Towards Integrating Rationalistic and Ecological Design Methods for Interactive Systems

Idris Hsi
Center for Human-Machine Systems Research
School of Industrial and Systems Engineering
Georgia Institute of Technology,
Atlanta, GA 30332-0290, USA
idris@chmsr.isye.gatech.edu

Colin Potts
College of Computing
Georgia Institute of Technology,
Atlanta, GA 30332-0290, USA
potts @cc.gatech.edu

Abstract

Interactive systems design based on rationalistic methods can benefit from the integration of ecological methods that gather information about the user and the task environment. In this paper, we begin to discuss how such an integration of methods can be brought about and what benefits can be derived from it. Using meeting scheduling as an example, we show how workplace data gathering, in the form of user interviews, can significantly alter the specification of a collaborative interactive system. Our discussion of rationalistic methods is restricted to goal refinement approaches, and we discuss a series of design issues that are most pertinent in this approach, specifically: agency and responsibility, obstacle identification, obstacle avoidance and recovery, volume, frequency and repetitiveness issues, generic scenarios and critical incidents, artifact analysis, and deliberate fuzziness. However, we conclude by outlining how ecological methods could be integrated similarly with other rationalistic methods.

Key Words: requirements, ecological methods, meeting scheduling, goal refinement, design

There are at least two paradigms for approaching software design. Many system developers and software engineering researchers adopt a *rationalistic* perspective [Kling, 1980]. According to this perspective, the purpose of a system is to fulfill the goals contained in a requirements specification. Design is a purely technical activity aimed at efficiently implementing the requirements, usually by representing them in a formal language.

A second approach to system design, the *user-centered approach* [Gould and Lewis, 1985] concentrates on the capacities of the people interacting with the system. User-centered methods emphasize the *cognitive model* of the user [Norman, 1988], which can only be understood through studying the environment and context that governs the user's work. Studying the user in the work environment requires a number of *ecological* methods [Flach, 1990] that help the designer obtain empirical, real world information about the user. Ecological methods include task analysis, user interviews, video taped observations, and activity tracking.

A pure rationalistic perspective often fails to account for the nuances of the user's work practices, thereby leading to systems that lack flexibility; whereas a pure ecological perspective, by emphasizing analysis over synthesis, does not cover all the issues that the designer needs to consider, and thus may fail to guide the design process toward a system that meets the user's needs. We believe, therefore, that a synthesis of the two perspectives is essential.

Integrating two fundamentally different perspectives is an ambitious, long-term goal. In this paper, we illustrate how such a synthesis can arise for one rationalistic method (goal refinement) and one ecological method (user interviews). Using meeting scheduling as an example work domain, and a collaborative meeting scheduler as a corresponding interactive system, we show how the analytic tasks of the goal refinement method can be supplemented by user interviews. Drawing on interviews with several office workers, we illustrate where data about meeting scheduling can undermine conclusions obtained from goal refinement.

The rest of this paper is structured as follows: First, we describe goal refinement and the ecological approaches in more detail. Next, under a series of design categories derived from the goal refinement approach, we present excerpts from transcripts of interviews with office workers about how they schedule meetings, showing how data about scheduling activities should be incorporated into the *synthesis* of an interactive meeting scheduler. Finally, we summarize our conclusions and their implications for the synthesis we seek between rationalistic and user-centered perspectives.

Complementary Perspectives

Systems Rationalism

Systems rationalism is enshrined in popular development methods, such as object-oriented analysis and design [Rumbaugh et al., 1992]. We have recently been developing and evaluating a development method that starts with enterprise goals and explicitly refines these goals into system and user operations and constraints [Potts, 1995; cf. Dardenne et al., 1993]. Like structured and object-oriented methods, goal refinement is rationalistic: In developing the approach, we have assumed (simplistically) that the designer is supplied with goals to fulfill, that some decompositions of goals into more concrete subgoals are superior to others, and that goals can be refined into executable goal-achieving and goal-maintaining actions by following systematic guidelines. We believe that emphasizing the achievement and maintenance of goals is inherently superior to modeling system functions (as in structured analysis) or environment or domain concepts (as in object-oriented analysis), because a system's purpose is more likely to be met if explicit attention is paid to goals. However, the precise nature of the method is not important for current discussion: Much of what follows could be translated to more orthodox rationalistic methods, and we return to this claim in the Discussion section.

Refinement of goals into system design solutions is a multistage process consisting of the following design actions:

- Defining and decomposing goals
- Assigning goals to system and environmental actions,
- Providing operational definitions of goal-achieving and goal-maintaining actions,
- Identifying obstacles to goal fulfillment,
- Elaborating subgoals or actions to defend against occurrence of obstacles or mitigate their effects.

For example, one of the subgoals of a collaborative meeting scheduler when scheduling requested meetings is to ensure that all invitees are aware of the request for the meeting. This could be achieved in several ways. The person calling the meeting might inspect an on-line shared calendar maintained by the system and then call each person (perhaps at telephone numbers supplied by the system) to advise them that a meeting is planned. Alternatively, the calendar could act as an agent for the person calling the meeting and would send a predefined message to each invitee automatically. Thus, a single goal could be assigned to the system and its users in many ways (i.e. degrees of automation), and the detailed actions required of the system and its users could be entirely different for different design scenarios. Which solution is best for a community of users depends on their responsibilities and ownership of information in the working environment.

Rationalistic thinking naturally emphasizes expected or normative behavior, and it idealizes and simplifies the likely behavior of users. Exceptions tend to be treated as afterthoughts. But interactive systems, such as meeting schedulers, must cope with many types of system and environment imperfections: Email goes astray; people forget to enter information on calendars; they ignore requests for appointments. We refer to these occurrences as *obstacles* because they block the fulfillment of goals. One of the principal objectives of our goal refinement method, and a distinguishing feature, is the identification of obstacles and the definition of secondary design solutions to avoid or recover from them. Considering obstacles forces the designer to investigate situations in which the working environment does not meet idealized assumptions, and therefore to think about robust and flexible design solutions. Obstacles that occur often in one environment might not arise in another; and those that are most obstructive to goal fulfillment in one environment could be ignored in another. Thus, the identification of obstacles, and the design of practical solutions, depends critically on a knowledge of the work environment for which the system is being designed.

Ecological Methods

Ecological methods provide a structured and scientific process for studying the user and the environment under actual or "real world" conditions. An ecological study may begin with some surveys of the user's environment, and a series of extended observations of the users as they work within that setting. The developer may also conduct interviews with the users to investigate how they perform their task or to discover other interactions that may not have been observed during the environmental survey. The data is sometimes videotaped or recorded, then transcribed to provide a concrete record of these observations.

In usability work, this information can be used to determine how users reason their way through a task and perform it using the tools available to them. By including the environment in the study, the developer may also learn about some of the related social, technical, and ergonomic issues that may aid or hinder the performance of the task [Nielsen, 1993]. Often, decisions and actions executed by the user in the performance of the task may seem irrational or inexplicable until the context is analyzed. Sometimes the behavior is socially mediated, such as an office procedure that explicitly states that a function be carried out a certain way, or artifactually mediated [Norman, 1991] where information may be stored in artifacts that have nothing to do with the system being used. A task analysis of the ecological data can then be used to sketch the user's mental model of the task and to determine whether there are some regularities of behavior that can be automated or supported by a system. By understanding the user's intentions and goals, the designer can derive interface requirements for mediating human-system interactions. The final result is a system that combines functionality with usability, made possible through this early focus on the users.

Integrating Rationalistic and Ecological Methods

What would an integrated approach be like? There are three plausible approaches to integrating the rationalistic and ecological perspectives:

- Start with an existing rationalistic method, and enhance it with ecological methods.
- Start with a collection of ecological methods, and make them more systematic.
- Start afresh, and invent a wholly new approach.

We have adopted the first approach, because the rationalistic goal refinement method that we discuss provides a prescriptive strategy into which empirical data gathering and analysis methods can be plugged. This is also the alternative most likely to give user-centered design a voice in most development organizations. It would be more difficult to start with ecological methods, because they provide less an overall strategy than (1) a motivating user-centered philosophy (which is laudable, but too vague), and (2) a collection of empirical methods (which are essential, but too specific to act as an organizing framework). We reject the third alternative as being impractical.

In this paper, we present the first part of our approach which is to refine each of the major design issues in the goal refinement method with empirical considerations. Because goals and their fulfillment through actions are central to the rationalistic method from which we start, we must consider how ecological methods can explicate the goals, actors, responsibilities, and actions in the task environment. Because obstacle identification and the analysis of strategies for coping with obstacles are so central to the method, we must look at how empirical data can illuminate what obstacles are problematic in the work environment, and how they are dealt with. Because goals and actions affect information and objects, we must suggest ways to identify real-world objects in the task environment.

Rationalistic And Ecological Issues In The Design Of Meeting Schedulers

Meeting scheduling affords a good example of how the nuances of a work environment strongly affect the feasibility of automating a task and how ecological methods can be used in conjunction with a rationalistic design perspective. Meeting schedulers have rapidly become the white rat of the requirements engineering community (e.g. see [Van Lamsweerde et al., 1995]), because meeting scheduling is a problem domain familiar to everyone, yet potentially complex. Among others, Grudin [1989] has used meeting schedulers as a canonical example of collaborative systems that are likely to fail precisely because rationalistic designers underestimate the complexity of the work environment.

The excerpts below are taken from taped interviews with three employees of an academic institution whose work involves the scheduling of meetings and other appointments. One (M.) spends most of her time scheduling meetings and arranging travel. Another (J.) schedules meetings for both a single key individual and for the department. The third (G.) schedules meetings for groups of people as part of much broader responsibilities.

In addition to conducting interviews, which inevitably included some informal workplace observation (see the subsection about artifact analysis), we have also been collecting a corpus of email messages about proposed meetings and visits. However, in this paper, we restrict ourselves to illustrating the potential role to be played by a single ecological method, the workplace interview.

Our discussion of this example is structured according to design issues that arise in the goal refinement method: agency and responsibility, obstacle identification, obstacle avoidance and recovery, volume, frequency and repetitiveness issues, generic scenarios and critical incidents, artifact analysis, and deliberate fuzziness.

Agency and Responsibility

An agent who is responsible for an activity has been given authority to carry it out. In this sense, responsibility is not a measure of accountability but rather a way of defining control. A rationalistic approach might make assumptions about idealized agents that have responsibilities for certain actions and decisions. however, the behavior of the actual participants may be more involved or very different from the abstract interpretation.

I: O.k. Who do you usually schedule meetings for?
J: Our whole center and also Dr. F.'s individual personal meetings.
I: So he'll tell you, "Well we need to schedule a meeting for X?" or "Could you schedule a meeting for...."

> J: Well, lots of times I'm doing it on my own. We want to have a general faculty meeting. I have to schedule a meeting for [this group of] faculty [who] are in ten different locations on campus.

From this dialogue, we learn that the center is spread out across a large area and that the user is responsible for scheduling meetings for an immediate supervisor and a much larger group of people. We can also infer that the user has some knowledge regarding what kinds of meetings take place and possesses the authority to initiate these meetings.

Responsibility for information may also have very little to do with the organization structure. In the scheduling domain, people may guard their own schedules as personal and privileged information. A person may not be permitted to go directly to another's schedule, even a subordinate's, but instead must contact an intermediary.

> M: People cannot come in here and just make an appointment unless they go through me ...It can't be accessible to just everybody in the college because everything in the world would get scheduled and we have to prioritize. We have to maintain some sort of control over that. There has to be some way to block out some people or just have these people have access or whatever. Someone shouldn't be able to pull up his schedule and just schedule a meeting. He would say "I don't need that meeting. They need to see so and so." And sometimes that happens. I would say about a third of the phone calls that I get from people who want to meet Dr. F., he's not really the right person. And I'll redirect them to the right person. That's what I do when I screen. Do they really need to talk to Dr. F. or do they really need to talk to someone else.

M. is responsible for screening incoming messages and phone calls to protect Dr. F's schedule. Some of these requests are redirected to others. A system that encroached on M's discretion for purely rationalistic reasons (for example, allowing users, even senior people, to schedule Dr. F's time directly by seizing slots on his calendar) would not fit the context of use of this organization, causing numerous problems. The most obvious difficulty would be a finished schedule that contained too many unnecessary meetings involving too many people. One of the authors has observed this particular problem at a software development organization that is attempting to use a commercial multi-user scheduler that lets users seize each others' time in this way. Users in that organization have taken to reserving blocks of time for "meetings" with themselves to avoid being disturbed. One user we talked to unwillingly schedules many of her important meetings over lunch, because she knows that she and the people she needs to meet are likely to be available then. This stratagem is likely to be a temporary ploy only as other people start adopting it themselves.

Obstacle Identification

Information about naturally occurring obstacles can help to make the requirements for a system more complete by defining what the system should do or what tasks it should support in the presence of environmental imperfections.

> I: Do they all have computers?
> J: Mostly, they have computers available to them.
> I: O.k.
> J: They don't always have them in their office or for some reason, their network is not maintained as well and they have more problems with their mail being flaky and so they just aren't as tuned into it as we are because we're a little bit more technology based and so that can be a problem because you have to know who those people are and let them know that there's a message there for them to read...

Because email is such an integral part of the way that meetings are scheduled in the academic environment, these people who fail to read their email pose an obstacle to successful meeting scheduling. However, as the excerpt reveals, not all users exhibit the same motivation, inhabit the same environment, or possess the same artifacts. Therefore, a recovery activity other than sending a second message (which would be a rational solution for dealing with an unresponsive participant) is required .

Another element of obstacle identification is prevention. One method for preventing the occurrence of the error in the first place is through automation. By automating a routine task, the likelihood of an error through a slip performed by the user is reduced.

> J: I save all my messages at least for a year or six months or whatever seems to be a reasonable amount of time. That way if I make a mistake, I have them to refer back to.
> I: Like what kind of mistake?
> J: Well, if I checked the wrong box for somebody when I'm going through, doing my spreadsheet, and then somebody sends me an answer and I already have an answer filled in for them, I can refer back to my messages.

The system could automatically verify and record an answer, reducing the need for the user to have to backtrack and determine whether an error had been committed.

Another goal of obstacle identification is to have the informant help the designer close off design paths that might appear promising but which would run into problems in practice.

> J: And that's how I do my scheduling now. I don't usually just pick one time and try to set up a meeting with one time in mind because then you end up having to send back messages multiple times because that one doesn't work, you have to start over again.

Because of their familiarity with the task and their experiences, users tend to have already encountered situations that the developer may attempt to investigate.

Obstacle Avoidance and Amelioration

Identifying obstacles is one thing. Overcoming them with a system design that robustly operationalizes goals is another. There are two generic ways to overcome obstacles: avoid them by pre-planned short-circuiting of the conditions that can lead to them or recover from them or ameliorate their effects by introducing special recovery behaviors. These mechanisms in turn require that we work out four types of system behaviors: those that predict obstacles before they occur, those that avoid the obstacles once predicted, those that detect that an obstacle has occurred, and those that recover from an obstacle. Although automation provides opportunities for shifting the balance from recovery/amelioration to detection/prevention, it is often worthwhile to find out how office workers currently cope with obstacles.

> J: Email sort of helps in that respect because you can send out one message to everybody at the same time. And the only problem there is people who don't read email faithfully, you know, and there are some. <u>So you have to find out who doesn't read their email regularly and make sure that you call them to make sure you say "I sent you an email message about [an appointment], please check your email."</u> or something to get them to read it.

The likelihood that a specific meeting invitee may not read his or her email, a potential obstacle, is predicted in advance. These predictions are derived from the user's experience. We also learn about the obstacle avoidance behavior, which is to send either a replacement message or a short warning message, such as a telephone call, to tell the person to check his or her mailbox.

> I: How do you find this out?
> J: Just by trial and error. <u>People who don't respond to you, eventually you call them up and say "Did you read this email message I sent?" And you find out that they normally don't read it very often and then you have to go back and say I need to find another way to reach these people.</u> I wish everyone read email, it would make this job easier.

This time, the obstacle detection behavior is a "time-out", a long delay before a reply eventually makes the office worker suspicious that the recipient never read the message. The amelioration behavior is again a warning message.

Volume, Frequency, and Repetitiveness

Inquiring about the frequency of task occurrences and task 'volume' (numbers of similar instances) helps the designer to derive nonfunctional requirements, such as response time or availability. These task demands also help the designer to assign development priorities. Events that rarely or never occur could be left to later stages of development, helping to constrain the design.

> M: I only work for Dr. F. So all the scheduling that I do are for meetings that he's calling for one or more people.
> I: Up to how many people?
> M: It can be anywhere from one other person, like say Dr. T. or a group of fifteen or sixteen people or so and they can be all in Georgia Tech or maybe scattered across the country.

> J: So I want to schedule a meeting and I have to contact 35 people in ten different departments to set up a meeting. We try to do this two or three times a quarter. So [I] try and find a time that's best for the majority of the people and that type of thing.

It is important to have some idea of the frequency of obstacle occurrences. Rare obstacles can be ignored if their effects are insignificant. It may be possible to recover from these obstacles creatively or just accept that the enterprise goal cannot always be met.

> J: No. Most of the people that we deal with are computer literate people. They're either in hardware or software positions or they have email. Most of the companies we deal with have email. We have visitors coming in and we set up a lot of schedules for visitors, you know, and we do a lot of email that way but almost everything we do is email. Occasionally there will be a [visitor that will call to set up a meeting] but very seldom. Even with our outside visitors we do email.

One of the standard benefits of automation is that frequently recurring tasks can be packaged into single operations with the system performing (or appearing to perform) the replication of the tasks behind the scene. For example, in scheduling, a replicated scheduling goal for regular appointments can be operationalized as a single command.

> M: Things that are internal like faculty meetings are pretty regular. There are a lot of regular kinds of meetings like faculty meetings. [Points to a highlighted piece of the calendar] He's teaching a course this quarter so I know this time is automatically blocked out. Things like that. Regularly scheduled meetings I go ahead and block a quarter at a time.

It may be important, however, to retain the individuality of the goals so that, for example, one of the regularly scheduled appointment can be rescheduled without affecting the remainder of the series.

Generic Scenarios and Critical Incidents

Generic scenarios are descriptions of typical courses of actions. The purpose behind eliciting generic scenarios is to find out how the enterprise goals are currently operationalized.

> I: How do you go about scheduling a meeting?
> G: <u>I send email to everyone on the committee and ask them what is a good time for them and what is not.. Then I look at everybody's responses. I see if there's a best time based on the majority of the schedules. Then I see if I can get a room for that particular time. Then when I get the room, I send out an email saying this is the time and place that it will be.</u>

Finding this out has at least two purposes: by describing actual behaviors, informants can help identify the underlying goals that the behaviors are supposed to support (by asking "why do you do that?" follow-ups); and the concrete behaviors help identify agency/responsibility issues and nonfunctional requirements (by asking "how often?" follow-ups).

> M: It can be anywhere from one other person, like say Dr. T. or a group of fifteen or sixteen people or so and <u>they can be all in [the institute] or maybe scattered across the country.</u>
> I: Really. How does the across the country work?
> <u>M: Well, I just have to get on the phone and start writing or getting people to give me numbers for various dates and times based on what Dr. F's schedule is.</u>

Critical incidents are narratives describing specific occurrences that were noteworthy. The purpose behind eliciting critical incidents is to help identify causes of successful or unsuccessful task performance.

> M: One meeting I have that I'm in the process of working on right now is a meeting with all the senior faculty in the [college]. This involves ten people, who also have schedules about as busy as he does. So what I do, <u>he and I get together and decides on dates</u>, <u>depending on when he needs the meeting to occur by such and such a deadline and then look at his availability.</u> Then <u>I send some email out to this group of people</u>, these *people have an alias*, and I ask <u>"Okay, what is your availability for these dates."</u> Like *I start out with a grid [shows another piece of paper]*, and they start emailing me back, this is lunch and dinner, are you available for lunch, are you available for dinner on these dates? <u>Then I just kind of wait until I hear back from everybody then I got to see what the best possible date looks like.</u>

Inquiring about generic scenarios may uncover the sources of the information that the users use in performing the tasks. These may not be obvious or what a rationalistic analysis of the problem domain would suggest are the logical sources of information. These scenarios and critical incidents are also appropriate for inquiring about obstacles.

> J: Yeah, like for example <u>if someone at [that department] does not read email, I'll take my message, print it down to a Word file and then fax it, because I have a fax machine on my computer in addition to the fax</u>

machine out there, so then I will fax him a copy of the message that I e-mailed.
I: So what happens over there? Do they have a secretary or someone who picks up the message and takes the message to them?
J: Yeah who takes it to him and then he responds that way. That's what I'm learning to do. I try to do as little as I can by telephone because I think that takes much longer. Then you have to wait for people, and have to wait for responses...

Another purpose behind inquiring about scenarios and incidents is to test one's understanding about responsibilities.

I: If someone contacts you and says "I need to meet with the dean." How do you know that they take priority?
M: I've been here long enough. I've been here almost two years. So I have a pretty good feel, on-campus, for who gets priority and also because of knowing who they are, I can sort of figure out the nature, I mean, I will screen. I will ask.

One may want to know which decisions are at the discretion of the informant, when they would defer to another person, or when they would follow some procedures, and whether these procedures are firm policies or just rough guidelines.

Artifact Analysis

Office workers often develop ways of working that exploit properties of an artifact that were not deliberately designed into it for that purpose. The work environment is full of concrete artifacts that support tasks or constrain the way they are done. Sometimes a generic scenario or critical incident can point to relevant artifacts. Sometimes artifacts are apparent in the work setting and can be asked about directly.

I: You have three types of calendars there?
M: Three types? I just printed that one out and reformatted it, it's the same thing. But there are different formats. There's like a month at a time. There's the week at a time, this is what I use because again I can block out the length of time. The thing about the monthly, it doesn't give me that luxury. It only says at 8:00 you've got this meeting but it doesn't tell me how long it lasts which is what I need to know and the nice thing about the monthly, though, it gives this banner option that says he's going to be gone for this conference over a period of three days. So I can put a banner across the whole week. But I can't do that in this [refers to the weekly] particular view.

The user uses the calendar as a visual mnemonic. Each affords a different kind of display of time and availability that is required to make a scheduling decision. We can also see that hard copies obtained from an on-line system are often more useful than the on-line version. For example, we learn that M. will print out a small copy of the schedule for her employer. The small copy allows portability and is easy to read. Some of the incidental properties of the hard copy make it preferable even though it contains the same information as the on-line version and may be out of date.

Deliberate fuzziness

Many work activities involve a relaxed or incremental approach to commitment and task performance. For example, in scheduling, a person may wish to blank out a period for most purposes, but would accept interruptions that were important or urgent. Sometimes, multiple conflicting commitments, or double bookings, could be made with the expectation that some of them would later become moot. For example, a much-postponed meeting will be postponed again or attendance delegated to another at the last minute.

> M: Because he's trying to do several different things on one trip so he might be flying to two or three different cities like the trip he's on today, he's flying out to Denver for a meeting in Boulder but then he's going out to California to meet with other people so it's never just a direct round trip, he's always making three or four stops along the way...A difficult thing that I have is maybe he says "I have got to see Dr. C. before such and such a date." Well, we look at the dates and I've got Dr. C.'s available dates so of course I take whatever times he's available and I may have to bump two or three people in order for him to do that. So I have to cancel whatever might be scheduled to try to work out other times.

A system that treated all commitments as binary events may be insufficiently flexible for the demands of real working practices. Even a series of decision branches may not allow for all possible combinations of behavior. This suggests that goals should sometimes be operationalized by extended sequences of behaviors that gradually increase commitment and support its negotiation, rather than enforcing fixed operation sequences that are difficult to undo. Sometimes, tasks may be performed within certain guidelines that are not intended to be firm policies. For example, some periods may be preferred for certain activities and will be used first for scheduling, without it being a hard and fast rule that those activities must occur then.

Discussion

Ecological investigation reveal that the users' behavior and the system environment exhibit regularities that the designer should not take for granted. For example, our informants told us about variations in how meetings are really scheduled, obstacles that can prevent scheduling, and creative recovery behaviors that far exceed the bounds that a fully automated system would be able to mimic. However, without some kind of guiding methodology from rationalistic methods, the designer may find it difficult to sift through this information to identify the nuggets that help enhance system usefulness and usability. Without this kind of framework, ecological data simply confirms existing guidelines.

How do the categories of design issues that were used to organize the meeting scheduler illustration facilitate the design process? Bearing in mind that we have restricted our attention mainly to user interviews, and their role in facilitating the early requirements establishment and specification activities of a project, we can identify five specific contributions:

- Actor and action identification is likely to be more accurate if a careful ecological analysis is performed that elucidates user's responsibilities and their range of discretion. One of the most problematic decisions in interactive system design is determining the system boundary. It is important that the designer does not automate actions that are never performed or that incorporate subtle judgments unlikely to be discovered by rationalistic methods.
- Generic scenarios and critical incidents also help the designer to specify the action sequences that fulfill goals in the work environment before system implementation.
- Artifact analysis suggests subtle uses of artifacts in the environment that would not be apparent from a functional analysis that presupposes a logical purpose to all artifacts and seeks to abstract away from their physical properties.
- Obstacle identification and the analysis of existing obstacle avoidance and amelioration strategies help make the requirements for the system more complete. These strategies make more apparent the properties that the system must have for it to be sufficiently flexible for the working environment.
- Finding out about typical frequencies and volumes should help to identify critical nonfunctional requirements, and eradicate technically attractive but contextually unnecessary requirements ("gold plating").

Other rationalistic approaches would benefit from the incorporation of ecological methods. For example, in the Object Modeling Technique [Rumbaugh et al., 1992] designers discover object classes and associations by preparing or obtaining short documents about the problem domain and then underlining noun and stative verb phrases. They then prune the resulting lists by applying standard heuristics. In other object-oriented methods, designers model the problem by analyzing scenarios of use [Jacobson, 1992]. But these methods do not guide the designer toward analyzing the right documents or scenarios; instead, the authors tacitly assume that designers will use their expertise and imagination appropriately. We believe that not only would incorporating ecological methods into the design process benefit object-oriented methods, but that it would be straightforward. Indeed, the seeds were planted long ago: Ten years before object-oriented analysis was even conceived, Spradley [1979] wrote a guide for analyzing ethnographic field data that sounds uncannily like modern object-oriented methods.

Our intention in this paper has not been to show that rationalistic methods are irrelevant to ecologically valid system design. That view would be too extreme. Nor have we set out to show that the user-centered perspective is important for developing usable systems. That much is obvious. What we are arguing is that ecological methods must be incorporated into rationalistic methods if they are to have a systematic place in system design. Furthermore, we have indicated, for one rationalistic approach and one style of ecological data gathering, how user-centered perspective can focus a rationalistic design process on issues that matter in the work environment.

References

A. Dardenne, A. Lamsweerde, and S. Fickas, "Goal-Directed requirements acquisition," *Science of Computer Programming*, **20**, 1993, 3-50.

J.M. Flach, "The Ecology of Human-Machine Systems I: Introduction," *Ecological Psychology*, **2(3)**, 1993, 191-205.

J.D. Gould and C.H. Lewis, "Designing for Usability: Key Principles and What Designers Think," *Comm, ACM,* **28(3)**, 1985, 300-311.

J. Grudin, "Why Groupware Applications Fail: Problems in Design and Evaluation," *Office: Technology and People*, **4(3)**, 1989, 245-264.

I. Jacobson, *Object-Oriented Software Engineering: A Use-Case Oriented Approach*, Addison-Wesley, 1992.

R. Kling, "Social Analysis of Computing: Theoretical Perspectives in Recent Empirical Research," *Computing Surveys,* **12(1)**, 1980, 61-110.

J. Nielsen, *Usability Engineering,* Academic Press Professional, 1993, 207-225.

D.A. Norman, *The Design of Everyday Things*, Doubleday, 1988.

D.A. Norman, "Cognitive Artifacts," in J.M. Carroll (Ed.), *Designing interaction: Psychology at the Human-Computer Interface*, Cambridge University Press, 1991, 17-38.

C. Potts, "Requirements Completeness, Enterprise Goals, and Scenarios," submitted for publication, 1995. [Note for reviewers, by camera-ready date, this will be citeable as a TR or conference publication, in press]

J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *Object-Oriented Modeling and Design*, Prentice Hall, 1991.

J. Spradley, *The Ethnographic Interview*, Holt Rinehard-Winston, 1979.

A. Van Lamsweerde, R. Darimont, and P. Massonet, "Goal-Directed Elaboration of Requirements for a Meeting Scheduler: Problems and Lessons Learnt," *PRoc. RE'95: 2nd International Symposium on Requirements Engineering*, IEEE Computer Society Press, 27-29 March, 1995, York, UK.