BEHAVIOUR & INFORMATION TECHNOLOGY, 1990, VOL. 9, NO. 6, 493–502

Coherence- and correspondence-driven work domains: implications for systems design

KIM J. VICENTE

Department of Mechanical and Industrial Engineering,
University of Illinois at Urbana-Champaign, Urbana IL, USA

Abstract. A distinction is made between coherence- and correspondence-driven work domains. This novel domain taxonomy is used to argue that the widely accepted goal of making the interface representation compatible with the user's mental model is not always appropriate. For correspondence-driven domains, it is more meaningful to constrain design from the side of the work domain rather than from that of the user. The implications of the coherence/correspondence distinction for the modelling of work domains, for interface design in computer supported co-operative work, and for the development of a multidimensional taxonomy of work domains are also briefly pointed out. The discussion suggests that the correspondence/coherence taxonomy provides a powerful conceptual tool for addressing fundamental issues in human–computer interaction.

1. Introduction

Design is fundamentally concerned with constraints (cf. Alexander 1964). In the design of human–computer systems there are many sources of constraint. Three are of particular concern here. First, there are constraints that arise from the capabilities of existing interface technologies (e.g., what I/O devices are available). Second, there are also context specific constraints related to the functionality that the particular system is supposed to achieve (e.g., for word-processing, to compose a piece of text). Third, and finally, there are psychological constraints on design imposed by human capabilities and limitations (e.g., short-term memory limitations). Interestingly, interface technology serves as the mediating point between the context specific constraints and the psychological constraints. Thus, the objective of interface design can be viewed as revealing the functionality of the work domain in a manner that is consonant with human capabilities and limitations (Rasmussen and Vicente 1989).

A question which naturally arises in systems design is: In which direction should design proceed? That is, should the designer begin by considering psychological constraints, or domain constraints? Often, the psychological constraints are given priority. One often hears the phrase, 'Make the interface representation compatible with the user's mental model of the domain'. In this paper, it is suggested that this is not always an effective way to proceed. There are certain cases where it is more appropriate to begin by investigating the constraints imposed by the work domain. Which set of constraints should form the starting point for systems design depends on whether the work domain is *correspondence-driven* or *coherence-driven.* The purpose of this paper is to describe this novel way of classifying work domains and to draw some implications for systems design.

The paper begins by comparing two approaches to human–computer system design, and then proceeds to illustrate how the two perspectives can lead to conflicting design recommendations in some situations. The correspondence/coherence taxonomy is introduced to elucidate the conditions under which conflict arises. Finally, the power of the taxonomy is illustrated by using it to clarify some claims and distinctions made in the literature.

2. Two perspectives on human–computer systems design

The primary concern in the design of human–computer systems is to construct an artifact that achieves a given set of purposes at a desired level of effectiveness. Thus, the design process is normative in that it is always goal-oriented. Explicitly or implicitly, design is always *for* something (e.g., design a power plant to produce electrical power in a safe manner). Despite this generally well-accepted definition, at least two different perspectives on the design of human–computer systems can be identified. The *systems* approach originates from the applied discipline of systems engineering (e.g., Rasmussen 1986), while the *user-centred* approach originates from basic psychological research (e.g., Norman and Draper 1986). This section will describe and compare these two approaches.

To appreciate the user-centred perspective, it is necessary to understand the objectives of the basic science from which it is derived. Basic psychological research is descriptive and explanatory in nature. Typically, the goal is to find out how the human mind works. Thus, the focus is on identifying the general capabilities and limitations of the human organism. The user-centred approach attempts to apply results and knowledge from basic psychological research to the design of human–computer systems. Naturally enough, the primary emphasis is on the human (i.e., the user). According to the user-centred approach, the design process should begin with the user (Norman 1986). Certainly, this is a laudable goal since computer system designers have traditionally paid little attention to the end user.

The systems approach to design is quite different. The primary emphasis is not on the user, but on the system, which is defined as an organized whole composed of a number of interacting elements that share a common purpose (c.f. Meister 1989). As a result, the design effort begins by investigating the global context in which work takes place. The reasoning behind this claim is that knowledge of the entire system cannot be built up solely from knowledge of the parts. Emergent properties exist as a function of system structure, and understanding of these properties is only possible by knowing how the parts are organized. As a result, from the systems perspective the human is viewed as one of many system elements, not as a privileged entity. But because the human element has such a great influence on system performance, a great deal of attention is – and should be – devoted to human factors in design. If this were not the case, there would be little justification, from a systems perspective, to be interested in the properties of human cognition, perception, and action.

It is clear that there are strong differences between the systems and user-centred approaches. The most obvious contrast is that the user-centred approach is more local in that it gives privileged status to the human component. The justification for this is that 'the purpose of the system is to serve the user' (Norman 1986: 61). While this is certainly true, focusing primarily on the user could lead to a lack of attention to the global factors that act on the local concerns of the user in interacting with his or her computer (cf. Moray and Huey 1988). For instance, the introduction of information technology in a hospital is ultimately intended to increase the organization's capabilities to achieve its goals, not to serve the idiosyncratic goals of each doctor or nurse using the system. If the organization as a whole is to function effectively, the immediate goals of the individual users need to be subservient to, or at least consistent with, the overall purposes of the organization. Consequently, if information technology is to contribute to effective operation, the design of human–computer systems must begin by considering the functions of the global setting in which work takes place.

This difference between the user-centred and systems approaches can manifest itself in many different ways (see Meister 1989 and Rouse 1989). One such manifestation pertains to the issue described in the introduction. There it was pointed out that it is often argued that the computer should present information in such a way as to match the operator's mental model of the process being controlled (e.g., Falzon 1982). Such an approach is consistent with the user-centred perspective, since it reflects a choice to adopt the operator's mental model as the primary constraint on design. This contrasts with the systems perspective, which constrains design from the side of global system functionality. Such an approach would lead to the recommendation to design the interface so that the operator can acquire a veridical mental model of the domain. Is there a fundamental conflict between these two recommendations, or do they complement one another? As it turns out, the answer to this question depends upon the type of domain for which one is designing.

3. Classes of work domains

As the term is used here, *domain* refers to the 'world' or problem space in which operators act (cf. Rasmussen and Goodstein 1988, Dowell and Long 1989). As such, a domain can be described as a set of constraints, both physical (e.g., material objects) and abstract (e.g., functional relations). There are many criteria according to which one could classify work domains. A simple and novel criterion, which nevertheless is quite powerful conceptually, is to adopt the distinction between coherence and correspondence theories of truth as a metaphor for classifying domains. In this way, a work domain could be classified as being either correspondence-driven or coherence-driven.

Before discussing how this distinction can be applied to systems design, the notions of correspondence and coherence as they are used in philosophy need to be described. These two theories have been used by philosophers to refer to two fundamentally different but complementary conceptions of truth (e.g., Newton-Smith 1981). According to the *correspondence theory of truth*, a proposition is true if it accurately describes the state of the world, independent of any particular perspective. To adopt the correspondence theory of truth is to also adopt a form of *realism*. That is, if one is to evaluate the truth value of a statement according to this criterion, one must also accept that an observer-independent, objective reality (what Nagel [1986] calls 'the view from nowhere') actually exists, otherwise the correspondence cannot be evaluated. The *coherence theory of truth*, on the other hand, does not require a realist stance. According to this criterion, a proposition is true if it is coherent with a formalized framework of concepts. The key is that the choice of concepts is arbitrary. Whether there is correspondence with reality is not considered. Thus, the coherence criterion is associated with *relativism*, since the truth value of a statement can vary according to the set of concepts against which it is evaluated.

While theories of truth may not be directly relevant to design, the distinction between correspondence and coherence is. In the following sections, this distinction is used as a basis for defining two broad classes of work domains.

3.1. *Correspondence-driven domains*

Domains that are driven by the correspondence theory of truth involve an objective reality which acts as the driving constraint for human–computer interaction. This idea is made more specific in figure 1. The key criterion for identifying correspondence-driven work domains can be identified by drawing an imaginary boundary around the human–computer system. If there are goal-relevant, dynamic (i.e., time-dependent) constraints that are external to this dyad, then the domain is correspondence-driven. For instance, in flying a plane, the position of the airplane with respect to the ground defines an objective reality that is independent of any observer's subjective viewpoint. There are relevant, dynamic constraints (e.g., the position of surrounding buildings or terrain) that exist outside of the boundary around the pilot and his cockpit instrumentation that need to be taken into account if system objectives are to be achieved. More generally, successful action in a correspondence-driven domain requires that the agent's beliefs be in correspondence with the actual state of the world.

In terms of design implications, the important point is that the domain representation embedded in the computer should reflect the objective state of the world so that the agent's assessment of that state corresponds as closely as possible to the actual situation. This is just a restatement of the systems perspective described above: design the computer system so that the agent acquires a veridical mental model of the system. Notice that the user-centred perspective is clearly inapplicable to correspondence-driven domains. *What good is it having the computer be compatible with the agent's mental model if that model does not correspond with reality?*

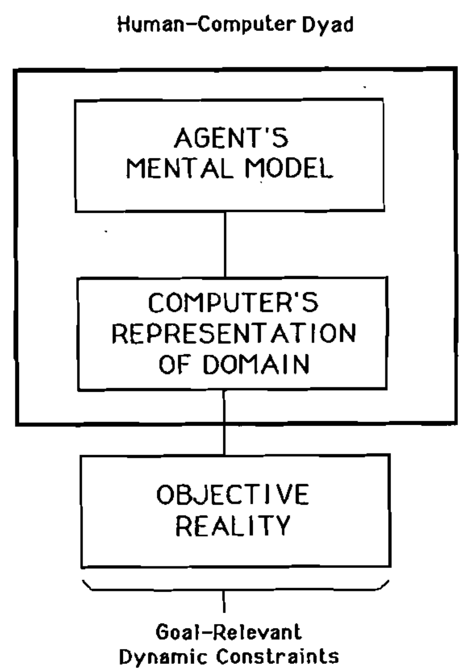

Figure 1. In correspondence-driven work domains, there are goal-relevant, dynamic constraints that are external to the human–computer dyad. In this case, the computer representation must ensure that the agent's mental model corresponds with objective reality.

This is not to say that psychological considerations are not important. This is clearly not the case. With correspondence-driven domains, the psychology comes into the design process when deciding how to reveal the structure of the domain to the agent (Rasmussen and Vicente 1989, Vicente and Rasmussen 1990). In other words, the *content* and *structure* of the domain is already predetermined (e.g., by thermodynamic laws for process control systems, or by the laws of physics for aircraft), but there are still considerable degrees of freedom in how to reveal that structure. As mentioned earlier, the challenge is to display the domain properties in a *form* that is compatible with the properties of human cognition, action, and perception.

3.2. *Coherence-driven domains*

Domains that are driven by the coherence theory of truth do not involve an objective reality. Instead, the driving constraint for human–computer interaction is some arbitrarily defined set of concepts. This situation is illustrated in figure 2. If there are no goal-relevant, dynamic constraints external to the human–computer dyad, then the domain is coherence-driven. It is important to note that there can be static constraints outside of the dyad. For example, in the domain of word-processing, there are external constraints which need to be taken into account (e.g., the size of the paper). However, because these constraints are static, they can be factored in as initial conditions to the design process. Since the constraints can be built into the software, there is no need for the user to continually consider them. This possibility is not available in a correspondence-driven domain because the constraints vary with time, and therefore, cannot be factored in as static initial conditions.

Chess is an example of a coherence-driven domain. The rules of the game define a 'language' for communication. When a person is playing chess with a computer, there are no external board or pieces that need to be considered. The interaction is contained within the human–computer dyad. The criterion for effective interaction is that the player's intentions be expressed within the boundaries defined by the rules of the game, otherwise his intentions will not be interpretable. Thus, successful action in coherence-driven domains requires that the agent's beliefs be coherent with the set of concepts, or the language, that has been arbitrarily adopted for communication. Unlike correspondence-driven domains, there is no need for that language to correspond with an objective reality.

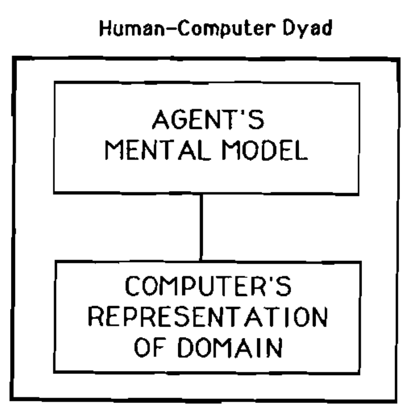

Figure 2. In coherence-driven work domains, there are no goal-relevant, dynamic constraints that are external to the human–computer dyad. In this case, the computer representation can be tailored to be compatible with the agent's mental model.

Consequently, the design of information systems for coherence-driven domains can proceed in a more flexible manner than for correspondence-driven domains. Not only is there freedom in how to reveal the structure of the domain, as there is in correspondence-driven domains, but there is also freedom in determining that structure itself. Thus, the designer can define the concepts which people find more natural and easier to use for accomplishing the task (cf. Pejtersen 1988, 1989, Goodstein and Pejtersen 1989 for an example from the domain of information retrieval). Therefore, the conflict previously observed between the systems and the user-centred perspectives disappears. Both perspectives lead to a design that provides agents with the information they require to satisfy domain objectives, and in a manner that is compatible with their mental model.

3.3. *Applying the taxonomy*

In this section, the power of this domain taxonomy will be illustrated by using it to clarify some claims and distinctions made in the literature. We will begin with the example provided by Falzon (1982) for the domain of air traffic control (ATC). It should be evident that ATC is a correspondence-driven domain. Nevertheless, Falzon adopts a user-centred perspective for design, stating that the computer's representation of the domain should be compatible with the operator's mental model. He supports this claim by stating that ATC operators tend to think of the domain in terms of the *separation* between pairs of aircraft rather than in terms of the *absolute position* of each aircraft. Thus, he recommends that displays should also show the relative separation between aircraft, thereby achieving compatibility between operator and computer.

Although this may appear to be a logical argument, it is flawed. Aircraft separation is a property of the objective world which is relevant to accomplishing the goals of ATC, and it needs to be taken into account by any entity acting as an air traffic controller, human or otherwise. It is a physical entity not a psychological entity. While Falzon is correct in concluding that aircraft separation is a key variable in ATC, he does so for the incorrect reason. The premise of having the display be compatible with the controller's mental model does not lead to the conclusion of displaying separation unless one assumes that the controller has the correct mental model (i.e., that he or she knows that separation is a key goal-relevant variable). This is a reasonable assumption to make in this particular case, although Falzon does not explicitly state it in his argument. The problem is that one cannot always assume that the operator's mental model corresponds with reality, particularly if the relevant objective variables are more complex, and therefore less obvious (e.g., the functional relations in a nuclear power plant). The systems design perspective of designing an interface to induce a veridical mental model is more appropriate in this case because it drives design from actual system functioning rather than from the knowledge of the operator, which may or may not be veridical.

A second application of the coherence/correspondence domain taxonomy further illustrates the novelty and utility of this distinction. Winograd (1987) provides a discussion of realism which is relevant to the concerns here. As a preface, it should be pointed out that Winograd's research interests have primarily been the study of language. He contends that the correctness of an utterance by a speaker 'cannot be judged in terms of objective reality... but only in terms of the speaker and hearer's understanding of the relevant background and purposes' (Winograd 1987: 9). In other words, Winograd is claiming that language can only be judged according to the coherence theory of truth – a controversial claim (see Barwise and Perry 1983 for an opposing view), but defendable nonetheless. In discussing the pros and cons of a realist position, he extends his conclusions about language to the design of computer systems in general: 'How is [realism] relevant to anything we might care about as cognitive scientists or computer system designers?... If we are trying to come up with causal explanations of mental processes, the claim of realism may not be wrong, but it is vacuous. It simply doesn't make any relevant difference' (Winograd 1987: 11). Note that Winograd is generalizing a claim to both cognitive scientists *and* computer system designers. But as stated before, the goals of basic cognitive research are not the same as those of design, thereby suggesting that conclusions derived from the former may not always be applicable to the latter.

A hypothetical scenario from a correspondence-driven work domain shows that Winograd's generalization does not hold. Take the case of two operators talking over an abnormal situation in a nuclear power plant. Based on the readings they get from the control panel, they agree on a diagnosis. This situation meets Winograd's (1987: 12) requirement of 'intersubjective correlation' (i.e., coherence) for effective communication. However, in spite of the coherence between the two operators, if there is no correspondence between their evaluation and the objective state of the plant, then the consequences are likely to be not only relevant, but also potentially disastrous (see also the Three Mile Island example discussed later). Contrary to Winograd's claims, the claim of realism, while (arguably) not crucial for the study of language, is certainly *not* vacuous for designers of computer systems. It can be a very important and very relevant claim, if one is designing for a correspondence-driven domain.

The distinction between coherence- and correspondence-driven domains can also be used to refine the relationship between what Norman (1986) refers to as *physical* variables and *psychological* variables. Physical variables are those that the machine uses to represent the state of affairs, whereas psychological variables are those that users adopt to think about their goals. Based on this characterization, Norman describes the human–computer interaction process as an interaction between the user's goals and the physical system. In the terms of this paper, Norman seems to be describing human–computer interaction in coherence-driven domains, as shown in figure 2. However, this characterization does not take into account the unique characteristics of correspondence-driven domains, as shown in figure 1.

The coherence/correspondence distinction forces one to realize that there are actually two kinds of physical variables. There are the physical variables represented in the computer and the physical variables describing the objective state of the world. Ideally, these two sets of variables should correspond with each other, but this may not always be the case if the designer does not specifically consider this objective. For example, in the Three Mile Island accident there was a discrepancy between the interface's display of the state of a relief valve and the valve's actual state (Rubinstein 1979). This discrepancy was a significant contributor to the ensuing accident because the operators were led to believe that the valve was closed when it was actually still open. This example points to the importance of distinguishing between the computer's representation of the domain and the work domain itself. In correspondence-driven domains, it is critical that the interface be designed so that there is a correspondence between the computer's representation and the actual state of the world. Otherwise problems such as the one that occurred at Three Mile Island are likely to occur.

The ideas expressed by Falzon (1982), Winograd (1987) and Norman (1986) indicate that the distinction between correspondence- and coherence-driven domains is not widely recognized or accepted. This is an unfortunate situation since confusing the two domain classes can lead to the adoption of a design perspective that is inappropriate for the work domain being designed for.

4. Conclusions

The design of human–computer systems must take into account the functional constraints imposed by the work domain, the technological limitations put forth by existing computer hardware and software and the psychological capabilities and limitations of the end users. There is no question that all three classes of constraints are important. The central question addressed in this paper is: With which set of constraints should the design process begin? Two prospective approaches were identified: the systems perspective, which starts with the work domain, and the user-centred perspective, which begins with the user. The distinction between correspondence- vs. coherence-driven domains was introduced to shed some light on the applicability of these two perspectives. More specifically, it was argued that in correspondence-driven domains, the two perspectives conflict. For this class of work domains, the systems perspective is a more fruitful approach because correspondence with objective reality is a prerequisite for effective action.

The coherence/correspondence distinction can also be applied to other issues in human–computer interaction. For example, the distinction points to the general importance of distinguishing between the computer's representation of the domain and the work domain itself. As the Three Mile Island example points out, this distinction is of critical importance in correspondence-driven domains. Because the study of human–computer interaction has typically taken place in coherence-driven domains (e.g., text editing), the work domain has not received a great deal of attention as an object worth studying in its own right. In contrast, the area of *cognitive engineering*, which is primarily concerned with the design of computer systems for correspondence-driven domains (e.g., aviation, process control, hospitals), has recognized the distinction between the computer's representation and the actual work domain (e.g., Rasmussen and Goodstein 1988, Woods and Roth 1988). As a result, cognitive engineers have developed methods for modelling the functional structure of the work domain (Rasmussen 1986).

The domain taxonomy presented in this paper also has important implications for computer supported co-operative work. If several individuals are working together on a common problem via information technology, then the actions and beliefs of each actor pose a set of external, dynamic constraints that must be taken into account in the interface of each individual. For example, the current beliefs and goals of the other actors with whom one is working should be represented so that co-ordination can be facilitated. Also, an up-to-date representation of the current state of the domain should be maintained because that state can continually change through actions performed by other individuals. Imagine coming in to work one morning to finish writing a program that had been modified overnight by another programmer.[1] This would be an extremely difficult and frustrating task if the interface did not directly highlight the changes that had been made since you last worked on the program. Clearly, the unique properties of correspondence-driven work domains have important design implications for computer supported co-operative work.

Finally, the correspondence/coherence distinction can also contribute towards a multidimensional work domain taxonomy. Such a comprehensive taxonomy could lead to the identification of the psychological demands induced by different classes of work domains, as well as serving as a basis for generalizing results from one domain to another (cf. the work of Hammond 1988 on a taxonomy of tasks).

These potential applications suggest that the coherence/correspondence distinction provides a simple yet powerful conceptual tool for addressing fundamental issues in human–computer interaction.

Acknowledgments

I would like to thank John Flach, John Lee, Neville Moray and an anonymous reviewer, all of whom provided very useful comments on earlier versions of this paper.

Note

1. While this may seem to be a ridiculous example, it is actually based on a true case of computer supported co-operative work where several different programmers were working together to write a piece of software.

References

ALEXANDER, C. 1964, *Notes on the Synthesis of Form* (Cambridge, MA: Harvard University Press).

BARWISE, J. and PERRY, J. 1983, *Situations and Attitudes* (Cambridge, MA: MIT Press).

DOWELL, J. and LONG, J. 1989, Towards a conception for an engineering discipline of human factors, *Ergonomics*, **32**, 1513–1535.

FALZON, P. 1982, Display structures: compatibility with the operator's mental representation and reasoning processes. In *Proceedings of the 2nd European Annual Conference on Human Decision Making and Manual Control* (Bonn: Forschungsinstitut für Anthropotechnik), 297–305.

GOODSTEIN, L. P. and PEJTERSEN, A. M. 1989, *The Book House: System Functionality and Evaluation* (Risø-M-2793) (Roskilde, Denmark: Risø National Laboratory, Department of Information Technology).

HAMMOND, K. R. 1988, Judgement and decision making in dynamic tasks, *Information and Decision Technologies*, **14**, 3–14.

MEISTER, D. 1989, *Conceptual Aspects of Human Factors* (Baltimore, MD: Johns Hopkins University Press).

MORAY, N. P. and HUEY, B. M. 1988, *Human Factors Research and Nuclear Safety* (Washington, DC: National Academy Press).

NAGEL, T. 1986, *The View From Nowhere* (New York: Oxford University Press).

NEWTON-SMITH, W. H. 1981, *The Rationality of Science* (Boston: Routledge & Kegan Paul).

NORMAN, D. A. 1986, Cognitive engineering. In D. A. Norman and S. W. Draper (eds) *User Centered System Design: New Perspectives on Human–Computer Interaction* (Hillsdale, NJ: LEA), 31–61.

NORMAN, D. A. and DRAPER, S. W. 1986, *User Centered System Design: New Perspectives on Human–Computer Interaction* (Hillsdale, NJ: LEA).

PEJTERSEN, A. M. 1988, Search strategies and database design for information retrieval in libraries. In L. P. Goodstein, H. B. Andersen and S. E. Olsen (eds) *Tasks, Errors and Mental Models: A Festschfift to Celebrate the 60th Birthday of Professor Jens Rasmussen* (London: Taylor & Francis), 171–190.

PEJTERSEN, A. M. 1989, *The Book House: Modelling Users' Needs and Search Strategies as a Basis for System Design* (Risø-M-2794) (Roskilde, Denmark: Risø National Laboratory, Department of Information Technology).

RASMUSSEN, J. 1986, *Information Processing and Human–Machine Interaction: An Approach to Cognitive Engineering* (New York: North-Holland).

RASMUSSEN, J. and GOODSTEIN, L. P. 1988, Information technology and work. In M. Helander (ed.) *Handbook of Human–Computer Interaction* (Amsterdam: North-Holland), 175–201.

RASMUSSEN, J. and VICENTE, K. J. 1989, Coping with human errors through system design: Implications for ecological interface design, *International Journal of Man-Machine Studies*, **31**, 517–534.

ROUSE, W. B. 1989, Human centered design of complex systems, *IEEE SMC Newsletter*, November, 8–9.

RUBINSTEIN, E. 1979, The accident that shouldn't have happened, *IEEE Spectrum*, **16**, 33–42.

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.

WINOGRAD, T. 1987, *Three Responses to Situation Theory* (CSLI-87-106) (Stanford, CA: Stanford University, Center for the Study of Language and Information).

WOODS, D. D. and ROTH, E. M. 1988, Cognitive engineering: Human problem solving with tools, *Human Factors*, **30**, 415–430.