Proceedings of the Human Factors and Ergonomics Society Annual Meeting
http://pro.sagepub.com/

Task "Versus" Work Domain Analysis Techniques: A Comparative Analysis

Christopher A. Miller and Kim J. Vicente

Proceedings of the Human Factors and Ergonomics Society Annual Meeting 1999 43: 328
DOI: 10.1177/154193129904300342

TASK “VERSUS” WORK DOMAIN ANALYSIS TECHNIQUES: A COMPARATIVE ANALYSIS

Christopher A. Miller
User Centered Design Technology Group
Honeywell Technology Center
Minneapolis, Minnesota

Kim J. Vicente
Cognitive Engineering Laboratory
Department of Mechanical and Industrial Engineering
University of Toronto
Toronto, Ontario

Work analysis techniques are the primary methods for designers to obtain knowledge for good interfaces. The majority of current techniques are either task-centered or system- or work domain-centered. In previous work (Miller and Vicente, 1998) we maintained that each technique focuses on different aspects of the design problem and has complementary strengths and weaknesses, thus mandating a unification for completeness. Here, we compare the results of two analyses of the same work environment, one using a work domain-centered technique (Rasmussen’s (1985) Abstraction Decomposition Space or Abstraction Hierarchy) and the other using a task-centered technique (Hierarchical Task Analysis—Shepherd, 1989). We compare the requirements produced by each analytic technique and demonstrate their complementary nature. We argue for examining a work domain from both perspectives and discuss interface concepts that would satisfy both analyses. We argue that these interfaces would provide better user support than one designed from either analytic technique alone.

INTRODUCTION

The discipline of Human Factors Engineering is dependent on our ability to determine what information a human operator needs in order to perform specific work, and then to provide that information in a usable form. Over the years, the field has developed two broad classes of techniques for performing the first of these two steps—information needs determination. Task analysis (Kirwan & Ainsworth, 1992; Diaper, 1989) describes the actions that an actor can or should take to accomplish goals. Work domain analysis techniques (Rasmussen, Pejtersen & Goodstein, 1994; Rasmussen, 1985), by contrast, examine the functional structure of the domain (specifically, the plant or system) in or on which work must be done.

COMPARING ANALYSIS METHODS

Prior to the work described here, we claimed (Miller & Vicente, 1998) that each approach has strengths and weaknesses, though ultimately they reflect different perspectives on (and different avenues to) the full set of knowledge needed for good human-centered system design. A comparison of the strengths and weaknesses of these techniques (derived primarily by conceptual analysis) is presented in Table 1 (Vicente, 1999). One purpose of the work described in this paper was to validate and refine the claims made in Table 1. Since it is rare to analyze the same application problem twice using separate tools, a direct, ‘face-to-face’ comparison of the results produced by two representative analysis methodologies was important as a step in validating the claim that separate and complementary results may be obtained by each analytic approach.

We chose to investigate the interface *requirements* produced by different analytic tools (instead of interfaces produced from those requirements) because the analyses naturally produce requirements; interfaces must be developed from them through human creativity. We chose to do this comparative analysis on Vicente’s (1996) DURESS II system because it is representative of many industrial process control systems, yet is small enough to be manageable. Furthermore

Table 1. Relative advantages and disadvantages of Task Analysis and Work Domain Analysis forms of information requirements analysis (and, by extension, of interfaces designed from information obtained via these analytic techniques).

	TASK	WORK DOMAIN
Mental economy	efficient	effortful
Ability to adapt to unforeseen contingencies	brittle	flexible
Scope of applicability	narrow	broad
Ability to recover from errors	limited	unlimited

work domain analyses using Rasmussen's (1985) Abstraction Decomposition Space (ADS) analysis technique (more commonly known as the Abstraction Hierarchy) had already been repeatedly performed on it. To provide the basis for comparison between the two analytic techniques, we performed a task analysis on the DURESS II system using a representative, familiar, yet easy-to-use task-centered technique known as Hierarchical Task Analysis (HTA-- Shepherd, 1989).

We were not attempting to conduct a 'pure,' side by side, 'shoot off' comparison designed to show which analytic method was 'better'. To have performed such a comparison fairly and accurately would have demanded double-blind experiments with equally trained design engineers. Not only did we not have such individuals available, but we were ultimately uninterested in which approach was 'better' than the other. Instead, we were interested in the *complementary* information produced by the two types of analyses when used in conjunction. Ultimately, if we are to justify the creation of a modeling technique and/or representation which unifies task- and work domain-based analyses, we must show that there are unique contributions from each perspective. In essence, performing one analysis after the other is a conservative approach to demonstrating that point. It might be expected that two separate analyses would produce different results, but if a second analysis can be performed with the full knowledge of the first and *still* produce novel information, that would be stronger evidence for the unique contribution of each approach.

COMPARATIVE ANALYSIS RESULTS

The DURESS II system for which we performed the analyses is a thermal-hydraulic process feedwater system simulation extensively modeled using ADS techniques (Bisantz & Vicente, 1994). Although there was substantial local expertise in operating DURESS II, task-centered modeling techniques had not been applied and task-based interfaces have not been designed for it. Since both the ADS and the HTA techniques are well-documented and reasonably well-known, they will not be described in depth here except to note that each technique was performed with the primary purpose of deriving information requirements for human users of DURESS II. The objective of the analysis is important since both ADS and HTA can be put to other uses (cf. Diaper, 1989; Vicente, 1999) with somewhat different resulting outputs. Nor do we have the space to present the detailed results of either analysis. Instead, we will summarize the types of information produced by the two analyses in Table 2 below.

Table 2. Comparison of the types of requirements knowledge produced by each analytic technique.

Information Requirements Identified in Analysis	Identified in ADS analysis?	Identified in HTA analysis?
Physical appearance and location of work domain components	X	
Physical connections between components	X	
The function and current state of physical components	X	X
Range of possible states for physical components	X	Implicit from multiple comparisons
Actual current behavior of components (Generalized function states: flows and quantities)	X	X
Range of possible behaviors of components	X	Implicit from multiple comparisons
Capability to achieve (and constraints on) general functional behaviors given the states of physical components	X	Implicit (and partial) in procedures and expectation generation
Causal relationships between general functions	X	Implicit (and partial) in procedures and expectation generation
Aggregation of generalized functions into subsystems	X	X (with notion that subsystem definition might be dynamic)
Actual current generalized function state at subsystem level	X	X (with notion that subsystem definition might be dynamic)
Range of possible functional states at subsystem level	X	Implicit from multiple comparisons
Causal connections between subsystem behaviors	X	Implicit (and partial) in procedures and expectation generation

Table 2 (con't). Comparison of the types of requirements knowledge produced by each analytic technique.

Information Requirements Identified in Analysis	Identified in ADS analysis?	Identified in HTA analysis?
Current state of abstract functions at the subsystem level	X	X (with notion that subsystem definition might be dynamic)
Range of possible abstract function states at subsystem level	X	Implicit from multiple comparisons
Capability to achieve (and constraints on) abstract functional behaviors given generalized functional states	X	Implicit (and partial) in procedures and expectation generation
Causal connections between abstract functions	X	Implicit (and partial) in procedures and expectation generation
Current state of functional purpose variables for the system as a whole	X	X
Range of possible states for functional purpose variables	X	Implicit from multiple comparisons
Capability for achieving (and constraints on) overall functional purpose behaviors given abstract functional states	X	Implicit (and partial) in procedures and expectation generation
Specific expected or goal value for physical functions	Implicit from functional behavior capability and constraint information	X
Specific expected or goal value for general functions	Implicit from functional behavior capability and constraint information	X
Specific expected or goal value for abstract functions	Implicit from functional behavior capability and constraint information	X
Specific expected or goal value for functional purpose	X (demand values)	X
Extra-system goal information (duration or cumulative volume; start, stop and change requests)		X
Social-organizational priority and tradeoff information		X
Social-organizational information about operational expectations (likelihood of faults, demand changes, etc.)		X
Explicit strategy choices and functional implications		X
Explicit information to support strategy selection (e.g., sum of D, interface availability)		X
Configuration-dependent subsystem groupings and capacities	Static groupings and implicit (derivable) capacities	X
Distinction between monitoring and controlling information elements	Capabilities discriminated but no information about when which was needed	X
Task dependent, temporal information clustering (sequential vs. parallel presentation, etc.)	Some capability via means-ends relationships	X

GENERAL CONCLUSIONS

For each row in Table 2, the leftmost column describes a type of information that one or both analytic technique identified as required for humans interacting with the DURESS II system. The second column records whether the ADS analysis identified this information as required, while the third column records whether the HTA analysis identified the requirement. An 'X' in either the second or third column indicates that, in the judgement of the analyst, the corresponding technique clearly, obviously and completely identified that type of information as required. By contrast, entries which claim that an information type was "implicitly" identified mean that some sensitivity to the type of knowledge was required in order to complete the analysis, but that the requirement was not as complete or deep, or as easily or explicitly represented in the 'implicit' technique's outputs as it was in the more 'explicit' one. Therefore, the designer using the 'implicit' analytic technique might do as thorough a job of understanding and capturing that knowledge, but the nature of the technique itself made this less likely. For example, the procedures produced in the HTA require an understanding of the underlying functioning of the DURESS II system, but this knowledge could come in the form of reported procedural rules from domain experts. There is no guarantee that such reports would be complete or even necessarily accurate. Furthermore, the understanding of the system's general capabilities and constraints required to produce accurate procedures is not explicitly captured anywhere in the HTA analysis. Instead, this knowledge is 'compiled' (which necessarily means that it is obscured) into procedural rules by the HTA. Thus, an HTA 'implicitly' conveys knowledge about the DURESS II system functions, but it does not 'explicitly' capture or convey that knowledge in depth.

Since the generation of information requirements is only a contributor to the ultimate display that is designed, the fact that an information type is missing from either column leaves open the possibility that a smart designer could intuitively fill that information in. On the other hand, the absence of that information type in the requirements list places a heavier burden on the designer's intelligence and creativity, thereby making errors of omission more likely.

The most general conclusion from our comparison of the two analytic techniques is that the analyses do have unique contributions to offer the interface design process, even when performed sequentially. As can be seen from Table 2, not only are the sets of display requirements produced by the two analyses substantially different, they are also highly complementary. Loosely speaking, the following general conclusions are valid:

The ADS work domain analysis:

- Does a much better job at providing 'deep knowledge' about the full set of constraints and capabilities for system behavior that are inherent in the work domain.
- More readily and directly identifies information requirements for monitoring, controlling and diagnosing the system—by contrast, the task analysis tends to reduce the granularity of tasks to an increasingly finer size, making it progressively easier for the analyst to infer information requirements without actually identifying them.
- Is more independent of the specific context in which the system is used (e.g., its interface, organizational goals, social structure, etc.)

The HTA task analysis:

- Provides 'compiled' procedural knowledge which, while being easier to learn and follow for anticipated cases, hides the deeper rationale for why procedures work and risks unexpected behavior in unexpected situations.
- Is more 'human-centered' in that it focuses more on what the operator must or can do and how s/he naturally thinks about the domain, dividing the set of operational behaviors into discrete chunks (i.e., tasks).
- More readily identifies when, how and with what priority information will be needed to perform expected tasks—and thus is more applicable to prioritizing, sequencing and dynamically configuring information presentations.
- Is less independent of context, but therefore requires a more comprehensive consideration of the full set of factors which influence operator behavior.

APPLICATION TO INTERFACE DESIGN

Finally, although we have not yet designed or evaluated interfaces which attempt to meet both task-centered and work domain-centered sets of requirements, ecological interfaces have been designed for DURESS II on the basis of the ADS analysis alone. These interfaces are described, and performance improvements stemming from using them are documented in Vicente (1996). By concentrating on meeting the unique information requirements presented in the HTA column of Table 2, we were able to suggest the following task-centered enhancements to the ecological interfaces Vicente has developed for DURESS II:

- *Strategy Guides:* explicit templates for selecting an appropriate water or heat input strategy, accompanied by an overlay on the interface to guide control actions in keeping with that strategy and/or intelligent 'critics' (Guerlain, 1995) to aid in strategy implementation.
- *Expectation Indicators*: The prevalence and importance of the expected value for a parameter *given* system state, strategy and goal, implies that marking expectations explicitly might be helpful. User settable 'bugs' or computer-generated expected trend lines are two different implementation strategies.
- *Sequencing Information*: The order in which some actions should be done (for safety, simplicity, or team coordination reasons) could be flagged using left-to-right ordering, sequenced equipment highlighting or 'unitizers', or machine-mediated hard lockouts.
- *Information Suppression or Filtering*: The task analysis shows that some DURESS II information is not needed during some phases of operation. For example, controls for input flow valves and heaters are not needed once normal operations mode is achieved (though presentation of corresponding heat and water flows is still needed).
- *Information Prioritization*: In current DURESS II interfaces, temperature and flow demand states and actual output states are presented in the same size and color as, say, the flow through any input valve. The task analysis shows that these values are used much more frequently. Thus, they should be made more salient.
- *Dynamic Organization*: The ADS analysis identifies a fixed set of subsystems within the DURESS II equipment, but the task analysis makes it clear that the set of components in which a user may naturally be interested depends on the task. Thus, dynamic organization of subsystems (such as showing by grouping or coloration the *current* set of pipes and valves feeding a specific reservoir rather than, or in addition to, the potential set) may be helpful for some diagnosis or control tasks.

- *Additional Information*: Some types of information, such as the social and organization factors that determine how to make tradeoffs among viable strategies (e.g., minimize upsets vs. minimize control actions) and the need for the sum of demands and total mass input capacity, are simply not identified by the ADS, though the HTA indicated such information would be useful.

Again, it is important to stress that these findings in no way show that a task analytic approach is 'superior' to a work domain analysis. The fact that the ADS uncovered many types of information requirements that the HTA 'missed' is ample evidence that if we had been seeking improvements to an existing task-centered display, the list would have been at least as long.

IMPLICATIONS FOR ANALYTIC TECHNIQUES

These results illustrate the importance of a unified analytic approach for interface design. We have shown that the two different approaches each produce different, but complementary, knowledge about the information users will need to operate within the work domain, and about how they will need that information. Furthermore, we have provided hints that it will be possible to integrate the requirements produced by each approach in a single interface. There are, however, a wide variety of ways in which this could be done. More work needs to be done to understand exactly how this can be accomplished most effectively in various domains and, ultimately, to ascertain whether such an integrated interface design will produce better human-machine performance than an interface designed from either perspective alone. Furthermore, the work reported above was conducted on a laboratory simulation of limited complexity. We are concurrently investigating the application of each analytic technique to a practical, real-world system (Acetylene Hydrogenation Reactors as a part of NOVA Chemicals ethylene refineries) to establish whether similar results will obtain in those domains.

ACKNOWLEDGEMENTS

This work has been partially funded by a research grant from NOVA Chemicals Ltd., through the support of the Canadian Natural Sciences and Engineering Research Council, and through internal research support of the Honeywell Technology Center. The authors would like to thank Joanne Cundict, Jamie Errington, Dave Weizenbach, Greg Jamieson and Peter Bullemer for their assistance in developing, steering and conducting this project.

REFERENCES

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

Diaper, D. (1989). Task analysis for human-computer interaction. Ellis Horwood; Chichester, UK.

Guerlain, S. (1995). Using the critiquing approach to cope with brittle expert systems. Proceedings of the Human Factors and Ergonomics Society 39th Annual Meeting, 1. Santa Monica, CA. 233-237.

Kirwan, B. and Ainsworth, L. (1992). A Guide to Task Analysis. Taylor and Francis; Bristol, PA.

Miller, C., and Vicente, K. (1998). Toward an Integration of Task- and Work Domain Analysis Techniques for Human-Computer Interface Design. In Proceedings of the 42nd Annual Meeting of the Human Factors and Ergonomics Society. Chicago, IL; October 5-9.

Rasmussen, J. (1985). The role of hierarchical knowledge representation in decision making and system management. IEEE Transactions on Systems, Man and Cybernetics, 15, 234-243.

Rasmussen, J., Pejtersen, A., and Goodstein, L. (1994). Cognitive Systems Engineering. John Wiley & Sons; New York.

Shepherd, A. (1989). Analysis and training in information technology tasks. In Diaper, D. (Ed.). Task analysis for Human-computer Interaction. Ellis Horwood, Ltd.; Chichester, 15-55.

Vicente, K. J. (1996). Improving dynamic decision making in complex systems through ecological interface design: A research overview. System Dynamics Review, 12, 251-279.

Vicente, K. J. (1999). Cognitive work analysis: Towards safe, productive, and healthy computer-based work. Erlbaum: Mahwah, NJ.