HUMAN-COMPUTER INTERACTION, 1999, Volume 14, pp. 159–189

Metaphors and Models: Conceptual Foundations of Representations in Interactive Systems Development

David Benyon and **Manuel Imaz**
Napier University, Edinburgh

ABSTRACT

When system developers design a computer system (or other information artifact), they must inevitably make judgments as to how to abstract the worksystem and how to represent this abstraction in their designs. In the past, such abstractions have been based either on a traditional philosophy of cognition or cognitive psychology or on intuitive, spontaneous philosophies. A number of recent developments in distributed cognition (Hutchins, 1995), activity theory (Nardi, 1996), and experientialism (Lakoff, 1987) have raised questions about the legitimacy of such philosophies. In this article, we discuss from where the abstractions come that designers employ and how such abstractions are related to the concepts that the users of these systems have. In particular, we use the theory of experientialism or experiential cognition as the foundation for our analysis.

David Benyon is Professor of Human–Computer Systems at Napier University, Edinburgh, with interests in all areas of interactive systems design, particularly methods and models for design and the application of knowledge-based techniques for HCI. **Manuel Imaz** is a professional computer system designer undertaking research into the conceptual foundations of software engineering methods within the HCI group at Napier University.

CONTENTS

Experientialism (Lakoff, 1987) has previously only been applied to human–computer interaction (HCI) design in a quite limited way, yet it deals specifically with issues concerned with categorization and concept formation. We show how the concept of metaphor, derived from experientialism, can be used to understand the strengths and weaknesses of alternative representations in HCI design, how it can highlight changes in the paradigm underlying representations, and how it can be used to consider new approaches to HCI design. We also discuss the role that "mental spaces" have in forming new concepts and designs.

1. INTRODUCTION

Human–computer interaction (HCI), or more generally information systems, has developed and employed many alternative methods and representations for design over the years. Although the subject of HCI did not really enter onto the agenda until the early 1980s, the previous 2 decades had seen several attempts at developing methods, approaches, and modeling techniques for the development of interactive systems.

The early days of systems analysis and design evolved methods and models that focused on how the existing system worked. Representations of the system, usually in terms of a flowchart that provided a graphical representation of the existing system, were produced that led to a computer system and file structure to support a particular application (Benyon & Skidmore, 1987). When systems were well understood and primarily cyclical in nature, such as payroll or accounting systems, they could be computerized without too much trouble. However, these methods of analysis began to break down with the introduction of more interactive working—that is, when other people wanted to use the data that had been gathered and stored for a specific application and when computers were being applied to less well-defined tasks. During the 1970s, there was a major change in the representations used in systems analysis and design, following the publication of Codd's (1970, 1982) seminal articles on the relational data model. The data-centered movement in information systems development looked to distance itself from current implementations, and this led to data flow diagrams (DFDs; DeMarco, 1979), the relational model, and the entity-relationship (E-R) diagram (Chen, 1976). The movement faltered with the development of many competing methodologies (Olle, Sol, & Tully, 1983; Olle, Sol, & Verrijn-Stuart, 1982) and finally gave way when the object-oriented (OO) paradigm, inherited from software development, began to be applied to analysis and design as well as to the construction of systems. Since the early 1990s, OO methods have been paramount (Booch, 1992; Larman, 1997).

The concerns of HCI as we know it today were most clearly expressed in the late 1970s through the participative and sociotechnical approaches to systems development, a movement that remains important today (Greenbaum & Kyng, 1991; Karat, 1991). Methods and models encouraging user participation in design developed alongside models derived from task analysis of users interacting with computers (Diaper, 1989; Moran, 1981). As graphical user interfaces began to dominate HCI, so task-based and OO methods followed, and the rather difficult, grammar-based task models were replaced by graphical notations with the focus on user objects. These became embedded in more proscriptive HCI methodologies such as STUDIO (Browne, 1994) and tool-supported systems such as ADEPT (P. Johnson, Johnson, & Wilson, 1995). The somewhat different tradition of cognitive (systems) engineering (Hollnagel & Woods, 1983; Rasmussen, 1987) has also contributed to the current position in HCI.

All this took place as the philosophy of mind was developing with Norman's (Norman & Draper, 1986) everyday psychology and seven-stage model of interaction. The strong cognitive psychology of the 1960s and 1970s put the focus on user tasks, and the detailed analysis of tasks derived from the workflow analysis from the 1960s. Influential philosophical developments by, for example, Winograd and Flores (1986), later work by Hutchins's (1995) on distributed cognition, and the introduction of activity theory into HCI (Bannon, 1991; Bødker, 1990) have changed the influence from cognitive tasks to situated action. The current position remains muddled with a number of alternative conceptions about HCI ("Peer Commentary," 1998), several different types of representations, and variations on the form that those types take. For example, Benyon (1996) identified task-based, data-centered, and OO as major types of representation. Within the task-based approaches, some adopt the form of a grammar and others a structure chart notation.

Our ongoing work (Imaz & Benyon, 1996) is to try to understand why these changes have come about and examine the conceptual foundations of different representations that are used in HCI and more generally in software engineering. To achieve these aims, we apply concepts arising from experientialism or experiential cognition (Lakoff, 1987). Experientialism is a reaction against the classical cognitive science tradition of the 1960s and 1970s (much as distributed cognition is a different reaction). *Experientialism* argues that there is no direct relation between the categories we build and an objective world. In terms of our thinking, *experiential cognition* emphasizes the role of both bodily and sociocultural experience in characterizing concepts and in the human imaginative capacity for creating concepts and modes of rationality that go well beyond any "mind-free" external reality (Lakoff, 1988). Indeed, the title of Lakoff's (1987) book, *Women, Fire and Dangerous Things,* was chosen because these seemingly diverse things are classified as belonging to the same category in the Australian aboriginal language, Dyirbal: "The structure inherent in our experience makes conceptual understanding possible and constrains the range of possible conceptual and rational structures" (p. 120).

In this article, we apply some of the concepts and approaches of experientialism to representations in interactive systems development. We examine some philosophical and practical aspects of developing human–computer systems from the perspective of the representations that they employ. In Section 2, we present a brief discussion of the representations or models (we use these terms interchangeably) that are a necessary part of any design. Section 3 presents a brief review of experientialism and shows how this philosophy can be used to understand more about the conceptualizations that our discipline has employed. In Section 4, we examine some applications of experientialism to three different levels of description of human–computer systems: the organizational level, workplace level, and operational level. Section 4 also describes the foundation of some "cognitive artifacts" that we use to undertake our analysis. In Section 5, we review a number of popular representations for both the process and products of systems development and their underlying concepts. Section 6 provides a brief conclusion and indicates how we can take this type of analysis further.

We believe that experientialism provides a different way of thinking about representations that takes us away from an objectivist approach to cognition. In developing interactive systems, we do not believe that there is a natural set of objects to be understood, designed, and implemented. We see evidence for this belief in both the competing methods that have evolved over the years and the wide variety of perspectives that still exist on what HCI actually is. The ideas presented here are important because of the impact that representations have on the resulting designs; an inappropriate (for some purposes) representation will lead to a poor design or at least make a good design more difficult to achieve. We offer a way of thinking about representations in general, providing a lingua franca with which to talk about models in systems development.

2. REPRESENTATIONS IN DESIGN

The term *design* refers to both the process of developing a product, artifact, or system and the various representations or models (e.g., physical artifacts, conceptual models, simulations, prototypes) of the product that are produced during the design process. Designers need representations that will help them to understand users' requirements, and they need to represent this understanding in different ways at different stages of the design. Selecting suitable models is important for exploring, testing, recording, and communicating design ideas and decisions both within the development team and with users.

Models abstract some domain of interest by hiding some details so that the important aspects stand out. They are "professional" languages that both constrain and focus a discourse by limiting the range of concepts that can be expressed in the language (Benyon, 1997; Kangassalo, 1983). Models are mediating artifacts in the activity of design. Models provide a certain perspective (we return to this later) on the domain by employing abstraction mechanisms that are reflected in the content and the structure of the concepts employed by the model. Abstraction mechanisms involve combinations of categorization or classification (treating a class of objects as a single object) and aggregation or composition (grouping related things together and treating the whole group as a whole). Modeling is important and difficult. Although there are principles of categorization that can be derived from classification theory (Parsons & Wand, 1997), it is the conceptual, philosophical basis of these classifications and aggregations in the context of developing human–computer systems that we explore in this article.

A model will be more or less effective for a given purpose according to the characteristics possessed by the model and the relations between those characteristics, the modeler, and the recipient of the model. A model must possess the necessary structure and processing capability to fulfill its purpose. The analytic, explanatory, and communicative power of a conceptual model arises from the structure, operations, and constraints that the model is able to capture (Kangassalo, 1983). A model must also have suitable physical characteristics: the notation employed by the model and its overall usability for the purpose at hand.

Our purpose is to develop effective human–computer systems. It follows that any model that we use must be orientated toward that purpose. Even with this declared purpose, there are many levels at which we can view human–computer systems. At the organizational level, we might consider the design of working practices and the impact of new technologies on organizations, health and safety issues, or office layout. At the workplace level, we want to design for the activities that users undertake, at the structures and artifacts that enable people to pursue their goals. At an operational level, we want to look at information flow, at how knowledge is constructed and represented or how it is distributed through the system, or consider the usability of systems and the cognitive and emotional demands that systems make on people. Similarly, there are different activities involved in design, and a model that helps in understanding requirements may not be so useful when it comes to formally communicating or documenting a design.

Representations are more or less useful for understanding or designing a system at some level of abstraction. At a given level of abstraction there are a variety of representations that can assist the designer in understanding or developing the system from one or more of three perspectives. The intentional, or goal, level of description is concerned with how the representation relates to entities external to that system. The conceptual, or logical, level focuses on the function and structure of the system—on its semantics. The physical characteristics of a model are equally important; conceptually, a model might have an appropriate structure for a purpose, but physically it may be so poorly designed that it is unusable. Mappings between the levels are required to progress from one level of description to another. At whatever level of abstraction we decide to model the system and whether we decide to focus on an intentional, conceptual, or physical model, we can consider the appropriateness of our model in terms of the structure (and, hence, constraints) that it can represent, the functions that it supports, and the structure–function relation (the behavior).

Consider maps as models of a terrain. Maps come in a variety of physical manifestations. Maps are abstractions from the domain (the terrain) that emphasize some features while suppressing others. If one wants to find a path from one village to another, then a relief map (one that shows only the height of the land) is unlikely to be of much help because it does not include the concept of a path. Its purpose is not appropriate for the purpose at hand. If we have a map with a scale of 1:50,000, this may be more suitable for the purpose of finding a path (but is less suitable for seeing the distribution of hills and valleys).

The map may have been designed to show fields, fences, rivers, and paths and so contains the required functions (the ability to follow a path on the map) and the necessary structure (e.g., the concepts of path) to fulfill its purpose. However, the map may be so poorly designed at the physical level (e.g., it uses black lines to show boundaries and paths and rivers) that it is not usable for its purpose. However, it is not simply a question of including all functions in order for the map to fulfill its purpose, to be objectively suitable. It is also a social or ideological issue. As Wood (1992) said about maps, “maps work by serving interests" (p. 4). The choice of things to represent on a map is not purely a technical problem; it is an economic, sociological, and political one (Suchman, 1995).

The question for interactive systems development, then, is which representations should be used for which purposes. Furthermore, we need to be confident that the representations that we use in HCI have an appropriate structure and an appropriate physical representation so that they can serve their purpose. Because HCI is concerned with developing human–computer systems, the concepts that underlie any given representation must be concepts that are suitable for representing people and their interactions with the other people, artifacts, and information that collectively make up the human–computer system. As we see, the word *suitable* applied to a representation means suitable for some social purpose. The underlying concepts must be consistent with the broader representations that define the social role that the artifact will play and the social role of users of the artifact.

3. EXPERIENTIALISM

Experientialism, or the theory of experiential cognition, developed out of Lakoff and Johnson's (1980) work, *Metaphors We Live By*. It continues to develop and be applied in a variety of areas (Fauconnier, 1997; M. Johnson, 1987; Lakoff, 1987). Within HCI, it is just starting to make an impact in terms of design (Waterworth, 1998) and has also been used to examine user concepts in using the Web (Maglio & Matlock, 1998).

Experiential cognition is an approach to understanding what meaning is to humans. The traditional objectivist view of cognition sees it as the algorithmic manipulation of abstract symbols that provide internal representations of an external reality. In contrast, experientialism sees cognitive activity as motivated and constrained by our experiences (sensory–motor, emotional, social) in the world. In particular, experientialists argue that we have an innate capacity to shape such experience and make it possible. Lakoff (1988) identified three main differences between objectivist and experiential cognition. Cognition in the traditional sense is disembodied, whereas, in experiential cognition, it is a function of human bodies. Traditional views of cognition see meaning coming from the association of symbols with external objects, whereas experiential cognition sees meaning coming from the application of "imaginative projections" to some basic concepts, these basic concepts being meaningful because of their roles in bodily experience. Experiential cognition distances itself from the idea that people reason by having many structured algorithms that work on their internal symbols, instead proposing a small number of general processes that are applied to abstract cognitive models.

It is not our intention to provide a full description of experientialism in this article. Much of the debate in Lakoff (1987) is concerned with the details of philosophical arguments concerning the nature of cognition. A brief introduction to the main concepts is in order, however, as we use these as the basis of our analysis in subsequent sections. The uneven treatment of the concepts contributing to experientialism provided here reflects the importance of the concepts to our analysis.

In terms of experientialism, rational thought is the application of the imaginative processes and basic cognitive processes to the basic concepts and image schemas that we possess. Meaningful structures arise from the structured nature of bodily and social experience and our innate capacity to imaginatively project from the bodily, social, or other interactional experiences to abstract conceptual structures.

Basic-Level Concepts. Experientialism accepts that we, as humans sharing a common experience and culture, possess a number of concepts or categories that are recognized as "basic level." Basic-level categories are formed at a level of abstraction "at which humans interact with their world most effectively" (Lakoff, 1988, p. 133). These concepts are basic in terms of

- Perception (they are easy to identify and correspond to a single mental image).
- Function (at both a motor and cultural level).
- Communication (the words for the concepts are typically common root words in the language that children learn first).
- Knowledge (most attributes of the category members are stored at this level).

Basic-level concepts represent significant discontinuities in the world for a population. Lakoff (1988) gave examples such as "table" and "giraffe." "Table" is a basic-level category, whereas "furniture" is not. It is a superordinate category. People do not have a single mental image that covers the entire category of furniture. You do not have a general motor program for using furniture in general, whereas you do at the level of table. You probably do not have a lot of knowledge about different species of giraffe (subordinate level), and you do not communicate often at this level. Similarly, you do not have a mental image of the superordinate category animal. Basic-level categories are a function of our experience in interacting with the world.

Image Schemas. In addition to basic-level concepts, people possess fundamental notions of spatial organization called *image schemas*. Image schemas are embedded in and structure our direct experience of the world.

However, they are more general than basic-level categories. They are common, experienced relations that pertain to humans and their existence in and movement through space. For example, we recognize a container as having an inside, an outside, and a boundary. A link joins two things together. Many things have a center and a periphery. We experience the front and back of things, we go in and out, up and down, move toward, follow a path, and so on.

Image schemas are "abstract patterns in our experience and understanding that … are central to meaning and to the inferences we make" (M. Johnson, 1987). They are described as being *nonfinitary,* which means that they are continuous, analogue representations and, thus, facilitate certain cognitive processes (discussed later). There is a container schema (things that have an inside, an outside, and a boundary), a part–whole schema (something can be seen as a whole or as its constituent parts), a link schema (two or more things have a link between them), and a source–path–goal schema (or sometimes just a path that goes from a source along a path to a destination). There is an up–down schema, a back–front schema, and so on. Schemas are gestalts—structured wholes—that structure our direct experiences.

> Take for example, the CONTAINER schema—a schema consisting of a boundary distinguishing an interior from an exterior. The CONTAINER schema defines the most basic distinction between IN and OUT. We understand our own bodies as containers—perhaps the most basic things we do are ingest and excrete, take air into our lungs and breathe it out. But our understanding of our own bodies as containers seems small compared with all the daily experiences we understand in CONTAINER terms. (Lakoff, 1988, p. 140)

Image schemas also have a main role in producing categories. To recognize several elements as structured by the same image schema is to recognize a category.

Imaginative Processes. Imaginative processes or imaginative projections act on the basic-level concepts and image schemas and allow us to form abstract conceptual models. Lakoff (1987) identified the following imaginative processes: schematization, metaphor, metonymy (when one member of a class or subclass stands in for the whole class), and categorization. It is important to distinguish concepts such as metonymy and metaphor as used in experientialism from their linguistic use. In experientialism, these concepts, along with schematization and categorization, are fundamental to cognition. Thinking and reasoning involve the application of the imaginative processes, linking and transforming the bodily basic-level categories and image schemas into abstract concepts.

The main process that we will be using is metaphor. To reiterate, the notion of metaphor here is not simply a linguistic expression. On the contrary, the locus of metaphor is the conceptualization of a domain. Metaphor is a cross-domain mapping—conceptualizing one domain in terms of another—and is central to our thought processes (Lakoff, 1993). Metaphors are sets of conceptual correspondences between domains. As an example of metaphor, we may consider a love relationship as a journey.

Our relationship has hit a dead-end street.

In this example, we defined two different domains: the source domain (travels) and the target domain (love). We can refer to this metaphor in terms of names that gives the following structure: TARGET-DOMAIN IS SOURCE-DOMAIN. In this case we have LOVE IS A JOURNEY and the mapping; the set of correspondences are

The lovers correspond to the traveler.
The love relationship corresponds to the vehicle.
The lovers' common goals correspond to their common destinations on the journey.

Mental Spaces. Mental spaces provide a medium in which cognitive activities can take place. Cognitive models created through imaginative processes structure those spaces. We think by connecting different mental spaces. For example, we may have a space that structures our experienced reality, another that structures future situations, and another that structures fictional situations.

The concept of *mental space* refers to the partial cognitive structures that emerge when we think and talk. It is in these mental spaces that domains are defined, altered, and merged: "Spaces are linked, or may be linked, to one another by 'connectors'. A connector establishes counterpart relations: it maps an element of one space onto one of more elements of another" (Fauconnier, 1988, p. 63).

Clearly, this concept of a connector is present in the use of metaphor. There is a source mental space, a target mental space, and connectors that map elements from both spaces. However, the concepts of mental spaces and connectors apply to more general situations involving more than two spaces.

Blends. A metaphor connects two different mental spaces. When a connection is established between more than two spaces, it is termed a *blend*. *Blending*—integrating partial structures from different domains—is another special case of imaginative projection. Blending receives a partial structure from two or more input spaces, producing a new space that has emergent structure of its own. Fauconnier (1997) saw the process of blending as follows:

- There is a partial mapping of counterparts between two input spaces.
- There is a generic space, which maps onto each of the inputs. This generic space reflects some common, usually more abstract, structure and organization shared by the inputs and defines the core cross-space mapping between them.
- The input spaces are partially projected onto a fourth space, the blend.

The blend has emergent structure not provided by the inputs. New relations arise that did not exist in the separate inputs and, taken in the context of background cognitive and cultural models, allow the composite structure projected into the blend to be viewed as part of a larger self-contained structure. The structure in the blend can then be elaborated in terms of its own logic. This is "running the blend."

Basic Cognitive Processes. In addition to imaginative processes, there are cognitive processes that act on the basic concepts. These processes are high level compared to traditional cognitive processes and include things such as scanning, focusing, vantage-point shifting, and so on. "A vantage point is the position from which a scene is viewed: as I walk along the sidewalk looking at a house my vantage point with respect to the house gradually shifts" (Langacker, 1987, p. 123). The concept of *scanning* is related to our ability of comparing and registering events: "Such comparison is at work when we perceive a spot of light against a dark background, for example, or when we catch a spelling error" (Langacker, 1987, p. 101). So, basic cognitive processes are related to sensory or attention experiences of looking, listening, observing, and so on and not to the general imaginative capacity of projecting between mental spaces that characterize metaphor and blends.

Applying Experientialism. Fauconnier provides many examples of these processes in action (see Fauconnier, 1997). He sees the process as cognitively complex, involving several stages of transitions and projections. In an analysis of the concept of computer viruses, he identifies six stages. Through analogy between biological and computer viruses, we induce a schema that classifies key features of viruses. This allows us to categorize certain computer programs as viruses and to develop and name a new conceptual structure. This structure is a blend that now has its own properties. A new domain is created integrating knowledge about computer and biological viruses and generic knowledge that results in the word *virus* taking on different meanings. Finally, the new domain of computer viruses emerges as a distinct domain of its own.

The concept of a blend is important because it helps to explain where new structures come from. For example, the famous desktop "metaphor" is in reality a blend from the mental space of offices and the space of computer operations. Some of the strange behaviors in the Macintosh user interface (e.g., dragging a diskette onto the trash can) are not so much a result of applying a poor metaphor but of producing some odd elements in the blend. Adding an icon that represents the floppy disk unit with a button to eject diskettes would have solved the problem of considering metaphors as harmful (Halasz & Moran, 1982).

As another interesting example of blend, consider the case of *visual formalism*. The definition was given by Harel (1988):

> The intricate nature of a variety of … systems and situations can, and in our opinion should, be represented by *visual formalisms;* visual, because they are to be generated, comprehended, and communicated by humans; and formal, because they are to be manipulated, maintained, and analyzed by computers.

The term *visual formalism* is defined by a blend space. This blend space is connected to two input spaces; one is the space of human activity in general, in which tools and artifacts—in this case conceptual artifacts—need to be considered from an ergonomic perspective. They are to be generated, comprehended, and communicated by humans, so a visual presentation is better than an abstract one. On the other hand, the second input space is that of computing, in which objects that are to be manipulated, analyzed, and maintained by computers have to be formally defined. It is in blend spaces that we construct artifacts such as tables, graphs, maps, control panels, and so on. Some of these artifacts are derived from successful metaphors and others from projections of mental spaces directly related to workplaces. The construction of these objects is a concern of computing science, whereas cognitive science tries to explain the mechanisms of the design and construction of these conceptual artifacts.

4. APPLYING EXPERIENTIALISM TO THE ACTIVITY OF HCI

Experientialism provides us with a set of conceptual tools that we can use to understand several aspects of the development of representations in interactive systems development. Imaz (1995) used experientialism to explain the paradigm shift that occurred in moving from procedural to OO programming.

Fauconnier and Turner (1994) used the blend concept to explain the theory of complex numbers in mathematics, and Imaz and Benyon (1996) used the metaphor concept to explain pilot error. Lakoff (1987) provided three detailed case studies applying experientialism to the concept of anger, the word *over,* and grammatical constructions. Maglio and Matlock (1999) focused on the image schemas that people use when navigating the Web, and Waterworth (1996, 1999) used experientialism to inform the design of a Web-browsing environment.

The main reason why we take experientialism into consideration is that it provides a different way of thinking about the effectiveness and impact of representations in HCI. Lakoff (1987) provided a powerful and detailed critique of the objectivist tradition of cognition and classification. This traditional approach places the concept of category at the center of its philosophy. However, although we might all agree about certain everyday experiences—for example, that I am sitting at a table and typing on a keyboard—the classical concept of category, that there are natural kinds of things in the world and that these entities form objectively definable categories based on them having shared characteristics, is

> untenable as a fully general approach … Such commonsense assumptions about physical objects do not necessarily extend to other domains. When we use them to deal with political movements, inflation, friendships, marriage, our emotions, and our foreign policy, … the entities and properties are by no means so clear, nor is the distinction between what is essential and what is accidental. (Lakoff, 1987, pp. 174–175)

We believe that interactive systems design is such a domain. Computer users, other computer users, designers, and all the other various stakeholders involved in interactive systems development do not share a simple, objective set of concepts and categories. Requirements and designs have to be negotiated, and the language used for that negotiation is made up from the models and concepts employed in the design process.

Our aim, then, is to look at representations (*re-present:* to put something again or in place of another thing; to present or picture to the mind; to be a sign or symbol for; to present in words, describe, state, or set forth, etc.) considered as models or artifacts. It is important to be aware of different types of representations; *explicit* representations, such as models and artifacts, and *implicit* representations contained in the underlying metaphors we use to describe a given domain. The implicit representations are associated background, roles, and culture. They form part of (or are embedded in) discourses, "a way of knowledge, a background of assumptions and agreements about how reality is to be interpreted and expressed, supported by paradigmatic metaphors, techniques, and technologies and potentially embodied in social institutions" (Edwards, 1996, p. 34). The meaning of categories is determined by the context of the discourse and, ultimately, by its embedded metaphors.

We are aware of some criticisms against the use of metaphor in design. One of the most recent proclaims "Designers of the world: Forget the term 'metaphor'" (Norman, 1998, p. 181). Norman argued that because the objects on the display of his computer are not the same as real objects that have the same name, metaphor will not help. However, just prior to this in his book he wrote:

> When I encounter a new situation, how do I know what to do? I look, listen, and copy. I try to understand what is happening. I see if I can find anything that looks familiar, and if I do, then I'll perform the actions that work in the familiar situation. … When I encounter a new piece of technology I do the same thing. I look at it and try to see if anything looks familiar. (p. 176)

What Norman said here is the essence of metaphor: to conceptualize one domain in terms of another.

With respect to his argument about objects on the computer screen being different from real objects, no one would disagree. A *window* in the user interface means something different from the *window* of my room, but this is not a true metaphor; it is a blend. One of the input spaces corresponds to a metaphor. However, being a blend, in this case a visual formalism, it has a different meaning from the word *window* when referring to my room. Ordinary language is full of the same type of example: "Norman is a shining scholar." Does this mean that Norman is shining like metal? No, as a consequence of using a metaphor, we get a blend where we combine the shining of objects with intellectual qualities of people. The result is a new meaning of the word *shine*, the basis of polysemy.

At this point, it is useful to show the difference between metaphor and analogy. The main difference between both concepts was pointed out by Presmeg (1997) when she said that the word *analogy* covers two different concepts: *similes,* which are explicit analogies ("*a* is like *b*") and *metaphors,* which are implicit analogies ("*a* is *b*"). "Metaphor is a specific form of analogy" (p. 267), but showing similarities in an implicit way has a strength that is lost in the explicit form. In this sense, Sfard (1997) wrote that

> The main point to remember is that *metaphor has a constitutive power* and thus functions *a priori*: It brings the target concept into being rather than just sheds a new light on an already existing notion … *the act of creation itself is a matter of metaphor.* … Analogy, or simile, on the other hand, makes a comparison between two already constructed concepts, even if this comparison does not leave our understanding of either the target or the source unchanged. (pp. 344–345)

As discussed previously, the concept and use of metaphor are some of the cognitive tools that experientialism offers. Of course, metaphor has been applied to HCI previously (Carroll & Mack, 1985; Erickson, 1990), and Lakoff and Johnson's (1980) book, which emphasizes the linguistic aspects of metaphor, is cited frequently. However, we feel that the later work of Lakoff (1993), M. Johnson (1987), and their colleagues has established the concept of metaphor not simply as a linguistic device but as one of the fundamental features of cognition. Indeed, we have found that during the writing of this article, the fact that we were discussing which metaphors to use to describe different concepts and using different metaphors allowed us greater insight into possible interpretations of the concepts.

In Section 2, we described how HCI may deal with any of several levels of abstraction at which we want to describe the system and identified the organizational level, the workplace level, and the operational level. We can now see that this consideration of levels implies that a spatial metaphor has been used to describe the concerns of HCI. It implies that depending on where we observers place ourselves, we have different perspectives or points of view. It is evident that the spatial metaphor refers to a "social place" in the complex network of social relationships, but this social place is rarely explicated with all its consequences. At the same time, the rationalist approach of a neutral observer with an objective point of view and without spurious interests immediately appears questionable. The most objective point of view is considering the observer along with all the biased ideas and interests embodied in the social role being played.

4.1. Organizational Level

At the organizational level, there are at least two different perspectives that can be taken: the organizational view and activity-based view (Sachs, 1995). Adopting one or the other of these perspectives leads to different visions of actors, relationships, skills, commitments, and so on. The foundation of this approach is the spatial metaphor, the view we adopt. The organizational view takes the point of view of management, about workers needing training, working in tasks with procedures, using techniques, and so on. The alternative point of view is of activities with people discovering problems, learning in informal conversations, creating communities of practice (Lave & Wenger, 1991), and developing knowledge and skills.

It is not the aim of this article to analyze the collection of metaphors that could build the core of the ideas of these different views. We can only quote some of the metaphors used by Lakoff (1996) in relation to work. He pointed out that there are two different common metaphors for work, each of which uses moral accounting: the work reward metaphor and the work exchange metaphor.

> [The first] can be stated as follows:
> - The employer is a legitimate authority
> - The employee is subject to that authority
> - Work is obedience to the employer's commands
> - Pay is the reward the employee receives for obedience to the employer …
>
> [The second] can be stated as follows:
> - Work is an object of value
> - The worker is the possessor of his work
> - The employer is the possessor of his money
> - Employment is the voluntary exchange of the worker's work for the employer's money. (pp. 54–55)

Both metaphors define work from the organizational point of view. It would be interesting to include some additional metaphors regarding other aspects of each point of view. A last example shows how a given problem (the "set-up-to-fail" syndrome) could be based on a metaphor: THE EMPLOYEE IS A CHILD (and, hence, lacks knowledge and skills). One consequence of the metaphor is the representation the boss has of the situation: "when an employee fails—or even just performs poorly—managers typically do not blame themselves. The employee doesn't understand the work … or isn't driven to succeed, can't set priorities, or won't take direction" (Manzoni & Barsoux, 1998, p. 101).

The main point about representations derived from an organizational view is the danger of failure produced by a poor vision of workplaces. Sachs (1995) described a particular technology that was introduced in a telephone company to provide a trouble ticketing system (TTS). This system replaced an old one, which

> allowed workers to talk to one another. In these conversations, they compared notes about what was going on at each end of the circuit. If there was a problem, they figured out what it was and worked on it together. These trouble-shooting conversations provided the occasion for workers to understand what was actually going on in the job, diagnose the situation and remedy it. (p. 39)

The organization took the view that *employees are children,* abandoning their responsibilities and engaging others in conversation without any usefulness for the job. This view led to the apparently "objective" need and requirements for the TTS system, the aim of which was to eliminate conversation. However, we argue that such views are not independent of any observer. They are derived from metaphors embodied in a given discourse and are suitable for some social interests, in this case the organizational point of view. At this organizational level, just detecting and modifying the underlying metaphor should help to reorganize the whole domain of needs and requirements.

4.2. Workplace Level

Another level of HCI is the workplace level. Madsen (1994) provided a summary of the use of metaphors in system design. One of the cases extracted from a set of five included in the article presented a design of a bank automated teller machine (ATM) taken from Maclean, Belotti, Yound, and Moran (1991). The authors described that designers had personal experience of a bagel store that

> handled its lengthy queues by having an employee work along the queue, explaining the choices available and helping fill out their order on a form. The customers would hand over their forms when they reached the counter, enabling their requests to be processed more speedily. (p. 169)

"Their familiarity with the bagel store arrangement lead the designers to the innovative idea of having bank cards the customers could preprogram while waiting in line" (Madsen, 1994, p. 58).

In the case of the ATM example, it is evident that we are not comparing two different existing systems (the bagel store vs. the bank ATM), as the latter is not implemented yet. When we apply the metaphor ("The bank ATM system is a bagel store"), we create two mental spaces: one corresponds to the similar elements (the *ground*) and the other to dissimilar elements (the *tension*). There is a dynamic relationship between both ground and tension that allows some elements of tension to pass to the ground, giving this process its constitutive power in the metaphor. Although considering that the employee working along the queue explaining the choices available belongs to the tension, there is no new conceptualization about the ATM system. It is when we move this idea to the ground (similar elements) that the new vision of the system appears: It is possible to replicate the employee task of helping to fill the form in its equivalent way of preprogramming the bank card.

The metaphor suggests some possible generic elements in the tension mental space. One of these possible generic elements is the employee working along the queue, but the equivalent in the ATM system is not immediately identified. Sometimes the generic space, with the common, usually more abstract, structure and organization shared by the inputs, will work as an intermediate phase before arriving at the final blend space. The generic space contains what both input spaces (the bagel store and the ATM system) have as equivalent abstract structure: the possibility of doing some task while waiting in line for the counter to speed up the process. Finally, the blend space will contain the actual form of speeding up the process as a programmable bank card. The blend space is built on structure coming from both input spaces and other spaces as well, in particular from that of the current bank card technologies. This last mental space is important, as it allows the proposal of a programmable bank card to be actually evaluated and implemented.

What the metaphor contributed to is the *envisioning* of new functionalities and, consequently, the modification of the requirements of the system being designed. The design process is a complex one, and we have been warned (Davis, 1993) about the difficulties of completely defining the requirements before beginning the design phase. During design, users uncover new possibilities that may result in additional functionalities. The role of metaphor is to anticipate what normally can be detected in the design phase, even if not all new functionalities can be anticipated by metaphorical design. On the other hand, some of the functionalities detected by a metaphorical design may be recognized by users because the possibilities offered by new technology are not available to them. The blend offers a wide choice of technological spaces to be taken into account.

We can conclude that metaphors are the original generative force, but we have to use blend spaces (and generic spaces) as a way of making triggering concepts workable or elaborating these triggering concepts. In the same way as general software engineering methods advocate documenting some critical design decisions through design rationale, we advocate including in the ontology of HCI an additional issue about metaphors used in design with a trace to all triggering concepts and the resultant blend spaces.

4.3. Operational Level

At the organizational level, metaphors can help in generating needs and requirements, even if in the TTS such needs were inadequate to solve the (apparent) problem of employees engaged in time-consuming conversations. An alternative metaphor (EMPLOYEES ARE MATURE INDIVIDUALS and engage in conversation when needed to solve their problems) would have determined that trouble-shooting conversations were not a problem but the solution. At the workplace level, metaphors can help us to envision new aspects and opportunities of the systems we are designing. In this case, metaphors are artifacts used to produce new functions during design. However, it is not metaphor on its own that gives the design solution; different blends offer different design alternatives.

At the operational level, our focus is to consider the artifacts we use in constructing systems. The concepts and notations employed in systems design are artifacts for constructing new artifacts: the models themselves. In terms of activity theory, we differentiate between the artifact used as a means (the cognitive artifact, or concept) and the object to be built (the conceptual artifact), even if that object will subsequently be used to construct new objects (the domain artifact).

For example, we describe the E-R diagram as follows:

- We use the cognitive artifacts to build the diagrams. E-R diagrams use the cognitive artifact of an *entity,* which consists of the entity concept and the associated notation (usually a rectangle), and a *relationship,* which consists of the concept of a relationship and a notation (usually a line).
- Conceptual artifacts, or conceptual models, that can be constructed using the cognitive artifacts.
- Domain artifacts (models or representations in a given notation, such as an E-R diagram, for a given system, such as a bank system) that have been built using conceptual artifacts.

The activity of constructing conceptual artifacts is based on primitive cognitive artifacts such as image schemas. M. Johnson (1987) argued that the term *image schema* (*schema* or *embodied schema*) "reminds us that we are dealing with schematic structures that are constantly operating in our perception, bodily movement through space, and physical manipulation of objects" (p. 23). A schema is a "*recurrent pattern, shape, and regularity in, or of, these ongoing ordering activities*" (p. 29), so image schemas are artifacts that derive from our everyday elementary activities; they are produced in such activities. However, as (cognitive) artifacts, we employ them in higher cognitive processes to conceptualize more abstract aspects of reality.

To understand the application of experientialism to the operational level, we can look at the following generic frame (presented in terms of activity theory). This shows how the activity of producing conceptual artifacts may proceed. In such activities, there are different elements involved:

- Cognitive artifacts, such as image schemas.
- Cognitive processes acting on the cognitive artifacts, such as metaphorical projection from image schemas.
- A subject: the community of participants in the design of cognitive artifacts.
- Blends that define the structure of new objects such as entity objects, relationship objects, state objects, and data flow objects.

We call the objects produced in this activity *blends* to focus attention on the mental spaces that are involved in the activity and on the new emergent structures that result. Let us consider the case of the entity concept in terms of the previous frame:

1. The image schema on which the entity concept is based is the CONTAINER. This image schema is the source mental space. An entity is something containing a name and its attributes. There is an inside (its name and attributes) and an outside (other entities) with which the ent ity may maintain relationships.
2. The metaphor used is AN ENTITY IS A CONTAINER.
3. The system designer and users of the system constitute the community of participants.
4. The blend produces the entity object by bringing together aspects from (a) the structure of the image schema (interior, exterior, boundary), (b) the logic of the image schema (if Container A is in B and Container B is in C, then Container A is in C), (c) the mental space of geometrical forms (the visual notation associated with an entity is usually a rectangle), and (d) the mental spaces of the domain (attributes).

In the next section, we take this conception of cognitive artifacts and apply it to a number of representations that are commonly used in the design of interactive systems. We use it to shed light on the differences between various modeling methods and alternative views on HCI. In doing so, we find that we can see the structure, functions, and constraints that different models are effective at capturing more clearly. In turn, this allows us to look at the influence that representations have on possible designs.

5. REPRESENTATIONS AND CONCEPTS FOR HUMAN–COMPUTER SYSTEMS DESIGN

As we indicated in the introduction, there have been many alternative methods and representations proposed in the field of HCI. In this section, we look at a few of the more common representations and concepts, using experientialism as our cognitive tool. These representations belong to the operational level of description of HCI as described in Section 4.3. The metaphors, image schemas, and blends that we use later are not intended to be *the* definitive way of conceptualizing the domain of HCI (this would imply an objectivist approach to reality, not an experientialist one); rather, they are different ways that have provided useful insights. They provide an explanation of how a given representation highlights some aspects of the domain of interactive systems design, allowing some characteristics to be shown and others to be hidden.

The tension between *visible* and *invisible* in this description (*highlights, hidden*) is inherent in the way that we examine representations. A collection of metaphors gives a wider view of the domain as some of the issues that are suppressed using one metaphor (recall that a metaphor only implies a partial mapping between spaces) are illustrated by others.

Entities. Let us return to the notion of an entity. In Section 4.3, we used a metaphor based on the container image schema, allowing us to see data elements as *contained in* the entity. The same schema means that we view entities as categories of events or things and collections of characteristics of these events (the attributes of the entity) that belong to these entities. This view leads us to conceptualize entities first and attributes after; attributes *belong to* entities.

An alternative view is to consider an entity as an aggregation of data elements (Benyon, 1997). Here we are conceptualizing the notion of an entity based on the PART–WHOLE image schema. The structural elements of this schema are a whole, parts, and a configuration. The basic logic of the schema allows all the PARTS to exist but still not constitute a WHOLE. It is only when the PARTS exist in the CONFIGURATION that the WHOLE exists. So, we can consider data elements as isolated PARTS, but, when focusing on a CONFIGURATION, these groupings could be established and the designer can refer to the collection of data elements—the entity, the WHOLE—by name. This suppresses detail that would otherwise clutter the model. Relations between entities are derived from the LINK schema.

States. A state transition diagram is based on a SOURCE–PATH–GOAL schema, where each node represents a location (it may be a source of a path and a goal of another path). The diagram is a network of such schemas. Even the word *transition* is derived from the spatial metaphor, *transit* (from Latin *trans,* over or across, + *ire,* to go). It is to move, to travel from one location to another. A state transition representation of a system will help the designer to focus on the movement, on the paths, and on beginnings and ends. An E-R diagram, on the other hand, helps to focus on configurations and links.

Data Flows. State transition representations focus on how to get from one state to another and the paths that exist or are otherwise allowable. Another way to model a domain is to look at how data flow between processes, and here a DFD is often used. The metaphor underlying DFDs may be expressed as THE PROCESS IS A MACHINE. A data flow is seen as the raw material that is transformed by the process into some other, more useful data. In this case

- The input data flows are collections of data.
- The output data flows are transformed collections of data.
- The process is the making of output data flows from collections of input data flows.

Object Orientation. In OO methods of systems development, the domain is represented in terms of the objects that exist, the relationships between objects, and the messages that are passed between objects. OO methods employ a very different metaphor: OBJECTS ARE PEOPLE. The language used in OO methods reveals the "Objects are people" metaphor clearly. Objects "pass messages" to one another, they have "responsibilities," play different "roles" so that they may "collaborate," and so on. An object also clearly employs the CONTAINER schema, as an object contains (encapsulates) structure and behavior:

> This relative fixing of location within the container means that the contained object becomes either accessible or inaccessible to the view of some observer. It is either held so that it can be observed, or else the container itself blocks or hides the object from view. (M. Johnson, 1987, p. 22)

We see that the information-hiding aspect of OO is a direct consequence of a coherent application of the container schema. The OO model also uses the link schema to connect objects and represents by this connection responsibilities, message passing, and so on.

The concept of object corresponds then to a blend from the following mental spaces: abstract data types (structure, behavior), people (responsibilities, collaborations, roles, messages), containers (encapsulation, tension between public, reserved, and private), and links (responsibilities, acquaintance, message passing). The OO approach thus demonstrates some emergent properties. One of these, it is often claimed, is that the approach leads to a more "natural" design. However, we do not find it natural that, for example, orders can answer us when we ask them the name of their customer or clients can print themselves when requested to do so. However, perhaps like trash cans on desktops, the blended space will go beyond the OBJECTS ARE PEOPLE metaphor.

Tasks. The concept of *task* has dominated the ontology of HCI since the earliest days (Card, Moran, & Newell, 1980; Moran, 1981), resulting in many task analysis methods and task-based approaches to HCI that continue to be popular today (e.g., Browne, 1994; Lim & Long, 1994). A task may be defined as a goal together with some procedure or ordered set of actions that will achieve that goal. Hence, other basic concepts include *goal* (a state of the environment or agent that is desired by the agent), *action* (a task that involves no problem-solving or control component), and *procedure* (a sequence of tasks, subtasks, actions, or a combination of these).

The schema underlying the notion of task is SOURCE–PATH–GOAL, as every time we move anywhere there is a place we end up at, a sequence of contiguous locations connecting the starting and ending points, and a direction (Lakoff, 1988, p. 144). People are given plans and undertake tasks to achieve goals. There are procedures, operations, and methods. The structural element of the schema underlying the task concept is the path, so the spatial metaphor could be THE USER IS A FOLLOWER OF PATHS, with a source state, a set of tasks (the path), and a goal state. The tasks are represented by the incidents the traveler has to suffer during the travel.

Activities. The idea of task may be contrasted with the notion of an activity, which may be characterized by the activity theorists' own metaphor THE USER IS AN ACTOR (e.g., Bannon, 1991). In her keynote talk at HCI '97, Nardi (1997) spoke about bringing people to the center stage of design, and, of course, Laurel (1991) in *Computers as Theatre* utilized exactly this metaphor. She pointed out that

> the key to understanding and designing what is going on in a human–computer activity is an understanding of work as human action … it is not simple *work* that we do with computers, but *work in a representational context.* And clearly, we do other things with computers, too—we learn, explore, noodle around, play, and entertain ourselves. (p. 134)

As we argue later, these alternative metaphors can have a significant impact on how we undertake design and how we view people and the things that they do with computers. Are users actors on a stage, part of a play, or followers of paths, or are they something different?

Representations of Requirements. The use of different terms to refer to the activities associated with system requirements implies different ways of conceptualizing them. As Jirotka and Goguen (1994) pointed out, there are many ways of describing what requirement engineers do: to "capture," "specify," "elicit," or "construct" requirements.

REQUIREMENTS ARE A CAPTURE makes us think that requirements are like an animal. The mobility of the capture implies that requirements are elusive but can be eventually captured. Requirements are continuously "escaping" from our hands. To specify means that REQUIREMENTS ARE AN EXISTING ARTIFACT. In this metaphor, requirements are represented by something that is "out there" (existing independently of ourselves). By using adequate methods we can "analyze" and after that take the artifact to pieces to "specify" it. Elicitation suggests that REQUIREMENTS ARE MENTAL REPRESENTATIONS (MODELS) to be extracted from the users' heads to have a complete list of requirements. Normally, the representation metaphor implies that these representations are conscious, so it is a matter of asking users about their mental models to get a complete requirements elicitation. Construction implies that REQUIREMENTS ARE AN ARTIFACT TO BE BUILT. This is the only metaphor not implying something existing "out there" but something to be constructed, such as a building. Just as the new system is something to be designed and implemented, requirements are also to be derived from the experience we have with the workplace.

These metaphors are not used in isolation but mixed in our everyday language. Some of them are mutually contradictory, such as capturing and specification. Whereas capturing requirements suggests an ever changing process, specification relies on the assumption of stability or fixed conditions. A blend of such mental spaces would allow us to have a richer image of what requirements could be. This new blend would consider requirements as having structure from all these input spaces, so there are some existent aspects of the domain we can detect in the workplace. Others are to be built as a consequence of the process of construction (they are not existent, yet they have to be designed in the new system), although at the same time, we are dealing with a capture, as requirements are elusive, changeable, and so on.

Conceptions of HCI. The paradigm shift between a data-centered process method and OO methods that occurred in the 1980s and 1990s corresponds to a change in the underlying metaphor. We moved from the PROCESSES ARE MACHINES metaphor to OBJECTS ARE PEOPLE. In a similar way, we have seen a paradigm shift from task-based to activity-centered approaches to HCI, from PEOPLE ARE FOLLOWERS OF PATHS to PEOPLE ARE ACTORS.

There is, no doubt, another paradigm shift around the corner. The direct manipulation, WIMP interface, and task-based analysis and design will soon be superseded by the indirect management, delegated approach of agent-based interaction (Kay, 1990). The passive retrieval of information from systems will give way to a more active involvement of people within their "information space" (Benyon & Höök, 1997).

These changes are already resulting in new metaphors. For example, Maes (1997) proposed a biological metaphor in connection with intelligent personal agents: AGENTS ARE INSECTS. The important thing is that the metaphor brings to us an input mental space: the world of insects with a set of actions they perform. We think immediately of some of their characteristics—for example, their mobility and the way they interact. This is a valuable source of inspiration, because it is possible to project some of the structure of this input space to the blend: "Agents could leave the digital equivalent of a pheromone at documents they deemed relevant to their users, thereby attracting more agents toward those same documents" (p. 17).

We also expect to see attention shift from application-based systems to domain-oriented environments (Fischer, 1989). In such a situation, our metaphors for systems development must similarly shift. Benyon (1998) presented a conception of HCI as THE CREATION OF INFORMATION SPACES. Such a view encourages us to look to the designers of physical, geographical spaces—architects, city planners, and the like—to help us understand our discipline. In the design of physical space, we have seen a move away from a utilitarian view of engineering toward a recognition of the social, cultural, and political environment that people inhabit. Postmodernism has taught us that engineers cannot dictate the nature of space. It is people who produce spaces (Lefebvre, 1991). Seeing HCI as people engaged in cognitive, personal, and social activities in information spaces leads us to a new metaphor, PEOPLE ARE NAVIGATORS. This takes us away from the staged, scripted performance suggested by PEOPLE ARE ACTORS and into the active exploration of domains.

Elsewhere, Benyon and Höök (1997) have provided a variety of alternative metaphors for thinking about information spaces that help to focus on different characteristics of information spaces and the various activities in which people might engage. A wilderness may be frightening, confusing, or enchanting; a desert is intimidating, is beautiful, and has few landmarks. In terms of the activities that people undertake in information space, these metaphors might encourage forging a path, enjoying the scenery, or exploration. Thinking about information space as the night sky supports the activities of mapping and identifying objects, clusters, and configurations. The open sea as a metaphor encourages a distinction between the surface and depth. Thus, it is natural to think of a lot of information as being hidden beneath the surface and only available for viewing if the user dives down. Currents can link continents and islands and take people to unexpected places. People can look for islands of information; archipelagos provide clusters. A breakfast bar (so popular in holiday hotels) offers another metaphor for thinking about information space. Here, a whole range of choices is laid out for the users. The user can see how far the space extends and can be offered choice. Users are encouraged to feed a need, take what they want, and come back for more. It is a constrained, help-yourself environment.

6. CONCLUSIONS

Experientialism gives us a powerful cognitive tool, the metaphor, which can be useful at different levels of study (organizational, workplace, and cognitive), thereby allowing us a deeper understanding of some underlying (implicit) knowledge. It is in the organizational discourse that we have to discover, or capture, needs and requirements of new computer-based systems. We have seen that there is no homogeneity in the organizational level discourse and that each point of view or perspective—corresponding to groups with different practices in the organization—is based on different metaphors and results in different solutions to a given problem.

At the workplace level, the use of metaphors in disciplines such as metaphorical design (Madsen, 1994) generates ideas for new functionalities (or services) in systems to be implemented. The systematic use of metaphors has a positive impact in design, even if metaphors have to be chosen in a given context. There are no universal metaphors to be applied; instead they are related to the community that will use the implemented system. In the example we presented, it is the personal experience of a bagel store that could be used, because this is the experience of the community developing the system. The specific issue is the metaphor employed; the generic pattern is that of searching useful metaphors. We could say that metaphorical design is a way of structuring what has been called *brainstorming,* taking into account that this last term is a blend based on the metaphor IDEAS ARE A METEOROLOGICAL PHENOMENON, sometimes stable, sometimes variable. In the blend, we have also an input space corresponding to a CONTAINER image schema where ideas are contained in the brain.

The operational level may be explained in terms of metaphors, blends, and image schemas, providing the conceptual richness as a consequence of the underlying metaphors. As we have seen, whereas a structured method (E-R models) is based on blends produced with image schemas such as CONTAINER or PART–WHOLE, an OO method is based on blends derived from the same mental spaces with addition of new metaphors such as OBJECTS ARE PEOPLE. At the workplace and operational levels of HCI, although metaphor continues to be the driving force, it only allows a superficial analysis on its own. So, we need to think in terms of blends and image schemas to give a more complete explanation of underlying conceptual foundations of the representations.

The complexity of HCI requires us to employ a range of models to gain the variety of insights that are necessary if we are to design successful human–computer systems. At different points in the design process, different models will be more or less useful for designers. We need to consider the social construction of interaction and the use of language and thought in situations, and here experientialism may give us a set of concepts with which to do this.

The examples provided in Section 5 are not intended to be exhaustive but rather illustrative of the type of analysis that experientialism can bring to bear. Specifically in this article, we looked at what the concept of metaphor can offer. Experientialism allows us to model concepts and representations using metaphor. As we mentioned previously, we found that thinking about concepts in this way enables us (as designers) to discuss alternative models (i.e., alternative metaphors) and to gain an insight into the conceptual foundations, strengths, and weaknesses of representations.

NOTES

Background. The work reported here is part of the ongoing doctoral research being conducted by Manuel Imaz.

Acknowledgments. This article has been significantly improved by the comments of the editors of the special issue and of the anonymous referees. Their questions and remarks have helped us revisit many of the issues and have contributed to the development of this work.

Authors' Present Addresses. David Benyon, School of Computing, Napier University, 42, Craiglockhart Avenue, Edinburgh, EH14 1DJ, United Kingdom. E-mail: d.benyon@dcs.napier.ac.uk. Manuel Imaz, School of Computing, Napier University, 42, Craiglockhart Avenue, Edinburgh, EH14 1DJ, United Kingdom. E-mail: imaz@compuserve.com.

HCI Editorial Record. First manuscript received September 22, 1997. Revisions received April 6, 1998 and October 16, 1998. Accepted by Eamonn O'Neill, Peter Johnson, and Hilary Johnson. Final manuscript received November 9, 1998.—*Editor*

REFERENCES

Bannon, L. (1991). From human factors to human actors. In J. Greenbaum & M. Kyng (Eds.), *Design at work: Cooperative design of computer systems* (pp. 25–44). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Benyon, D. R. (1996). Domain models for user interface design. In D. R. Benyon & P. Palanque (Eds.), *Critical issues in user interface systems engineering* (pp. 3–19). London: Springer-Verlag.

Benyon, D. R. (1997). *Information and data modelling*. Wokingham, UK: McGraw-Hill.

Benyon, D. R. (1998). Cognitive ergonomics as navigation in information space. *Ergonomics, 41*(2), 153–156.

Benyon, D. R., & Höök, K. (1997). Navigation in information space: Supporting the individual. *Proceedings of INTERACT'97,* 39–46. London: Chapman & Hall.

Benyon, D. R., & Skidmore, S. R. (1987). A tool-kit approach to information systems development. *The Computer Journal, 31*(1), 2–7.

Bødker, S. (1990). *Through the interface.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Booch, G. (1992). *Object-oriented design with applications*. Menlo Park, CA: Benjamin/Cummings.

Browne, D. (1994). *Structured user design interaction: STUDIO methodology.* Englewood Cliffs, NJ: Prentice Hall.

Card, S., Moran, T. P., & Newell, A. (1980). *The psychology of human–computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Carroll, J. M., & Mack, R. L. (1985). Metaphor, computing systems and interactive learning. *International Journal of Man–Machine Studies, 22,* 39–57.

Chen, P. (1976). Towards a unified theory of data. *ACM Transactions on Database Systems, 1,* 9–36.

Codd, E. F. (1970). A relational model of data for large shared databanks. *Communications of the ACM, 13,* 377–387.

Codd, E. F. (1982). Relational database: A practical foundation for productivity. *Communications of the ACM, 25,* 109–117.

Davis, A. M. (1993). Software requirements: Objects, functions and states. Englewood Cliffs, NJ: Prentice Hall.

DeMarco, T. (1979). *Structured analysis, systems specification*. Englewood Cliffs, NJ: Yourdon.

Diaper, D. (Ed.). (1989). *Task analysis for human–computer interaction*. Chichester, UK: Ellis-Horwood.

Edwards, P. (1996). *The closed world: Computers and the politics of discourse in cold war America*. Cambridge, MA: MIT Press.

Erickson, T. (1990). Working with interface metaphors. In B. Laurel (Ed.), *The art of human computer interaction* (pp. 65–73). London: Addison-Wesley.

Fauconnier, G. (1988). Quantification, roles, and domains. In U. Eco, M. Santambrogio, & P. Violi (Eds.), *Meaning and mental representations* (pp. 61–80). Bloomington: Indiana University Press.

Fauconnier, G. (1997). *Mappings in thought and language*. Cambridge, UK: Cambridge University Press.

Fauconnier, G., & Turner, M (1994). *Conceptual projection and middle spaces* (Technical Report 9401). San Diego: University of California–San Diego, Department of Cognitive Science.

Fischer, G. (1989). Human–computer interaction software: Lessons learned, challenges ahead. *IEEE Software, 1,* 44–52.

Greenbaum, J., & Kyng, M. (Eds.). (1991). *Design at work: Cooperative design of computer systems*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Halasz, F., & Moran, T. P. (1982). Analogy considered harmful. *Proceedings of the ACM Conference on Human Factors in Computer Systems,* 383–386. New York: ACM.

Harel, D. (1988). On visual formalisms. *Communications of the ACM, 31,* 514–520.

Hollnagel, E., & Woods, D. D. (1983). Cognitive systems engineering: New wine in new bottles. *International Journal of Man–Machine Studies, 18,* 583–600.

Hutchins, E. (1995). *Cognition in the wild*. Cambridge, MA: MIT Press.

Imaz, M. (1995). *Object oriented methods: An epistemological approach*. Unpublished master's thesis, Open University, UK.

Imaz, M., & Benyon, D. R. (1996). Cognition in the workplace: Integrating experientialism with activity theory. *Proceedings of the Eight European Conference on Cognitive Ergonomics,* 103–108. Le Chenay Cedex, France: European Association of Cognitive Ergonomics, INRIA.

Jirotka, M., & Goguen, J. (Eds.). (1994). *Requirements engineering. Social and technical issues*. London: Academic.

Johnson, M. (1987). *The body in the mind. The bodily basis of reason and imagination.* Chicago: University of Chicago Press.

Johnson, P., Johnson, H., & Wilson, S. (1995). *Rapid prototyping of user interfaces driven by task models.* In J. M. Carroll (Ed.), *Scenario-based design: Envisioning work and technology in system development* (pp. 209–246). New York: Wiley.

Kangassalo, H. (1983). Structuring principles of conceptual schemas and conceptual models. In J. Bubenko (Ed.), *Information modelling.* United Kingdom: Chartwell-Bratt.

Karat, J. (1991). *Taking software design seriously.* London: Academic.

Kay, A. (1990). A personal view. In B. Laurel (Ed.), *The art of human–computer interface design* (pp. 191–207). Reading, MA: Addison-Wesley.

Lakoff, G. (1987). *Women, fire and dangerous things: What categories reveal about the mind.* Chicago: Chicago University Press.

Lakoff, G. (1988). Cognitive semantics. In U. Eco, M. Santambrogio, & P. Violi (Eds.), *Meaning and mental representations* (pp. 119–154). Bloomington: Indiana University Press.

Lakoff, G. (1993). The contemporary theory of metaphor. In A. Ortony (Ed.), *Metaphor and thought* (2nd ed., pp. 202–251). Cambridge, UK: Cambridge University Press.

Lakoff, G (1996). *Moral politics: What conservatives know that liberals don't.* Chicago: University of Chicago Press.

Lakoff, G., & Johnson, M. (1980). *Metaphors we live by.* Chicago: University of Chicago Press.

Langacker, R. (1987). *Foundations of cognitive grammar: Theoretical prerequisites* (Vol. 1). Stanford, CA: Stanford University Press.

Larman, C. (1997). *Applying UML and patterns: An introduction to object-oriented analysis and design.* Englewood Cliffs, NJ: Prentice Hall.

Laurel, B. (1991). *Computers as theatre* (5th printing, 1997). Reading, MA: Addison-Wesley.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation.* Cambridge, England: Cambridge University Press.

Lefebvre, H. (1991). *The production of space.* Oxford, England: Blackwell.

Lim, K. Y., & Long, J. (1994). *The MUSE method for usability engineering.* Cambridge, UK: Cambridge University Press.

Maclean, A., Belotti, V. M. E., Young, R. M., & Moran, T. P. (1991). Reaching through analogy: A design rationale perspective on roles of analogy. *Proceedings of CHI'91 Conference on Human Factors in Computer Systems,* 107–172. New Orleans, LA: Addison-Wesley.

Madsen, K. H. (1994). A guide to metaphorical design. *Communications of the ACM, 12,* 57–62.

Maes, P. (1997). Pattie Maes on software agents: Humanizing the global computer. *IEEE Internet Computing, 4,* 10–19.

Maglio, P., & Matlock, T. (1998). Metaphors we surf the Web by. *Navigation in information space. Proceedings of the Workshop held March 1998, Sweden* (SICS Technical Report T98:02), Stockholm, Sweden.

Maglio, P., & Matlock, T. (1999). The conceptual structure of information space. In A. Munro, K. Höök, & D. R. Benyon (Eds.), *Footprints in the snow: Personal and social navigation of information space.* London: Springer.

Manzoni, J. F., & Barsoux, J. L. (1998). The set-up-to-fail syndrome. *Harvard Business Review, 2,* 101–113.

Moran, T. (1981). Command language grammar: A representation for the user interface of interactive computer systems. *International Journal of Man–Machine Studies, 1,* 5–30.

Nardi, B. (Ed.). (1996). *Context and consciousness. Activity theory and human–computer interaction.* Cambridge, MA: MIT Press.

Nardi, B. (1997). *HCI'97 keynote address.* Keynote address given at the BCS HCI'97 conference, Bristol, England.

Norman, D. (1998). *The invisible computer: Why good products can fail, the personal computer is so complex, and information appliances are the solution.* Cambridge, MA: MIT Press.

Norman, D., & Draper, S. (1986). *User-centered systems design.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Olle, T. W., Sol, H. G., & Tully, C. J. (Eds.). (1983). *Comparative review of information systems design methodologies: A feature analysis. Proceedings IFIP WG8.1 Working Conference.* Amsterdam: North-Holland.

Olle, T. W., Sol, H. G., & Verrijn-Stuart, A. A. (Eds.). (1982). *Comparative review of information systems design methodologies: A feature analysis. Proceedings IFIP WG8.1 Working Conference.* Amsterdam: North-Holland.

Parsons, J., & Wand, Y. (1997). Choosing classes in conceptual modeling. *Communications of the ACM, 6,* 63–69.

Peer Commentary: Target paper—Conception of the cognitive engineering design problem and commentaries. (1998). *Ergonomics, 41*(2), 126–179.

Presmeg, N. (1997). Reasoning with metaphors and metonymies in mathematical learning. In L. English (Ed.), *Mathematical reasoning: Analogies, metaphors and images* (pp. 267–279). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Rasmussen, J. (1987). Cognitive engineering. *Human–Computer Interaction—INTERACT '87,* xxv–xxx. Amsterdam: North Holland.

Sachs, P. (1995). Transforming work: Collaboration, learning and design. *Communications of the ACM, 9,* 36–44.

Sfard, A. (1997). Commentary: On metaphorical roots of conceptual growth. In L. English (Ed.), *Mathematical reasoning: Analogies, metaphors and images* (pp. 339–371). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Suchman, L. (1995). Making work visible. *Communications of the ACM, 9,* 56–64.

Turner, M. (1996). *The literary mind.* Oxford, UK: Oxford University Press.

Waterworth, J. A. (1996). A pattern of islands: Exploring public information space in a private vehicle. In P. Brusilovsky, P. Kommers, & N. Streitz (Eds.), *Multimedia, hypermedia and virtual reality. Springer Verlag Lecture Notes in Computer Science, 1996.* Berlin, Germany: Springer-Verlag.

Waterworth, J. A. (1998). Experiential design of shared information spaces. *Navigation in Information Space. Proceedings of the Workshop Held March 1998, Sweden* (SICS Technical Report T98:02), Stockholm, Sweden.

Waterworth, J. (1999). Spaces, places, landscapes and views: Experiential design of shared information spaces. In A. Munro, K. Höök, & D. R. Benyon (Eds.), *Footprints in the snow: Personal and social navigation of information space*. London: Springer.

Winograd, T., & Flores, F. (1986). *Understanding computers and cognition: A new foundation for design*. Norwood, NJ: Ablex.

Wood, D. (1992). *The power of maps*. London: Routledge.