Principles for integrating voice I/O in a complex interface

M. M. Taylor
Defence and Civil Institute of Environmental Medicine, Box 2000, North York, Ontario, Canada M3M 3B9
D. A. Waugh
Andyne Computing Ltd., 552 Princess St., Kingston, Ontario, Canada K7L 1C7

SUMMARY

The integration of voice into a complex interface like that between a pilot and an aircraft is not trivial. In this paper, we try to address some of the factors affecting the use and integration of voice in human-machine interfaces. We describe general principles for merging different kinds of human-machine interaction, and apply them to voice interaction in the cockpit. We do this despite published opinion that psychological principles cannot be applied in the design of human-computer interaction (e.g., Landauer, 1991).

The theory of Layered Protocols (LP) is introduced in context of the more general Perceptual Control Theory of behaviour (PCT). LP theory provides a model for describing interaction between complex partners based on a layered structure of protocols that differ in levels of abstraction. The proper use of feedback is fundamental to both LP and PCT.

Voice interaction is useful mainly for the control of tasks requiring discrete information. Failure of voice recognition systems is often caused by inappropriate feedback. Providing feedback and forcing correction word by word may increase the mental load on a user, often leading to instability in the interaction. Such inefficient, and often frustrating, use of voice interaction can often be overcome through the use of feedback at higher, more abstract, layers of interaction. Successful adoption of voice interaction depends on allocating the appropriate tasks of communication to the voice protocol, the dynamic modeling of the partner, and the use of higher level protocols to help control potential instability.

INTRODUCTION

Many people think that the problem of getting our machines to do what we want would be greatly eased if we could talk to them. Experimental aircraft such as a BAC-111 airliner in the UK and the Mirage IIIB fighter in France have been provided with voice I/O for this reason. But not all attempts to use voice have been as effective as their proponents hoped.

The blame for failure of voice I/O is often placed on the error rates of word recognition, but other factors may actually be more important. For example, in an effort to allow the talker to correct recognition errors, the designer may place a visual display of the recognized speech somewhere the talker can see it. But if there are only a few errors the talker may well miss them on the display. In addition, the display may both distract the talker from other visual tasks and take up valuable space in the cockpit. Such a visual read-back of the words recognized may be more harmful than helpful to the use of voice in the cockpit. But under some circumstances, it could be exactly what is needed.

A general approach to communication

The theory on which we base our analysis is known as the Theory of Layered Protocols (LP; Taylor, 1987, 1988a, 1988b, 1989). LP is a general theory of communication, compatible with an approach to psychology known as Perceptual Control Theory (PCT; Powers, 1973), and is readily described in its terms. PCT is well suited to the description and analysis of interactions with inanimate objects, whereas LP emphasizes the mutuality of interaction between "intelligent" partners, and can be seen as a specialized form of PCT.

The theory of Layered Protocols (LP) is based on a long tradition in psychology that people perceive, remember, and act at a number of levels of abstraction, the lower levels supporting the higher. LP theory, like PCT, is based on the properties of a hierarchy of control systems, but focuses on control loops that incorporate a partner with some independence of perception and of action.

LP theory is the focus of this paper, but before discussing it in any depth, we present a brief introduction to PCT, to help lay the groundwork.

Perceptual control theory

According to PCT, all behavior is directed to the control of perceptions at a variety of levels of abstraction simultaneously. A perception might be the tension in a muscle involved in the turning of a steering wheel that allows a driver to perceive the car as staying on the road during a trip for money for the purchase of food that allows the driver's body chemistry to stay within survivable bounds.

In the PCT view, all living things consist of a hierarchy of control systems that maintain their percepts at desired levels by means of actions they perform on the outer world. In very simple organisms such as bacteria, there may be only one level in the hierarchy, and only one possible action (e.g. "wiggle"), but this suffices to move the bacterium from noxious environments into suitable ones more often than not. In more complex organisms, there are more levels of control. Actions are, like the perceptions they control, organized at different levels of abstraction. The driver is "doing" many different things at once: following a life plan, performing a job, getting wages, driving, following a bend in the road, turning the steering wheel, and tensing certain muscles, among other behaviors (Vallacher & Wegner, 1987). A feedback loop controls each perception of the situation, as shown in Figure 1.

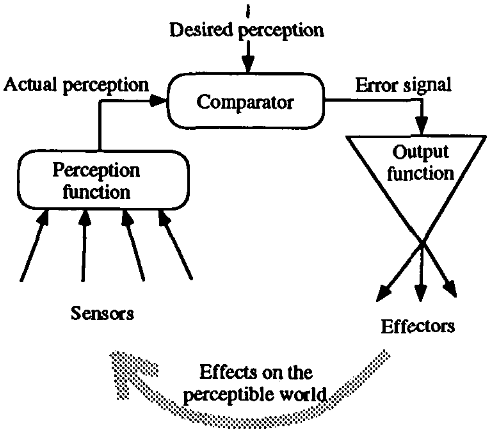

Figure 1. An elementary control system compares its perceptual input with a desired (reference) percept, and outputs the difference as an error signal that causes effects to happen in the world. These effects modify the perceptual signal in the direction of the desired percept.

An elementary control system (ECS), such as shown in Figure 1, is a unit of a control hierarchy. An ECS has a percept that is derived by some function from the sensory input from a variety of sources, including lower-level ECSs; it has a reference signal derived from a multiplicity of signals from higher-level ECSs, and it has a error signal that is the difference between its reference and its percept. Some amplifying operation converts the error signal into a set of reference signals for lower order ECSs, or, at the lowest level, into action signals for effectors (muscles), as shown in Figure 2. The amplified error signal, usually caused by unpredicted events in the environment, leads to behavior that alters the perceived state in, the protagonist hopes, the desired direction.

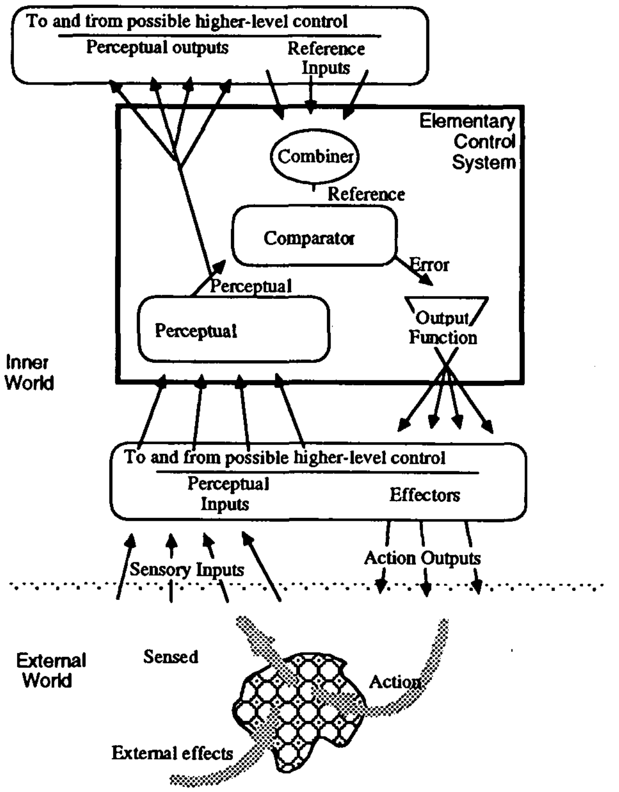

Figure 2. An Elementary Control System is normally connected to other control systems at both higher and lower levels, permitting the control of arbitrarily complex percepts.

ECSs are connected in a hierarchy in which the perceptual signals of low-level control systems combine to provide the sensory inputs to higher level ones. The connections among the perceptual elements in the hierarchy could be seen as a multi-layer neural network. Such a hierarchy could develop perceptual functions of arbitrary complexity (e.g., Lippmann, 1987).

In the same way that the perceptual function of a lower-level ECS provides one of the inputs to the perceptual funtion of a higher one, so do the action outputs of high level ECSs combine to provide the reference signals for low-level ECSs. Figure 3 shows a sketch of such a hierarchy, in which the lowest ECSs affect and sense the world of perceptible things directly. These "perceptible things" include the aircraft being flown, as well as communicative partners.

In a "classical" PCT hierarchy, the signals (percept, reference, and error) are all scalar. If, however, one looks at a set of ECSs at the same hierarchic level acting in parallel, the effect is almost the same as if there were one ECS with a vector percept, reference, and error. More complex structures for the signals can also be considered, so that we can talk about the control of perceptions of arbitrary complexity.

In the hierarchic control system depicted in Figure 3, each of the control systems at any level of the hierarchy attempts to

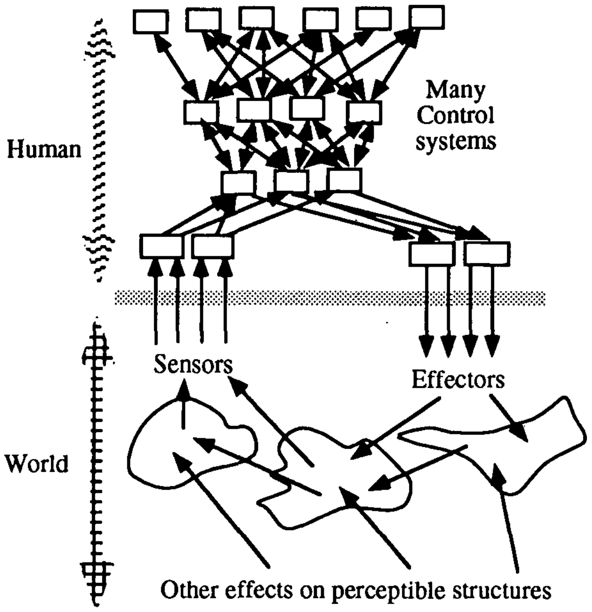

Figure 3. A generic view of hierarchic perceptual control structures by which the human uses behavior to control perceptions. Higher level control systems receive perceptual inputs based on lower-level perceptual patterns, and provide outputs corresponding to the differences between what they perceive and their reference percepts. These outputs represent references, or goals, for lower level control systems, the lowest of which control the effectors that act on the world, changing the percepts experienced by all the control systems.

act so that its percept agrees with its reference percept. The higher level control systems do this by adjusting the reference signals sent to lower level systems. The lower level control systems have very concrete percepts (and references, therefore) that depend directly on the signals impinging on the sensor organs. At higher levels, the percepts represent more abstract structures, not directly observable by the sense organs.

One can see a structured control system as a goal-seeking device. It works toward ever changing goals defined by its structured reference signals. For example, the reference percept for a pilot may be the perception of a safe landing at the destination airport. The achievement of that goal involves setting several other reference percepts for lower-level ECSs, such as the perception of passing a given sequence of waypoints, the perception of flying at a particular altitude, and so forth.

Communication

In the framework of either Perceptual Control Theory or Layered Protocol theory there is no structural difference between controlling an inanimate tool and communicating with an intelligent partner. A communicative partner is part of the perceptible outer world that is affected by behavior. But unlike a tool, an "intelligent" communicative partner has its, his, or her own control systems that perceive and act. We give the name "virtual messages" to communications between the high-level structures of the protagonist and corresponding ones of the partner.

We take communication to cover a wide range of interaction, from simple use of a tool at one pole of a continuum, to complex discussions such as philosophical argument at the other. We describe both sorts of interaction in the same way, using the same principles, but with great differences in the quantitative aspects of the interaction structure.

Whether the partner is a machine or a person, the protagonist performs some action that affects the partner's behavior in some detectable manner, as shown in Figure 4. A pilot may pull back on the stick, and the partner (the plane) begins to climb; or a philosopher may present an argument to which the partner responds with a counter-argument. In either case, the fundamental construct is a feedback loop. If the climb is too steep or not steep enough, the pilot changes the angle of the stick; the philosopher changes some part of the argument, augmenting, explaining, correcting in a way calculated to bring the partner to agreement. In either case, the protagonist behaves in such a way as to perceive a desired situation.

Using a different terminology, the protagonist has a perceptual goal that may be achieved through behavior. The success of the behavior in bringing the goal closer is monitored, and the behavior altered accordingly. If the achievement of the goal involves communicating with a partner, there is a communicative goal, which may be achieved through either dialogue or non-dialogue methods, depending on the protagonist's model of the partner and on other circumstances, such as whether the protagonist wants the partner to realize that the communication is happening, or whether the partner is even capable of engaging in dialogue.

There is a critical difference between using a tool and communicating with an intelligent partner. The tool has no goals of its own—no reference states that it uses to set desired percepts that might conflict with those desired by the protagonist. The tool is neither cooperative nor antagonistic, whereas a human communicative partner may be either, but is unlikely to be passively neutral in respect of the communication. The tool is reactive, whereas the person may be proactive. An "intelligent" machine falls somewhere between, in that it can be controlling sensory inputs to accord with references that are set independently of, and may be unknown to, the human user. A computer-controlled aircraft that has flight envelope limits built into its program may "deliberately" thwart the intentions of a pilot to perform some manoeuvre. Such a machine is the kind of partner that we consider in this paper.

Background to Layered Protocol theory

The principles we will be describing are part of the Layered Protocols (LP) theory of communication between "intelligent" entities. In this context, "intelligent" implies not cleverness so much as a degree of independence between the entities in three respects:

- independence of design,
- independence of sensing mechanism, and
- independence of action.

Independence of design means that neither partner can be sure precisely how the other will interpret any specific communication. Independence of sensing mechanism means that neither partner can know exactly what information the other has available. Independence of action means that neither partner can know at any moment all of what the other is doing. Independence of action lies at the heart of the link

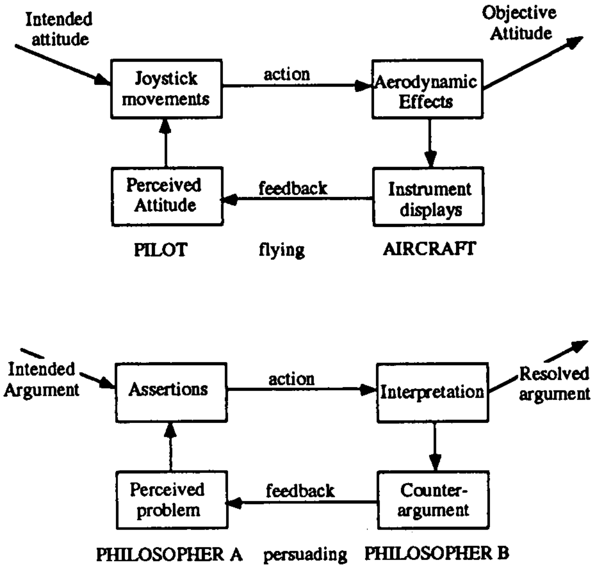

Figure 4. The formal similarity between controlling the attitude of an aircraft and communicating a persuasive argument. In either case, the important issue is whether the protagonist (on the left, in each part of the figure) can come to perceive that the effect on the partner is as desired.

between "intelligence" and independence, because it is the ability of an entity to perform actions that bear a useful relation to its circumstances that normally leads one to classify it as "intelligent."

Communications between a pilot and an aircraft largely fulfil the three independence criteria, and will increasingly do so as computerization of aircraft proceeds. The aircraft and the pilot clearly were not designed together; the aircraft has many sensing systems not directly accessible to the pilot, and vice-versa; and the aircraft can perform actions, such as manipulating information, that are not known in detail to the pilot. As a trivial example, when the aircraft is on autopilot, the pilot may not even be aware of some of the flight events it performs.

The independence criteria mean that neither communicating partner can be aware of the exact effect of any particular communication. Under these circumstances no message can be encoded with sufficient redundancy to guarantee error-free reception, in contrast to the classical situation in which the only communicative difficulty is noise in the communication channel and reception delay associated with error-correction coding. With independence, the receiving partner must provide the transmitter with feedback that helps each partner to believe that the intended message was the one received, and to correct the situation if it was not.

When feedback enters the situation, questions of stability arise. If the feedback is delayed, has too strong an influence on the forward channel, or changes too rapidly, the whole connection may be dominated by instability in the feedback loop, to the exclusion of the information that is supposed to be transmitted. The stability criteria for feedback systems would oppose those for rapid and effective information transmission, if a single error-correcting feedback loop were used (Taylor, 1989). But the effects of the opposition can be reduced by dividing the work among a hierarchy of feedback systems, each of which occur over successively longer time scales. In voice, for example, corrections can be made at the level of word ("what word was that?'), proposition ("You mean alter the map display?"), or higher-level constructs ("You mean to suggest that we will have very little fuel reserve?").

Before considering the Layered Protocol theory in more detail, let us discuss some considerations relating to the voice channel through which much human-human (and very little human-machine) communication is passed.

Voice in aircraft

What kind of material is suited for voice input? In human communication, voice is used to communicate subtle relationships among a wide range of concepts, a range much wider than can be accommodated by gesture or by any other means of communication, except possible gestural language such as American Sign Language. Only writing offers a comparable range of possibilities for communication, and it is restricted by its inability to convey nuances of affect.

In communicating with a machine, voice cannot be used in its "natural" function, for two reasons. Firstly, current and projected recognizers have neither the range of vocabulary nor the ability to deal with intonational modulation demanded by normal conversation; secondly, the machines with which we might wish to communicate do not have the intelligence to interpret the kind of subtleties that humans are accustomed to conveying by voice. In dealing with machines, humans must use voice in an unnatural way, restricting the vocabulary and relying on words by themselves to convey the intent of the communication. Indeed, voice recognition by machine has ordinarily been taken to be a problem in word recognition, and only occasionally until recently has there been any significant interest in speech understanding (e.g. Klatt, 1977). Why, then, should anyone want to use voice in interacting with machines?

Flying an aircraft is fundamentally a question of making it go where the pilot wants it to go, and perhaps to perform other manoeuvres while it is doing so. From moment to moment, going where the pilot wants is a problem in continuous control, but on a longer time scale the problem is discrete—the plane should arrive at this airport or that, maintain this altitude or that, pass through this waypoint or that. Voice is adequate for communicating discrete information, but not for continuous control. It follows, then, that the tasks for which voice should be used are not those in which the pilot needs continuous control. It also follows that the tasks that use voice must be integrated with those under continuous pilot control, and in an "intelligent" future aircraft with tasks controlled by largely autonomous subsystems.

Voice is not the only means by which the pilot can communicate discrete symbolic information to the aircraft. Voice must compete with pushbuttons, keyboards both soft and hard, and even possibly with voice communication to other humans. If there are few choices and very clean error-free communication is required, a pushbutton may be more appropriate than voice, provided that the pilot has a hand free to push it, and perhaps eyes free to see it pushed correctly.

One may not want to use voice when the timing of an event is crucial. Fingers are much better than voice for exactly timed events. It may be appropriate to arm a weapon by voice, but not to activate it. An error in arming can be retracted, but a gun cannot be unshot, or a bomb undropped. A weapon used at a time 200 msec from optimum might as well not have been used. Even in far future aircraft with excellent voice recognition capacities, it is unlikely that spatially discrete buttons and switches will be totally superseded by voice.

What, then, is voice good for? Primarily for strategic information, communication that is not time-critical, the control of other interactions and displays, and in general those things that involve selection among many discrete possibilities that can combine in different ways and in which errors are recoverable. Voice is particularly useful when the hands and eyes are otherwise occupied, and this is the main reason for wanting voice in the cockpit at all.

Voice recognition equipment has its limitations, especially in complex, rapidly changing situations such as are encountered in an aircraft cockpit. Natural human voice communication is less syntactically constrained than is written text, but recognizers for continuous speech usually demand adherence to a fairly rigid syntax as the price of recognizing an adequate vocabulary. In stressful situations, people may forget to use the correct highly constrained syntax. On the other hand, it has been reported that in emergency conditions when the pilot has a chance of landing safely, the language may revert to the correct form, perhaps because it takes less cognitive resources to use the highly trained forms than to invent new dialogue. Even in conditions of relatively low stress, such as normal air traffic control (ATC) interactions, pilots and controllers often go well beyond the bounds of the prescribed syntax, and this has presented significant problems to researchers attempting to use voice recognition as a component in training systems for ATC (F. Néel, Personal Communication, April 1990).

Speech recognizers tend to have trouble during conditions of stress, although the few existing studies of stressed voice show no consistent trends in the parameters of speech (C. Weintstein, Personal Communication, April 1992). It is probable that the stressed voice is more variable than the unstressed voice, and this causes recognition errors. But it is particularly in stressful conditions that the accuracy of the recognizer is most important.

People tend to stop talking while they deal with immediate control problems, so it is likely that voice is best used for tasks that can be postponed until immediate stresses have passed, or that involve preparation for later periods of stress. This will not necessarily be true if the use of voice is part of the trained behaviour to deal with the stress condition, but it otherwise imposes some restrictions on the machine side of the interface, because synchronization between voice and other controls may not be reliable.

Voice in the cockpit faces a hurdle not placed in the path of traditional means of communication between the pilot and his aircraft. Voice is psychologically required to be error-free, whereas keyboard entry for the same task is not. Why? Could it be that error detection and recovery is thought to be easier and quicker for keyboard entry than for voice? In fact, voice entry for numbers may well be more accurate than through a keyboard, even under office conditions in which keyboard entry is easy. Or is it that the pilot knows that the keyboard will accurately report the character corresponding to the pressed key, whether the key be right or wrong, whereas the voice recognizer will sometimes report the wrong word even when the pilot speaks the correct one? In military action, operators (pilots) must have confidence in their abilities and their equipment, whether or not either be warranted.

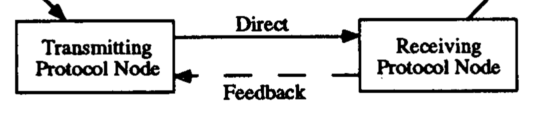

Figure 5. A Protocol converts information to be transmitted into a less abstract form that is communicated to the partner. The protocol may require feedback to ensure correct transmission, but the virtual message goes only one way, and is interpreted by the recipient at the same level of abstraction as the original information that was to be transmitted.

STRUCTURE OF A MULTIMODAL INTERFACE

To see where voice fits into a Layered Protocol structure, we must first deal with the general question of how the theory of Layered Protocols can be used to describe a complex interaction between "intelligent" partners. The principles are quite general. To apply them to specific situations, the details must be filled in, but the structure remains the same. The Layered Protocol structure we describe here is somewhat evolved from that presented by Taylor (1988a, b; 1989), but most of what was described there is still valid.

Support structure

A Layered Protocol interface has two important aspects: the protocols, and the layered support structures that link the protocols. Let us first briefly consider the support structure, before we deal in more detail with the nature of protocols. For now, assume that a protocol is a means whereby a chunk of information to be communicated is transformed into some less abstract form by the transmitting partner, and interpreted by the receiving partner, possibly after much feedback and correction, into a similarly abstract (though possibly different) form, as shown in Figure 5. The "chunk" might be the information whose transmission constitutes the original communicative goal of the transmitter, or it might be the result of a transformation performed by a higher-level protocol, as shown in Figure 6. In either case, the result of the transformation performed by the protocol is a "virtual message" that passes between the two parties. Only when the successive transformations have arrived at physical phenomena such as sound waves or photon streams can the messages be termed "real."

Virtual messages are realized by means of actions that may take different forms under different circumstances; the same message may be transmitted by voice on one occasion, by gesture on another, and on a third by a combination of the two. It does not matter which mode is used, provided that the result satisfies the protagonist's requirement to perceive that the virtual message was received. It is the beliefs or percepts of the originator, not the recipient, that determine the need for, the means of, and the success of, the communication.

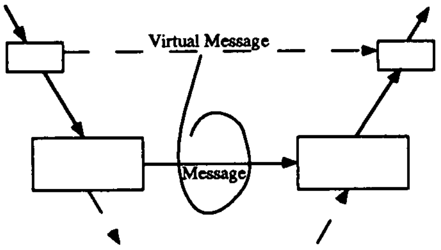

Figure 6. The information that is transmitted by a protocol may be the virtual message of a higher level protocol. The lower-level, less abstract, protocol is said to support the higher-level one. Feedback connections are important at both levels, but are not shown in this figure.

Those beliefs are most strongly affected by feedback from the recipient, and it is the highest level of feedback that matters most. If high-level control systems are satisfied, lower-level ones need not do anything. But higher-level systems are more likely to be satisfied if lower-level ones perform properly.

Sometimes information is more easily transmitted in one form, say verbal, and sometimes in another, say pictorial. Or perhaps part of a higher-level message may be transformed into words, and another part into pictures, as happens, for example, with labels on a map. In this situation, two protocols support the transmission of a single chunk of information, as in Figure 7. Such an arrangement is known as "diviplexing." The converse, multiplexing, also is common; one lower protocol supports two higher ones simultaneously. Multiple windows on a computer screen provide a well-known example. The screen is a single visual display channel, but different processes communicate with the user through it, distinguishing their outputs by locating them within different windows on the screen surface.

The whole structure of support within the network is an acyclic graph, in which converging and diverging threads of support allow multiple kinds of high-level messages to be communicated through a variety of physical media. Virtual messages are transformed and transformed again, into ever less abstract forms, before being physically transmitted to the partner, where they are reassembled and interpreted.

An important issue is how the partner can determine the interrelations of the partial messages being transmitted by the various different means. What signals how two diviplexed messages ought to be recombined, or in a multiplexed message how the components with different destinations should be identified? The necessary information must be in one of two places: in the message forms, or in specific prior knowledge of the receiving partner. If it is in the message form, then it is part of the protocol for that type of message, as, for example, in the protocol that whatever appears within the frame of one window on the screen must belong to a single task.

Protocol Structure

A protocol has a simple job: to take as a goal the transfer of some chunk of information from one partner to the other by transforming it into some less abstract form that the protocol is able to transmit. The protocol is executed by a pair of "protocol nodes," a transmitting node in the originating partner, and a receiving node in the recipient partner. The protocol may, but need not, take advantage of situation-specific information such as information about the partner, the task, the local state of the world, the recent and planned dialogue, and so forth. In its simplest form, the protocol may simply translate the originator's intent to pass a character, say, into the depression of a key on a keyboard, which is directly interpreted by the recipient (computer) as the transmission of the character. Similar simplicity would attend the transmission of a spoken word through a very reliable recognizer that gave no feedback (if such a thing were to exist); the user would speak the word (translating it into a pattern of acoustic waves), and the recognizer would reconvert it into a computer representation of the same word with no further ado.

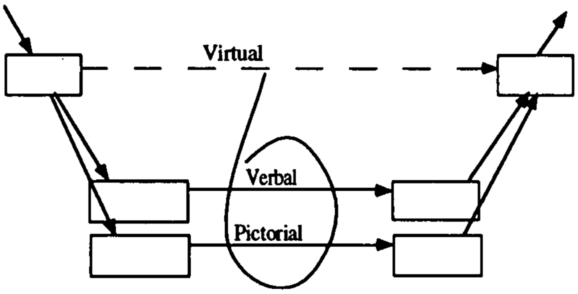

Figure 7. It is possible for a higher-level protocol to be supported by more than one lower-level protocol. This arrangement is called "diviplexing." At the higher level, information may be transmitted in the form of a map, for example, which at a lower level is transmitted partly in pictorial and partly in verbal form. Feedback connections are omitted for clarity in this figure.

Such error-free transmission is rare, and perhaps impossible even between partners that lack the three independences (Elias, 1953). People do mistype, hitting the wrong key even though they know which one they intend to strike; speech recognizers do make errors, even though the user speaks the correct word clearly. In either case, if the errors are not detected and corrected, higher-level protocols must be designed to accept the erroneous keystroke or word, and to perform acceptably in spite of the error. The options, then, are to correct errors before they are taken as truth by higher-level protocols, or to accept that errors will be made, and to ensure that the higher-level protocols are flexible enough to handle them.

People do not normally speak with the expectation that their partner will query each dubious word. They expect that the partner will try to make sense of what they are saying, despite the fact that many words are slurred, shortened, or even omitted. Think of the typical American pronunciation of the word "President," as in "Pres'n Bush." The word "and" usually is pronounced as "n" if it is not acoustically omitted entirely. Such "flaws" at the word level cause no difficulty to a fluent English speaker listening to them, though they may well make it impossible for a less fluent listener to understand the speech at all. The internal redundancies in the structure of the sentences allow the missing or deficient words to be reconstituted with almost total accuracy. Listeners rarely are even aware that the reconstruction has occurred, unless they listen very critically. Word-by-word feedback of what the recognizer has recognized is therefore not likely to be part of a good set of protocols for dealing with voice input, unless the recognizer performance is very poor.

A second conclusion about these simple protocols without feedback, implicit in the foregoing discussion, is that the results are almost never used directly, but are part of a larger message. It is true that in a simple tool-using protocol the push of a button is the signal for something to happen, often independently of any prior tool manipulation, but even there, the effective action is often part of a larger sequence. Feedback can be omitted if and only if (a) the message is with high probability going to be properly interpreted, and (b) the message will be incorporated into a higher-level message that does have feedback, or will be used in an observable and correctable way to affect the outer world.

What is the point of feedback? To understand this, we must go further into the whole process of action and communication.

Feedback

A protocol node has functions quite analogous to those of an ECS, as shown in Figure 8. The protocol node in the originator has a goal that the recipient's receiving node should have some chunk of information, and a reference belief about what information the partner has; it performs an action in the form of a virtual message intended to affect the partner's belief in the desired direction, and acquires feedback that allows it to modify its beliefs and perhaps alter the way it presents the virtual message. Parallel relationbships hold for the receiving node.

The key to the action of the transmitting protocol node is in the relation between the Model on the one hand and the Coder and Decoder on the other. The Model is the analogue to the comparator of the ECS. It compares the chunk of information to be sent with that believed to be known to the recipient. How complex is the comparison that leads to the output message? At very low levels, the Model is vestigial, and the Coder and Decoder do all the work. By design, a low level protocol node always assumes that the partner does not have the information, so the virtual message carries it. For example, if a keyboard command requires the letter "p," the user presses the "p" key. All of the behavior of such a low-level protocol node is determined by the Coder (or, in a receiving node, by the Decoder).

In contrast, at very high levels, almost all the work is done in the Model, and the Coder and Decoder are relatively simple. At these levels, the partner probably already knows much of the information that the originator wishes to get across, or at least has analogous information that can be used to assist in the encoding process.

The question of what the partner knows or ought to know is crucial in determining the need for feedback. From the viewpoint of the recipient, if the originator probably believes that the message has been successfully received, there is no need to provide feedback. On the other hand, if the recipient believes that originator quite probably is uncertain what message was received, then informative feedback is required. As Pask (e.g. Pask, 1980) points out, informative feedback should normally not be a mere echo of the overt content of the message. Rather, it should be some reinterpretation of the message in a different form that the originator can recognize as having the same intention.

For voice, the worst feedback probably is a vocal echo, which is subject to a set of error probabilities very like those that applied to the original vocal message, so that the speaker may well misinterpret the echo of a wrong recognition as being correct. What kind of feedback is best for voice will depend on the situation and the probability of error in the voice recognition equipment. It might even be a vocal paraphrase of the spoken material, but is more likely to be an entirely different kind of presentation.

As an example of appropriate feedback, consider the use of voice to control a map display of variable scale that shows the region centred on the aircraft, such as in the experimental BAC-111 flown by RAE Bedford. In this aircraft, the pilot can issue a command something like "Map range fifty miles" to set the range of the outer ring of the map. The experimental recognition system incorporates a visual display that shows the words recognized, but this display is ordinarily used only by pilots for whom recognition performance is bad (Moderator's comment in Taylor, 1987). A pilot for whom recognition is normally good discovers whether a particular recognition was correct through the action of the map display, which may change to the wrong range, if, for example, "fifty" is mistaken for "fifteen." In such a case, the pilot merely corrects the effect of the message by using a new command, perhaps repeating the original. The error would not be repeated if the system were designed to make the pragmatic assumption that a "map range" command is intended to change the range.

The primary reason for introducing voice into the cockpit is supposed to be to reduce the load on the pilot's resources, not to give him more capability. After all, there is very little that voice can do that cannot be done tactually. However, keying a number can distract the pilot from flying the aircraft more than does saying the number. In low-level flight, it is important that the pilot fly accurately, and voice is therefore a desirable means of entering numbers, provided the numbers are usually recognized correctly. But if the pilot is not assured that the number will be assigned to the proper function, the use of voice to enter it will not be very helpful.

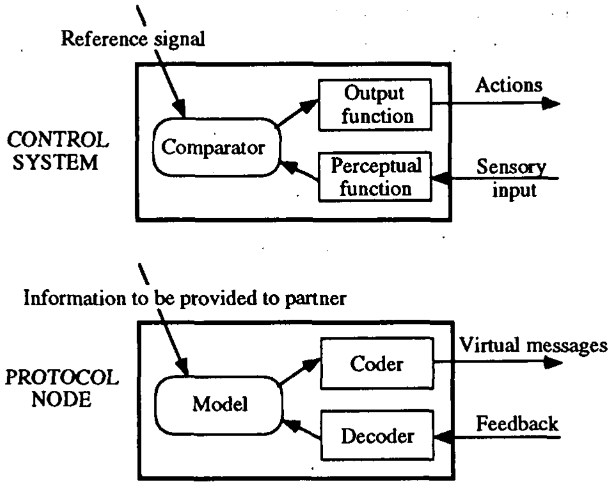

Figure 8. Analogy between an elementary control system and a protocol node. In the control system, incoming perceptual data are evaluated and compared with some reference that represents the desired percept. The difference between the two results in some virtual actions that are actually performed by lower-level control systems. In the protocol node, the information that the partner should have is compared with a model of what the partner does have, the difference resulting in a virtual message implemented by lower supporting protocols. The model is affected by the resulting feedback, which may result in further virtual messages.

An error in the command funtion is likely to be more serious and hard to correct than an error in a numeric parameter, because it may be hard to detect, and may cause unwanted actions to occur. It would be unlikely to be corrected by the pilot without a substantial period of confusion, annoyance, and expenditure of mental resources. Accordingly, there might be a need for feedback, if there is a significant probability that the command itself might be misrecognized or misunderstood. Alternatively, the command might be entered tactually, with voice to provide the argument to the command.

Misrecognition is not the same as misunderstanding. It is possible to understand a message at a high level while at a lower level misrecognizing some of its words.

There are two ways in which the structure of a high-level message can be used to increase the probability of its being correctly understood: the probabilities for different low-level structures such as words may be changed according to higher-level expectations so that they are likely to be correctly recognized, or misrecognized words that do not fit into the high-level structure can be changed for confusable ones that do fit.

The structure of a high-level message involves everything that the higher-level protocol can use, including the situational context. For example, at a particular stage in a flight, the pilot may normally switch the map display to the more local environment. When a flight reaches this stage, the probabilities for the appropriate command may be changed within the recognizer itself, or, if the recognizer produces a nonsensical word string that might easily have been derived from a sensible one that performs the expected command, the revision might be understood to have been spoken, as suggested in Figure 9. Typically, in such a situation, the higher-level protocol would determine that the message was ambiguous or problematic, and would ask for a check of its putative understanding, or for a correction. This feedback might well be vocal.

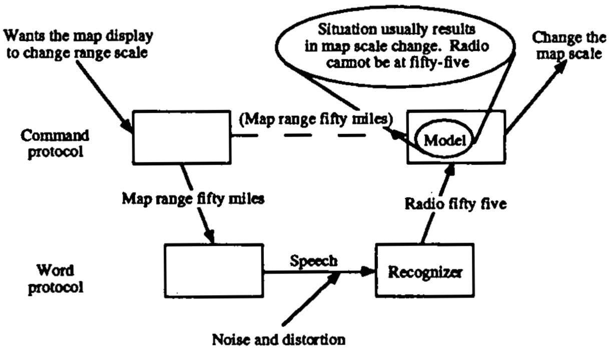

Figure 9. One way that correct understanding can come from misrecognition. The mission state normally calls for a change in the map display. The pilot speaks a command for one, but because of noise and distortion, the recognizer provides a string that would cause an illegal operation on the radio. Because there is a known possibility of this kind of recognition, plus a much higher probability for the map command than for the illegal radio command, the high-level protocol interprets the command as the one that was really intended. Naturally, this interpretation could be checked by further feedback, such as "Did you mean map display?"

In order to discuss the possibilities for feedback, a digression is needed on the General Protocol Grammar (GPG) that we believe underlies all dialogue. Once the GPG is understood, issues of how to choose appropriate feedback may become clearer.

The General Protocol Grammar

Many authors argue that the idea of a grammar for dialogue is absurd, and that there cannot be such a thing, because dialogue is so varied in different situations (e.g. Good, 1989). We agree, but we argue that there can nevertheless be a grammar that describes the interactions within a protocol. In fact, we argue that every protocol incorporates such a grammar, and that it is the same grammar for every protocol. For this reason, we call it the General Protocol Grammar (GPG).

We will describe the GPG in two stages. First, we treat it as a standard node-and-arc state transition grammar, as shown in Figure 10, but only for expository convenience and mnemonic assistance. A state transition grammar works under the assumption that a state is occupied until a discrete transition to a new state is made, and then that new state is occupied. In dialogue, what matters is information transmission, which does not happen instantaneously.

In the more exact approach to the GPG, the focus is on the effect of information transmission on various belief states. Changes in these belief states correspond to state transitions in the node-and-arc grammar.

The term "belief" should be taken broadly. Of course, the aircraft has no beliefs, and makes no overt deductions. It just behaves. The aircraft designer has built the contingencies in, and the aircraft has no possibility of changing them. In the Layered Protocol theory, this kind of "built-in" behavior is a property of the Coders and Decoders, whereas beliefs and deductions are among the properties of the Models. An outside observer, however, can rarely tell the difference. All the same, it seems reasonable to work as if the lower protocols tend more to "just behave," while the higher ones are more susceptible to analytic modelling that justifies the terms "believe" and "deduce." In both cases, we can use the same symbolism for the analysis.

GPG as a state transition network

A message is initiated within a protocol when the originator (O) has a chunk of information that the recipient (R) should have. The chunk may be a part of a higher-level message supported by this protocol, or it may result from some task goal on the part of O. In either case, we call it the "primal message" that the protocol is being asked to transmit. It is the job of the transmitting protocol node to determine how to transmit it, given the beliefs it has about the current situation, the history of the dialogue, the recipient's beliefs, and so forth.

At one extreme, O (meaning O's transmitting protocol node) may believe that R (the corresponding receiving protocol node) already has the information, and will therefore transmit nothing. For example, if the goal is that the plane should fly straight and level, and it already appears to be doing so, then the "joystick protocol" should do nothing.

At the other extreme, O may believe R to know none of the message. This might be the situation associated with the entry of a flight plan into a mission database. All the information might have to be installed explicitly. As an intermediate case, the mission plan might be of a kind partly known to the aircraft, with only some elements to be explicitly transmitted, such as waypoints or regions of potential threat such as SAM sites.

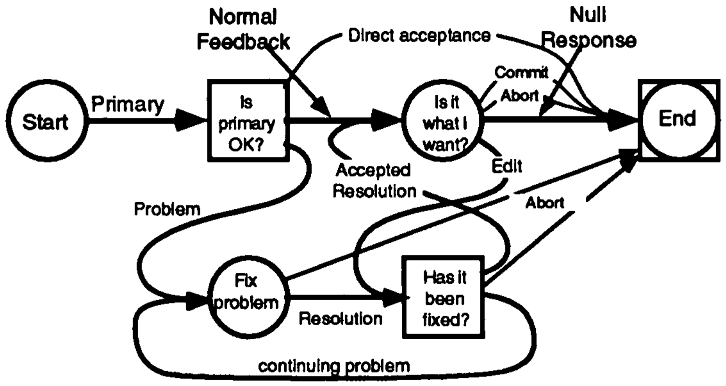

Figure 10. A sketch of the General Protocol Grammar, showing the major arcs. Circles represent stages at which the originator of the primary message must do something, squares stages at which the recipient of the primary must do something.

In the initial state of the grammar, O has some belief about what information actually needs to be transmitted, but how it is to be transmitted may be a matter for choice, especially at higher protocol levels. Perhaps O attempts to encode the whole message and transmits the encoding all at once; perhaps O transmits only enough that R is alerted to the fact that a message is being sent. In either case, what is sent is the "Primary" in Figure 10, and on receiving it R must determine whether it could represent the entire primal message. If it could, then R provides whatever feedback is necessary to get O to move to the node marked "Is it what I want." If not, R provides feedback that indicates to O that there is a problem.

Consider just these two possibilities in the case in which R's part of the dialogue is taken by an isolated word speech recognizer running with a wide open syntax. O speaks a word, and R, the recognizer, produces both a best candidate recognition and a "goodness" measure. If the recognition has a high goodness measure, then R should "want" O to move to the "Is it what I want" node at which O might accept the recognition as correct. Otherwise, R would want O to move to the "Problem" node at which O could solve the problem. Clearly, these two situations demand distinct feedback so that O can determine which possibility R intends. But most recognizers that provide word-by-word feedback do not indicate any difference between high-probability and low-probability recognitions. O can find the problematic ones only by monitoring each word fed back. If the errors are few, as they often are for good recognizers, O is likely not to detect them. But almost certainly, errors will be more probable for the recognitions given a low goodness measure by the recognizer than for those given a high goodness measure. There may be occasional errors even in the recognitions "believed" by the recognizer to be good, but it will be hard for the user to detect them among all the good ones. They are probably best passed through to a higher protocol for detection and correction.

This analysis suggests the reason for a practice that is commonly found useful: to feed back only those words with a low goodness measure, and to accept the rest silently, passing them up to a higher level as if they were correct. O can then readily detect R's use of the "Problem" arc in the grammar, and can make appropriate corrective or accepting responses. An additional benefit is that R should not expect any response to the well recognized words, but should expect a confirmatory or corrective message about each potential misrecognition. O is thereby relieved of the need to identify for R which word is being corrected, as O would have to do if R reported each word. There is a double benefit for the resource loading on O: the monitoring of feedback for incorrect words is eased, and O has less to do to correct a false recognition.

In the GPG, the different arcs may be instantiated in different ways, depending on what the partners believe each other to believe. For example, suppose that R (in any protocol, not just word recognition) believes that the primary message has provided enough information to permit a satisfactory interpretation of the primal message. R wishes O to move to the node "Is it what I want" in Figure 10. If R believes that O will be sure that R did get a good message and would interpret it properly, then no overt feedback is necessary. We call this a "null instantiation" of the Normal Feedback arc. A second possibility is that R believes that O does not know whether R has made a good interpretation, but would trust R's judgment in the matter. All R need do is to indicate that the message was received. We call this a "neutral instantiation" of the Normal Feedback arc. There are at least two other instantiations for this arc, "informative" and "corrective"; most arcs in the GPG have more than one instantiation.

The state transition description of the GPG is in principle adequate for many purposes. In its full form, there are many more arcs than are shown in Figure 10—about 24 arcs in one recent version, with an average of about two instantiations for each arc. But a state transition diagram is inadequate for real communication, in which partial interpretations are continuously being made at all protocol levels. To handle this situation, we must consider the models and beliefs that underlie the state transition description of the GPG.

Belief structure of the GPG

The GPG is a description of the process of communicating a virtual message. It is not concerned with the content or level of abstraction of the message. Accordingly, the information that directs the state transitions is concerned not with the message itself, but with the satisfaction of the partners about its transmission.

We identify three propositions dealing with the message transmission:

- P1: R has made an interpretation of the message.
- P2: If P1, then R's interpretation is satisfactory.
- P3: It is not worthwhile to continue trying to transmit the message.

These three propositions are statements of fact. Facts are unknowable, but anyone is entitled to believe them with any degree of certainty from strong disbelief through total ignorance to implicit faith. We assign the degree of belief that someone has about one of these facts a number from -1 to +1. We notate the degree of belief A holds about P as $A(b,P)$, or simply $A(P)$, omitting the b because by default we are dealing with belief. On other occasions we use a similar notion for the strengths of goals and intentions, using g or i, which are ordinarily not omitted in the notation.

The final element of notation is to divide the degrees of belief into five categories, as a matter of convenience rather for any reason of theory: Disbelief (D), Negative belief (N), ignorance (X), weak belief (W), and Strong belief (S). The letters correspond to numerical ranges that are not well specified. The boundaries between the ranges occur at points where the behaviour of one or other partner may change.

Given this notation, we can write statements such as $W<A(P)$, which, by convention, means that A holds at least a weak belief in P, or $S=A(b,W<B(P))$, which means that A strongly believes that B at least weakly believes P. $S=A(g,W<B(P))$ means that A holds a strong goal that B should at least weakly believe P.

Now consider the three propositions of the GPG. So long as a message is still being transmitted, each partner must at least weakly believe that P3 is false:

- $W<O(b,\neg P3)$ & $W<R(b,\neg P3)$

Implicit in this is that at least one of the partners believes that either P1 or P2 is false and that each has a goal that both P1 and P2 should become true.

This notation may seem a little removed from the problem of flying an aircraft, but let us apply it to the simple matter of adjusting the climb angle, using the joystick. We shall call the pilot C (commander) and the aircraft A.

The design of the aircraft suggests that both A and C "believe" that P2 is always true (in the absence of malfunction). In other words, if the pilot manipulates the stick appropriately, the aircraft will assume the correct attitude. But the variety of flight situations assure that simply setting the joystick to some condition for a fixed duration will not work. The pilot must receive feedback from the aircraft as to its present (and perhaps predicted) attitude. This is normally done through an instrument display in current aircraft, though in earlier days the pilot had to look outside. We can take the instrument display as feedback for the joystick message.

In the state transition diagram of the GPG, the behavior is only crudely described. The pilot makes an initial adjustment ("primary"), the plane assumes some attitude ("Normal Feedback"), the pilot determines whether it is the desired attitude, and if not, uses the "Edit" arc followed by "Accepted Resolution" as the plane assumes a new attitude. The "Edit" loop is continued until the attitude is correct.

Such discrete control is a very poor way to handle an aircraft. Effective control is continuous, the pilot judging the attitude and the rate of change of attitude, and manipulating the joystick continuously to achieve as quickly as possible the perception that the attitude is correct. The protocol feedback loop and the types of message involved are shown in Figure 11.

In terms of the belief structures, the pilot starts with a strong belief that the aircraft has not interpreted his intention that it take some particular attitude

- $S=C(\neg P1)$

The aircraft has no belief, as the word is ordinarily used in English, since models are not built-in to most aircraft, but a future computer-controlled aircraft might well develop some beliefs. In fact, one might consider that the Mulhouse A-320 crash as being due to a belief held by the aircraft that differed from the intentions of the pilot about the future path of the plane. Whether or not this is the case, one can in the formalism assert that by design, if the pilot moves the joystick the aircraft "believes" that it holds the wrong attitude. Hence, the joystick movement leads to:

- $S=A(P1 \,\&\, \neg P2)$

It has interpreted (and acted upon) the message (joystick movement) but "believes" its interpretation is incorrect because the joystick continues to be moved.

As the aircraft's attitude approaches the desired one, the pilots's belief structure changes, so that strong belief in the wrongness of the attitude gives way to weak belief, indifference, and disbelief. At that point, the joystick is neutralized, so that the aircraft can now "believe"

- $S=A(W<C(P3))$

and hence

- $S=A(W<C(P1 \,\&\, P2))$.

The aircraft "deduces" from the pilot's cessation of moving the joystick that the pilot is satisfied with its interpretation of

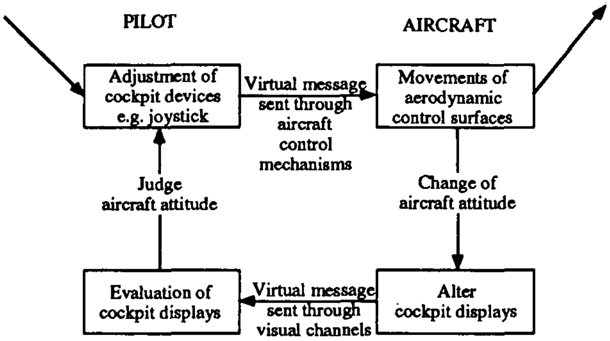

Figure 11. The protocol loop for the control of aircraft attitude. The pilot sends a continuous stream of messages by manipulating the joystick and other controls, which are sent by means of whatever control mechanisms the aircraft provides (cables, hydraulics, computer-driven servos) to the aerodynamic control surfaces. The aircraft signals the interpretation of this message by means of changes to the cockpit displays.

the message, which is to say that he is satisfied with its current attitude.

It is important to note that the changes in belief are continuous, and occur in parallel with the actions of both partners—the pilot and the aircraft. It is this continuously developing and parallel communication that cannot be handled by the state transition description of the GPG. Neither can it be handled by a turn-taking analysis of dialogue. It is, however, characteristic of most realistic communication, whether among people or between people and their "intelligent" machines.

Voice and belief structure

When we come to consider voice interaction, the concepts of "belief" and "deduction" become more plausible, since voice recognition equipment is normally based on probabilistic models. The recognizer starts with some set of expectations about the possible things the user might say, and these expectations often include probabilities that might well be situation-dependent. In the LP structure, the probabilities will depend also on the momentary state of interpretation of the higher-level message that it supports.

Let us suppose that the higher level message is a flight plan consisting of a set of waypoints identified by name from the hypothetical map in Figure 12. Let us also suppose that the aircraft is "intelligent" in that it has a database of locations and some concept of what a normal flight plan might look like. In particular, a normal flight plan does not double back on itself. Now the pilot starts to enter waypoints: Astal, Birland, Demick, Coltaine, Endow.

Suppose the pilot makes a mistake and enters Demick before Coltaine. The recognizer correctly identifies the words from the acoustic signal. The protocol at the next level translates these words into coordinate strings that will be entered into the navigation system. But the Model in this protocol could incorporate the knowledge that normal routings do not include consecutive turns near 180°, and this knowledge would allow it to propose that the message is possibly incorrect. In the state transition grammar, the aircraft would use the "Problem" arc in Figure 10. In the belief notation,

- $W<A(P1 \,\&\, \neg P2 \,\&\, W<C(P1 \,\&\, P2))$

which indicates a difference between its view of the correctness of the message and the view it believes the pilot has. The result should therefore be to inform the pilot not only that it finds the message dubious, but also in what way it finds it dubious. Accordingly, its feedback should be to propose a reasonable reinterpretation: "Do you mean Birland, Coltaine, Demick?" The pilot, of course, might have intended the zigzag route that was originally entered, and could say so, by voice.

None of the above indicates the manner in which the feedback is provided. There are some criteria that could help to determine that; the method must alert the pilot that the feedback exists, and it must be able to convey the information as to what the problem might be. It cannot, therefore, be provided by a non-obvious visual display, unless the probability of error is high enough that the pilot will normally look to see whether one had occurred. If such is the case, the same display could support feedback from both the word recognizer protocol and the waypoint construction protocol.

Integration

The idea of "integration" is closely bound with that of diviplexing, but the integration of voice with non-vocal interaction between pilot and aircraft usually does not involve diviplexing at any low level. The voice usually replaces rather than supplements the non-vocal interactions, though it could affect the interpretation the aircraft places on non-vocal actions. For example, a navigation display might be presented on a touch-sensitive screen, and the area displayed changed under voice control. Likewise, normally the non-vocal interactions do not supplement the voice. Each acts essentially independently, except that they both affect the behavior of the aircraft.

Consider the waypoint entry example illustrated in Figure 12. Suppose the plane were on the leg Coltaine-Demick and the pilot spoke a command such as "Next waypoint Birland." The flight situation would allow the waypoint protocol Model to determine that this would entail a turn near 180°, and might suggest the possibility of a misinterpretation that should be queried. This query would be independent of whether any waypoints had previously been entered by voice, but would be based on the present situation of the aircraft.

Another way in which the different modes of interaction relate to each other is that each imposes a load on the pilot's resources. The pilot must be kept aware of the local situation, and alerted to possible dangers and opportunities, but to provide too great a flood of information would risk the pilot ignoring important events. The problem of evaluating situation awareness and workload is a long-standing one in psychology, and no solution is near. Nevertheless, qualitative analyses could easily assist the interface designer in determining how and when to provide feedback or to request pilot action.

If a particular interaction would benefit from joint use of the symbolic information transmitted by voice and the continuous information transmitted by control movements, the structure of the messages that immediately support the high-level message must contain the information that allows them to be

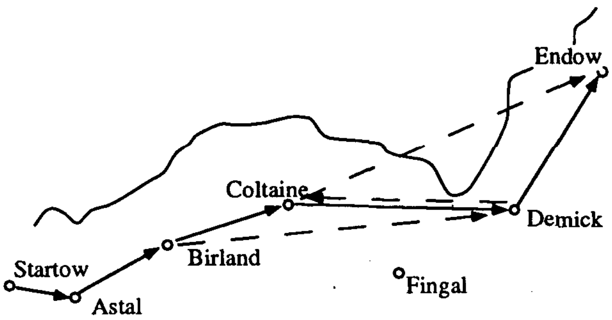

Figure 12. A hypothetical map with some named waypoints. The natural route from Startow to Endow follows the solid arrows. Erroneously ordered input could lead to the route shown by dashed arrows.

recombined. For example, a map display contains labelled entities that could well have descriptive information available in a database. The geographic spatial information is well displayed pictorially, and the items can be selected verbally, but the notion of "SAM sites north-east of Demick" involves a linkage between the verbal and the visual, through the coordinate location of Demick, which is at once symbolic and selected from a continuum of possible locations.

CONCLUSION

A basic principle of behavior is that purposive actions can be performed only with respect to some percept that the actions will affect, based on sensory data. A natural corollary, which leads to Perceptual Control Theory (PCT), is that actions are normally such as to make the percept closer to some desired state, or reference percept. Behavior thus controls perception.

PCT contains the concept of an Elementary Control System, a structure that accepts sensory input from a variety of sources, combines them into a percept through some perceptual function, compares the percept with a reference to create an error signal, and distributes some function of the error signal to effectors. Both the sources of sensory data and the effectors may well be themselves Elementary Control Systems at a lower level in a hierarchy, so that the reference signal in any Elementary Control System is a combination of the error signals from higher ones. The perceptual control system as a whole consists of many layers of Elementary Control Systems, the lowest ones acting directly on, or receiving data directly from, the physical world.

The theory of Layered Protocols asserts that a similar structure describes the interactions among "intelligent" entities, which are entities that have three independences: independence of design, independence of sensing mechanism, and independence of action. Messages are passed by means of creating effects in the physical world, but the physical messages represent messages of a hierarchy of levels of abstraction. The passage of a message at any level of abstraction uses a specific protocol, which includes a general protocol grammar (GPG) that determines the kinds of feedback that are appropriate to different conditions in the sending of the message.

The GPG can be applied to the manual control of an aircraft, as well as to both simple and complex voice interaction. The problem of integrating voice with other control mechanisms thus becomes partly one of determining which kinds of message suit the symbolist character of voice, and which suit the continuous character of manual control. In aircraft, the second issue of integration is less important: how to ensure that a high-level message transmitted by voice and manual control is properly reintegrated by the aircraft. Mission situation can be effective in determining the interpretation of high-level messages supported in part by voice, even when the recognizer correctly reports words erroneously spoken by the pilot.

Layered Protocols provides a principled framework for describing and integrating messages of different kinds, especially in complex interfaces such as the aircraft cockpit.

NOTE ADDED AFTER THE SYMPOSIUM:

Graceful control of automated systems

On the first day of the workshop, several talkers made the point that pilots had a difficult time accepting automated functions beyond the most simple, although they indicated in questionnaires that they wanted them. What the pilots did not want was for the automated functions to take decisions at critical moments that they would rather take for themselves, although the automated function could perform non-critical duties. This problem seems to be readily addressed within the PCT framework.

Both the plane and the pilot are conceived as hierarchic control systems, the plane's upper-level references being set either by the designer or by the pilot. For examples, the pilot gives the autopilot a reference to keep the plane on a certain heading at a certain altitude and with a stable attitude regardless of winds. The pilot resets this reference from time to time, or she might set a whole sequence of waypoints from which the plane computes references for the autopilot. In either case, if we think of the pilot and plane as one single hierarchic control system, the plane's chunk simply takes over the function of performing the task of satisfying the references provided by the pilot.

If the pilot has delegated control, the plane taking over the particular function, the pilot tends to lose "situation awareness" in respect of that function. He could control the function if he wanted to, but since he is not, neither is he acquiring the sensory information that would allow him to get the perception that he could be controlling. He may not perceive what is going on. The pilot's re-acquisition of situation awareness when retaking control of an automated function is a significant problem. To continue the autopilot example, the autopilot is switched out of the control loop, and the pilot's own lower-level control systems take over the maintenance of heading, altitude, and attitude. One significant case in which this might happen is collision avoidance, where situation awareness is critical.

The view of the automated function being switched in or out of the loop in alternation with the equivalent part of the pilot's hierarchy, as in Figure 13, is almost inevitable with conven-

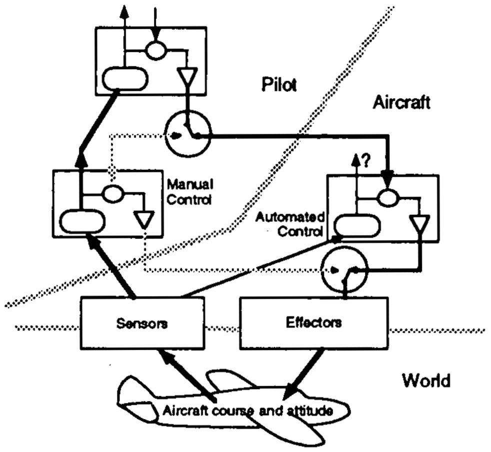

Figure 13. Conventional switched mode of operation of automated functions. When the aircraft is controlling a function, the pilot is not.

tional approaches to the problem. But PCT offers a different solution. Imagine that instead of a simple switch that sends a reference signal either to part of the plane's hierarchy or to part of the pilot's, the reference signal is sent always to both, as in Figure 14. If the pilot is choosing not to control, the gain in her part of the loop is zero. The gain in the aircraft's part of the loop is adequate to maintain course against external disturbances.

But it is possible for the pilot to set his gain to some low value other than zero, and "shadow" the aircraft's control. The aircraft could sense this in two ways. One way is that the pilot's attempts to control might set up a conflict in the lower-level systems that actually drive the plane's control surfaces. The result would be a persistent failure of the automated system to achieve the desired percept, if the pilot's references differed from those of the plane. The second is that in contrast to ordinary disturbances, the pilot's actions can be directly sensed by the plane, as shown by the dashed arrow in Figure 14. So long as the pilot's gain remains low, the automated system would keep its own gain high, but as soon as the pilot's gain increased, the plane would drop the gain of the automated system, perhaps to zero. The pilot is, at low gain, maintaining situation awareness, or regaining it preparatory to taking control.

There is a continuum, as the plane's gain decreases, between the plane performing the function, assisting (and perhaps training) the pilot to perform it, and getting out of the way to let the pilot do what she wants. There is no need for the pilot to switch automated functions in and out; they are in by default, but as soon as the pilot starts controlling what they control, they gracefully get out of the way.

What the pilot can switch in or out, or alter in a continuous way, is the sensitivity of the plane to the pilot's insistence on control. A novice pilot could set a high level, asking the plane to do what it thinks proper even though she requires it moderately strongly to do something else, whereas an expert would want it to get out of the way as soon as she started controlling.

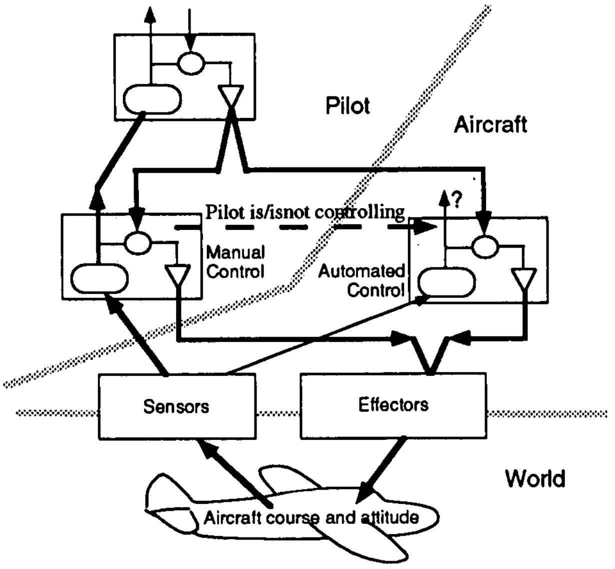

Figure14. A PCT-based view of conjoint manual and automated control. The automated system maintains control until it senses that the pilot is controlling the "same" percept, when it reduces its gain or even stops entirely, until the pilot ceases attempting to control that percept. At low gain, the automated system can assist or train a novice pilot.

Shifts of control locus need affect only a small part of the control hierarchy. The pilot's choice to control the course of the plane does not indicate that he must control the positions of individual control surfaces. And the plane can know at what level the pilot does desire to take control. The pilot may, for example, take quick evasive action (controlling the momentary attitude and course of the plane) while leaving the automated systems in charge of the control surfaces and the attainment of the waypoints that define the larger course of the plane.

REFERENCES

Elias, P. (1953) Coding for noisy channels. IRE Convention Record, 37-46.

Good, D. A. (1989) The viability of conversational grammars. In *The Structure of Multimodal Dialogue*, M. M. Taylor, F. Néel, and D. G. Bouwhuis (Eds.), Amsterdam: Elsevier North-Holland.

Klatt, D. H. (1977) Review pf the ARPA Speech Understanding Project. *J. Acoustical Society of America*, 62, 1345-1366.

Landauer, T. K. (1991) Let's get real: a position paper on the role of cognitive psychology in the design of humanly useful and usable systems. In *Designing interaction—Psychology at the Human-Computer Interface*. J. M. Carroll (Ed.), Cambridge, UK:Cambridge University Press.

Lippman, R. P. (1987) An introduction to computing with neural nets," IEEE ASSP Magazine, April, 4-22.

Pask, G. (1980) Developments in conversation theory—Part 1. *International Journal of Man-machine studies*, 13, 357—411.

Powers, W. T., (1973) *Behavior—the control of perception.* Chicago: Aldine.

Taylor, M. M. (1987) Layered protocols in voice interaction with computers. In *Information management and decision making in advanced airborne weapons systems*, AGARD Aeromedical panel CP-414.

Taylor, M. M. (1988a). Layered Protocols for computer-human dialogue. I: Principles. *International Journal of Man-Machine Studies*,.28, 175-218.

Taylor, M. M. (1988b). Layered Protocols for computer-human dialogue. II: Some practical issues. *International Journal of Man-Machine Studies*, 28, 219-257.

Taylor, M. M. (1989). Response timing in Layered Protocols: a cybernetic view of natural dialogue. In *The structure of multimodal dialogue*, (M.M. Taylor, F. Néel and D.G.Bouwhuis, Eds.), Amsterdam:Elsevier Science Publishers (North Holland).

Vallacher, R. R. and Wegner, D. M. (1987) What do people think they're doing? Action identification and human behavior. *Psychological Review*, 94, 3-15

NOTE

This paper owes much to discussions over several years within NATO AC243 (Defence Research Group) Panel 3 Research Study Group 10 on Automatic Speech Processing, and to electronic interactions with W. T. Powers and other participants in the Internet mailing list CSG-L.

Discussion

QUESTION R.M. TAYLOR

Your description of the application of perceptual control theory to the control of the aircraft's state seems clear and useful. However, the designer's main task in the future will be to help the pilot to solve mission problems. Can you forsee any changes to perceptual control theory that will be necessary in order to apply it to solving mission tactical problems and pilot tactical decision-making?

REPLY

Perceptual Control Theory (Powers, 1973) has many different kinds of abstraction - eleven at the current state of the theory, including categories, sequences, programs, principles, among others. Mission problems seem related to programs and sequences, which are components of current theory, so to that extent no changes are necessary. On the other hand, the precise behaviour of non-linear control systems is not well understood and neither is the behaviour of neural network meshes, which form a part of the control network. Hence the theory is likely to provide a framework for productive analysis and design rather than a precise predictive model. The key point is to realise that the stabilities of behaviour relate to the intentions of the actor and not to the actions taken to realize those intentions.