Plan-based Interfaces: Keeping Track of User Tasks and Acting to Cooperate

David Franklin, Jay Budzik, and Kristian Hammond
Intelligent Information Laboratory, Northwestern University
Evanston IL 60201 USA
{franklin, jlbudzik, hammond}@infolab.northwestern.edu

ABSTRACT
The ability to reason about the activity of a user is crucial to the implementation of any Intelligent User Interface. If it is able to recognize what a user is doing, a computer can act to cooperate. Most computer systems limit themselves to command-response interactions—their trivial understandings of their users cannot support a more complicated interaction. However, by looking at the tasks that their users are performing and reasoning about sequences of actions, a computer system can provide a more interesting level of interaction that is more efficient and does not demand as much of its users. Furthermore, the understanding of the user's activity provides a context within which to better understand future actions and to tune the sensing systems to look and listen for the actions that the user is most likely to take next. Finally, in many domains, such computer systems can recognize user tasks and act to cooperate without requiring a deep, goal-oriented understanding. In this paper, we look at the process-based interface used in the Intelligent Classroom, focusing on how a human lecturer can control it by simply going about her presentation. Also, we look at how the general ideas have been adapted to Jabberwocky, a speech-based interface to Microsoft PowerPoint that automatically switches slides, and how they are being applied to extend the functionality of Watson, an autonomous web research tool that uses the document a user is viewing as a search context.

Keywords: Plan-based intelligent user interfaces, perceptual user interfaces, representation of activity, task recognition

INTRODUCTION
The command-based user interface paradigm, which has been dominant in computer/user interaction from the very beginning, poses great difficulty to users, especially in applications with particularly rich interfaces. People often are incapable of remembering which of the potentially hundreds of discrete commands will accomplish what they want to do in interacting with a computer. The problem with systems employing this paradigm is simple: the more powerful the system is, the more commands the user is required to know in order to utilize its power. This is true for everything from a desktop application with menus and dialog boxes to a multi-modal Intelligent Environment with cameras and microphones. The entire burden of communication falls upon the user.

On the other hand, computer interfaces that allow a user to describe, in natural language, what they want the system to do for them are far beyond the state of the art. Such systems would have to sift through the vagueness inherent in human communication to discover the user's intentions, and then autonomously fill in any details that the user left out. It will likely be a long time before any computer system can begin to approach this level of cognition.

So instead, we propose an interaction paradigm based on the idea of providing competent assistance to a user. The software system is designed to assist its users in a particular set of tasks. It knows (i.e. has enumerated) the sorts of things that its users are likely to perform and knows the ways that it can assist in performing those tasks.

As a first step towards building computer systems that serve as competent assistants, we have built the Intelligent Classroom, a prototype automated lecture facility that serves as its own audio/visual assistant. As its user, a speaker, goes about her presentation, the Classroom watches and listens to her and, when appropriate, assists her. For example, the Classroom films the presentation that occurs with in it, producing a video feed that would be suitable for distance learning. To produce a useful video feed (i.e. to be a good assistant) the Classroom must capture all the activities in the Classroom that the speaker would wish her viewers to see. So, if the speaker starts writing on the board, the Classroom must focus its presentation camera on the words and figures that she writes. Finally, when the speaker wants the Classroom to play a particular video segment, the Classroom must cue up the video on the VCR, set the display to accept the VCR output, start the video and make sure that the video is also captured in the video feed of the presentation.

While all the algorithms and representations discussed in this paper were developed for use in the Intelligent Classroom domain, they were designed with an eye to future use in other domains. The Jabberwocky slide switcher is based loosely on the Classroom's implementation, and extended to deal with the extreme sensory noise that is inevitable with unconstrained speech recognition. Currently, we are also looking at ways of incorporating more knowledge about people's brows-

IUI'02, January 13-16, 2002, San Francisco, California, USA.

ing and writing activity into the Watson web research tool. The underlying idea that drives all this work is simple: the better a system is able to understand the user and what she is doing, the better it is able to understand her needs and to ultimately help meet them.

RELATED WORK

There has been much research addressing techniques for improving interaction between humans and computer systems. In this section we look at some of the work that is most relevant to this paper.

In one of the best examples of using context to make sensing easier and to facilitate interaction, the KidsRoom [1] immerses children in a fanciful adventure. Through computer vision and some simple audio analysis, the KidsRoom allows the children interact with virtual characters and scenes. At every point in the story, the computer system knows what it is reasonable for the children to do and uses this knowledge to aid its action recognition algorithms. As a result, the sensing tasks are rendered sufficiently reliable to successfully immerse the children in the story, unaware of and not concerned with what is going on behind the scenes. But, although the KidsRoom provides insight into how to tie sensing to tasks, it does not demonstrate interaction in the sense we are pursuing in this paper: the KidsRoom directs the actions of the children—not the other way around.

Projects like the KidsRoom, which utilize multiple sensing modalities to interact with people in non-electronic domains, are often referred to as Intelligent Environments. In [4], a few of these environments are discussed and the importance of capitalizing on the various environmental and contextual constraints is argued in detail.

In electronic domains, most research strives to strike a delicate balance: they wish to provide valuable assistance and yet not annoy their users. At Microsoft, the researchers behind today's most demonized user interface agents have done some of the most compelling research on how to assist their users. In [10], they look at many of the important issues in integrating autonomous action within a windows-based user interface and also formalize a decision process for determining whether help should be provided. In [9], they discuss probabilistic approaches for determining what the users are doing and for using this information in deciding how and if to assist them. The interface agents that have shipped with the various Office products over the last six years have been rather unreliable helpers, largely due to the sheer magnitude of the problem they are trying to solve. How can a help agent hope to use a five word query to select one of more than a thousand help topics, particularly if it employs only a beefed up term-spotting approach? A problem of that magnitude requires human-level cognition.

Probably the research that has the most in common with that described in this paper—both in philosophy and scope—is the Collagen project at MERL [12]. Collagen suggests techniques for designing user interfaces that allow the user and the computer to communicate about the tasks the user is performing (e.g. to ask things like "Where am I in this task?" or "What should I do next?"). They have used their approaches in building interface agents for a VCR, a thermostat, and a graphical interface development tool. In these domains (like those described in this paper), the notion of task is central and the domains are of a reasonable complexity. The research described in this paper differs from the Collagen research primarily in its use of physical sensors (cameras and microphones) as input devices.

REPRESENTING ACTIVITY

In reasoning about what is going on within it, the Intelligent Classroom keeps track of the various activities being pursued by the speaker as well as its own activities in control of its various autonomous components. Before we delve into our activity representation, it is important to consider all the ways that this representation will need to be used. There are three ways that we, as people, commonly use plans (in the non-technical sense of the word). To accomplish some goal, we may execute some plan that will achieve it (i.e. plan execution). To see what goal someone else might be pursuing, we may match her actions to the set of plans we know about (i.e. plan recognition). To guess what someone is likely to do next (so that we are somehow able to react), we can follow along with her in her plan and look at future actions (i.e. projection). For example, if someone picks up a box and walks towards the door, we can infer that they wish to exit through the door and therefore we should go and open the door for them. These ways of reasoning about plans are an essential part of how we interact with the world, and should therefore be supported by any intelligent user interface.

Indeed, the Classroom *executes* plans for such things as playing video segments and controlling the camera; it *recognizes* the plans that are pursued during the course of a lecture; and it *examines* those plans to see how it ought to cooperate with them. However, it is important to note that the Classroom is designed purposely to avoid reasoning explicitly about the goals underlying these plans. Instead, the Classroom reasons exclusively about the actions and events that make up the plans, which is sufficient to recognize what plans a user is performing and to take appropriate actions in response.

In the remainder of this section, we discuss how the Classroom represents the activities taking place in it—through sequences of actions called processes—and how the Classroom keeps its process set synchronized with what is actually happening.

Viewing the world as a set of processes

The Intelligent Classroom attempts to explain everything it senses it terms of its own actions and the actions of other agents, generally the speaker. These action explanations need not be very detailed; the Classroom is less concerned with why someone does something and more concerned with what it should observe when they are doing it. The Classroom maintains a set of activities going on in it (e.g. there is a video playing, or the speaker is talking at the podium). This is the

Classroom's understanding of what is going on in the presentation. When something happens that is not explained by any of the activities in its understanding, the Classroom must revise this activity set to include something that would explain the strange observation.

We define a *process* to be sequence of actions (or compound actions) that will be executed by a single agent. For example, the sequence of actions in a STRIPS plan for stacking blocks would be considered a process. This definition of process separates the actions from the goals they are intended to achieve. Later we will discuss how sets of one or more processes can be tied together with goals to form plans. Our language for describing processes supports partial orderings, simultaneous actions, conditional branching, and hierarchical composition.

A process is made up of one or more discrete steps, each of which specifies a number of continuous actions that make up the running of the step and a number of discrete events that may happen during its running. Figure 1 shows portions of the Classroom's representation of a speaker's process of going to the board and writing something. The first step in the process (labeled "_1") specifies that the speaker should start moving towards the board and that, as she moves, the distance between her and the board should decrease. Furthermore, it specifies that the step is complete when the process of moving to the board is finished. Conceptually, a process can be thought of as a finite automaton, with each state corresponding to a step in the process (with its continuous actions) and each transition out of a state corresponding to an event that might occur the associated step.

Typically, the Classroom will have three to five active processes (plus their sub-processes due to composition) in its set of activities. One of these will be the speaker's top-level process (e.g. lecturing at the podium, doing a slide presentation, or writing on the board), and the others will be the Classroom's processes in service of cooperating with the speaker. Figure 2 shows a representation of the Classroom's set of activities. At the given moment, the Classroom expects that the speaker will take an action that will produce events E_1 or E_2. If she does, the Classroom will advance the process

```
(define-process (move-to-board-and-write)
 (main-actor
  (person ?speaker))
 (roles
  (chalkboard ?board)
  (chalk ?chalk))
 (steps
  (_1 (do _move (move-to ?board))
    (track _delta (track-floor-distance ?speaker
                    ?board))
    (wait-for (_move :done) _2)
    (ensure _delta (:decreasing)))
  ...)
 (time-constraints
  (process-duration (range 30 3000) (expected 300))
  (step-duration _1 (range 0 30) (expected 5))
  ...))
```

Figure 1: A partial process definition for the speaker going to the chalkboard and writing.

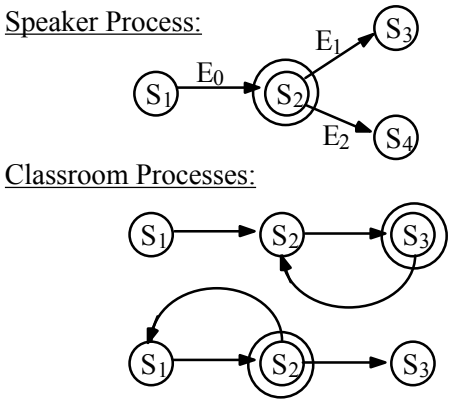

Figure 2: Finite automata representing the activities taking place in the Classroom

representing her activity to states S_3 or S_4. If instead the speaker takes an action that produces some other event, then the Classroom will need to revise its understanding of what the speaker is doing, performing the process recognition described later in this paper.

Following along with processes

The processes are designed in such a way that the Classroom can essentially use the same algorithm for executing a process that it uses for observing the speaker as she executes a process. The algorithm for executing a process is as follows:

1. Start up all the sub-processes that are a part of the current (initially, the first) step in the process. These substeps may include processes (as in the action in Figure 1 labeled "_move"), progress monitors (as in the "_delta" action), or "primitive" actions that manipulate the physical actuators in the Classroom.
2. Configure the sensing systems such that they are looking and listening for the events that are a part of the current step in the process. For some events (such as "the _move sub-step is done"), no additional sensing is necessary. But for other events, the computer vision system may need to be configured to detect a particular type of gesture or pose, or the speech recognition system may need to be configured to listen for a particular set of commands. Here, the situation of executing this process provides a context in which to understand what is observed in the world.
3. Wait for one of the events to occur and take whatever resulting action that event prescribes. Usually, this will involve going on to another step (go back to step 1 of this algorithm) or halting the process (succeed or fail).

To alter the algorithm to allow the Classroom to observe the speaker and to follow along with her in her plans, we only need to change a portion of the first step. Rather than actually performing the primitive actions that are a part of the step, the Classroom performs "observation" actions that complement the primitive actions. So, instead of physically picking up a pen from the marker board tray, the Classroom carefully

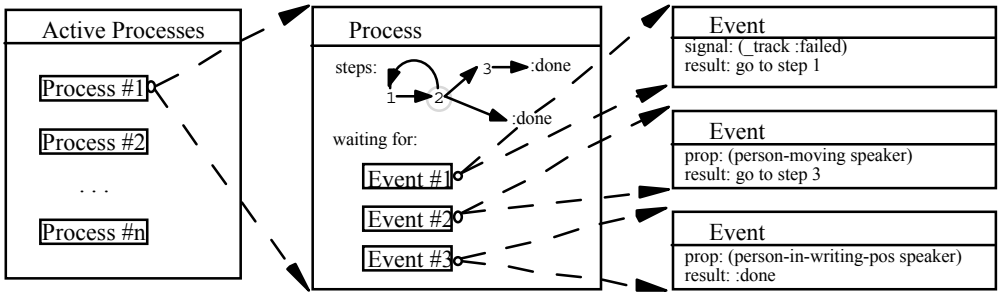

Figure 3: Important structures in the Process Manager

watches the speaker's hands to observe whether she is picking up a pen. For each primitive action the speaker might perform, the Classroom has a corresponding action for observing her perform it.

The basic algorithm for process execution is based upon Firby's RAPs reactive execution system, used in earlier robotics research [5]. This algorithm and its use in process recognition and cooperation are discussed in detail in [7]. Figure 3 shows a graphical representation of the important structures in the process manager, the system component that keeps track of all the processes being executed in the Classroom. On the left is the set of processes. In the middle is a process, with the steps drawn as a finite state machine. On the right is the set of events that the current step in the process is waiting for.

Recognizing what the speaker is doing

The algorithm outlined above allows the Intelligent Classroom to follow along with what the speaker is doing, but only if it already knows what she is doing. When a speaker is doing a presentation, the Classroom tries to keep its representation of what the speaker is doing consistent with what it observes the speaker do. In the process manager, this means continually stepping through its set of processes to keep them synchronized with the speaker and also revising the set of processes when necessary. The basic idea is that, when the Classroom observes the speaker take some action, there should be a process in the process manager that "explains" why she took that action. For instance, if the Classroom knows that the speaker is going over to the board to write, the speaker's action of picking up a piece of chalk is explained, because that is what the Classroom expects her to do after arriving at the board. However, if instead of writing, the speaker walks away from the board, the walking action is not explained—apparently the speaker is not writing on the board after all.

In situations where a speaker's action is not explained by any of the processes in the process manager, the Classroom must revise its understanding of what the speaker is doing; it must add a process that explains the speaker's action (i.e. the process must include the speaker's action in one of its steps). Initially, there may be several processes that adequately explain the speaker's single action, even though the speaker is actually only performing one of them. But, as the speaker continues in what she is doing, the Classroom will be able to rule out the incorrect ones.

So, when faced with an unexplained action, the Classroom selects all the top-level processes in its library that: (1) are appropriate for the speaker to execute, (2) have the unexplained action as a part of one their steps, and (3) are consistent with the speaker's actions. For the third condition, the Classroom may have to look at the last few actions the speaker has taken and ensure that they could lead to the speaker taking the unexplained action.

Once the Classroom has its set of candidate processes, it must reduce the set to the single correct process. This is not immediately possible because the Classroom simply does not have enough information to determine which process the speaker is performing. So, the Classroom eliminates the incorrect processes as the speaker performs more actions. There are two ways that the Classroom is able to eliminate incorrect candidate processes. For the first, the Classroom eliminates processes that are not progressing properly. This means either that one of the measurable quantities is not behaving properly or that some event that the process predicts does not occur when it ought to. (Process steps can have time constraints associated with them, giving a reasonable interval of time for the step to take.) For instance, if the Classroom has hypothesized that the speaker is going over to write on the board, it expects the speaker to get there within a few seconds and to be moving generally in the direction of the board. If she stops before getting to the board or starts moving away from it, the Classroom will be able to remove that process from consideration.

The second way that a process can be eliminated is if the speaker performs actions that are inconsistent with her executing the process. Since currently the Classroom assumes that the speaker is only doing one thing at a time, the Classroom expects all of the speaker's actions to be explained by the speaker's process. Therefore, if the speaker takes an action that is not explained by a given candidate process, that process can be eliminated from consideration.

The process manager provides the functionality for keeping track of all the candidate processes and eliminating them as

future speaker actions contradict them. When the Classroom observes the speaker taking an action, it tries to use this event to advance the processes in the process manager. In doing so, it also makes sure that the top-level speaker process (or one of its sub-processes) explained the speaker event. If the speaker event is not explained, then the Classroom will need to find a new explanation for what the speaker is doing. It gathers a set of candidate processes and adds them to the process manager, marking them as explanations. As the process manager runs, it checks that each candidate process is progressing normally, and eliminates those that are not. When the Classroom observes the speaker taking another action, the process manager will advance the candidate processes with this action. Any of the processes that do not explain the action are removed from the process manager. When only one candidate process remains, it is accepted as the correct explanation for what the speaker is doing.

One issue that often comes up in the Classroom domain, is that the Classroom will often need to take some sort of action before it has reduced the set of candidate processes to one (e.g. the Classroom always needs to be performing camerawork). Our solution has been discussed in earlier work [7], but the basic idea is that the Classroom chooses a explanation that generalizes the set of candidate explanations. For example, as the speaker approaches the board, the Classroom knows the she intends to write on it, revise something, erase something, or just refer to something. Each of these explanations requires a slightly different camera technique. Until the Classroom has ruled out the incorrect explanations, it will act on a general explanation: "the speaker is going to the board to do something."

The important idea behind the process representation and algorithms is that the Classroom always tries to understand the speaker's actions in the context of what it believes the speaker to be doing. It will not even consider other possible explanations unless it is absolutely necessary. This is crucial in the Classroom's role as an intelligent user interface. If the Classroom is continually second-guessing itself and vacillating between explanations as it deems one and then the other to be more probable, the speaker will be baffled by its resulting erratic behavior and find it frustrating to work with the Classroom.

Researchers often classify plan recognition problems as either "intended" or "keyhole" recognition, based on whether the observed person is deliberately acting to make her intentions clear. In the Classroom, it is safe to assume that the speaker is not purposely obscuring her intentions, but we must not assume that she will always make things obvious. The Classroom's task recognition uses an aggressive keyhole recognition approach, with the knowledge that the speaker will likely correct it (e.g. issue direct commands) if its recognition fails badly.

COOPERATING WITH THE SPEAKER

Given its understanding of what the speaker is doing, the Intelligent Classroom must figure out how to cooperate. Its cooperation will be in performing the roles of audio/visual assistant and cameraman. Each top-level process that the speaker might perform has a plan associated with it that describes how the Classroom should cooperate with her. The idea is that the speaker, in executing her part of a plan, expects the Classroom to do its part of the plan. Cooperation, as might be expected, is achieved through being aware of what others around you are trying to do and then seeing how you fit into their plans.

What is a plan?

We define *plan* as a set of processes (often to be executed by a number of different agents) that when run together successfully, accomplish some goal. In the Classroom, most plans have one process executed by the speaker and one or two processes executed by the Classroom. It is important to note that this is not really a new definition of plan—any plan that has a step of the form "wait for this event to happen" is implicitly representing processes external to its main actor. This definition makes explicit the presence of other agents or exogenous events. In the Classroom, these plans attempt to express a common understanding of how a speaker and an audio/visual assistant interact.

The plan the Classroom uses when the speaker walks over to the board and writes includes two processes. It includes the previously described process for what the speaker does (walking and writing) and a process that details how the Classroom should film the speaker as she walks and writes. Finally, the plan definition ties these processes to a particular goal (the speaker effectively communicating while at the board) and specifies how the processes should fit together. The Classroom's filming process specifies what camera techniques to use and the plan tells it when it is the right time to switch techniques. In doing its part of the plan, the Classroom must not only do the right thing, but also do it at the right time.

How does the Classroom use plans?

When the Classroom infers that the speaker is pursuing a new plan, it adds the additional processes to the process manager and also adds new events for the processes to wait for, enforcing the synchronization constraints imposed by the plan. Once the processes have been thus altered, the process manager is able to ensure that the processes fit together properly. In the plan for writing on the board, the Classroom will change camera techniques once when the speaker enters the vicinity of the board and another time when she starts writing. (Each time, the Classroom will focus on a more precise region of the board.)

In the plan for playing a video segment on the Classroom's VCR, the processes provide another important role: they capture the context of playing a video segment, providing a framework for interpreting the speaker's commands and gestures. If, while the tape is playing, the speaker holds out her hand or says, "pause", the Classroom knows what she wants it to do. Because of the context, what would otherwise be an ambiguous command becomes clear.

CLASSROOM SCENARIOS

Having provided this introduction to some of the important representational details of the Intelligent Classroom, we now look at how they effect the actual interaction that occurs in the Classroom. These scenarios demonstrate how the techniques discussed in this paper can be applied to provide a remarkable level of cooperation with a human user. In many cases, the speaker does not need to do anything that could even be construed as a command; she just goes about her presentation and trusts the Classroom to act at the appropriate moments. And, when more direct control is necessary, the speaker can use rather elliptic commands, since the Classroom knows the context in which she is making them.

Writing on the board

Imagine that the speaker is lecturing at the podium in the middle of the Classroom. The Classroom is working with the plan for standing in place and lecturing. Its part in this plan involves keeping the presentation camera zoomed in close on the speaker's face, trying to move as little as possible. And, if the speaker does much gesturing with her hands, the Classroom will zoom out a little to capture her gestures as well. The Classroom is not really listening to what the speaker says. (There are a few explicit commands that the Classroom is listening for, but otherwise, the actual words she is saying are ignored.) The Classroom is watching her head and hands and trying to keep everything of interest framed by the camera. The speaker can just ignore the Classroom, and the Classroom is only paying enough attention to the speaker to do its job.

Then the speaker leaves the podium. This is unexpected. There is nothing in the speaker process for standing and lecturing that explains why she would do this. So, the Classroom is forced to revise its understanding of what the speaker is doing. Given only that the speaker is moving, there are many possible explanations for what she intends to do. She may be going over to the demonstration table to illustrate a point; she may be going over to the board to write; she may simply be stretching her legs before returning to the podium. The one thing the Classroom can infer is that she is going somewhere. The Classroom knows its role in the plan for this (very abstract) top-level process and zooms the camera way out and follows the speaker as she moves.

When it is clear that the speaker is heading towards the board, the Classroom is able to reject all of the other candidate processes that do not involve being at the board, limiting its consideration to such activities as referring to something on the board, erasing something, revising something, or writing something. The Classroom has a plan for the (still rather abstract) activity of going to the board and doing something, and its role now is to wait until the speaker is almost at the board and then shift the camera frame to include the entire board.

When the speaker finally picks up a pen and begins writing on an empty space on the board, the Classroom finally knows precisely what she is doing. It zooms in on what she writes, remembering the precise region of the board, in case she later refers to it.

Doing a structured presentation

While the default behavior of the Classroom is sufficient for typical presentations, the speaker can gain greater control of how the Classroom will act by giving it a presentation script. This script gives an outline of many of the key components (slides, videos, desired camera shots) of the planned presentation and can tell the Classroom how to respond to particular events. It allows the Classroom to anticipate what will happen next and can even instruct the Classroom to take particular actions without being explicitly told to by the speaker.

For example, imagine that the speaker is preparing to give an anatomy lecture, utilizing a skeleton and a video with brief segments discussing a number of the important bones. The default behavior of the Classroom will not do a good job of framing the individual bones with the camera, making this a good lecture for utilizing a presentation script. In the script, the speaker provides an outline for the presentation: after a brief lecture, she will move over to the skeleton and alternate discussion of individual bones with the brief video segments. Furthermore, the presentation script specifies that she will touch each end of the particular bone she is discussing, giving the Classroom a region to zoom its presentation camera in on. Figure 4 shows the camera frames as the speaker touches each end of an ulna bone and after the Classroom has zoomed in on it.

When the speaker is giving her presentation, the Classroom is able to do much better camera work. After the speaker completes her initial lecture and moves towards the skeleton, the Classroom knows what is going to happen next. It carefully watches the speaker's hands, looking for the brief pauses over the skeleton, thereby specifying a region to zoom the camera

Figure 4: The speaker specifies a camera region.

into. Once the speaker has begun discussing a particular bone, the Classroom begins listening for an indication that the speaker wants the next video segment played. Since the Classroom knows that this is what will happen next, it is able to dramatically expand what it interprets as a command to play the next segment. So, even though "roll it" and "next" should not be considered commands to play the VCR in the midst of writing on board, in the context of this point in the lecture, both commands are unambiguous.

Lecturing from slides

While speakers may be understandably hesitant to go to the effort of preparing a detailed presentation script for their lectures, they are often willing to prepare a set of slides. These slides provide the same sort of structure as a presentation script. In fact, speakers often even insert slides into their talks indicating important non-slide events in the presentation such as the playing of a video segment. So, the Classroom, in analyzing a set of slides, can build its own presentation script for a lecture.

But, given the contents of a set of slides, the Classroom can do even more. Jabberwocky, the automated slide switcher that we developed as a part of the Classroom, follows along with the speaker, point-by-point, as she lectures from her slides, switching slides at the appropriate moments. Through a syntactic analysis of the slides, it extracts words and phrases from the slides that are representative of particular slides. Then, as the speaker lectures, Jabberwocky listens for these important words and phrases as indicators of where she is in her presentation.

This technique of word and phrase matching allows the speaker to deviate from the given slide ordering, spontaneously skipping over sections when time is running out, jumping to a particular slide to address a question, or even performing a completely freeform presentation using the slides as a secondary resource. (However in practice, due to the overwhelming unreliability of speech recognizers — especially in noisy rooms and with anxious speakers—it is generally best to provide some structure to a presentation.)

In giving such a talk, the Classroom is always interpreting what the speaker is saying in the context of where she is in the presentation. Also, commands are interpreted in the context of doing a slide presentation. So, when the speaker says, "Skip ahead to the conclusion", Jabberwocky skips to the first slide in the concluding sequence. And, when the speaker says, "Jump to the slide that talks about how Jabberwocky gets its name from the Lewis Carroll nonsense poem," Jabberwocky locates the slide that best matches that description.

MOVING OUTSIDE THE CLASSROOM

The techniques developed in implementing the Intelligent Classroom were intended for use in a broad range of interactive applications and we have the opportunity to apply them to other projects in our laboratory. In doing so, we discover many of the assumptions underlying the Classroom's design, and if these assumptions do not hold in a new domain, we are forced to revise the techniques to accommodate the differing constraints of the new domain.

Jabberwocky

As previously discussed, Jabberwocky was originally conceived as a slide advancer for the Classroom, but ended up being a project in its own right. In the Classroom, a slide show was transformed into a presentation script and each slide became a top-level process with slide points as steps and the words and phrases as events to advance from point to point and from slide to slide. So, if it heard what was expected, the Classroom would have no trouble following along with the speaker through the planned presentation. And, when the speaker said things other than what was expected, the Classroom would propose that the speaker had jumped to a different slide—one that contained the words and phrases that she said.

In theory, this should have worked splendidly—by talking carefully we could easily achieve the 95% accuracy our commercial speech recognizer promised and if we simply read the slides, the Classroom followed along perfectly. In practice however, the speech recognition produced accuracy results closer to 40% (due to the more expressive sort of speaking common in public) and speakers would invariably happen to say key phrases from one slide while discussing another slide.

As a result, while retaining the basic process structure, we changed the way that the noisy speech input was interpreted, employing a probabilistic approach to determine what slide the speaker wants to discuss [8]. The sensory results were interpreted based on the current context (including what slide the speaker was on, how long she had been discussing it, and what slides she had previously discussed) and the underlying operation was left unchanged.

Watson

Just as systems like the Intelligent Classroom and Jabberwocky are aimed at recognizing enough of what a person is doing in the physical world to be helpful to her, we are also working on systems that do the same thing in the electronic realm. Watson [3] is a system that observes users in document manipulation applications (e.g., word processors or Web browsers). Watson's goal is to automatically conduct online research for the user, without requiring any explicit intervention. It analyzes user actions to determine when to conduct a search, and what kind of information to search for, using simple finite state detectors. For example, in a word processor, if a user inserts a caption with no image to fill it, Watson will search for relevant images using the text of the caption entered.

The most recent version of Watson builds a profile of user interests. Unlike other profiling systems, however, Watson does not assume that any document viewed is relevant to the user. Instead, Watson judges relevance based on the user's sequence of actions before, during, and after viewing a document. For example, suppose a user is browsing a list of

links. She clicks on one, and then clicks back. She clicks on the next one, and immediately clicks back again. She clicks on the next one, reads it, and then clicks on a link somewhere in the middle. Watson infers that the user is more interested in the third document than in the others.

In his discussion of Letizia [11], Lieberman suggests many such processes that people browsing the web might use, and what can be inferred when they do. It is our belief that, by reasoning more deeply about how people browse the web and use word processors, we can build research tools for them that are much more useful.

CONCLUSIONS

This paper presents user interface work based on the principle that the more the system understands its users and their tasks, the more useful it will be to them. Interaction is facilitated through the system's knowledge of what a user is likely to do and of how it can assist. This task knowledge allows the system to clarify potentially ambiguous commands and to take cooperative action without being told explicitly what to do. Here the notion is that the computer system plays the role of a competent assistant.

The algorithms described in this paper have been implemented in the Intelligent Classroom, a research lab set up as the front portion of a university classroom. The Classroom currently performs camera work (aiming and zooming a fixed-location camera) and handles interaction with a marker-board, a VCR, and PowerPoint slideshows (through Jabberwocky). Gargoyle [6] and IBM ViaVoice provide visual and speech input.

The research is discussed as it is implemented in the Intelligent Classroom, but we argue that the same techniques can be applied to many other domains. In fact, some of the techniques serve as a basis for Jabberwocky, an automated slide switcher, and are in the process of being applied to Watson, a web research tool.

The work on Jabberwocky has suggested some ways that probabilistic approaches could be applied in the Classroom. Though the computer vision techniques we use are certainly more reliable than the speech recognition that Jabberwocky is forced to deal with, they are not infallible. The sensory noise does not invalidate the algorithms we use. In the worst case, it causes the Classroom to become briefly "distracted" when it hallucinates a user action. Most of these distractions occur when the speaker's head or hand momentarily disappears. Generally the missing limb reappears within a few camera frames and the Classroom will re-infer what the speaker is doing within a few more—all of this takes approximately one second.

In the Classroom, we are currently looking at ways to incorporate more information sources (e.g. Watson and Rosetta [2], an indexing system for locating relevant research papers), helping a teacher (or her students) locate useful secondary sources of information. Also, we are investigating an approach to recognize when a speaker is switching between two or more techniques (e.g. interleaving the playing of brief segments of video with writing on the board). The basic idea is that the Classroom will recognize when a speaker has abandoned a process (rather than completed it) and will look for events that indicate she has resumed it. While this is not necessary for providing good camera work, it does limit the scope of the process recognition (favoring resumption over starting something new) and, in presenting what happened in a lecture, it allows the Classroom to more closely model what the speaker was trying to do.

REFERENCES

1. Bobick, A.; Intille, S.; Davis, J.; Baird, F.; Pinhanez, C.; Campbell, L.; Ivanov, Y.; Schutte, A.; and Wilson, A. 1998. Design decisions for interactive environments: Evaluating the KidsRoom. In *Spring Symposium on Intelligent Environments*.
2. Bradshaw, S.; Scheinkman, A.; and Hammond, K. 2000. Guiding People to Information: Providing an Interface to a Digital Library Using Reference as a Basis for Indexing. In *Fourth Inter. Conference on Intelligent User Interfaces*.
3. Budzik, J. L. and Hammond, K. J. 2000. User interactions with everyday applications as context for just-in-time information access. In *Fourth International Conference on Intelligent User Interfaces*.
4. Coen, M. 1998. Design Principles for Intelligent Environments. In *Fifteenth National Conference on Artificial Intelligence*.
5. Firby, R. J.; Kahn, R. E.; Prokopowicz, P. N.; and Swain, M. J. 1995. An architecture for vision and action. In *Fourteenth International Joint Conference on Artificial Intelligence*.
6. Flachsbart, J.; Franklin, D.; and Hammond, K. 2000. Improving Human Computer Interaction in a Classroom Environment using Computer Vision. In *Fourth International Conference on Artificial Intelligence.*
7. Franklin, D. 1998. Cooperating with people: the Intelligent Classroom. In *Fifteenth National Conference on Artificial Intelligence*.
8. Franklin, D.; Bradshaw, S.; and Hammond K. J. 2000. Jabberwocky: You don't have to be a rocket scientist to change slides at a hydrogen combustion lecture. In *International Conference on Intelligent User Interfaces.*
9. Horvitz, E.; Breese, J.; Heckerman, D.; Hovel, D.; and Rommelse, K. 1998. The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. In *Fourth Annual Conference on Uncertainty in Artificial Intelligence*.
10. Horvitz, E. 1999. Principles of Mixed-Initiative User Interfaces. In *ACM SIGCHI Conference on Human Factors in Computing Systems.*
11. Lieberman, H. 1995. Letizia: An Agent That Assists Web Browsing. In *Fourteenth International Joint Conference on Artificial Intelligence*.
12. Rich, C.; Sidner, C.; and Lesh, N. 2001. COLLAGEN: Applying Collaborative Discourse Theory to Human-Computer Interaction. To appear: *AI Magazine*.