

Questioning Representations

Mike Robinson and Liam Bannon

Centre for Innovation & Cooperative Technology, University of Amsterdam, NL

Abstract. The role of models in the design of computer systems to support interpersonal and cooperative work is examined. It is argued that the current generation of models over-emphasise determinism at the expense of interpretation in the work process. It is further argued that there are many cases in which designs pass between many different professional groups (office workers, managers, analysts, designers, programmers). Each of these groups has its own worldview and specialised language, and hence they are termed "semantic communities". When designs pass between semantic communities, something is lost and something is gained -- but the objects on which each community works are not commensurable. The distinct objects of work (office problems, analyses, designs, programs) do not map onto each other, and cannot be mutually tested using simple true/false criteria. This is termed a problem of "ontological drift", and arises whenever several distinct semantic communities work on the "same" system. It is suggested that the disparity so often observed between design expectations and the ways systems are actually used is therefore quite normal. Current efforts are directed at eliminating the disparity. We suggest that a more fruitful approach might be to accept that the final determination of a system rests with the users. In the long run this might give rise to different types of design principles than those used at the moment. In the short run, even the consciousness of this perspective could make significant differences to design dialogues and attitudes to "users".

Introduction

The use of abstractions and modeling is ubiquitous in the development of computer-based systems for office work. A variety of methodologies have been developed to advise on how to perform such activities and ensure some veridicality between the model and actual work practices¹ -- with mixed results, as witnessed by

¹ Including at least the following: SADT (Ross & Schoman, 1977); Structured Analysis (Yourdon,

the results of evaluations of such systems in the workplace². While some simply argue for more powerful representational forms, and more resources to be provided to the systems analysts and designers, there has been a growing awareness that the problem is not simply one of forms or resources, but more fundamentally, of an inappropriate concept of what is "captured" in the model. Rather than viewing the models embedded in designs as implementable accounts of work processes, they should be conceived as heuristic devices that provide resources on particular occasions for particular groups of people.³

Our concerns developed while investigating the recent spate of design models that have been proposed for modelling group communication in the CSCW arena. Our investigation has three strands:

- a number of critical points on the theoretical grounding and practical use of models of people and work in the design of systems and applications to support cooperative and, more generally, interpersonal work processes;
- a developing account of the analysis, design, implementation, and use process in CSCW (and, more generally, computer support for interpersonal work).
- application of the critique to clarify the design process, analyse current applications, and identify some ways of improving CSCW system design.

This paper will concentrate on the second strand: an account of the design-to-use cycle. This is intended to capture some of the problematics and difficulties of design. It is *not* intended to be a full critique of current design models, or to provide pointers for improvement. The latter will be the subject of further papers.

Some Comments on Models and Modeling

Our goal is to develop an analytic case against an objective reality that can be usefully "captured" in a model and subsequently used as a sufficient basis on which to develop a computerized system. Our critique relies on the centrality of *interpretation* in the conduct of work⁴, and also on the fact that the development of computer-based applications requires the collaboration or involvement of a variety of distinct communities -- workers with different skills, analysts, developers, programmers, etc. This necessary heterogeneity poses a number of problems which cannot be removed simply by ensuring good "communication" between the differing groups. The issue is more fundamental, arising out of the different practices of the groups and the essential incommensurability of their world views and language.

1982); OPSL (Zisman, 1977); ICN (Ellis et al, 1980); OAM (Hammer & Sirbu, 1980); and JSD (Jackson, 1983).

² See reviews by Auramaki et al (ms) and by Schmidt (1990b).

³ in the sense that Suchman (1987) sees plans as resources for action

⁴ See, for instance, Berger & Luckmann, 1967, Latour & Woolgar, 1979, Wynn, 1979, Suchman & Wynn, 1984, Gerson & Star, 1986, Suchman, 1987.

A distinction between the nature of *description* and that of *interpretation* is relevant. Here, we wish to focus on practical issues, without detailing the philosophical discourses on realism vs. constructivism. A view of modelling that sees it as merely description of an accepted reality, or an abstraction of it, can lead to serious difficulties. Accepting that there is a sense in which all description is really interpretation (see Geertz, 1973) causes us to be more careful in ascribing veridicality to our models. Models are then seen as interpretations, as constructions, which for some purposes, under certain conditions, used by certain people, in certain situations may be found useful, not true or false. We thus see the modelling process as one of reframing rather than describing or abstracting.

We will now look at some examples with these distinctions in mind. For purposes of illustration, we will use activity/role models from some recent CSCW work. We intend to do a more thorough investigation of a comprehensive set of models in a later paper. As our example here we will refer to the specific group communication models cited in Hennessy, Benford and Bowers (1989). The intent of these European projects (COSMOS, AMIGO MHS+, AMIGO Advanced, and MacAll II) was to create "abstract models of groups communication" and to address "critical limitations" in existing services. We hope that the similarity to other CSCW models will be apparent, and that the outline of our general doubts can be gleaned from this limited case. Our specific concern was the way the concepts of "role" and "activity" are defined and used. Each time the term "role" appears in the examples below, one may ask whether it is intended as a description, as an interpretation, or as an element in a self-contained design dialogue that is independent of any office reality.

Example 1: Notes on COSMOS

Actions -- These are divided into exchanges, which are elementary communicative acts involving at least two participant roles and usually one object; and encapsulated actions (EA's) which do not involve any exchange (eg. creating objects)

Rules -- these define the conditions under which actions can occur"

Roles -- these are agents who initiate actions. A role may be played by an individual user, a collective, or an automated process"

Conditions -- these are expressions, resolving to true or false, which define the context in which rules are triggered."

Example 2: Notes on MacAll

"During the performance of an activity, people are assigned to roles, and they follow the rules associated with those roles. Messages are created, and their rules direct their transfer between workspaces. When a message arrives at a workspace, the workspace notifies the appropriate role instances that a new message has arrived, and one of the role instances processes the message according to its rules."

Example 3: Notes on Amigo Advanced

"Roles -- Within each activity, a set of role classes is defined. Each member of the group must be associated with at least one role that is an instance of one of these classes. The class definitions include the functions that instances of this class of roles may perform or request to be performed, and the message object types that can be sent or received."

"Rules -- These are used to define how each role is expected to behave during the execution of an activity."

In these examples there is a strong logico-mechanistic, or deterministic picture, consistent with the goal of machine implementation. Complementing this, in many CSCW papers, there is an absence of reference to specific office situations, or to the many sociological and anthropological studies from which the central notion of "role" is derived. This provided us with a catalyst for an initial set of comments.

- * There is a constant temptation for designers to confuse the models with an underlying reality.
- * The models impose an ordering on people and or events, often unilaterally.
- * The models are difficult for "users" to understand and thus
- * preclude people from appropriating and re-working the model in situations of use.
- * The models do not define their basic concepts adequately.
- * Emphasis appears to be on model form and elegance over actual coverage and practicality or usefulness.
- * Emphasis is on "determinism" at the expense of "interpretation" in work processes.
- * The models embody an inappropriate correspondence theory of truth, and thus
- * make the untenable assumption of a specifiable, one-to-one, decontextualized relationship between an instruction and the action that satisfies it.

Our first attempt to understand the design process, and our notes on role and activity models, resulted in a simple diagram shown in Figure 1.

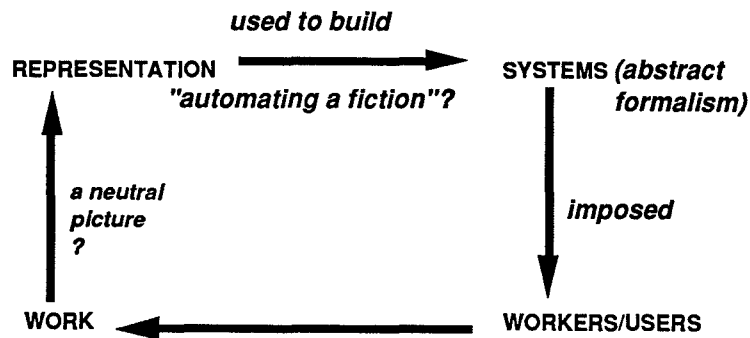


Figure 1: A sketch of some steps in the design-use cycle.

Here we attempt to summarise our understanding of a process that moves from a situation in the world to a representation, then to an implementation of the representation in a computer-based system, and finally to the re-entry of the computerised representation into the original workplace.

This understanding links several quite distinct perspectives. The language of work is abstracted in a language of representation, useful to analysts. This is transformed again into an abstract formalism, chosen for its usefulness to the system implementers. The resulting system is then imposed on workers/users, taking a critical perspective, and changes the nature of the work that the representation was built on. This is a cycle that has clear potential for catastrophic change via a positive feedback loop.

Apart from this obvious danger, Figure 1 suggests to us that there are aspects that need further exploration. The role of interpretation needs a larger place. The different groups involved in the design process need to be clarified. Following from this, the type of interpretation made by each of these groups also needs exploration.

Comments on Interpretations & Communities

Interpretations

Here we would like to suggest that the concept of a dialectical movement between an *object of interpretation* and *interpretation itself* can assist in understanding the problems and may be useful in designing applications. There are two central points. The first is that *all work involves interpretation*. This is often easiest to see when a misinterpretation happens. For instance:

"On being informed by her postmaster that she need no longer include "R.F.D.2" in her address, a Westport matron we know informed Bonwit Teller, among others, of the fact. Her next bill from Bonwit was addressed to:

Mrs. Hillary Jones

Eliminate R.F.D. 2

Westport, Conn." (Goffman;1974)

Even success in carrying out supposedly "low level" clerical tasks can be seen as having almost miraculous interpretive qualities. For example, Goffman (ibid) observes how infrequently secretaries, when taking shorthand, record as part of the text words that were meant as comment on it, especially given the subtlety of the cues which differentiate on-record and off-record streams.

Goffman is suggesting a different conception of role. A "role", for instance "secretary", is a way of stating what someone is doing with respect to other people in the same social milieu. When such as label becomes useful in the work context, it slides (often imperceptibly) into being seen as a description. This does not raise

problems when it is used *in the context* in which it arose. A problem does arise when the "role" in question does not originate with the group in question, and ceases to be the property (or part of the ontology) of the original actors (see the earlier CSCW examples). Making sense of each other and the work situation is not given *a priori*. This is admirably caught in the following quote from McDermott, Gospodinoff, & Aron (1978).

"In addition to sharing knowledge about each other, and whatever it is they are doing together, actors *struggle* to make sense of each other and *do work* to help generate the kinds of recognisable contexts for common sense to be achieved from one moment to the next. As Garfinkel has pointed out often, the problem facing people in interaction is never simply one of shared knowledge or overlapping interpretive grids. No matter how much people know in common, they must still work at constructing the environments that their mutual knowledge leads them to expect, and any relaxation of this effort can have disastrous consequences. People never know exactly how to make sense of each other. "

This introduces the second point: *knowing a person's "role" is rarely sufficient in order to understand what is happening in work interactions*. The actual interaction is achieved moment by moment, with respect to a local context, which cannot be determined in advance. Reducing the complexities of such interactions and ignoring the local negotiations required to perform the work leaves one with a gloss on the work process. This may be adequate at one level for certain descriptive purposes, but becomes positively disruptive if it is viewed as adequate to support the ongoing work process as the framework of a computer system. In such cases the actual way work is accomplished is not supported through the system, leading to ways of working around the system that have been documented by Gasser (1986).

Communities

The creation of computer artefacts almost invariably involves cooperation that crosses group, professional, and subcultural boundaries. The difficulties of working in situations where several groups have different practices, traditions, and working objectives are well known. Different groups, professions, and subcultures embody different perspectives. They communicate in different "jargon". Much of this cannot be translated in a satisfactory way into terms used by other groups, since it reflects a different way of acting in the world (a different ontology and epistemology). Distinct groups of this sort will be referred to as semantic communities.⁵

Recent efforts within the systems development process to emphasize the importance of good communicative practices between these differing semantic communities attest to the difficulties that are experienced. The problem is not resolved by promoting the necessity of open communication -- since this assumes the different

⁵ This is a term coined to describe a distinction of importance, but the term itself should be regarded as tentative. It has affinities to other concepts such as "communities of practice" or the feeling of "we".

groups can be framed in a single semantic world. The meaning of terms is not transparent across groups, for example:

"...each functional department has its own set of meanings for key terms....Key terms such as *part*, *project*, *subassembly*, *tolerance* are understood differently in different parts of the company" (Savage,1987, cited in Schmidt,1990a)

In such a situation the task of the systems analyst is not simply "getting it right", as there is no clear "right" answer. As Gerson and Star (1986) note:

"No representation is either complete or permanent. Rather any description is a snapshot of historical processes in which differing viewpoints, local contingencies and multiple interests have been temporarily reconciled."

In order to understand how representations (in particular the idea of "role" discussed in the CSCW designs earlier) come to be reified⁶, the concepts of *interpretation* and *semantic community* have been introduced. The interplay of these two concepts leads us to a third important concept that will be discussed next, which we shall call *ontological drift*.

Ontological drift

This is the shift in meaning that can occur when knowledge artefacts (maps, designs, models) move between semantic communities. The process starts when the *object of an interpretation* and the *interpretation itself* change places. This "flipover" (Robinson, 1990) has been noted by several authors. In the activity of authoring:

"Talk about dividing the labor of writing is likely to include plans for the paper (Kraut et al., 1987). Writers, however, do not "execute" plans in the same sense that programs do. Instead, writers use a plan as a resource in deciding what to do while they are writing (cf. Agre & Chapman, 1989). Often, the partially completed product plays an important role in the process: *the partially completed product becomes part of the task environment and constrains the subsequent course of the design* (Flower & Hayes, 1981; Kaufer et al., 1986)." Neuwirth et al. (1990) [our emphasis]

Here it is clear that the paper is, at first, an interpretation of a plan. The flipover happens when parts of the paper that are written become the object that the authors interpret in the subsequent parts of the paper.

A similar observation is made by in a discussion of how groups work with symbols they create on the whiteboard they can all see in Xerox PARC's COLAB:

".... whiteboard items hold a dual status as elements *in* the conversation and elements that may be conversed *about*." (Tatar et al. , 1991)

Such flipovers in the process of writing (and discussion) are often useful and creative, reflecting developing moments in the understanding of a subject matter on the part of the authors. Similar flipovers in the process of software creation are often positively dangerous. This is because the process almost invariably happens *between*, rather than *within*, semantic communities (office workers, managers, ana-

⁶ ie. Interpretation and self-contained dialogue replaced by simple "description".

lysts, designers, programmers, customers). In the process of writing, the authors can always recapture their original intentions in their plan by throwing away the draft text that is acting as a new constraint on them. However, when the divide between communities is crossed, such a recovery is no longer possible.⁷ The interpretation of one community becomes an object for interpretation by another -- but the original object of interpretation is lost in transit. There is nothing to refer back to. The original interpretation has been established as part of a different "reality" (object of interpretation) in the new group.⁸

Figure 2⁹ illustrates the problem of software development in a humorous but instructive fashion. Successive elaborations (interpretations) usually occur at an ever increasing distance from the basic need and use situation. When a feasibility study is passed to analysts, *it is convenient to assume that it is a veridical map of the work process*. The cartoon illustrates that a specification which respects the entities and relationships in the analysis is naturally assumed to apply to the work situation. We and the cartoon question this. In the end there may be no relationship between the ontology and epistemology of the artefact and that of its intended use situation. When such a situation arises, the people doing the work may respond in a variety of ways. For instance, Mackay (1990) in a field study notes that "a number [of users] spent a significant amount of time retrofitting new "improved" software to be like the old familiar software. Some avoided using the software altogether." Less commonly, but interestingly, recipients may invent their own uses that are outside the original design intentions.

⁷ We concentrate here on an exploration of what happens when "models" move between groups -- the discontinuities of meaning based on discontinuous realities termed ontological drift. None of this should be taken to imply that there is a stable "reality" within groups and communities. Here it is a common experience that meanings, emphases, and significances are mobile and negotiable. But this is a drift of an ontology, not a drift between ontologies. The emergence of new readings within a community may even exacerbate the gap between communities, making one community even more opaque to another.

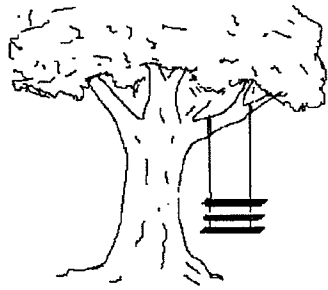
⁸ *A note on the epistemology of ontological drift:*

A major point here is that we should acknowledge our own perspective in talking about different semantic communities and ontological drift.... where are we speaking from? who *sees* ontological drift? are we asserting some vantage point on the moon where all human problems can be seen objectively? The answer is that the ideas are inferred from problems we have seen when one group attempts to utilize the terms, distinctions, or artifacts of another group.... From particular cases, which, once spotted, can often be resolved, we infer a general situation that is much more intractable.

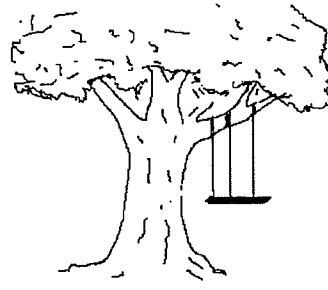
The status of the inferred construct is that of a fairytale. It would be naive in the extreme to believe on the one hand that there is a "warp" in the transit of representations between groups that precludes the applicability of a correspondence theory of truth -- and yet to claim that the construct that denotes this inherently mysterious process can itself be testable or measurable.

Ontological drift, like "phlogiston" or "paradigm", is a structuring metaphor that may play a useful role in design dialogues. It is not, to use one of Gregory Bateson's metaphors, an attempt to count the number of bats in a Rorschach inkblot test. The question is not whether there *are* "ontological drifts", but whether positing the concept gives a new or richer handle to understand the problem of mismatch between user needs and designed systems.... ie. is it "useful" as distinct from "true"?

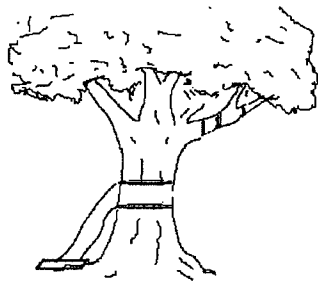
⁹ adapted from a worn and much copied cartoon whose origins are lost to us in the mists of time.



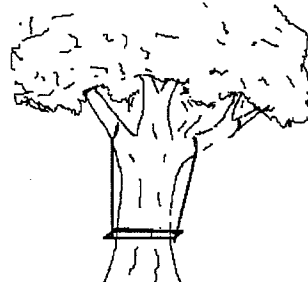
what the user asked for



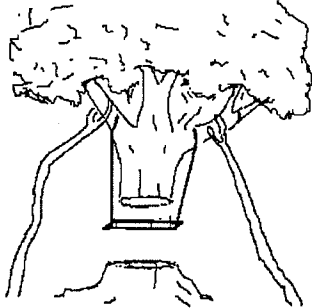
what the feasibility study said ...



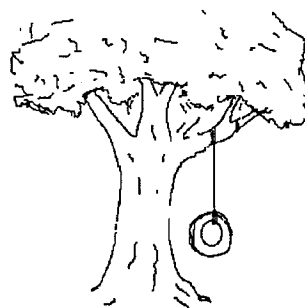
what the system analyst designed



what the programmer did



what was installed at the user's site.....



what the user really wanted !

Figure 2: Some products of ontological drift.....

One of our initial comments on design models was that: "there is a constant temptation for designers to confuse the models with an underlying reality." The comment now appears too simplistic. There are *several* "realities" to be taken into account. The objects of the work of each community are not usually related to the (commonsense) underlying office reality at all. Each deals with an object that may be several "flipovers" from the work process.

First there are interpretations of a work situation produced by those who do (or who are responsible for) the work. These interpretations are not the work itself, or

even a description of it. They are provisional hypotheses produced in a particular dialogical context. While they remain in their original context, connected to the hands-on experience of getting the work out by the end of the day, they are constantly subject to reconstruction.

"Oh, did I say that? Well really what happens is"

When the analysts take away their notes on these interpretations, and start to explore the inner connections and structures, the interpretations have been reified. They have to be "fixed" or it would not be possible to work with them. But this does not make them into veridical descriptions. The first "flipover" has happened. When the notes cross the boundary between the two semantic communities of office workers and systems analysts, the living connection to the original work process is lost. The notes become the object of interpretation in a different work process -- that of producing analyses.

Another "flipover" happens when the finished analysis is passed onto the design community. The analysis loses its status of provisional interpretation of notes, memories, doubts, and intuitions, and becomes a "fixed" object on which the design work can be performed. A further discontinuity occurs when the analysts pass their model on to the programmers.....

Our account causes us to revise another of the initial comments on design models. We said: "the models do not define their basic concepts adequately." This creates an illusion that design may be improved by better definitions. There is an assumption that a definition can "drop through" all the semantic communities to a common core. This may simply not be possible. Definitions may be quite adequate in their own communities, but do not translate or transfer between communities. The question is not one of adequate definitions in each domain, but of how these definitions might relate to each other. In other words, it may be a dangerous illusion to believe that models (especially those that have crossed semantic boundaries) can "correspond" with "reality".¹⁰ We are squarely in the area of Wittgenstein's (1963) language games. The question is not how to verify propositions. The essential problem is how to *integrate activities that are taking place on different ontological foundations*.¹¹ The question is one of competence rather than truth.

Some Further Issues and Questions

The analysis sketched above has lead us into a number of interesting areas, and we will mention a couple that we are currently pursuing. One is a review of the various design methods that have been developed; the other is a reflection on how the is-

¹⁰ related to our earlier comment that "the models embody an inappropriate correspondence theory of truth".

¹¹ "If a lion could talk, we would not understand him" (Wittgenstein, 1963: 223) nicely illustrates the extremes of different ontological foundations. In practice we would expect to find complex relations of intersection and differentiation among the multiple ontologies in question.

sues raised here might be better taken into account in the design process itself. Let us take these in turn.

Design Methods

It appears possible that the wide range of design methods can be understood as responses to different perceptions of the underlying phenomenon of ontological drift. The following paragraphs are initial pointers to work that might be done in this area in the future, and should not be taken as arguments or analyses.

* "Don't see the problem....."

The assertion, implicitly or explicitly, of a correspondence theory of truth in complex system design can be taken as a symptom that the issue of ontological drift is simply not recognised. It would be interesting to know whether, and under what circumstances, *any* system based on this assumption has been accepted as useful!

* "Make unifying dictionaries....."

It is most uncommon for epistemological and ontological premises in the design process to be examined, but the problem can often be sensed. A symptom of this is when designers attempt to provide a set of tight definitions that are assumed to transcend the differences between semantic communities. This is frequently a hopeless task, as noted earlier, since it ignores the interpretive processes illustrated in the Figures. A definition may be appropriate and work well in its own ontological domain -- but will not "drop through" onto the different ontological domains. Nevertheless, definitional structures proposed by methods that use successive formalisms (eg. Jackson's (1983) Structured Design) may work well in situations where ontological drift is low -- but this is not an assumption that can be generally made, especially where CSCW applications are concerned.

* "Bring the users closer to the designers...."

The strongest sense of the dangers of ontological drift can probably be found among researchers and developers within the broad church known as the Scandinavian Tradition. Here we find a serious diversity of attempts to "involve users", and thus, in our terms, bring about some closure on the process of ontological drift. For instance, Pelle Ehn (1988:116) says:

"If designers and users share the same form of life it should be possible to overcome the gap between the different language-games. It should at least in principle be possible to develop the practice of design so that there is enough family resemblance between a specific language-game of design and the language-games the design of the computer artefact is intervening in. A mediation should be possible."

Typically, work such as this engages in a struggle to recognise and legitimate the semantic diversity of different work communities and to link "end-users" (and hence the living work situation) into all phases of the design cycle. Some of the

practical problems experienced in this endeavour relate to the difficulty of maintaining *both* a recognition of distinct semantic communities *and* of evolving systems based on user correction mechanisms.

* Bring the designers closer to the use situation

Another response to the problem is the endeavour to make designers more appreciative of the nature of the work they are supporting with information systems. Such an approach, with its problems and prospects, is explicitly considered by Curtis et al. (1988). Their paper is a critique, and response to the lip service commonly paid to the idea that analysts and designers *should* understand the business they are currently working on. Recognizing these difficulties, and providing resources for more serious efforts to bring them closer to the work process would be an improvement -- but it is not clear that such efforts would avoid the dilemmas mentioned in the previous paragraph.

* Structural coupling

If it is accepted that the mix of problems concerning interpretation, different semantic communities, and the limited applicability of classical scientific verification procedures do raise major difficulties, then the complex conceptual apparatus elucidated by Maturana (1988) might be useful. In particular his idea of "structural coupling" might be used in design as it explicitly pays attention to different ontologies in action, and of interchanges between them. To date, the one major attempt to embed this framework in systems (the CO-ORDINATOR -- Winograd & Flores, 1986) has been the subject of much dispute and criticism (eg. Grantham & Carasik, 1988; Robinson, 1991).

Design Research

Switching now from design methods to design research issues, the process of ontological drift leads to an expectation of discontinuity, surprise, and unanticipated use of designed computer systems. While such experiences are endemic, they have rarely been subject to close examination or welcomed. Our approach argues for more sympathetic attention to be paid to these important phenomena. This is already beginning within CSCW -- for instance, Mackay's field study (1990) on the use of rules in prototypes of Information Lens; Tatar et al. (1991) on conversational disruptions in COLAB; Bullen & Bennett (1990) on the highly selective appropriation of features from a welter of office coordination tools. Faced with the "creative misuse" of designed artifacts, we are exploring what it would mean to undertake design work with a more paradoxical and self-critical expectation of "unanticipated use".

One particularly exciting area for further research lies in the exploration of "turn taking" and its support in CSCW systems. This complex yet universal aspect of human interaction, resting as it does on a mix of procedures, conventions and

moral orders offers an interesting nexus for the concerns of many disciplinary communities within CSCW. Further, the gaps between current protocols implemented in meeting support systems, diverse workplace social conventions, and ethnomethodological analyses of conversational flux seem *prima facie* examples of ontological discontinuity. This offers the possibility of examining the clash between embedded and enacted conventions, and the differences between premeditated support for work, and the facilitation of unanticipated use.

Conclusion

We have attempted to develop an analytic case against the implicit belief in a "reality" that can be usefully "captured" in a model, and used as a sufficient basis for the development of computerized systems to support interpersonal work processes. The process of interpretation and reinterpretation is central, not just to the work practices of offices, but to the work of analysis, design, and implementation. The passing of work between different semantic communities, each with their own ontologies, epistemologies, and conventions; each interpreting and recontextualising the products of other communities, generates a phenomenon we term ontological drift. There is no simple correspondence or mapping between analyses, designs, implementations, and the flux of work into which systems are placed. Better definitions have a role within semantic communities, but do not address the issue of the discontinuities that arise when models cross semantic boundaries. Here the problematic is one of integrating activities that take place on different ontological foundations.

It is suggested that closure is imposed on the process of ontological drift by those who end up using (or not using) the system. It is inherent in the process of ontological drift that the type of closure achieved cannot be anticipated with any certainty. If this is not taken into account in the design dialogues, the final outcome may come as a nasty surprise to all concerned. If it is taken into account, even in broad terms, expectations of the nature and results of the design process may change considerably.

Acknowledgements

The Authors would like to thank the ECSCW-91 reviewers, colleagues in the EC COTECH Working Group 4, E. Rønby Pedersen and J. Grudin for comments.

References

- Agre, P. & Chapman, D. (1989) What are plans for? MIT AI Memo 1050a, MIT, Oct. 1989.
- Auramaki, E., Hirschheim, R. & Lyytinen, K. (draft ms.) Modelling Offices in SAMPO: A Comparison and Evaluation with OSSAD and ICN.
- Berger, P. & Luckmann, T. (1967). *The Social Construction of Reality*. Harmondsworth, UK: Penguin.
- Bullen, C. & Bennett, J. (1990) Learning from user experience with groupware. In *Proceedings of Conference on Computer-Supported Cooperative Work, CSCW '90*, October, Los Angeles, CA, pp 291-302.
- Curtis, B., Krasner, H., & Iscoe, N. (1988) A Field Study of the Software Design Process for Large Systems. *Communications of the ACM*, 31 (11), 1268 - 1287.
- Ehn, P. (1988) *Work-Oriented Design of Computer Artifacts*. Stockholm: Arbetslivscentrum.
- Ellis, C.A., Gibbons, R., & Morris, P. (1980) Office Streamlining. In N. Naffah (ed.) *Integrated Office Systems - Burotics*. Amsterdam: North Holland.
- Fikes, R.E. & Henderson, D.A. (1980) On supporting the use of procedures in office work. In *Proceedings of the 1st Annual AAAI Conference*, Stanford University, CA, August, 1980, 202-207.
- Flower, L. & Hayes, J. (1981) The pregnant pause: An enquiry into the nature of planning. *Research in the Teaching of English*, 15: 229-243, Oct. 1981.
- Gasser, L. (1986) The integration of computing and routine work. *ACM Transactions on Office Information Systems*, 4,3, 205-225.
- Geertz, C. (1973) *Interpretation of Cultures*. New York: Basic Books.
- Gerson, E. M. and Star, S. L. (1986) Analyzing due process in the workplace. *ACM Transactions on Office Information Systems*, 4, 3 (July 1986), pp. 257-270.
- Goffman, E. (1959) *The presentation of self in everyday life*. New York: Anchor.
- Goffman, E. (1974) *Frame Analysis*. New York : Harper Colophon.
- Grantham, C.E. & Carasik, R.F. (1988) *The Phenomenology of Computer Supported Cooperative Work*. Interpersonal Software, Berkeley, CA.
- Hammer, M. & Sirbu, M. (1980) What is Office Automation? In *Proc. First Office Automation Conference*, Atlanta, Georgia. March.
- Henessey, P. , Benford, S. & Bowers, J. (1989) Modelling Group Communication Structures: Analysing four European projects. In *Proceedings of the First European CSCW '89 Conference, Part Two*, Gatwick, UK, pp. 406-420.
- Hewitt, C. (1986) Offices are open systems. *ACM Transactions on Office Information Systems*, 4, 3, 271-287.
- Jackson, M.A. (1983) *System Development*. Englewood Cliffs: Prentice-Hall.
- Johansen, R. (1988) *Groupware: Computer Support for Business Teams*. New York and London: The Free Press.
- Johnson, B., Weaver, G., Olson, M., & Dunham, R. (1988) Using a computer-based tool to support collaboration: A field experiment. In *Proceedings of Conference on Computer-Supported Cooperative Work, CSCW '86*, Austin, Texas, pp. 343-352.
- Latour, B. & Woolgar, S. (1979) *Laboratory Life: The Social Construction of Scientific Facts*. Beverley Hills: Sage Publications.

- Mackay, W. (1990) Users and Customizable Software: A Co-Adaptive Phenomenon. Doctoral dissertation, Sloan School of Management, MIT.
- Mackay, W.E., Malone, T., Crowston, K., Rao, R., Rosenblitt, D. & Card, S. (1989) How do experienced Information Lens users use rules? ACM CHI '89 Proceedings, Austin, Texas, 211-216.
- Malone, T. W., Grant, K. R., Lai, K.-Y., Rao, R., and Rosenblitt, D. (1987) Semistructured messages are surprisingly useful for computer-supported coordination. ACM Transactions on Office Information Systems. 5, 2 (April 1987), pp. 115-131.
- Malone, T. W., Grant, K. R., Turbak, K. R., Brobst, F. A., and Cohen, M. D. (1987) Intelligent information sharing systems. Communications of the ACM, 30, 390-402.
- Maturana, H.R. (1988) Ontology for Observing: The Biological Foundations of Self Consciousness and The Physical Domain of Existence. In Proc. Conference on Texts in Cybernetic Theory, American Society for Cybernetics, Felton, CA. Oct. 18-23.
- McDermott, R. , Gospodinoff, K. & Aron, R. (1978) Criteria for an ethnographically adequate description of concerted activities and their contexts. Semiotica 24, 3, 4, 245-275.
- Neuwirth, C.M., Kaufer, D.S., Chandhok, R. & Morris, J.H. (1990) Issues in the design of computer support for co-authoring and commenting. In Proceedings of Conference on Computer-Supported Cooperative Work, CSCW'90, Los Angeles, CA pp. 183-195.
- Robinson, M. (1990) Computer Supported Cooperative Work & Informatics for Development. In Proc. INFORMATICA '90. Havana. February.
- Robinson, M. (1991) Double level languages and co-operative working. AI & Society, 5, 34-60.
- Ross, D.T. & Schoman, K.E. (1977) Structured Analysis for Requirements Definition. IEEE Transactions on Software Engineering. Vol. SE3. No.1. January.
- Savage, C.M. (Ed.) (1987) Fifth Generation Management for Fifth Generation Technology (A Round Table Discussion), Society of Manufacturing Engineers, Dearborn, Michigan.
- Schmidt, K. (1990a) Analysis of Cooperative Work: A Conceptual Framework. Risø National Lab Report M-2890.
- Schmidt, K. (1990b) The Procrustes Paradigm. (Working Paper) COTECH WG4 Meeting: CEC Brussels, Nov.
- Suchman, L. & Wynn, E. (1984) Procedures and problems in the office. Office: Technology and People, 2.
- Suchman, L. (1983) Office procedures as practical action. ACM Transactions on Office Information Systems. 1, pp. 320-328.
- Suchman, L. (1987) Plans and Situated Actions. Cambridge: Cambridge University Press.
- Tatar, D. , Foster, G. & Bobrow, D. (1991) Design for conversation: Lessons from Cognoter. International Journal of Man-Machine Studies, 34, 185-209.
- Victor, F. & Sommer, E. (1989) Supporting the design of office procedures in the DOMINO system. In Proceedings of the First European CSCW '89 Conference, Gatwick, UK, pp.148-159.
- Winograd, T. & Flores, F. (1986) Understanding Computers and Cognition. Reading, Mass: Addison-Wesley.
- Wittgenstein, L. (1963) Philosophical Investigations. Oxford, Basil Blackwell.
- Wynn, E. (1979) Office conversation as an information medium. Unpublished Ph.D. dissertation. University of California, Berkeley, CA.
- Yourdon, E. (1982) Managing the System Life Cycle. New York: Yourdon Press.
- Zisman, M.D. (1977) Representation, Specification, and Automation of Office Procedures. PhD dissertation, Dept of Decision Sciences, The Wharton School, Univ. of Pennsylvania, PA.