as customer satisfaction and production policies. In terms of tasks, all categories of task from low-level detection to high-level goal evaluation must be supported. In terms of user characteristics, the system involves a team of users and these can be identified with all of the categories defined in the original WTU map. For example, technicians in the 200mm fab are operators serving the system, but a line manager can choose from a range of different strategies to achieve goals based on their personal management style making them more of an autonomous user. As a result this cognitive system occupies a region covering most of the WTU map. This mapping activity does little to explain why particular design artifacts were chosen, suggesting that a better means of describing cognitive systems is required. This is discussed further in the following chapter.

Chapter 8

Discussion

At the outset of this work, the primary research problem was to define a CSE approach for designing visual decision support systems in sociotechnical enterprises. Practice-led research was used to extract two separate methodologies from design projects conducted in the semiconductor-manufacturing domain. Although these methodologies share a number of steps and the same basic structure they are also quite distinct. The fact that separate design approaches are needed within the same sociotechnical enterprise requires further discussion. In particular it is necessary to identify in what way are these methodologies generic and how can they be *re-used*? To answer these questions it is necessary to discuss this research within the wider context of the CSE design problem. This chapter reviews how the design problem has been handled by the CSE discipline to date, it shows how the CSE meta-model can be used to develop a catalogue of design methodologies and it explains why an extended taxonomy of cognitive systems is required to support the re-use of design knowledge.

8.1 The Cognitive Engineering Design Problem

In their book, cognitive systems engineering, Rasmussen and colleagues propose that *the design activity is an inherently variable and opportunistic process* (Rasmussen et al., 1994). They cite Gould's assertion that design solutions depend on a host of unpredictable factors that are *unique to the particular problem* (Gould, 1988). In addition they state that experienced designers adopt *an intuitive, recognition-primed mode of decision mak-*

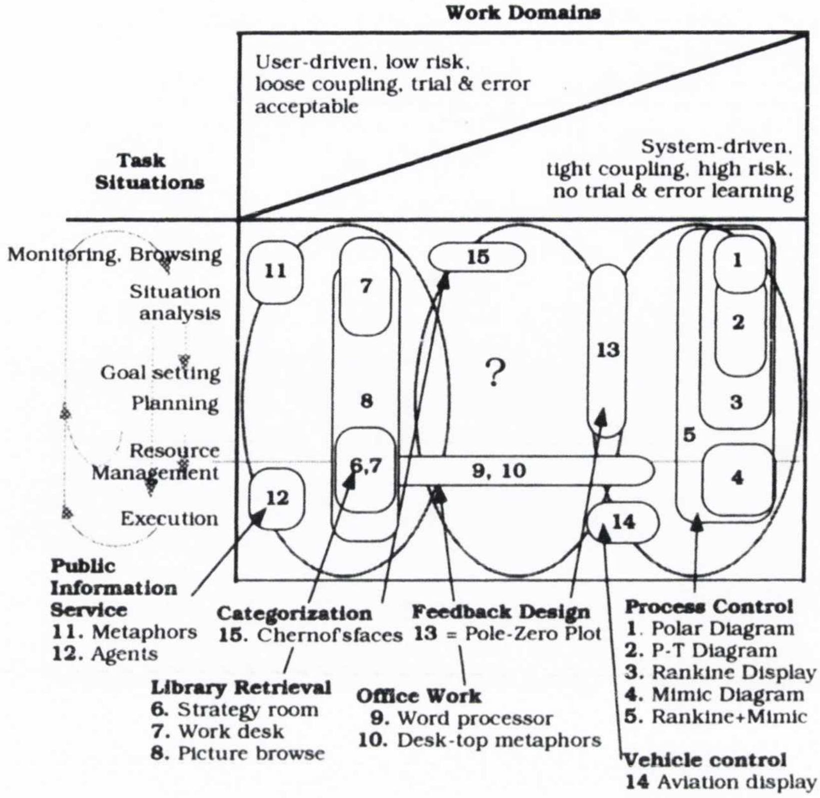


Figure 8.1: Catalogue of displays mapped to WTU map (Rasmussen et al., 1994)

ing. Given these qualities, they argue that design guidelines have limited utility and that designers should develop a catalogue of successful design solutions within their sphere of interest. They propose that the Work domain, Task situation and User characteristic (WTU) map, which was introduced in section 2.4.1 of this thesis, can be used to organise these designs in a manner that encourages their re-use for cognitive systems with similar properties (see figure 8.1). However the re-use of existing graphic solutions is problematic as they can only fall into two categories. They are either *highly contextual ecological displays* that have been designed for a specific system, in which case their domain-specific visual syntax does not support re-use, or they are *context-free interface components* such as charts or navigational elements, in which case they provide only a partial solution that must be integrated into an ecological display to provide an accurate system image.

This approach was based on an acceptance that the indeterminism, complexity and variability of cognitive systems preclude the application of prescriptive design knowledge and dictates a craft-based approach to design. A decade ago, Dowell & Long (Dowell and Long, 1998) challenged this view, proposing that while the softness of cognitive systems is a problem in itself, the lack of usable design principles is symptomatic of a discipline that is in its infancy. Rather than being inherently intractable, design knowledge requires the development of generic design principles and the provision of exemplars that illustrate and validate their utility. These steps, they argue, are critical for the establishment of cognitive design as an engineering discipline. Since then cognitive systems engineering has evolved significantly. The range of theories and principles has grown to the point where practitioners are now becoming overwhelmed with the selection of design knowledge available (Erickson, 2002; Halverson, 2002). However despite this abundance of design knowledge, the practice of design remains extremely complex and this complexity can be traced back to the initial problem relating to the softness and variability of cognitive systems.

8.2 A Catalogue of Design Methodologies

This research has demonstrated that many CSE theories and principles focus on particular aspects of cognitive design. These restricted views have resulted in a number of design gaps that must be bridged each time design theory is applied to design practice. As different work systems have different physical, social and technological characteristics, it is necessary to select theories and principles that suit domain characteristics. If a particular configuration of models and principles results in a successful design solution, this configuration may go on to form a design methodology. The last decade has witnessed the development of numerous design methodologies that have been built around domain-specific exemplars (Blandford and Furniss, 2005; Lepreux et al., 2004; Carayon, 2006). While these aim to provide a more holistic approach to supporting design practice, this approach poses two new issues for the CSE discipline. Firstly, how can we ensure that these design methodologies are *comprehensive* and secondly how can we ensure that they are *re-usable*? The CSE design meta-model developed through this research provides a conceptual tool for resolving the first issue. To ensure that a design methodology is comprehensive and can inform design practice through-out a design process it must:

- Cover each of the phases outlined in the design problem space

- Provide an exemplar that demonstrates how the analytical models inform representational design decisions

- Provide a design rationale that describes conceptualisation in each of the design phases

- Specify the theories or principles that control this conceptualisation activity

The second issue of re-usability is more complex. As methodologies are suited to particular characteristics of cognitive systems, it may be possible to adapt Rasmussen's original cataloging concept. However rather than mapping existing design solutions, a catalogue of design methodologies could identify suitable design processes for particular types of cognitive systems. In this way a designer could examine the characteristics of their target domain, identify the associated region on the WTU map and select the most appropriate methodology accordingly. This would allow a designer to identify analytical models and representational principles that can control design conceptualisation without actually dictating design solutions.

8.3 A Taxonomy of Cognitive Systems

One difficulty with this approach is that the WTU map may be too simplistic to provide useful distinctions between cognitive systems. Both of the cognitive systems associated with the design projects reported in this thesis were charted on the WTU map (figures 6.24 and 7.24). However, rather than identifying specific points, they occupied large, overlapping region. Furthermore, particular characteristics of the cognitive systems such as scale, distribution of work and temporal issues are not covered by this approach. If design methodologies are to be re-usable, *a more refined, design-focussed taxonomy of cognitive systems is required.* Here, an initial effort to describe such a taxonomy is outlined [1]. Eight characteristics are defined, two within each of the four main headings; work domain, tasks, agents and representability. Each characteristic describes the extent to which a particular quality is exhibited by a cognitive system. Figure 8.2 outlines

[1] This is not an attempt to develop a definitive taxonomy of cognitive systems, but more of an exercise to demonstrate how such a taxonomy could inform the selection of design methodlogies.

these characteristics and describes three alternative cognitive system using this taxonomy. These are the Duress II (Vicente, 1999) microworld, the PCS health monitoring system and the ROC system that are discussed in this thesis. Figure 8.3 outlines the three corresponding design methodologies. Each characteristic and its relationship to particular design artefacts are described in more detail below.

8.3.1 Work Domain

The work domain is described using two characteristics, environment and constraints. While the WTU map shows these characteristics as being directly correlated, the increasing use of embedded automated control systems is changing the nature of this relationship. As human decision-making in joint cognitive-systems moves to higher levels of abstraction, it become necessary to consider how other factors outside of environmental contraints can influence behaviour.

8.3.1.1 Environment

This describes how well the physical characteristics of a work domain describe the causal relationships used by its associated cognitive system. This characteristic is divided into three regions; natural, structured and engineered systems.

- *Natural systems* are those whose components are loosely coupled and may involve conceptual representations rather than real world objects. In these environments causal relationships are more dependent on interpretation and social conventions than physical relationships and therefore approaches such as activity modelling and intentional goal modelling are more appropriate. Examples include information retrieval systems such as libraries.

- *Engineered systems* have tightly coupled components that are configured to achieve well-defined goals. Causal relationships are fully described by the physical and functional structures of the system, so work domain modelling provides a suitable approach. Manual process control falls into this category.

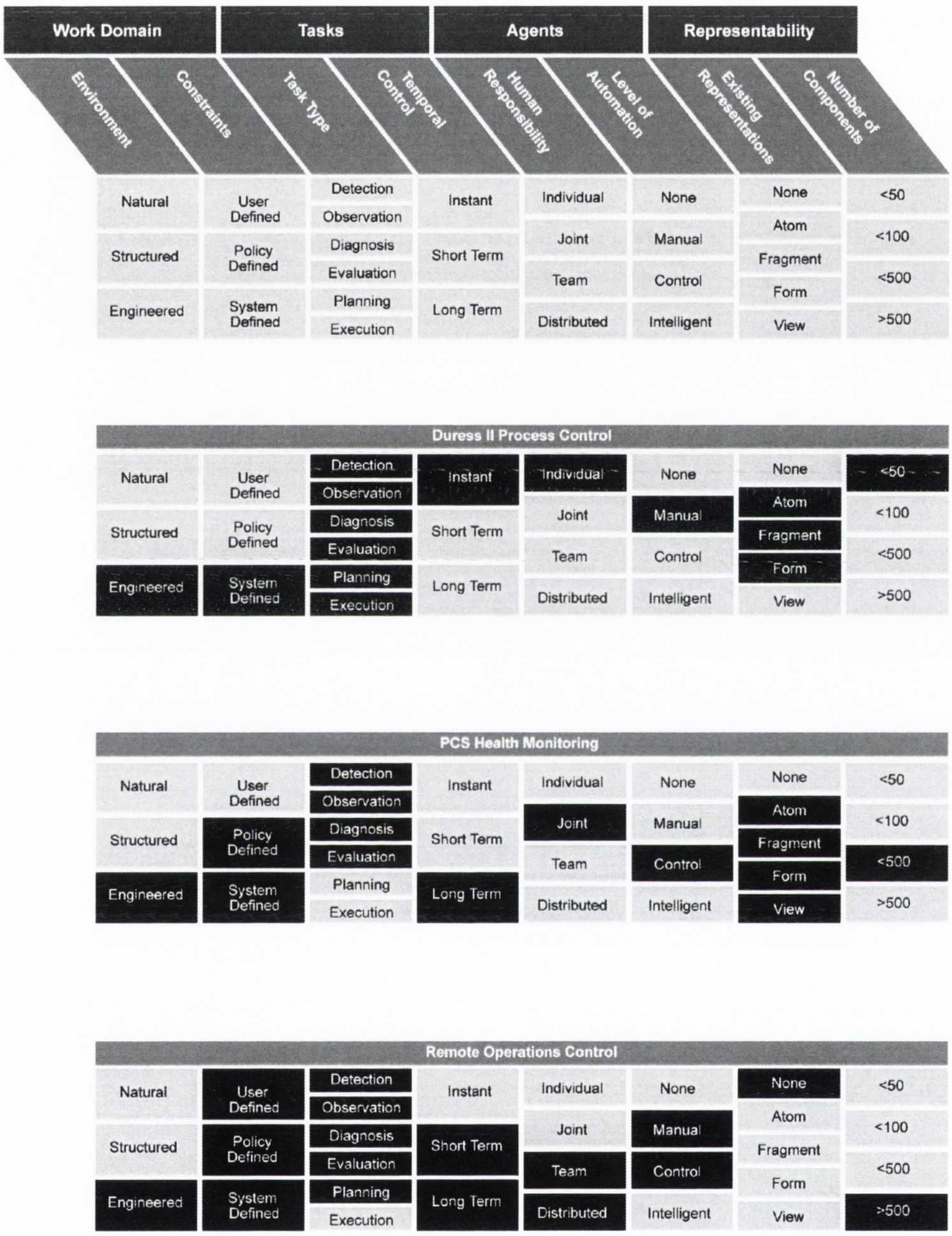


Figure 8.2: A taxonomy of cognitive systems with three examples

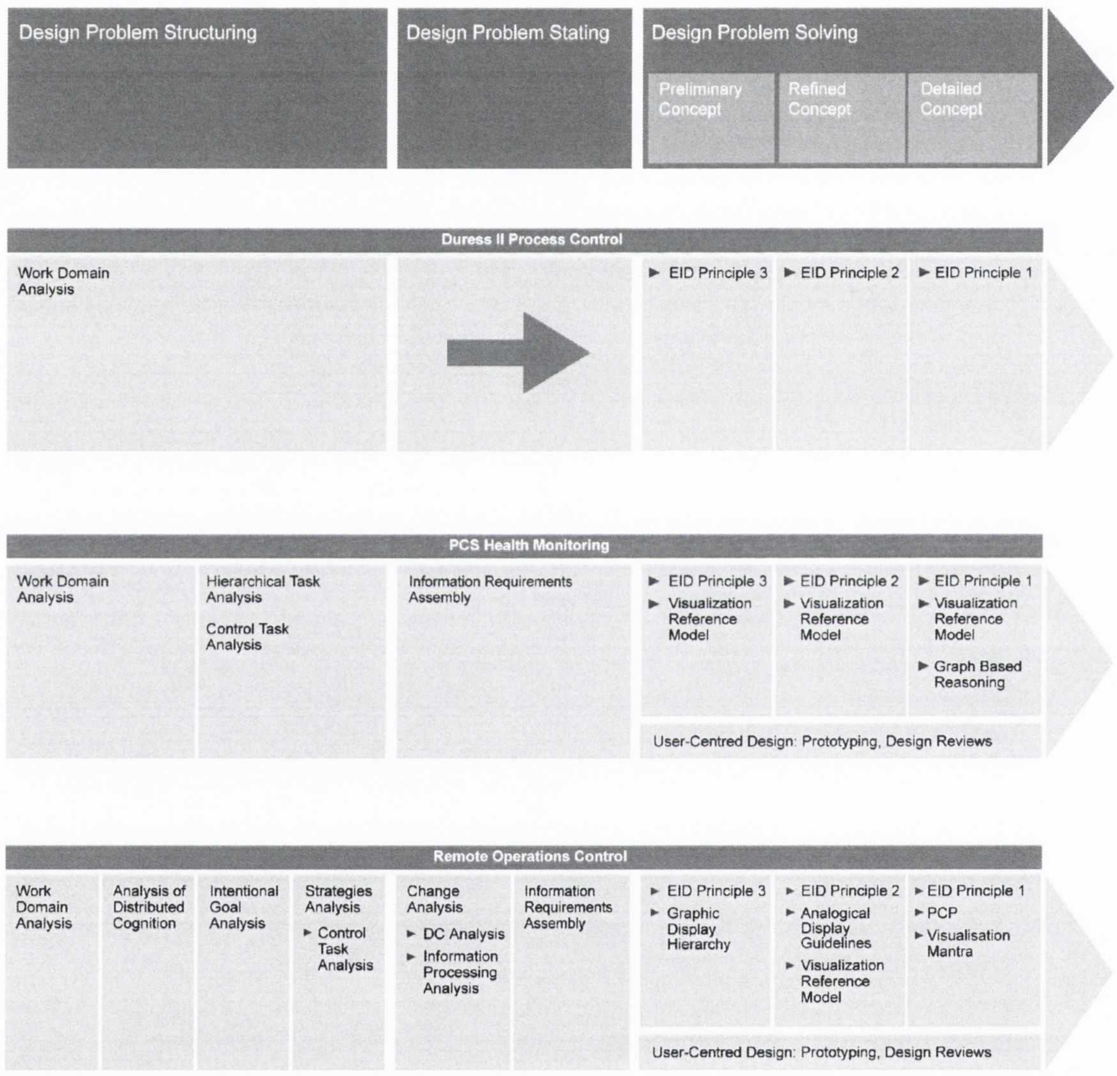


Figure 8.3: Three CSE design methodologies

- *Structured systems* describe work domains that fall between these extremes. Environments that involve controlling real world objects, but where the nature of control is influenced by policies or procedures fall into this category. Examples of structured systems include resource planning and strategic operations. A mixture of both work domain and activity analysis is needed to describe these properties.

8.3.1.2 Constraints

This characteristic describes the factors that dictate how specific decision outcomes are reached. Again three different regions are used; user-defined, policy-defined and system-defined. These can be used to indicate the predominant constraint that influences behaviour.

- *User-defined* describes systems where the users intention is of maximum importance for decision-making, for example image editing. Here activity models are required to understand a user's motivation.

- *Policy-defined* describes situations where the decisions must comply with social or operational norms. For example in planning and scheduling, procedures, checklists and targets all influence behaviour. In addition the manner in which these elements are represented dictates the cognitive strategies used to support decision making. This makes modelling distributed cognition an appropriate approach for understanding functionality.

- *System-defined* relates to highly structured work situations where the goals are known in advance but where unexpected events can affect performance. In this situation human activity revolves around achieving system goals and these take priority over their personal views or opinions, making work domain modelling a suitable approach.

8.3.1.3 Application to examples

The Duress microworld involved an engineered environment and system-defined constraints. Consequently work domain modelling can identify the information requirements necessary for design. The PCS health report involved an engineered environment and a combination of system and policy driven constraints. Work domain modelling was useful but did not fully describe the system constraints. Further analysis was required based on the tasks and agents involved. The ROC project involved an engineered environment and all three forms of constraints. Work domain modelling provided a useful first step in problem structuring but a range of additional analytical design artefacts were required to describe system functionality. Again the specific types of analysis were derived from the tasks and agents involved.

8.3.2 Tasks

Tasks are described using two characteristics; task type and temporal control. Again, these characteristics are somewhat related but their separation allows for greater distinction between types of cognitive systems and can better inform the selection of design artefacts.

8.3.2.1 Task Type

This characteristic describes six basic types of tasks derived from the WTU map and the decision ladder namely; *detection, observation, diagnosis, evaluation, planning, execution*. These are related to the different levels of cognitive control outlined in the skills, rules and knowledge taxonomy and can be used to identify how important each of the EID design principles are for a display. For example, a display may just need to support the detection and observation of changes for a range of parameters. If there is no need to present causal relationships then the second EID principle relating to supplying visual cues takes precedence. The purpose of this variable is to show what level of task support is required in the display.

8.3.2.2 Temporal Control

This relates to the pace at which action and feedback are required in a system. This is a continuum on which three regions are defined; real-time, short-term and long-term.

Real-time feedback is required to maintain control where the actions of a user result in an instant change to the system state, This can occur in both intentional domains such as desktop publishing and causal domains such as manual process control.

Short-term feedback can suffice where the actions taken by a controller take a number of minutes or even hours before their effects materialize in the system state.

Long-term temporal feedback is used in many complex work systems to understand process changes. For example, a change in a business workflow may affect productivity but this may only be apparent in the context of monthly performance metrics.

This characteristic can be used to select appropriate design principles. For example the real-time temporal control requires the designer to emphasise the third EID principle and to consider additional interaction principles.

8.3.2.3 Application to examples

The Duress II system involved the full range of tasks and required real-time control. Consequently all three EID principles had equal importance. The PCS health report primarily involves monitoring and diagnosis tasks and does not need to support planning or execution. Temporal control is long-term as the effects of many process changes are not noticeable for number of weeks. As the detection of change is the primary goal, the EID principle relating to rules-based behaviour and visual cues takes precedence. The ROC system is again more focused on monitoring and diagnosis than task execution however the system also needs to support the planning of operations strategies. In terms of temporal control, the feedback is not instant but short-term effects within a shift must be highlighted and long-term effects between shifts must also be made observable. The EID principles relating to knowledge-based and rules-based behaviour take precedence here.

8.3.3 Agents

The WTU map describes three archetypal users whose behaviour is influenced to different degrees by their work domain. This approach fails to take into consideration that a cognitive system may consist of multiple users who collaborate to achieve system goals. Furthermore the control of complex systems may no longer be handled exclusively by human users, but may be shared between human and automated controllers. The term "Agents" has been used to replace users and two characteristics are described under this heading; human responsibility and level of automation.

8.3.3.1 Human Responsibility

This describes the level of distributed support available to a human agent. It involves four categories; individual, joint, team and distributed.

- *Individual* responsibility describes situations where a problem solver has no support other than the information system. In this situation work domain or activity modelling is sufficient to identify the factors that influence user behaviour.

- *Joint* responsibility describes where a problem space is divided between the user and one other human or automated agent. Here control task analysis can be used to identify where information transformations and tranfers occur.

- *Team* responsibility describes where a group of co-located workers collaborate to achieve a goal. This may also involve support from automated systems.

- *Distributed* responsibility is used here to describe a work system involving a large number of workers who are distributed in terms of physical space or time.

For both team and distributed responsibility models of DC are required to identify how information representations are used to support system functionality.

8.3.3.2 Level of Automation

While numerous levels and scales of automation have been proposed (Parasuraman et al., 2000; Wickens et al., 1998; Sheridan and Verplank, 1978), this taxonomy provides a simple four level scale that covers both physical and data automation.

- *None* relates to natural decision making environments where workers act directly on the environment. Examples include fire-fighting crews and palliative care teams. The ill-defined and dynamic range of problems encountered by these workers requires a more interpretative approach to analysis such as activity modeling.

- *Manual* automation refers to situations involving physical automation but may be extended to automated interface tasks e.g. form filling, batch image processing. The well-defined nature of these systems means that the combination of work domain modeling and EID principles is sufficient to support design.

- *Control* automation refers to situations where a system includes automated data processing, monitoring of system variables and the ability to detect non-normal system behaviour.

- *Intelligent* automation refers to systems that include some form of artificial intelligence to enable higher levels of behaviour-monitoring and response.

For both control and intelligent automation a combination of work domain and control task analysis is required to identify how responsibility is divided between human and automated agents.

8.3.3.3 Application to examples

The Duress II system involves an manually automated process controlled by an individual operator. As such the combination of work domain analysis and EID principles provide sufficient design guidance. The PCS health report involved control automation in the form of computerized statistical analysis and joint responsibility shared between the human and automated agents. This required control task analysis to be conducted in order to identify how and where automated data processing supported system goals. The ROC system involved both team and distributed responsibility. It required information to be shared between users during the control activity and it also required information to be transferred between shifts. Manual and control automation all played a part in operations control as technicians instruct WIP scheduling and the AMHS controls the movement of wip through the process. The models of distributed cognition were required to understand how operations control is achieved. Control task analysis was again applied to understand how information was transformed and propogated around the system.

8.3.4 Representability

Factors that influence the representation of a cognitive system are not present at all in the WTU map. However these factors play a critical role in any design process and are addressed in this taxonomy under the characteristics of existing representations and number of components.

8.3.4.1 Existing Representations

This characteristic describes the degree to which standard representational formats are available in a work domain. Five different levels of representation, based on a modification of Wood's graphic display hierarchy (Woods, 1997b), are used. While the original hierarchy relates to all forms of representation, this is limited to iconic or analogical representations and only levels that can carry semantic meaning. The degree to which pre-existing representations are available will define the amount of visual design and representational design guideance required.

- *None* refers to domains that have no established visual vocabulary. In these domain propositional forms, such as spreadsheets or text, provide the predominant representational format. Consequently representational design will have to be carried out down to the syntactic level.

- *Atom* refers to individual graphic elements that carry semantic meaning such as icons, symbols, colour codes etc. These provide the basic level of visual vocabulary required to directly apply the EID principles.

- *Fragment* refers to a simple arrangement of atoms such as a scale or an indicator that encodes multiple variables.

- *Form* refers to a more complex arrangement consisting of fragments. Graphs and diagrams both fall into this category.

- *View* refers to a set of graphic forms displayed together that relate to a common aspect of the system.

Fragment, form and view levels all permit visual scale analysis to be carried out to identify what types of cognitive tasks they support.

8.3.4.2 Number of Components

This characteristic is used as an estimate of the maximum number of system components required in a view. This gives an early indication of the representational constraints that will be faced during visual design. While a precise figure may or may not be available, four categories are used to support approximations during initial project scoping; *<50, <100, <500, >500.* With smaller numbers of components iconic representations can be developed to carry semantic meaning. However when a system involves a large number of components that must be displayed together, information visualisation approaches will be required.

8.3.4.3 Application to examples

The Duress II system features a small number of components and existing representations at the graphic atom, fragment and form level, courtesy of its mimic display. These characterics means that the EID principles provide sufficient support to inform representational design. The PCS health report provided pre-existing representations from graphic atoms up to views and the individual indicator charts could feature over 100 data values associated with parameter sensors. Visual scale analysis revealed that the existing indicator charts did not support the full repertoire of cognitive tasks associated with monitoring and diagnosis. The visualisation reference model was applied to generate a design space of possible solutions and principles of graphed-based reasoning was applied to identify the most appropriate concept. The ROC system had no pre-existing visual vocabulary and involved over 500 individual components in the form of process steps and toolsets. Again the visualisation reference model played an important role in providing concrete design guidance when implementing the EID principles.

8.4 Building Design Knowledge

These eight characteristics describe properties of cognitive systems that influence the way in which CSE design can be carried out. This taxonomy extends the WTU map, whose implied correlation between environment, constraints and users placed too much emphasis on the work domain as the primary factor governing functionality. While the cognitive systems associated with the PCS and ROC projects appear very similar on the WTU map, their individual characteristics become evident in this new taxonomy (see figure 8.2). This explains why two separate design methodologies were developed; they each inform the practice of analysis and design for two different cognitive systems.

At the same time these two systems share a number of characteristics that differentiate them from more traditional process control domains exemplified by DURESS. An underlying theme in this work has been how work domains incorporating embedded systems differ from such traditional process control applications. The taxonomy highlights these differences, which have a number of implications for design that are supported by the reported methodologies.

Traditional process control systems are primarily concerned with manual and very basic control automation. A human operator controls such a system using a mental model of the constraints and these constraints tend to be defined by physical laws. In contrast to this embedded systems use advanced control and intelligent automation. A human operator must have knowledge of the rules and relationships that regulate an embedded system if they are to understand whether it is functioning correctly. These rules can be considered as policy driven constraints that are defined by performance targets. Control in both cases requires the user to access a model of system functionality however the constraints that define functionality are very different. The methodologies reported here combines work domain analysis with activity analysis to allow both environmental and policy constraints to be identified during design problem structuring.

In terms of task execution and feedback, traditional process control systems aim to support instant reaction to system events. In this sense direct manipulation of interface component are seen to correspond to manipulation of the real world. This is generally achievable due to the fact that the systems representation features a relatively small number of components with which to interact. With embedded systems task execution does not usually relate to manipulation of the real world but involves correspondence with or adjustment of an intermediary agent. This may involve changing a target that regulates an automated controller or requesting an action from a worker. The challenge in this case relates to the number of components (or agents) that make up the system. The shear scale of these systems can make it difficult to identify components and to understand their relationship to the system as a whole. The methodologies reported here attempt to overcome this issue by combining the reasoning supported by ecological design principles with the perceptual efficiencies supported by information visualisation techniques.

In terms of re-usability of design knowledge, as the methodologies describe design artefacts rather than design solutions, they should be applicable to other cognitive systems that exhibit similar characteristics. The taxonomy of cognitive systems provides a tool for analysing and classifying categories of cognitive systems and associated design methodologies. Figures8.2 and 8.3 provide a basic illustration of how this could be achieved. While each and every work domain is highly contextual in terms of environment, practices and vocabularies, this approach enables a designer to identify generic characteristics that can be used to guide the practice of cognitive systems engineering. This approach should enable and encourage the re-use of design knowledge across different work domains.

Chapter 9

Conclusions, Reflections and Future Work

Joint cognitive systems, where responsibility for control is shared between human and machine 'intelligence', are becoming increasingly important in all modern workplaces. System observability plays a critical role in the success of joint cognitive control as it ensures that human and automated agents can co-ordinate their actions and collaborate effectively. However, as automation takes a more pervasive role in large industrial enterprises, the challenge of achieving system observability increases.

This thesis set out to define a design methodology for visual decision support systems that can achieve observability in sociotechnical enterprises. This problem was broken down into a number of questions. Chapter 2 dealt with the question of what analytical approach should be applied to reveal the functionality of a sociotechnical enterprise. Chapter 3 dealt with the question of which representational principles can inform the design of a system image for a sociotechnical enterprise. Chapter 4 focussed on how these factors can be tied together in a comprehensive design process. Given the highly contextual nature of both work systems and design, two further research questions were outlined namely, how can generic design methodologies be generated and how can design knowledge be re-used? The CSE design meta-model developed in chapter 4 provides a means for extracting generic design knowledge from contextual design practice. The utility of this approach is illustrated in the reporting of two design projects in chapters 6 and 7. The re-usability of this design knowledge was discussed in chapter 8 where two related taxonomies of cognitive systems and design methodologies were proposed.

9.1 Contributions

A central theme in this research has been the contrasting agendas of research and practice. As a discipline, cognitive systems engineering aims to produce generic principles that can inform the design of cognitive tools. As a practice, the cognitive systems engineer aims to design highly contextual system images that allow humans to interact with work systems. As a result of this contrast, this research has generated both methodological and domain specfic contributions.

9.1.1 Methodological

Cognitive systems engineering has evolved significantly of the past two decades and many new techniques for analysing and supporting cognitive work have been developed. However, its focus on real-world systems or "cognition in the wild" means that the CSE discipline will continue to face new methodological challenges. Work systems can differ in a multitude of ways, requiring analytical and representational principles to be selected and adapted accordingly. However, to avoid having to re-invent the wheel for every work system, it is important that configurations and adaptations of principles are recorded and made available to other CSE practitioners.

In light of this, the meta-model of CSE design is a major methodological contribution. It provides a conceptual framework that supports both the practice and the reporting of design. By distinguishing between progressive phases of the design, the model requires the overall design process to be considered, from early analysis through to the final design solution. This structure ensures that design gaps are bridged and that the CSE design problem is dealt with in a more comprehensive manner. By distinguishing between primary and secondary design artifacts the model identifies both the contextual and generic characteristics of design. In this manner the key design activity of sketching is included in the model as a means to explore and commit to design concepts during contextual design practice. Analytical and representational principles are presented as generic secondary design artifacts that control this conceptualization process. As demonstrated through the two design projects reported here, this meta-model provides a conceptual tool for extracting generic design methodologies from contextual design practice.

Two new design artefacts were generated from this work each of which mark methodological contributions to the domain. While the abstraction hierarchy is a well established technique for modelling causal relationships in physical domains, the Intentional Goal Model applies this approach to situations involving intentional constraints. This artefact allows an analyst to identify how goal balancing occurs and how intentional decision making can be distributed across a team. The information requirements matrix is another design artefact that is useful for integrating requirements identified through a variety of knowledge elicitation techniques. It further supports the design process by structuring information requirements at different levels of abstraction and by explicitly indicating data relationships. Furthermore the IR matrix can be easily extended to specify data sources making it very useful during the development of the final data-driven displays.

A fourth contribution is the design methodology and associated exemplar reported in chapter 6. The methodology informs the design of visual decision support for monitoring large-scale, process control systems. This methodology extends existing CSE knowledge by using information visualization techniques to implement ecological interface design principles. This approach reduces the representational design gap and supports the application of EID to enterprise-scale systems. Furthermore, it shows how work domain analysis can be augmented with activity analysis in order to provide a level of detail that informs data transformations and visual encoding during representational design. Through the design exemplar, the manner in which these secondary design artifacts support conceptualization is made explicit.

A fifth contribution is the design methodology and associated exemplar reported in chapter 7. This methodology describes how to understand and support changes to operations control in a sociotechnical enterprise. It demonstrates how multiple analytical frameworks can be integrated to generate an enterprise-level model of functionality and how formative and descriptive models can be used to identify the impact of change on cognitive work. This reduces the analytical gap by presenting analytical models as sketches of functionality that can be applied concurrently. In addition, it shows how EID principles provide high-level guidance during design problem solving but require additional representational principles to support design decisions. The design exemplar illustrates how these analytical models and representational principles inform and control the design process.

A minor methodological contribution is the definition of the two related taxonomies of cognitive systems and design methodologies. These taxonomies provide a new approach for re-using design knowledge in cognitive systems engineering. To date efforts to communicate interface design knowledge have tended to focus on the re-use of pre-designed solutions (Rasmussen et al., 1994; Tidwell, 2005). However, the need to design a system-image that reflects system constraints restricts the utility of this approach. Rather than prescribing design solutions, the two taxonomies outlined in chapter 8 aim to prescribe design methodologies that can guide the practice of CSE design. While these taxonomies are only outlined in this research, their extension and population provides a future research goal.

9.2 Domain Specific Contributions

A number of contributions have also been made in relation to understanding and supporting cognitive work in relation to manufacturing operations control. The issue of achieving system observability is of considerable importance for the future of this industry. As automation becomes more pervasive and moves further into areas of system control, it is critical that the system state is communicated efficiently and accurately to support management and policy decisions. Even as developments in the area of intelligent automated control push production systems closer to a lights-out, fully automated model, representations of system functionality will continue be required. Ultimately, when supervisory-control is replaced by a management-by-exception approach, these representations will provide the only means for humans to inspect, comprehend and intervene in the operation of the functional system (Brann et al., 1996).

The models of functionality identified by this research make an important contribution to understanding the role of cognitive systems in the semiconductor manufacturing enterprise. While the technical factors associated with production are well understood, the cognitive factors associated with controlling the enterprise are less well defined. The social organisation has evolved in line with changing factory designs and much of the knowledge associated with production control exists in a tacit form. This research explicitly defines two cognitive systems within a semiconductor manufacturing enterprise. While the purpose of these analyses were primarily to inform the design of visual decision support systems, these models can support management decisions and future developments in the industry. The formative models describe stable aspects of the system around which workers can develop a system image. These provide structures for communicating information that should be consistent across the enterprise and can be used in future interface projects. The models of distributed cognition describe the configuration of physical, social and information resources involved in system control. These models provide a context for understanding how changes in any one area may affect other aspects of the system. The task models provide a low-level view of the information processing associated with control tasks. A massive increase in sensor systems has made data more available than anytime before. While workers have developed personal strategies for extracting important information and coping with data-overload, these models make these strategies explicit and can inform the development of future information systems.

A second contribution to this domain is a visual decision support system for process control health monitoring. In addition to the interface itself, the accompanying design rationale explains how cognitive systems engineering methods can be applied within the domain to provide a structured approach to visual design. The empirical evaluation verifies that this approach can result in improved performance for a range of cognitive tasks. While the system was developed for the semiconductor industry, the visualisation technique is broadly applicable to similar systems in other manufacturing domains.

A third contribution is a visual decision support system for remote operations control in the semiconductor manufacturing industry. Again the system makes use of ecological visualisations that integrate alternative views of the system and provide high-level indicators of the system state. The system should improve decision support by communicating system information in terms of manufacturing goals and removing the need for mental computation. Again the accompanying design rationale provides a detailed examplar of how CSE can be applied within a sociotechnical enterprise.

9.3 Limitations and Future Work

In addition to the contributions made, this work is subject to certain limitations and also highlights a number of open problems that require further investigation.

The original aim was to provide CSE design approaches for visual decision support in sociotechnical systems. Ideally these should help to guide guide information system developers by outlining a prescriptive step-by-step design process. However, the research presented here does not achieve this and instead suggests that the contextual nature of design makes such a goal unfeasible. Design remains a complex problem but the work does provide a model that outlines the structure of the design problem space. This model shows how prescriptive design principles can form generic, high-level methodologies that can guide and control design practice.

The re-use of the methodologies extracted through the CSE design meta-model are dependent on their association with classes of cognitive systems. However, the taxonomies that are needed to achieve this have only been outlined. In order to make CSE design knowledge widely accessible these taxonomies need to be populated and cross-referenced. Such an undertaking is beyond the scope of this research and would require co-operation from across the CSE community. A research workshop that examines the use of design principles across multiple domains would be a suitable starting point for achieving this and may be conducted in the future.

An empirical evaluation of user performance with the ROCC interface is yet to be completed. While this research has focussed primary on the development of design knowledge rather than evaluation of cognitive systems, an empirical investigation would provide further verification that a CSE approach can produce better designs. The complexity of line management makes it difficult to develop partial scenarios with defined tasks and a proper evaluation would require the development of a simulated microworld that captured the data volumes and relationships of a real world fab environment. While this was outside of the scope of the current research, it may be carried out at a future date.

Finally, the display developed in chapter 7 supports high-level line management activities, but ideally this should exist as part of an application that supports multiple levels of operations control. ROC technicians carry out WIP scheduling in response to higher-level goals but the current interface that supports this activity does not make these goals explicit. In order to develop a fully integrated application for operations control, further analysis of the lower-level goals, constraints and strategies is required. Some research has already been carried out at this level and a number of preliminary design concepts have been developed. Additional research would result in a complete model of operations control that would provide a valuable resource for the semiconductor manufacturing domain and a useful exemplar for the cognitive system engineering discipline.

References

M. Ackerman and C. Halverson. Considering an organization's memory. In *CSCW '98: Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pages 39–48, New York, NY, USA, 1998. ACM. ISBN 1-58113-009-0. doi: http://doi.acm.org/10.1145/289444.289461.

C. Alexander. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, USA, 1977.

J. Ammons, T. Govindaraj, and C. Mitchell. Decision models for aiding fms scheduling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 18(5):744–756, 1988. ISSN 0018-9472. doi: 10.1109/21.21601.

B. Archer. The nature of research. *Co-design*, 2:6–13, 1995.

L. Archer. Systematic method for designers. In N. Cross, editor, *Developments in Design Methodology*, pages 57–82. John Wiley & Sons, Chichester, 1984. Originally published by The Design Council, London 1965.

C. Argyris, R. Putnam, and D. Smith. *Action Science*. Jossey-Bass, San Francisco, 1985.

W. R. Ashby. *Introduction to Cybernetics*. Chapman and Hall, 1956.

K. B. Bennett and J. M. Flach. Graphical displays: implications for divided attention, focused attention, and problem solving. *Human Factors*, 34(5):513–533, Oct 1992.

K. B. Bennett and B. Walters. Configural display design techniques considered at multiple levels of evaluation. *Human Factors*, 43(3):415–434, 2001.

O. W. Bertelsen. Design artefacts: towards a design-oriented epistemology. *Scandinavian Journal of Information Systems*, 12(1-2):15–27, 2001. ISSN 0905-0167.

J. Bertin. *Semiology of Graphics*. The University of Wisconsin Press, Madison, 1983.

H. Beyer and K. Holtzblatt. *Contextual design: defining customer-centered systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA., 1998.

A. M. Bisantz and K. J. Vicente. Making the abstraction hierarchy concrete. *International Journal of Human-Computer Studies*, 40(1):83–117, 1994.

A. Blandford and D. Furniss. Dicot: a methodology for applying distributed cognition to the design of team working systems. In S. Gilroy and M. Harrison, editors, *DSVIS*, LNCS, pages 26–38, Newcastle upon Tyne, UK, 2005. Springer.

C. Bonaceto and K. Burns. A roadmap for cognitive engineering in systems engineering. In *48th Annual Meeting of the Human Factors and Ergonomics Society.*, 2004.

K. Braa and R. Vidgen. Action case: Exploring the middle kingdom in is research methods. In *Computers in Context: Joining Forces in Design*, Aarhus, Denmark, 1995.

D. M. Brann, D. A. Thurman, and C. M. Mitchell. Human interaction with lights-out automation: A field study. In *Symposium on Human Interaction with Complex Systems*, volume 00, pages 276–283, Dayton, OH, 1996. IEEE Computer Society.

C. M. Burns and J. R. Hajdukiewicz. *Ecological Interface Design*. CRC Press, Boca Raton, FL, 2004.

M. A. Buttigieg and P. M. Sanderson. Emergent features in visual display design for two types of failure detection tasks. *Human Factors*, 33(6):631–651, Dec 1991.

B. Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann, San Francisco, 2007.

P. Carayon. *Handbook of Human Factors and Ergonomics in Health Care and Patient Safety*. Routledge, 2006.

S. K. Card, A. Newell, and T. P. Moran. *The Psychology of Human-Computer Interaction.* Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, 1983. ISBN 0898592437.

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

C. Carlsson and E. Turban. Introduction: Dss: Directions for the next decade. *Decision Support Systems*, 33(2):105–110, 2002. ISSN 0167-9236. doi: http://dx.doi.org/10.1016/S0167-9236(01)00137-3.

C. M. Carswell. Choosing specifiers: an evaluation of the basic tasks model of graphical perception. *Human Factors*, 34(5):535–554, 1992. 152486.

P. Checkland and J. Scholes. *Soft Systems Methodology in Action.* John Wiley and Sons Ltd., 1990.

A. Cherns. The principles of sociotechnical design. *Human Relations*, 29(8):783–792, 1976.

K. Christoffersen and D. Woods. How to make automated systems team players. *Advances in Human Performance and Cognitive Engineering Research*, 2:1–12, 2002. Position Paper.

W. S. Cleveland. *The Elements of Graphing Data.* Chapman and Hall, New York, 1985.

W. S. Cleveland and R. McGill. Graphical perception and graphical methods for analyzing scientific data. *Science*, 229(4716):828–833, 1985.

K. Craik. *The Nature of Explanation.* Cambridge University Press, Cambridge, 1943.

M. Cummings. Can cwa inform the design of networked intelligent systems? In *Moving Autonomy Forward Conference*, Grantham, England., June 2006. URL `web.mit.edu/aeroastro/labs/halab/papers/Cummings2006_CWA.pdf`.

S. W. A. Dekker and D. D. Woods. To intervene or not to intervene: the dilemma of management by exception. *Cognition, Technology and Work*, 1:86–96, 1999.

J. Dowell and J. Long. Conception of the cognitive engineering design problem. *Ergonomics*, 41(2):126 – 139, 1998.

P. Duez and K. J. Vicente. Ecological interface design and computer network management: the effects of network size and fault frequency. *International Journal of Human-Computer Studies*, 63(6):565–586, 2005. ISSN 1071-5819. doi: http://dx.doi.org/10.1016/j.ijhcs.2005.05.001.

Y. Engeström. *Learning by expanding: An activity-theoretical approach to developmental research.* Orienta-Konsultit., Helsinki, 1987.

T. Erickson. Theory theory: A designers view. *Computer Supported Cooperative Work (CSCW)*, 11:269–270, 2002.

F. Flores, M. Graves, B. Hartfield, and T. Winograd. Computer systems and the design of organizational interaction. *ACM Transactions on Office Information Systems*, 6: 153–172, 1988.

N. R. T. Force. Next generation manufacturing: A framework for action. NGM Project Office, Bethlehem, PA, 1997.

W. R. Garner. *The processing of information and structure.* Lawrence Erlbaum Associates, Potomac, Md., 1974.

J. Gibson. *The Ecological Approach to Visual Perception.* Houghton Mifflin, Boston, 1979.

V. Goel and P. Pirolli. The structure of design problem spaces. *Cognitive Science*, 16 (3):395–429, July-September 1992.

E. Goldratt and J. Cox. *The Goal.* Creative Output Books, USA, 1986.

G. Goldschmidt. Capturing indeterminism: representation in the design problem space. *Design Studies*, 18(4):441–455, October 1997. Descriptive models of design.

J. Gould. *Handbook of human-computer interaction*, chapter How to design usable systems, pages 757–789. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. ISBN 1-55860-246-1.

C. Halverson. Activity theory and distributed cognition: Or what does cscw need to do with theories? *Computer Supported Cooperative Work (CSCW)*, 11:243–275, 2002.

P. Higgins. Extending cognitive work analysis to manufacturing scheduling. In *Proc. Australasian Computer Human Interaction Conference*, pages 236–243, 30 Nov.–4 Dec. 1998. doi: 10.1109/OZCHI.1998.732220.

R. Hoffman. Protocols for cognitive task analysis. on-line, March 2005. URL `/http://ihmc.us:16080/research/projectsCTAProtocols`. available at http://ihmc.us:16080/research/projectsCTAProtocols.

J. Hollan, E. Hutchins, and D. Kirsh. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction*, 7(2):174–196, 2000. URL `citeseer.ist.psu.edu/hollan00distributed.html`.

E. Hollnagel. From function allocation to function congruence. In S. Dekker and E. Hollnagel, editors, *Coping with computers in the cockpit*, pages 29–53. Ashgate, London, 1999.

E. Hollnagel and D. D. Woods. Cognitive systems engineering: new wine in new bottles. *International Journal of Man-Machine Studies*, 18(6):583–600, 1983. ISSN 0020-7373.

E. Hollnagel and D. D. Woods. *Joint cognitive systems: Foundations of cognitive systems engineering*. Taylor and Francis, Boca Raton, FL, 2005.

D. Howie and K. Vicente. Making the most of ecological interface design: the role of self-explanation. *International Journal of Human-Computer Studies*, 49:651–674, November 1998. doi: doi:10.1006/ijhc.1998.0207. URL `http://www.ingentaconnect.com/content/els/10715819/1998/00000049/00000005/art90207`.

E. Hutchins. *Cognition in the wild*. MIT Press, Cambridge, MA, 1995a.

E. Hutchins. How a cockpit remembers its speeds. *Cognitive Science*, 19(3):265–288, 1995b.

G. A. Jamieson, C. M. Miller, W. H. Ho, and K. J. Vicente. Integrating task and work domain-based work analyses in ecological interface design: A process control case study. *IEEE Transactions on Systems, Man and Cybernetics*, 37:887–905, 2007.

V. Kaptelinin. Activity theory: implications for human-computer interaction. In *Context and consciousness: activity theory and human-computer interaction*, pages 103–116. Massachusetts Institute of Technology, Cambridge, MA, USA, 1995. ISBN 0-262-14058-6.

V. Kaptelinin, B. A. Nardi, and C. Macaulay. Methods & tools: The activity checklist: a tool for representing the "space" of context. *interactions*, 6(4):27–39, 1999. ISSN 1072-5520. doi: http://doi.acm.org/10.1145/306412.306431.

D. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler. Challenges in visual data analysis. In *Information Visualization*, pages 9–16. IEEE, 2006.

B. Kirwan and L. Ainsworth. *A Guide to Task Analysis*. Taylor and Francis, London, 1992.

A. Kobsa. An empirical comparison of three commercial information visualization systems. In *INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, page 123, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-1342-5.

P. Koopman and R. Hoffman. Work arounds, make work, and kludges. *IEEE Intelligent Systems*, 18(6):70–75, 2003.

S. Kosslyn. *Elements of Graph Design*. W.H. Freeman and Company, 1993.

K. Kuutti. *Context and consciousness: activity theory and human-computer interaction*, chapter Activity theory as a potential framework for human-computer interaction research, pages 17–44. Massachusetts Institute of Technology, Cambridge, MA, USA, 1995. ISBN 0-262-14058-6.

J. Lamping, R. R., and P. Pirolli. A focus and context technique based on hyperbolic geometry for visualizing large hierarchies. In *ACM CHI '95*, pages 401–408, 1995.

J. H. Larkin and H. A. Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11(1):65–100, 1987.

P. Laseau. *Graphic Thinking for Architects and Designers*. Wiley, 3 edition, 2000.

A. N. Leontev. *Activity, Consciousness, Personality*. Prentice Hall, Englewood Cliffs, NJ,, 1978.

S. Lepreux, M. Abed, and C. Kolski. A human-centred methodology applied to decision support system design and evaluation in a railway network context. *Cognition, Technology and Work*, 5(4):248–271, December 2004.

K. Lewin. *Resolving Social Conflicts*. Harper, New York, 1948.

Y. Lin, W. Zhang, R. Koubek, and R. Mourant. On integration of interface design methods: Can debates be resolved? *Interacting with Computers*, 18(4):709–722, 2006.

M. Lind. Making sense of the abstraction hierarchy in the power plant domain. *Cognition, Technology and Work*, 5(2):67–81, June 2003.

N. Marmaras and D. Nathanael. Cognitive engineering practice: melting theory into reality. *Theoretical Issues in Ergonomics Science*, 6(2):109–127, 2005.

E. S. Meieran. 21st century semiconductor manufacturing capabilities. *Intel Technology Journal*, Q4:0, 1998.

G. Moore. Cramming more components onto integrated circuits. *Electronics Magazine*, 38:–, 1965.

T. P. Moran and J. Carroll. *Design rationale: concepts, techniques, and use*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, 1996. ISBN 0-8058-1567-8.

C. Mouli. Next generation manufacturing execution system (mes) enabling fully integrated fab automation in 300mm technology development and manufacturing. In *IEEE/SEMI, Advanced Semiconductor Manufacturing Conference*, pages 77–85, April 11-12 2005.

C. Mouli and K. Srinivasan. Intel automation and its role in process development and high volume manufacturing. In O. Graf, editor, *Advanced Semiconductor Manufacturing Conference, ASMC '04*, pages 313–320, 2004.

B. Myers, S. E. Hudson, and R. Pausch. Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction*, 7(1):3–28, 2000.

M. Nadin. Interface design: A semiotic paradigm. *Semiotica*, 69(3/4):269–302, 1988.

B. Nardi. *Context and consciousness: activity theory and human-computer interaction*. Massachusetts Institute of Technology, Cambridge, MA, USA, 1995. ISBN 0-262-14058-6.

D. Nathanael, N. Marmaras, B. Papantoniou, and N. Zarboutis. Socio-technical systems analysis: Which approach should be followed? In S. Bagniara, S. Pozzi, A. Rizzo, and P. Wright, editors, *Cognition, Culture and Design*, pages 137–142. Instututo di Sienze et Tecnologie dela Cognizione, 2002. URL `http://simor.mech.ntua.gr/ergou/Docs/Nathanael\&Marmaras\&Papantoniou\&Zarboutis_ECCE11.pdf`.

A. Newell. *Human Problem Solving*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1972. ISBN 0134454030.

A. Newell and S. K. Card. The prospects for psychological science in human-computer interaction. *Human-Computer Interaction*, 1(3):209–242, 1985. URL `http://www.informaworld.com/10.1207/s15327051hci0103_1`.

D. A. Norman. *The Design of Everyday Things*. Basic Books (Perseus), New York, 2002.

D. A. Norman. Affordances and design, June 2007. Available online at www.jnd.org.

O. Paez, J. Dewees, A. Genaidy, S. Tuncel, W. Karwowski, and J. Zurada. The lean manufacturing enterprise: An emerging sociotechnological system integration. *Human Factors and Ergonomics in Manufacturing*, 14(3):285–306, 2004. ISSN 1090-8471. doi: http://dx.doi.org/10.1002/hfm.v14:3.

R. Parasuraman, T. Sheridan, and C. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics*, 30:286–297, 2000.

D. Peebles and P. Cheng. Modeling the effect of task and graphical representation on response latency in a graph reading task. *Human Factors*, 45(1):28–46, Spring 2003.

C. S. Peirce. *Collected Papers*. MA: Harvard University Press, 1931.

M. Perry. The application of individually and socially distributed cognition in workplace studies: Two peas in a pod? In *European Conference on Cognitive Science*, pages 87–92, Siena, Italy, 1999.

J. Petersen and M. May. Scale transformations and information presentation in supervisory control. *International Journal of Human-Computer Studies*, 64(5):405–419, 2006.

J. J. Planta. 300mm fab design : A total factory perspective,. In *International Semiconductor Manufacturing Conference*, pages 5–8, October 1997.

M. Potter, B. Quill, G. Aglipay, E. Anderson, L. Rowitz, L. Smith, J. Telfair, and C. Whittaker. Demonstrating excellence in practice-based research for public health. Public Health Report 121, Center for Public Health Practice, Graduate School of Public Health, University of Pittsburgh, January 2006.

S. S. Potter, W. C. Elm, E. M. Roth, J. Gualtieri, and J. Easter. Bridging the gap between cognitive analysis and effective decision aiding. In M. D. McNeese and M. A. Vidulich, editors, *Cognitive Systems Engineering in Military Aviation Environments: Avoiding Cogminutia Fragmentosal*, pages 137–168. Human Systems Information Analysis Center, Wright-Patterson AFB, OH, 2002.

S. Pugh. *Total Design: integrated methods for successful product engineering*. Addison Wesley, 1990.

P. Rand. *A Designer's Art*. Yale University Press, New Haven, CT, 1985.

J. Rasmussen. Skills, rules, knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man and Cybernetics*, 13:257–266, 1983.

J. Rasmussen and A. Jensen. Mental procedures in real-life tasks: A case of electronic trouble shooting. *Ergonomics*, 17(3):293–307, May 1974.

J. Rasmussen and M. Lind. Coping with complexity. In *European conference on human decision and manual control*, The Netherlands: Delft, 1981.

J. Rasmussen, A. M. Pejtersen, and K. Schmidt. Taxonomy for cognitive work analysis. In B. M. M. d. L. J. Rasmussen, J.; Brehmer, editor, *Taxonomy for analysis of work domains*, volume 1 of *MOHAWC workshop*, pages 1–153. Risø National Laboratory., May 1990.

J. Rasmussen, A. M. Pejtersen, and L. P. Goodstein. *Cognitive systems engineering*. John Wiley and Sons, Inc., New York, NY, USA, 1994. ISBN 0-471-01198-3.

I. Report. Yield and yield management. Technical report, Integrated Circuit Engineering Corporation, Scottsdale, AZ., 2000.

Y. Rogers. New theoretical approaches for human-computer interaction. *Annual Review of Information Science and Technology*, 38(1):87–143, September 2005.

C. Rust, J. Mottram, and J. Till. Practice-led research in art, design and architecture. Technical Report version 2, Arts and Humanities Research Council, November 2007. URL `http://www.ahrc.ac.uk/apply/research/sfi/ahrcsi/research_reviews.asp`.

N. Sarter, D. Woods, and C. Billings. Automation surprises. In G.Salvendy, editor, *Handbook of human factors and ergonomics*, pages 1926–1943. Wiley, New York, 2 edition, 1997.

M. Scaife and Y. Rogers. External cognition: how do graphical representations work? *International Journal of Human-Computer Studies*, 45(2):185–213, 1996.

M. Scaife, Y. Rogers, F. Aldrich, and M. Davies. Designing for or designing with? informant design for interactive learning environments. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 343–350, New York, NY, USA, 1997. ACM. ISBN 0-89791-802-9. doi: http://doi.acm.org/10.1145/258549.258789.

T. Sheridan and W. Verplank. Human and computer control of undersea teleoperators. in man-machine systems laboratory report. Technical report, Cambridge: MIT, 1978.

B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proc. IEEE Symposium on Visual Languages*, pages 336–343, 1996. doi: 10.1109/VL.1996.545307.

H. Siirtola. Interaction with the reorderable matrix. In *IEEE International Conference on Information Visualization*, pages 272–277, 1999.

H. Simon. A behavioural model of rational choice. *Quarterly Journal of Economics*, 69: 99–118, 1955.

H. Simon. *The sciences of the artificial.* MIT Press, Cambridge, MA, USA, 1996. ISBN 0-262-69191-4.

K. Srinivasan. The evolving role of automation in intel microprocessor development and manufacturing. *Intel Technology Journal*, 5(3):–, 2001.

S. Stevens. On the theory of scales of measurement. *Science*, 103:677–680, 1946.

J. Tidwell. *Designing Interfaces: Patterns for Effective Interaction Design.* O'Reilly Media, 2005.

E. Trist. *Experimenting with Organizational Life: The Action Research Approach*, chapter Engaging with large-scale systems, pages 43–75. Plenum, New York, 1976.

E. R. Tufte. *The Visual Display of Quantitative Information.* Graphics Press, Cheshire, Conn, 1983.

C. Upton and G. Doherty. Adapting the ads for high volume manufacturing. In M. F. Costabile and F. Paterno, editors, *INTERACT*, volume 3585 of *Lecture Notes in Computer Science*, pages 1038–1041, Rome, Italy, 2005. Springer.

C. Upton and G. Doherty. Evaluation of an ecological design for process control health monitoring. Technical Report TCD-CS-207-32, Department of Computer Science, Trinity College Dublin, 2007.

C. Upton and G. Doherty. Designing decision support in an evolving sociotechnical enterprise. *Cognition, Technology and Work*, 2008. doi: 10.1007/s10111-008-0124-1.

K. Vicente. Representation aiding for problem solving in process control systems. In *Proc. IEEE International Conference on Systems, Man, and Cybernetics 'Decision Aiding for Complex Systems*, pages 1189–1194 vol.2, 1991. doi: 10.1109/ICSMC.1991.169849.

K. J. Vicente. Coherence- and correspondence-driven work domains: Implications for systems design. *In Behaviour and Information Technology*, 9(6):493–502, 1990.

K. J. Vicente. *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work*. Erlbaum and Associates, Mahwah, NJ, 1999.

K. J. Vicente and J. Rasmussen. The ecology of human-machine systems ii: Mediating 'direct perception' in complex work domains. *Ecological Psychology*, 2(3):207 – 249, September 1990. doi: 10.1207/s15326969eco0203_2. URL `http://www.informaworld.com/10.1207/s15326969eco0203_2`.

K. J. Vicente and J. Rasmussen. Ecological interface design: theoretical foundations. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(4):589–606, 1992. 0018-9472.

G. Waloszek. Fighting (with) hierarchies part ii: Presentation, 25/07/2006 2004.

M. W. Wartofsky. Perception, representation, and the forms of action: toward an historical epistemology. In M. W. Wartofsky, editor, *Models*, pages 188–210. Dordrecht: D. Reidel Publishing Company, 1973.

M. Wattenberg. Visualizing the stock market. In *CHI '99: CHI '99 extended abstracts on Human factors in computing systems*, pages 188–189, New York, NY, USA, 1999. ACM. ISBN 1-58113-158-5. doi: http://doi.acm.org/10.1145/632716.632834.

G. Weinberg. *The psychology of computer programming*. Van Nostrand Reinhold, New York, 1971.

C. Wickens and C. M. Carswell. The proximity compatibility principle: Its psychological foundation and relevance to display design. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(3):473–494, 1995.

C. Wickens, S. Gordon, and Y. Liu. *An introduction to human factors engineering*. New York: Longman, 1998.

J. Womack. The future of u.s. manufacturing: Auto assemblers and suppliers. Technical report, Joint Economic Committee, One Hundred Second Congress, First Session Hearing, December 1991.

B. Wong. The cognitive engineering design gap. In *Australian Conference on Computer Human Interaction OzCHI99*, pages 196–198, Charles Stuart University, Wagga Wagga, Australia, 1999. Ergonomics Society of Australia CHISIG.

W. Wong, P. Sallis, and D. O'Hare. The ecological approach to interface design: applying the abstraction hierarchy to intentional domains. In *Australasian Conference on Computer Human Interaction*, pages 144–151, Adelaide, SA, Australia, 1998.

L. Wood. *User Interface Design: Bridging the Gap from User Requirements to Design*. CRC Press, Inc., Boca Raton, FL, USA, 1997. ISBN 0849331250.

D. Woods. The cognitive engineering of problem representations. In G. W. . J. Alty, editor, *Human-Computer Interaction and Complex Systems*. Academic Press, 1991.

D. Woods. Towards a theoretical base for representation design in the computer medium: Ecological perception and aiding human cognition. In J. Flach, P. Hancock, J. Caird, and K. Vicente, editors, *Global perspectives on the ecology of human-machine systems*. Erlbaum, Hillsdale, NJ, 1995.

D. Woods. Discovering how distributed cognitive systems work. In E. Hollnagel, editor, *Handbook of Cognitive Task Design*. Erlbaum, Hillsdale, NJ:, 2003. URL http://www.itee.uq.edu.au/~cerg/misc/Woods-inHollnagel2003.pdf.

D. D. Woods. *Human centered systems: Information, interactivity, and intelligence*, chapter Human-centered software agents: Lessons from clumsy automation, pages 288–293. National Science Foundation., Washington, DC, 1997a.

D. D. Woods. The theory and practice of representation design in the computer medium (ver. 4.2). Technical report, Cognitive Systems Engineering Laboratory, The Ohio State University, 1997b.

D. D. Woods. Designs are hypotheses about how artifacts shape cognition and collaboration. *Ergonomics*, 41:168–173(6), 1998. URL http://www.ingentaconnect.com/content/tandf/terg/1998/00000041/00000002/art00011.

D. D. Woods, L. J. Johannesen, R. I. Cook, and N. B. Sarter. *Behind human error: Cognitive systems, computers and hindsight.* CSERIAC, Columbus, Ohio, 1994.

D. D. Woods, E. Patterson, and J. Corban. Apollo 13 where's waldo game. Technical report, Cognitive Systems Engineering Laboratory, Ohio State University, 1999. URL csel.eng.ohio-state.edu/woods/design/principles/wheres_waldo_Apollo13.pdf. accessed 12/04/2006.

D. Yeaman and P. Stachura. Reducing costs with the farmed fab layout. In *Semiconductor Equipment and Materials International*, San Francisco, USA., 2002.

C. Yen, M. Woolley, and K. Hsieh. Action case research: a method for the accumulation of design theory/practice knowledge in practice. *Working Papers in Art and Design*, 2:1, 2002. URL http://www.herts.ac.uk/artdes/research/papers/wpades/vol2/yenfull.html.

J. Zhang. A representational analysis of relational information displays. *International Journal of Human-Computer Studies*, 45(1):59–74, 1996. 238922.

J. Zhang and D. A. Norman. Representations in distributed cognitive tasks. *Cognitive Science*, 18(1):87–122, 1994.

J. Zhang and V. L. Patel. Distributed cognition, representation, and affordance. *Pragmatics and Cognition*, 14:333–341(9), 2006.