

A theoretical note on the relationship between work domain analysis and task analysis

JOHN R. HAJDUKIEWICZ†‡ and KIM J. VICENTE†§*

†Cognitive Engineering Laboratory, Department of Mechanical and Industrial Engineering, University of Toronto, Canada

‡Advanced Technology Laboratories, Honeywell, 3660 Technology Drive, Minneapolis, MN 55413, USA

§Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, USA

Keywords: Work domain analysis; task analysis; abstraction hierarchy; cognitive work analysis.

The purpose of this note is to clarify the theoretical relationship between work domain analysis and task analysis, two classes of techniques that have been used by cognitive engineers to identify information requirements for systems design. The transformation from a work domain analysis (i.e., from a description of the object of control to a description of control itself) can be conceived as a discrete set of transformations. Work domain analysis identifies the set of all structural degrees of freedom that are available to any actor. Only a subset of these will be relevant for a particular context. At any particular point in time, actors will have to choose which of these relevant degrees of freedom to utilize. Finally, the utilized degrees of freedom will have a dynamic state that can usually be described quantitatively. Task analysis is the function that maps current states onto desired states via a set of human or automated control actions. By making these transformations explicit, the relevance of work domain analysis to worker (or automation) goals and actions becomes more clear.

1. Introduction

Cognitive engineers are all familiar with *task analysis* and the role that it can play in identifying information requirements (Kirwan and Ainsworth, 1992). *Work domain analysis* is a less familiar technique that also aims to inform systems design and that is receiving an increasing amount of attention (Rasmussen *et al.* 1994, Vicente 1999a, Naikar and Sanderson 1999). It is important to understand the relationship between these two types of techniques so that both researchers and designers have a better idea of their relative advantages and disadvantages for systems design. Several attempts have been made to compare task analysis and work domain analysis theoretically (e.g., Vicente 1999a, 1999b). However, these attempts have not been as clear or as thorough as they could, and should, have been. In particular, the generalized set of discrete transformations that link work domain analysis and task analysis have not been fully clarified in previous work. Consequently, there is still some uncertainty about the relationship between the two techniques, even among experienced researchers who have used work domain analysis in their work (e.g., Terrier *et al.* 2000).

*Author for correspondence. e-mail: vicente@mie.utoronto.ca

The purpose of this note is to resolve this theoretical ambiguity. First, a case study illustrating how it is possible to transition from a work domain analysis to a task analysis for a representative laboratory microworld will be provided. Second, the lessons learned from this case study will be generalized theoretically by induction so that they are applicable to other problem domains. This analysis will show that the relationship between work domain and task analysis can be generically conceived as a series of discrete transformations, each of which has a clear conceptual meaning. At one end, work domain analysis identifies information requirements that are event- and time-independent, providing a robust basis for supporting worker adaptation to novelty and change. At the other end, task analysis identifies information requirements that are event- and time-dependent, providing an efficient basis for supporting worker performance to anticipated situations. The discrete transformations that connect these techniques are described, providing a stronger basis for understanding the relationship between work domain and task analysis. The end results are deeper insights into the utility of each technique and an understanding of the relevance of work domain analysis to worker (or automation) goals and actions.

2. Case study

The thermal-hydraulic process control microworld illustrated in figure 1 will be used as the focus of the case study. This simulation consists of two feedwater systems, which can supply water to two reservoirs. Operators have control over eight valves (six input valves: VA, VB, VA1, VA2, VB1, VB2, and two output valves: VO1 and VO2), two pumps (PA and PB), and two heaters (HTR1 and HTR2) within the numerical ranges specified for each component. The operators are required to achieve the dual purposes of satisfying external, dynamic output demands for water (D1g and D2g) while maintaining each of the reservoirs at their respective temperature goals (T1g and T2g).

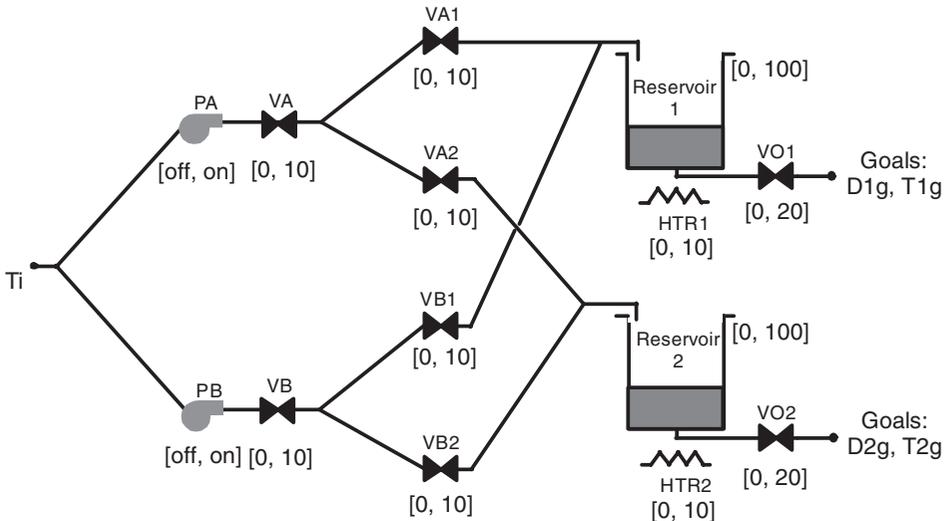


Figure 1. A schematic diagram of the process control microworld. The maximum and minimum ranges of each component are shown in square brackets. Adapted from Vicente and Rasmussen (1990).

2.1. Task analysis

While there are many different ways of doing a task analysis, all of them deal with identifying the goals or actions that need to be pursued by an actor, whether it be a worker or automation. Figure 2 shows a normative sequential flow task analysis for starting up the process in figure 1. This task requires the actor to take the process from a state where all of the components are off and the reservoirs are empty to a steady state where the demand and temperature goals for both reservoirs are all satisfied. If the goal to be achieved were a different one (e.g., shut down the process), then the results of the task analysis could be markedly different. Thus, task analysis is *event-dependent* (Vicente 1999a). And because the goals that the operator is pursuing change over time, task analysis is also time-dependent.

To simplify the discussion, we will focus only on the mass flow and omit the energy flow aspects of the startup task. Starting from the top left of figure 2, the actor will need to check the state of the work domain. If the target water demand exceeds the combined capacity of the two feedwater streams (i.e., 20 units/s of flow), then the task cannot be performed. Next, the states of the valves, pumps, and reservoirs are checked. If the process is totally shutdown, then VA, VA1, VA2, VB, VB1, VB2 must be opened to their maximum settings, then pumps PA and PB turned on. Next, the actor must check the level in the reservoir until it exceeds 20% of its capacity. At that point, the actor checks the operating demands again and sets the output valves (VO1 and VO2) to their respective demand values. Once set, the input valves and pumps need to be configured based on the specific demands. As discussed in more detail later, three categories of configurations are possible depending on the values of the demands ($D1 + D2 \leq 10$; $D1 + D2 > 10$, $D1$ and $D2 < 10$; $D1$ or $D2 > 10$). After making the appropriate input valve and pump adjustments (see figure 2), the levels in the reservoirs are checked to ensure that they are within a safe range. If they are, then the startup task is complete.

What is the relationship between this type of analysis—dealing with the familiar conceptual currency of actor goals, tasks, and actions—and a work domain analysis, which does not explicitly represent any of these constructs? To answer this question, we will begin by describing a work domain analysis for the process in figure 1, and then show how, through a set of discrete transformations, it is possible to eventually connect back to the task description in figure 2.

2.2. Complete work domain structure

A work domain is the ultimate object of action—the system being controlled, independent of any particular worker, automation, event, task, goal, or interface. Accordingly, work domain analysis identifies the structural constraints associated with the equipment and its functional capabilities, showing all of the relevant action possibilities that align with the work domain purposes. Figure 3a provides an outline of the work domain analysis that was developed for the process in figure 1 (Vicente and Rasmussen 1990, Bisantz and Vicente 1992). There are three levels of resolution in this space connected by part-whole links (System, Subsystem, and Component). Also, there are five levels of abstraction connected by structural means-ends links (Physical Form, Physical Function, Generalized Function, Abstract Function, and Functional Purpose). The bottom level of Physical Form is not used here because it refers to the physical location and appearance of the work domain, features that are not particularly meaningful in a microworld simulation.

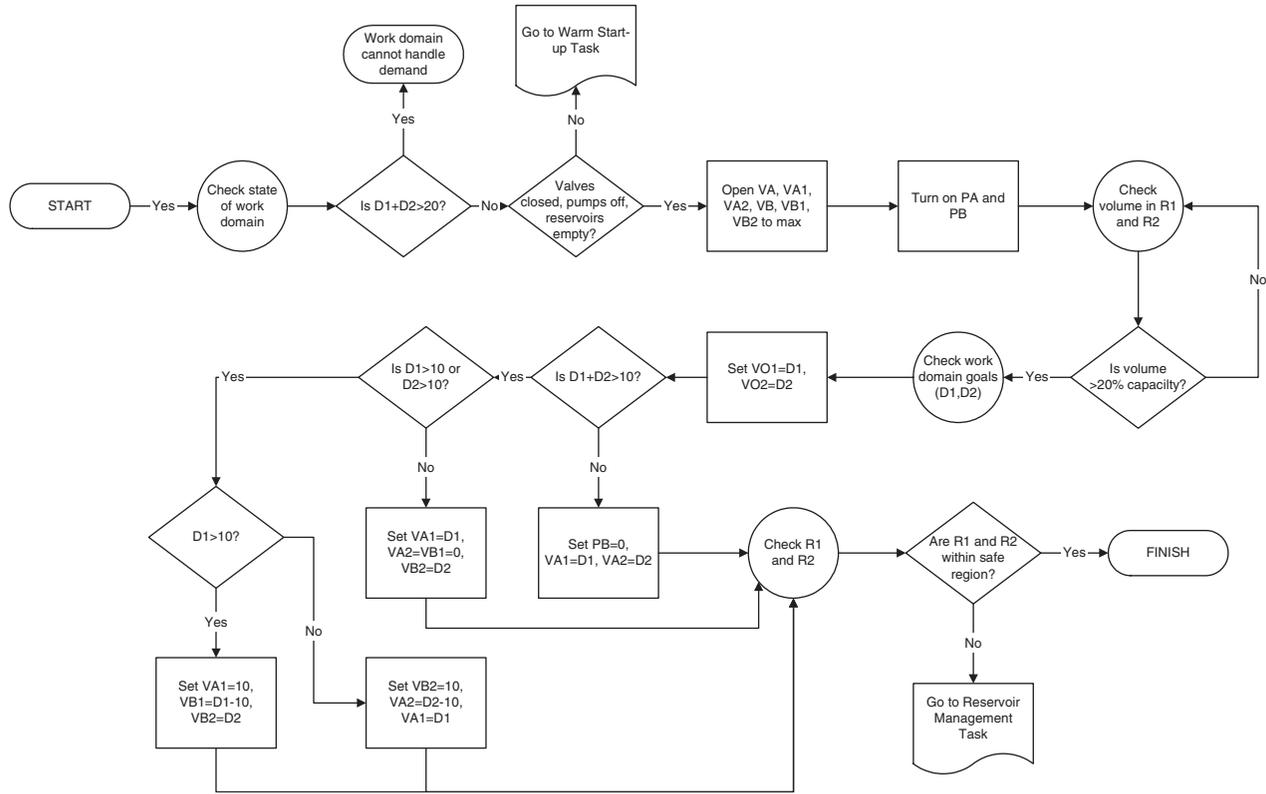


Figure 2. An example of a task analysis for starting up the microworld in figure 1.

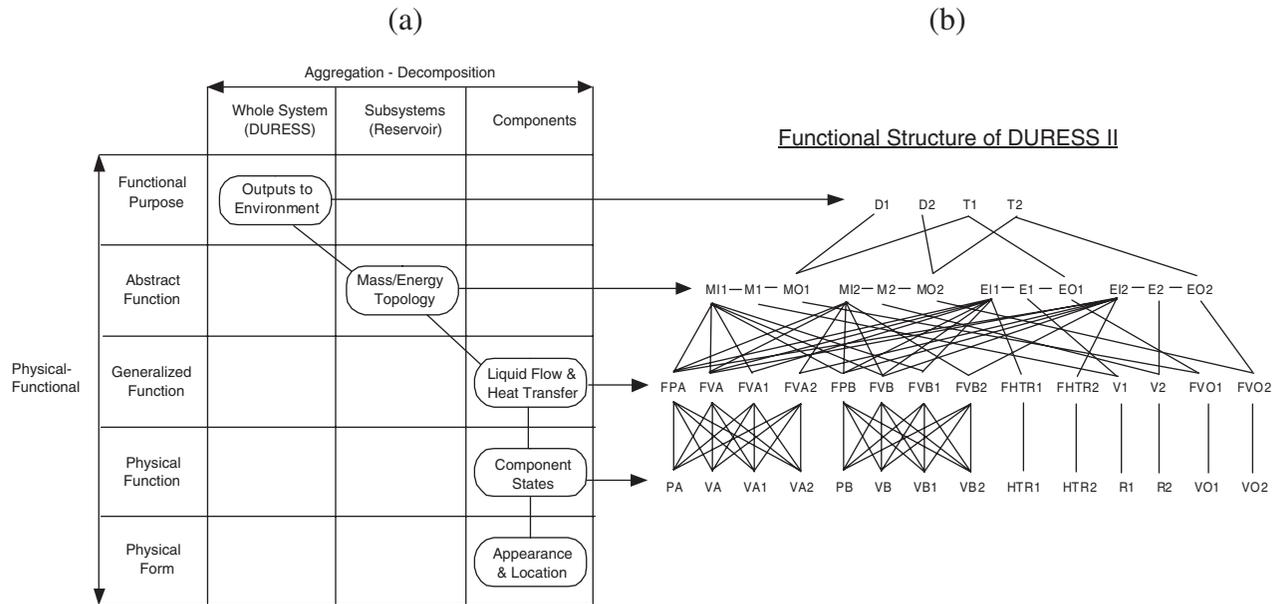


Figure 3. Work domain analysis of the microworld (adapted from Bisantz and Vicente, 1992; Hajdukiewicz, 2000).

Figure 3a shows that the abstraction and part-whole dimensions, while conceptually orthogonal, are coupled in practice. At higher levels of abstraction (e.g., Functional Purpose), participants tend to think of the work domain at a coarse level of resolution (e.g., System), whereas at lower levels of abstraction (e.g., Physical Function), participants tend to think of the work domain at a detailed level of resolution (e.g., Component). Thus, certain cells in the space are not very meaningful (e.g., Functional Purpose/Component). In the specific case of the process shown in figure 3a, four cells were identified as being useful for the purposes of process control. Each of these cells contains a different representation of the same work domain. Figure 3b lists the variables associated with each representation, and specific structural (means-ends) links between adjacent levels of abstraction (Hajdukiewicz 2000).

Accordingly, figure 3 shows the structural opportunities for aligning action possibilities with the work domain purposes. Any path using these structural links can be used to achieve the stated purposes. This representation is *event-independent* in that it is not contingent on any particular situational context or state in time. Regardless of what the operator's goals happen to be and regardless of what is going on in the process at any particular moment, the structural relationships identified in Figure 3b are the ones that have been built into the process by designers. These action possibilities show the requisite variety of the work domain (Ashby 1956), and thus represent an unavoidable bedrock of constraint that must be taken into account by any actor, worker or automation.

This event-independence can be made more concrete by developing a phase diagram for each of the levels of abstraction identified in figure 3. For example, figure 4 shows a partial phase diagram for the level of Abstract Function for one of the two reservoirs. The horizontal axis represents the mass level (M) and the vertical axis represents its gradient ($dM/dt = MI$ [Mass Input Rate] $- MO$ [Mass Output Rate]). The constraint boundaries are derived from the physical and safety constraints of the process. For example, the pumps and valves have a limited capacity (see figure 1), defining maximum and minimum values for dM/dt representing how quickly the reservoir level can be increased or decreased, respectively. Similarly, the reservoir has a limited capacity (see figure 1), defining a maximum value for M representing how much water can be stored without spilling over. These work domain constraints exist, independent of any particular worker, automation, event, task, goal, or interface. They are the fundamental foundation of knowledge for controlling the process. Analogous state space representations could be developed at each of the other levels of abstraction in figure 3.

Given that there are no tasks or goals represented in figures 3 and 4, one might reasonably ask whether a work domain representation is task-relevant, and if so, how?

2.3. *Relevant work domain structure*

For any category of events, only a subset of the action possibilities in Figure 3b are likely to be relevant. Take the case of a startup task described earlier. The goal in this context is to take the process from a shut down state (i.e., all components turned off and reservoirs empty) and fill the reservoir with water (mass) to a safe level. Note that the heaters, and their associated functions, are not relevant for this context because they cannot be used to change the mass level. Figure 5 shows the subset of the full structural degrees of freedom that are relevant for filling up the two

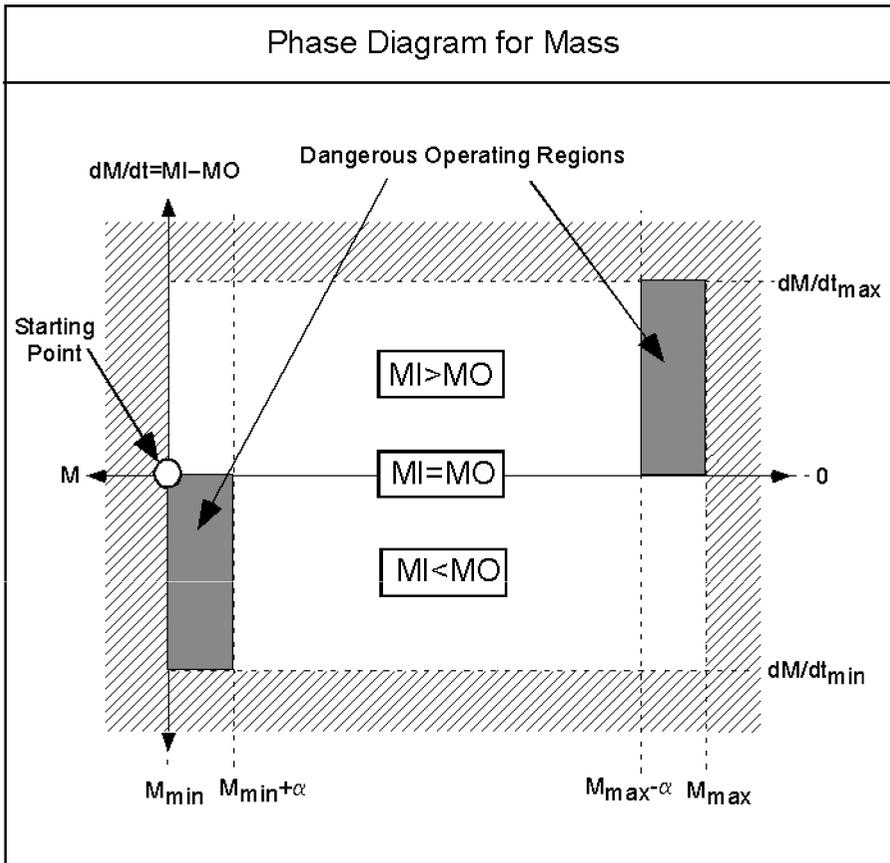


Figure 4. Phase diagram for mass in the microworld.

reservoirs. These are the action possibilities available to any actor for this particular task context. Note that the task is still underdetermined—there is more than one way to get the job done.

2.4. Utilized work domain structure

At some point in time, actors must decide which of the structural possibilities in figure 5 to utilize, given their current task goals. Here too, there are certain constraints that must be obeyed. Particular strategies are feasible for different sets of conditions. Thus, strategies can be seen as categories of action opportunities that further constrain how a task may be accomplished (Vicente 1999a).

This fact can be illustrated by examining the various ways in which the feedwater streams (i.e. pumps and inflow valves) in figure 1 can be used to fill up the reservoirs. Figure 6 shows three general configurations that can be utilized depending on the output demands: *single*, *decoupled*, and *full* (Vicente 1999a). The ‘Description’ column describes the configuration strategy. Below this description is a partial AH relevant to the configuration strategy in supporting mass inflow (MI) to each reservoir. The structural links that are utilized are highlighted in this representation. The ‘Topological Example’ column shows an instance for each strategy category, depicting the resultant topological mass flow through the work domain. Many

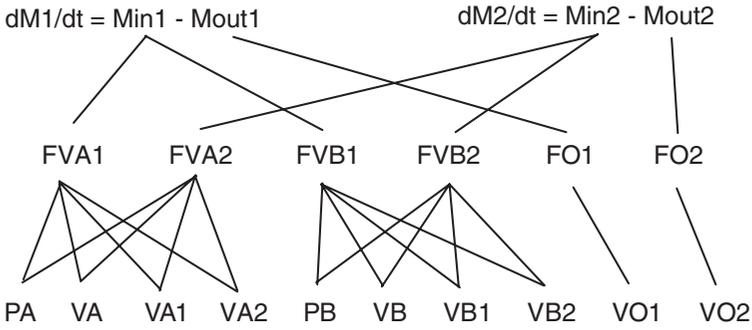


Figure 5. Structural opportunities to configure the microworld during start-up to control dM/dt for reservoirs 1 and 2.

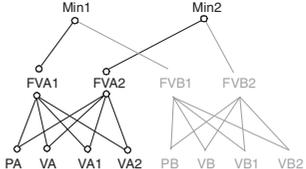
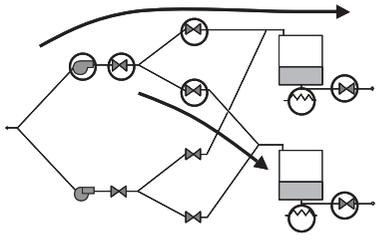
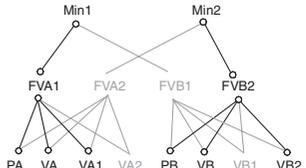
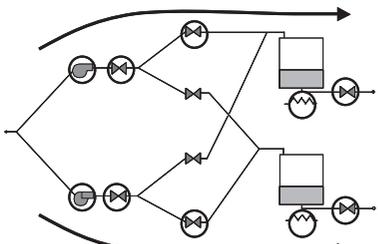
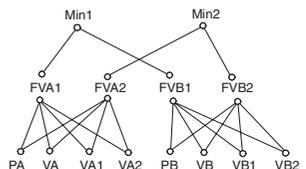
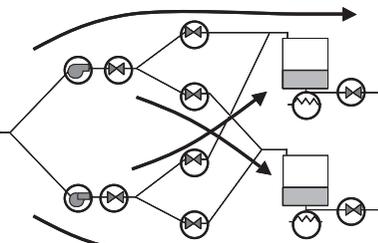
Feedwater Configuration Strategy	Description	Example
<p>a)</p> <p>Single</p>	<p>One pump and its connected input valves are used to feed both reservoirs.</p> <p>Feasible when $D1g+D2g \leq 10$</p> 	
<p>b)</p> <p>Decoupled</p>	<p>Each pump and its connected input valves are used to feed different reservoirs.</p> <p>Feasible when $D1g$ and $D2g \leq 10$</p> <p>Necessary when $D1g+D2g > 10$</p> 	
<p>c)</p> <p>Full</p>	<p>Both pumps and their connected valves are used to feed the reservoirs. At least one pump feeds both reservoirs.</p> <p>Feasible when $D1g+D2g \leq 20$</p> <p>Necessary when $D1g$ or $D2g > 10$</p> 	

Figure 6. Feedwater configuration strategies are a) single, b) decoupled, and c) full (circles represent utilized components). Adapted from Vicente (1999a).

instances can be developed for each category, so the categories still underspecify action.

As shown in figure 6a, the single feedwater configuration strategy uses only one pump and the valves connected to it to fill both reservoirs. This strategy can be sustained when the outflow demands add up to less than 10 units/s because the capacity of each pump is 10 units/s. As shown in figure 6b, the decoupled feedwater configuration strategy uses both pumps and their connected valves to feed different reservoirs. This strategy can be sustained when the outflow demand for each reservoir does not exceed 10 units/s because the capacity for each pump is only 10 units/s. Finally, as shown in figure 6c, the full feedwater configuration strategy uses both pumps and their connected valves to feed both reservoirs. This strategy is sustained when the output demands do not exceed 20 units/s in total, since the capacity of the two pumps is 20 units/s in total.

2.5.. *Current and desired work domain state*

Up to this point, we have been describing the qualitative structure of the work domain with an increasing level of specificity. We have not referred to any particular work domain state. In this final step, the degrees of freedom can be narrowed even further by representing a specific work domain state (rather than just structure) at a particular point in time. This involves annotating the nodes in one of the utilized work domain structure diagrams shown in figure 6 with specific, quantitative values, showing the current state of each component and function. An example is shown in figure 7. Because this is a state description, it represents what is going on at a particular point in time, and thus, is likely to change from one moment to the next. At this level of description, all of the degrees of freedom for the particular task context of filling up the reservoir will have been resolved—the state is uniquely specified.

An analogous transformation can be used to describe a desired (i.e., goal) state because it too can be represented by annotating the nodes in a utilized work domain structure diagram with specific, quantitative values. The only difference is that, in this case, the desired state of each component and function is shown rather than the current state. However, because desired states will also change over time (e.g., for shutdown vs. startup vs. normal operations), the representation of desired work domain state is also time- and event-dependent. In this way, it is distinctly different from the complete work domain structure described earlier.

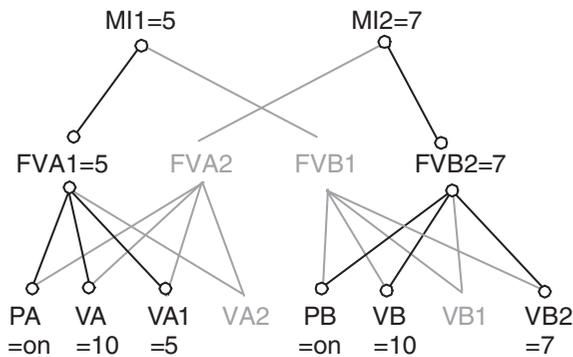


Figure 7. Current work domain state for feedwater configuration.

With a quantitative description of both the current and desired work domain states in hand, we are now in a position to close the circle back to the task analysis in figure 2. The state description in figure 7 is the input to a task description. For example, the work domain nodes in figure 7 would serve as the input into the circle labelled “check state of work domain” in figure 2. Conversely, the state description in figure 7 is also the output of a task description. For example, the work domain nodes at the bottom of figure 7 would serve as the output of the box labelled “Open VA, VA1, VA2, VB, VB1, VB2 to max”. Finally, the goal state represents a version of figure 7 that has the desired values in all of the nodes, rather than the current values. Note that the task analysis in figure 2 does not contain an explicit description of the work domain. It merely refers to work domain nodes and states, not the structure of the work domain itself, as shown in figure 7. In other words, figure 2 is a function that maps current work domain states onto desired work domain states via a set of actions.

3. Theoretical generalization

Using the case study described in the previous section, we can describe the relationship between work domain analysis and task analysis in a more generalized fashion. Five levels of analysis were identified and described:

1. Complete work domain structure
2. Relevant work domain structure
3. Utilized work domain structure
4. Current and desired work domain state
5. Final set of actions and work domain states (outputs of a Task Analysis).

As depicted in figure 8, each level is connected by a discrete transformation. More specifically, the transformation from a complete work domain structure (e.g., figure 3, showing all relevant action possibilities) to a relevant work domain structure (e.g., figure 5, showing all relevant action possibilities for a particular category of events) takes us one step away from a work domain analysis and closer to a task analysis, essentially by using a task context as a filter for examining the work domain representation. The transformation from a relevant work domain structure to a utilized work domain structure (e.g., figure 6, showing the subset of utilized action possibilities at a particular point in time) takes us a second step away from a work domain analysis and closer to a task analysis, essentially by using the operator’s strategy choices as an additional filter for examining the work domain representation. The transformation from a utilized work domain description to a current or desired work domain state (e.g., figure 7, showing the work domain state at a particular point in time) takes us a third step away from a work domain analysis by using the effects of the actor’s actions or goals as a filter for examining the work domain representation. Given these transformations, task analysis is a function that maps current states onto desired states via a set of human or automated actions.

The five levels of analysis just described can probably be applied to any socio-technical system with a known structure. However, the generalizability of the transformations needs to be determined through additional research. They clearly have a broad range of applicability (e.g., physical engineering systems), but they may require modification or reconsideration for other types of systems (e.g., intentional

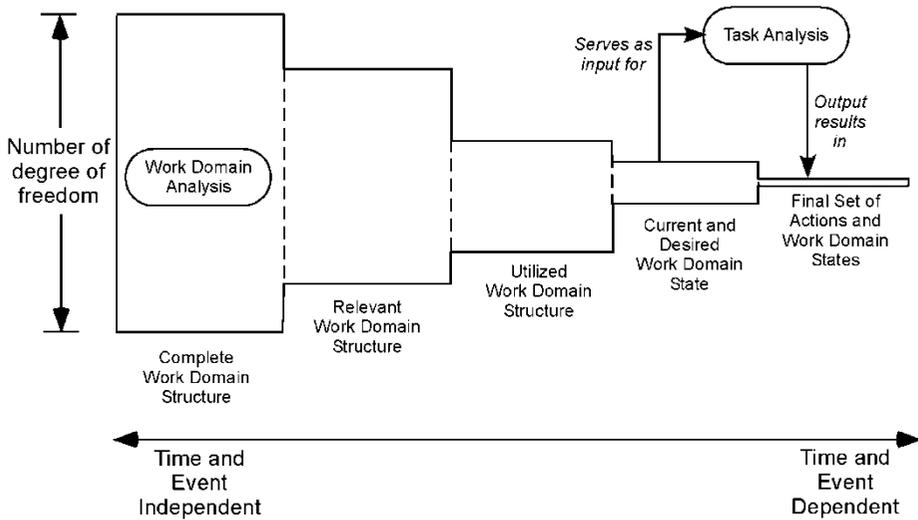


Figure 8. Generalized relationship between work domain analysis and task analysis.

systems that consist of people rather than physical components; Rasmussen *et al.* 1994). This remains a topic for future research.

4. Conclusions

Work domain analysis does not explicitly deal with any particular worker, automation, event, task, goal, or interface. These are fundamental concepts in cognitive engineering, so it is reasonable to ask: If work domain analysis does not represent any of these key issues, let alone all of them, then of what use is it? By making the relationship between work domain analysis and task analysis more clear and explicit, the answer to this question becomes apparent.

Work domain analysis represents information that is only implicitly captured, if at all, in a task analysis. And by identifying information requirements that are event- and time-*independent*, work domain analysis provides a robust basis for supporting worker adaptation to novelty and change. This is something that task analysis cannot do, by definition, because a class of events must be explicitly identified (or implicitly assumed) before a task analysis can even be started. There is now a substantial body of research to show that making work domain constraints visible in an interface enhances human performance under unanticipated situations requiring discretionary problem solving (Vicente 2002, Hajdukiewicz and Vicente 2002). Therefore, even though it does not deal with goals and actions, work domain analysis can play an important and unique role in shaping how well actors achieve their goals and select their actions.

Acknowledgements

This research was sponsored in part by the Jerome Clarke Hunsaker Distinguished Visiting Professorship from MIT, by a research grant from the Natural Sciences and Engineering Research Council of Canada, and by Honeywell International.

References

- ASHBY, W. R. 1956, *An Introduction to Cybernetics* (London: Methuen).
- BISANTZ, A. M. and VICENTE, K. J. 1994, Making the abstraction hierarchy concrete, *International Journal of Human-Computer Studies*, **40**, 83–117.
- HAJDUKIEWICZ, J. R. 2000, Adapting to change in complex work environments, *Proceedings of the Conference of Human Factors in Computing Systems (CHI 2000)* (New York: ACM), 89–90.
- HAJDUKIEWICZ, J. R. and VICENTE, K. J. 2002, Designing for adaptation to novelty and change: The role of functional information and emergent features, *Human Factors*, **44**, 592–610.
- KIRWAN, B. and AINSWORTH, L. K. 1992, *A guide to task analysis* (London: Taylor & Francis).
- NAIKAR, N. and SANDERSON, P. M. 1999, Work domain analysis for training system definition and acquisition, *The International Journal of Aviation Psychology*, **9**, 271–290.
- RASMUSSEN, J., PEJTERSEN, A. M. and GOODSTEIN, L. P. 1994, *Cognitive systems engineering* (New York: Wiley).
- TERRIER, P., CELLIER, J.-M. and CARRERAS, O. 2000, Task analysis does count in making the abstraction hierarchy concrete: Evidence from a pressurized water reactor situation, *Proceedings of the IEA 2000/HFES 2000 Congress, Volume 1* (pp. 161–163) (Santa Monica, CA: HFES), 161–163.
- VICENTE, K. J. 2002, Ecological interface design: Progress and challenges, *Human Factors*, **44**, 62–78.
- VICENTE, K. J. 1999a, *Cognitive work analysis: Toward safe, productive, and healthy computer-based work* (Mahwah, NJ: Erlbaum).
- VICENTE, K. J. 1999b, Wanted: Psychologically relevant, device- and event-independent work analysis techniques, *Interacting with Computers*, **11**, 237–254.
- VICENTE, K. J. and RASMUSSEN, J. 1990, The ecology of human-machine systems II: Mediating “direct perception” in complex work domains, *Ecological Psychology*, **2**, 207–250.

About the authors

John R. Hajdukiewicz received a PhD (2001) in Mechanical and Industrial Engineering at the University of Toronto. He is currently Principal Research Scientist at Honeywell’s Advanced Technology Laboratories in Minneapolis. He is also a research lead of the ASM[®] consortium (<http://www.asmsconsortium.org>). With Catherine Burns, he co-authored *Ecological Interface Design*, published by CRC Press (Taylor & Francis).

Kim J. Vicente received a PhD (1991) in Mechanical Engineering from the University of Illinois at Urbana-Champaign. He is currently Jerome Clarke Hunsaker Distinguished Visiting Professor of Aerospace Information Engineering at MIT; Professor of Mechanical and Industrial Engineering, Biomaterials and Biomedical Engineering, Computer Science, and Electrical and Computer Engineering at the University of Toronto; founding director of the Cognitive Engineering Laboratory there; and a licensed professional engineer in Ontario. His latest book, *The Human Factor: Revolutionizing the Way People Live With Technology*, will be published by Knopf Canada in 2003.