

Adapting the ADS for High Volume Manufacturing

Connor Upton, Gavin Doherty

Distributed Systems Group, Dept of Computer Science, Trinity College Dublin

Abstract. Cognitive Work Analysis (CWA) is a methodology for analyzing complex socio-technical systems. It aims to structure system information in a manner that is meaningful for human control and interaction. The Abstraction Decomposition Space (ADS) is an important tool used during the first phase of CWA to describe the work domain. In this paper we create an ADS for a Semiconductor Fabrication Plant (Fab). This is a High Volume Manufacturing environment and its complexity necessitates a number of adjustments to the original ADS technique. The physical decomposition of the system is de-emphasized and a number of alternative decomposition hierarchies are used instead. The analysis aims to produce artifacts that can aid in the design of decision support systems. These artifacts not only help to assess the information needs of workers, but also structure the work domain in a manner that will inform display design.

1. Introduction

The correct visual representation of data has been shown to improve user performance and reduce human error in a range of domains [1] and many guidelines exist for the correct visual encoding of quantitative data [2,3]. Advances in sensor and communications technology means that more data is now being generated than ever before. Automated control systems are frequently used to process this data but human operators are often relied on to step in and assume system control if required. In these cases operators must examine data to evaluate the system state and make decisions. The complexity of these domains means that the challenge is not only how to encode the data visually, but also how to decide what data is required for the tasks at hand and how to navigate through the information space. These systems, described by Vicente [4] as Complex Socio-technical Systems, generally involve: large problem spaces, multiple users, conflicting constraints, dynamic data, coupled components and unanticipated events. These attributes make it difficult to apply a purely task-oriented analysis approach when designing user interfaces. Cognitive Work Analysis (CWA) [4] is an alternative approach that attempts to structure system information in a manner that is meaningful for human control and interaction. It produces a number of design artifacts that can inform a UI designer about both the system and the user's information requirements.

2. The Abstraction-Decomposition Space

CWA structures system information using multiple levels of abstraction. The aim is to support reasoning about a system rather than providing a set path of interaction towards a predefined goal. The approach is closely tied to Rasmussen's SRK Taxonomy

[5] which demonstrates the relationship between causal reasoning and abstraction hierarchies. A concept of the taxonomy is that low level data goes through a series of abstractions into higher levels of information until it can be judged against performance criteria. The Abstraction Decomposition Space (ADS) is an important tool used in work domain analysis, the first phase of CWA. The ADS combines two views of a system, a functional means-end hierarchy, ranging from high-level functional purpose to low-level physical form, and a physical decomposition hierarchy, ranging from overall system to individual components. These hierarchies are flattened and placed orthogonally in a matrix, essentially mapping function to form at different levels of granularity. The result is that each cell in the matrix describes the entire system at a different level of abstraction. This tool allows us to chart the information requirements of a user at various levels of abstraction during in a problem solving task. The most cited application of the ADS is DURESS [4], a micro-world designed to be representative of industrial process-control scenarios. While it successfully demonstrates the use of the ADS for mapping information needs, the scale and complexity of the system is relatively small.

3. Applying the ADS to High Volume Manufacturing

Modern High Volume Manufacturing (HVM) environments are examples of extremely complex Socio-Technical Systems. They combine sophisticated factory automation with the changing demands of dynamic markets. A common constraint across HVM is the conflicting goals of achieving high volumes of production while ensuring that machinery continues to operate within acceptable control limits. High production volumes place machinery under stress, requiring them to receive more maintenance and repair. Repair causes more downtime leading to lower levels of production. This conflict is resolved by humans who must reconcile manufacturing-focused and engineering-focused priorities. A visualization that could present system state information from both perspectives would benefit users trying to deal with such conflicts. We apply a CWA analysis to a HVM system to structure system information in a way that can inform our visualization design.

Our study focuses on in Semiconductor Fabrication Plants (Fabs), involving hundreds of machines (described as tools) and a highly complex process-flow. The overall *process* is divided into a number of *segments*. Segments consist of a number of functional *operations* that build components of the semiconductor. These operations may be repeated with slight variations in different segments, introducing circulation and re-entries into the process-flow. Operations are carried out on specific *tools* which are categorized according to specific functional activities. Multiple tools carrying out the same operation are gathered together into a *toolset*. Groups of toolsets that carry out the same general function form a *functional area*. Product moves through the process on an automated material handling system. This complex relationship between process-flow and functional areas is shown in fig. 1a.

Two basic structures are evident. A Process hierarchy allows us to organize the system into different levels of granularity according to position in the process-flow. This equates to the manufacturing view mentioned earlier and facilitates a horizontal view across the process-flow. A Functional hierarchy allows us to think about the system in

terms of functional areas. It equates to an engineering view and gives a vertical view down into areas, toolsets and tools.

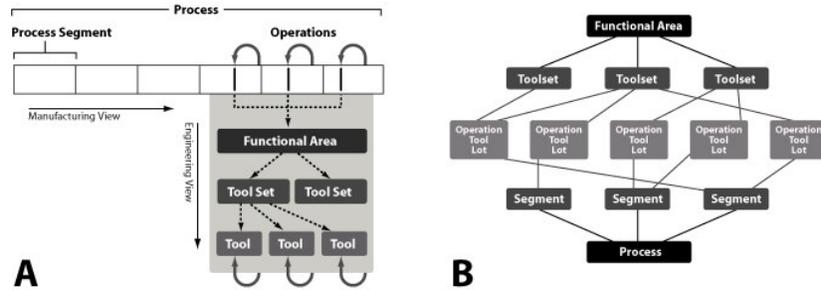


Fig. 1. A) Process-Flow & Functional Area Structures B) Abstraction Lattice

		Decomposition →		
		Process	Segment	Operation
↑ Abstraction	Functional Purpose	Efficient Product Manufacturing		
	Abstract Function	Move Product through Process	Advance wafer production	
	Generalised Function		Carry out Operations	Carry out an operation
	Physical Function			Lot/Tool States (Production)
	Physical Form			Wafer in Tool

		Func Areas	Toolsets	Tool
↑ Abstraction	Functional Purpose	Maximise Tool Availability		
	Abstract Function		Max. Uptime Min. Downtime	
	Generalised Function		Toolset Health Toolset Availability	Carry out PM's Find Faults Fast
	Physical Function			Tool States (Health)
	Physical Form			Wafer in Tool

		Process	Segment	Operation
Functional Purpose	Produce a Technology			
Abstract Function	Move Product through Process	Advance wafer production		
Generalised Function		Carry out Operations	Carry out an operation	
Physical Function			Lot/Tool States (Production)	
Physical Form			Tool, Lot Operation	
Physical Function			Tool States (Health)	
Generalised Function		Toolset Health Toolset Availability	Carry out PM's Find Faults Fast	
Abstract Function		Maximise Uptime Minimise Downtime		
Functional Purpose	Maximise Tool Availability			
		Func Areas	Toolsets	Tool

Fig. 2. Two ADS's for alternate views & Final ADS

4. Adaptation of the ADS

This system causes a number of problems for the construction of an ADS. The ADS combines a functional abstraction hierarchy with a physical decomposition, but in this case the physical decomposition has limited use. While physical tools match functional operations at the lowest levels, recirculation in the process-flow means that physical and functional relationships no longer equate at higher levels of abstraction, e.g. segment variables do not equate to toolset variables. The usefulness of a physical decomposition as a natural system model is also challenged by the automated transportation of wafers around the fab. If a physical decomposition is no longer valid what other structures can we use? Both of the views mentioned above are valid system decompositions but their relationship is non-analogous. How can we generate a single ADS of the system that uses both structures?

As a first step, two ADS's (fig.2) were constructed and examined, one for each view. While they are very different at the abstraction level of functional purpose, they share the same properties at the level of physical form. This commonality can act as a bridging point between the two views. While a single physical decomposition causes us to think of the ADS as an Abstraction hierarchy, using two conceptual decompositions allows us to think of the ADS in terms of an Abstraction Lattice (fig.1b). An Abstraction Lattice allows us to reason our way down through levels of abstraction in one view and then up through levels of abstraction in an alternative view of the same system. This approach allows us to reflect the Abstraction Hierarchy across the level of physical form joining up the two ADS representations. Our ADS (fig.2) captures all of the system variables from both view at multiple levels of abstraction.

5. Evaluation & Observations

In order to evaluate our adjustments we mapped a use-case scenario for shutting down an Out-of-Control Tool to our ADS. The mapping revealed a number of interesting observations. Firstly, although the user was operating in the engineering area, information from both sides of the ADS was referred to during the use-case. Secondly, information at different levels of abstraction was combined from different sides of the ADS in order to gain a better understanding of understand system state. Thirdly, while causal reasoning enables movement between states of knowledge in either view, this cannot explain movement between abstraction-levels that occurs independently in both views. These observations are particularly interesting for display design as they force us to think about visual representations that can encompass different levels of information abstraction within a display and movement between different displays.

6. Conclusion

While the ADS is a useful tool for structuring system information it has difficulty dealing with the Fab environment. Circulation in the process-flow makes physical/functional relationships problematic at higher levels of abstraction. This reduces the effectiveness of physical decomposition as a natural structure. By combining alternative decompositions of the system it becomes possible to structure information in a manner that is more meaningful to users. This modified ADS is a useful tool for charting users information needs when interacting with the system. By embedding the structures revealed by the ADS into a display design we can provide visual interfaces that better match the system constraints. While a preliminary use-case mapping has been completed, more are being carried out to further test the ADS. This is being used as part of an approach to analysis and design of displays for a HVM environment.

References

1. Woods DD, Johannesen L, Cook RI, Sarter N. Behind Human Error: Cognitive Systems, Computers and Hindsight. CSERIAC, WPAFB, Dayton OH. (1994).
2. J. Bertin. Semiology of Graphics. The University of Wisconsin Press, Madison, (1983)
3. Tufte, E. R. The Visual Display of Quantitative Information. Graphics Press, Conn. (1983)
4. Vicente, K. J., Cognitive Work Analysis: Towards Safe, Productive, and Healthy Computer-based Work, Mahwah, NJ: Erlbaum (1999).

5. Rasmussen, J., 1979, On the Structure of Knowledge - a Morphology of Mental Models in a Man-Machine System Context, Riso Report Riso-M-2192 (Roskilde. Denmark : Riso National Laboratory).