Sociology in machines:

applying sociology to software design

Nigel Gilbert *et al*

Social and Computer Sciences Research Group
University of Surrey
Guildford GU2 5XH
UK
(email: gng@uk.ac.surrey.soc)

September 16, 1990

This is a revised version of a paper originally presented to the workshop on 'Social Perspectives of Software', Oxford, January 13-14th 1990.

[Steve: I don’t suppose you want an abstract for a chapter of a book; but just in case you do, I’ve left this (amended) abstract in]

Abstract

Applied sociology is usually taken to mean the application of sociological theories and findings to issues of social policy. In this paper, we present some examples of a quite different notion of applied sociology: one in which sociological insights are applied to the design of computer software. Sociology’s primary concern as a discipline can be considered to be the study of the interaction of agents, where ‘agents’ are generally assumed to be people or human collectivities. The kind of application of sociology which we discuss trades on the possibility of widening the scope of agency to include non-animate systems.

The success of this kind of applied sociology will depend in part on the degree to which artifacts can be considered to be agents susceptible to sociological analysis. This is a question which has been exercising philosophers for many years. Rather than joining directly in this debate, we illustrate the possibilities and difficulties with examples of the application of sociological insights to their design.

One example is a project, now almost complete, to develop a knowledge-based system for public use. The interaction between computer and user is controlled by a set of rules which are derived from some of the basic findings of conversation analysts. Another project, still in its early stages, is developing a module for a computer-based telephone information enquiry system which will monitor the oral dialogue between a caller and the computer and will generate the computer’s side of the ‘conversation’. The design of this module will also be influenced by the findings of conversation analysis of human-human dialogues. As a third example, we point to the potential of computer simulations of collectivities of agents for illuminating issues in sociological theory.

Part 1: Sociology and machines

Much of computer science is, despite its name, essentially an engineering discipline, concerned with the design and construction of artifacts, albeit often only in software. Like practitioners of many engineering disciplines, computer scientists are always on the look out for new ideas to borrow from other areas. The field of 'artificial intelligence' in computer science, for example, has always had parasitic relationship with cognitive psychology. And software engineers in the UK are increasingly looking to some branches of mathematics for new approaches to the specification of systems.

There are now signs that the latest source of ideas might be sociology. This suggests intriguing possibilities, some of a practical and empirical nature and some in relation to fundamental sociological issues such as the nature of action and agency. In this chapter, we will mainly be describing research in which we are personally involved. We hope, nevertheless, that the general points extend more widely, to other projects and to other areas of computer science.

For a designer of complex computer systems, the interface with the human user often presents particular difficulties. For practical and commercial reasons, systems need to have a high degree of usability. Precisely what this term means is the subject of debate (Ankrah, 1989), but it is generally accepted that interfaces should be easy to learn, quick to operate, give rise to low rates of error and be pleasant to use. Unfortunately for the designer, stating these objectives gives few clues about how they might be achieved. Some practitioners have tried to help by proposing guidelines for the designer (e.g. Shackel et al, 1988) or by suggesting methodologies and software tools (e.g. Foley and Van Dam, 1982).

However, these proposals have been widely criticised on two counts. First, they lack a well-defined theoretical base: they are essentially pragmatic in nature, embodying accumulated experience and commonsense (Maguire, 1982). Second, they are too vague. In order to be seen as applicable to a wide variety of systems, the guidelines have to be couched in general terms . But their very generality means that they leave a great deal of room for interpretation when the designer attempts to apply them to the design of a particular interface.

The HCI community is therefore always on the look out for ideas which could help them with *principled* ideas for *specific* recommendations for design. Cognitive psychology, the obvious source for such ideas, has already been extensively mined (e.g. Draper and Norman, 1986). Recently, it has been realised that sociology has some attractive concepts which could be pressed into service. Attention has turned in particular to conversation analysis (CA), an area of research which is concerned with the fine-grained description and analysis of sequences of human interaction. CA provides a set of observations which might, by analogy, be applied to the design of interaction between people and computers. A major advantage of this approach is that it could provide the coherent and principled view of interaction which the HCI community has been looking for. In particular, it could provide a perspective on human-computer interaction which is independent of the technology used to implement systems and independent of the specific psychological characteristics of individual users (Norman and Thomas, 1990).

Designs based on the work of conversation analysts have other potential advantages. First, interactional procedures and strategies which are grounded in everyday life are presumably reasonably effective as means of communication. If they were not, we might assume that they would have fallen out of use in human-human interaction. Thus the designer can tap into strategies which have already withstood extensive practical testing. Secondly, since people are already observed to be using these procedures, we may assume that they are tacitly familiar with them. They may therefore have less difficulty in communicating with computers which have been designed to capture the characteristics of these same procedures; in particular, they should have less to learn, make fewer errors and find the interaction more comprehensible than if some quite new and strange principles of interaction had been designed into the interface.

Such is the manifesto for using CA for the design of computer interfaces. Does it fulfil its promises? We have recently completed the construction of a fairly complex system, the 'Advice System', designed to help members of the public gain access to advice and information on welfare benefits (Gilbert et al, 1990). A number of possibilities for the design of the interface were considered. Any kind of iconic or graphical interface could be ruled out by the difficulty of representing graphically the complex and abstract concepts which are inevitably involved in social security (for example, 'income', 'weekly rent' and being 'responsible for a child'). The user interface has to be appropriate for users with no prior training. This excludes command driven interfaces which people would need to learn how to use. Natural language has been proposed as the most obvious and easiest medium for communicating with computer systems (e.g. Boguraev, 1985). This suggested a design based on the idea of a conversation or consultation with an adviser. The 'conversation' would be conducted through textual, rather than verbal interactions, but would nevertheless be a recognisable conversation, with conversational openings and closings, and questions and answers from both system and user.

The user interface consists of three parts: a scrolling dialogue window on the left of the screen in which all the input and output is displayed; a set of three buttons at the bottom right of the screen; and an area on the right of the screen in which menus are displayed by the system. The menus allow users to compose, in restricted natural language, any of the statements or questions which the system is capable of understanding. The items in the menus change to suit the current dialogue context. The buttons enable users to request a rephrasing of a system utterance, to ask the system for a justification of its reasoning and to demand an immediate termination of the interaction (an 'emergency exit').

This interface provides the resources which can be used to support a 'conversation' between the user and the system. Unlike the majority of current user interface management systems, which limit the interaction possibilities to predetermined routes through a network of utterance options, the interaction design for the Advice System is based on the idea of 'local management'. Each utterance, whether from the system or the user, is constrained only by the immediately preceding conversational context. Constraints on the next conversational turn are imposed by a set of 'interaction rules' derived from CA.

An example of the nature and effect of these interaction rules can be seen in how they control the overall development of the dialogue through the pre-closing sequences occurring after each topic has been completed (*cf* the analysis of telephone conversation closings in Schegloff and Sacks, 1973). A pre-closing sequence first provides the user with the opportunity to ask further questions on the same topic and then, if this has been refused, offers the chance to ask questions on a different topic. If this is also declined, the user is asked whether he or she wants to ask a question on any topic and, finally, if the user does not want to ask any further questions, the system begins a terminal exchange which culminates in the closure of the dialogue and the end of the interaction.

System:	OK, from the information you have given, you should be able to get Child Benefit for Helen.
System:	Do you want to ask another question about getting Child Benefit?
User:	Yes.
System:	Go ahead.
User:	What are the conditions for getting Child benefit?
System:	To get Child Benefit you must have a child who is living with you and you must be responsible for bringing it up.
System:	Do you want to ask another question about what Child Benefit is?
User:	No.
System:	You may be able to get Family Credit and Housing Benefit and Income Support and One-Parent Benefit. Do you want to ask a question about one of these?
User:	No.
System:	We can stop talking now unless you want to ask anything else. If we stop talking I won't remember what you have told me if you come back later. Do you want to ask anything else?
User:	Yes.
System:	Go ahead.
User:	What is take-home pay?

Figure 1 Pre-closing dialogue

The start of a pre-closing sequence is illustrated in Figure 1. The system has answered the user's question about Helen's eligibility for Child Benefit. It then starts the pre-closing, enquiring whether the user wants to ask another question about getting Child Benefit. When the user says no, the system generates a list of those other benefits which, given its current knowledge, the user might be eligible for and asks whether she wants to ask a question about any of these. This offer is also declined, so the system begins the third and final pre-closing sequence, indicating that the interaction can either now come to an end or the user can ask a question about 'anything else'. In the excerpt shown in Figure 1, the user takes up this invitation and asks a question seeking a definition of 'take-home pay'. If she had had no further questions, the system would have begun a closing sequence which concludes with an exchange of "Good-byes".

The effect of the interaction rules governing the pre-closing sequence is to segment the dialogue into coherent units, each of which is concerned with only one topic. The rules thus create a global structure although they only act within the local context. The rules also constrain the user to keep to the topic and avoid possibly confusing diversions. For example, while the system is in the course of responding to a question, further questions from the user which the system cannot immediately answer (and which would therefore disrupt the dialogue) are set to one side with "Sorry, I'm trying to answer your other question. I cannot answer that one yet." The same occurs if the user asks a question about any topic other than the current one. The system returns to held-over questions about the current topic at the beginning of the pre-closing sequence, asking whether the user now wants to ask the question she asked before. The user can return to held-over questions about other topics at the second and third stages of the pre-closing sequence.

As a consequence of this design, the user of the Advice System can obtain a wide range of types of information by asking questions during a dialogue in which the initiative passes back and forth between the user and the system. The dialogue allows the user to explore topics in whatever way seems best, mirroring spoken discourse, where generally conversations are not pre-planned but proceed from 'one thing to another'. The intention was to provide a system which supports a style of advice giving where, although both the content and the structure of the dialogue respond directly to the individual moment-to-moment utterances of the participants, extended interactions are coherent and cumulative.

A small scale evaluation of the Advice System was carried out to assess its usability. Further details can be found in Luff and Frohlich (1989) and Gilbert and Jirotka (1990). The behaviour and comments of five single parents and 20 undergraduate students were recorded as they each interacted with the system for approximately one hour. Lone parents were simply asked to explore the system for any information which might be useful or interesting to them. Students were asked to discover specific kinds of benefit information on behalf of a hypothetical lone parent called Maggie. Half the students were given the job of finding out eligibility and assessment information for Maggie, while the other half were asked to obtain procedural information to help her in the process of claiming a benefit. A procedure known as 'cooperative evaluation' was used in which subjects are encouraged to describe their experience of interacting with the system to an evaluator who is present throughout the session (Wright & Monk, 1989). Subjects were also interviewed afterwards. All dialogue between the users and the evaluator was tape recorded and all user and system actions were automatically logged by the system.

A preliminary analysis of the data suggests that on the whole the design meets its objectives. The system's answers were considered to be clear and direct, and users appreciated their ability to ask as many questions as they liked. One user said the best thing about the system was that "You can keep going round without it getting fed up. ...You can go through different areas and different angles on the same thing". The negative side of this was that the sessions were often quite long, frequently lasting an hour or more. Users reported trusting the system's answers and only became interested in the reasons underlying the advice if it meant that they could not get a benefit they had expected to be eligible for.

The fact that users trusted the advice did not mean that they were always satisfied with it. In fact a single answer rarely gave enough information of the desired kind. Users often had trouble phrasing exactly the right question, and almost always went on to ask a follow-up question on the same or a similar topic. This led to a dialogue in which users explored an issue from a number of different angles. In the post-session interview, users confessed to being surprised at this, having expected a much more system-led interaction in which they would answer a long string of questions before being given a single monologue on the benefits they could get. Several users spontaneously commented on the 'human qualities' of the dialogue, but reacted against the suggestion that it was "like a conversation". One user reflected a general feeling that the interaction was more formal or business-like than a conversation, by saying the "It was more like a question and answer interview". Another commented that "It's better than a conversation 'cos you can tell it to shut up and butt in".

The Advice System is an example of the use of sociological findings in the weak sense, of borrowing specific technical ideas and tricks from the sociological literature in order to solve specific interactional problems. Nevertheless, as the comments of the evaluation subjects suggest, because of the nature of the interaction, it is common for users to invest the system with some of the characteristics of 'agency'; one can consider telling it to "shut up" and not worry about it "getting fed up".

Part 2: Sociology in machines

The fact that users are apparently prepared to ascribe agency to what is, in all honesty, a fairly primitive looking interface to a not very 'intelligent' program suggests that it might be worth looking more deeply into what would be involved in carrying off the trick for real; that is, deliberately trying to design systems which do behave in a way which suggests that they have intentionality. This immediately raises a host of interesting and contentious questions. For example, what does it mean to say that a machine has intentionality? There are people who assert that the question is meaningless – the concept of intention is such that it cannot be applied appropriately to entities other than humans. Others, while likewise reluctant to admit that machines can have intentionality, also argue that there are human capacities which cannot and could not ever be reproduced by a machine.

These critical voices are useful, if only as an inspiration to show that they are wrong, through an existence proof consisting of at least one machine which does have features of intentionality. On the way to this goal, we will need to be rather more precise about what intentionality could actually mean in practical terms. A project in which we are currently involved will serve to illustrate some of the issues which we are having to address. The aim of the Sundial project[1] is to develop a computer system which is capable of 'conversing' with members of the public over the telephone to answer simple queries such as the time of arrival of a particular airline flight or the times of trains. One of the possibilities in designing this system is to make it emulate

[1] Sundial, ESPRIT project P2218, is a collaborative research project involving Logica Cambridge Ltd and the University of Surrey (UK), CAP-SESA, CNET, and IRISA (France), Daimler-Benz, Siemens and the University of Erlangen (FDR), and CSELT and Sarin (Italy).

as closely as possible the human enquiry clerk. In order to do so, the system will need to be able to recognise and 'understand' in 'real-time', that is, as quickly as the words are spoken, dialogue directed at it through the public telephone system (which adds distortion to the voice signal) from callers (who may have strong accents) who are speaking normally. To be successful at the basic recognition task, the computer will need to be able to understand connected word speech from a moderately large vocabulary (about 1000 words). This goal is likely to be achieved in laboratory prototypes within two to three years.

Most current speech understanding systems being developed in research laboratories can be divided for convenience into a number of functional modules (Holmes, 1988; Ainsworth, 1988). Analogue electrical signals from a microphone or from the telephone line are conditioned to remove as much extraneous line noise ('crackles and pops') as possible and smoothed to eliminate gross amplitude variations. The analogue signal is then converted into digital form and passed into an 'acoustic-phonetic' module. The major tasks of this module are to chop the continuous signal into segments (corresponding roughly to phonemes, although different techniques use different units for segmentation) and then categorise the segments with labels. However, the process of segmentation and labelling is very prone to error.

The output of the acoustic-phonetic module provides the input to a lexical processor whose purpose is to extract words from the segment stream. This is a difficult task because speakers blur and change the pronunciation of words depending on the surrounding speech context. Words are also pronounced differently depending on their position in the sentence and their degree of stress. For these reasons, the output from this module is not a simple sequence of words, but a 'word lattice' which gives the probability of any of a large number of combinations of words being the ones which the caller actually spoke. The word lattice is passed to a syntactic and semantic parser. This has the task of filtering the word lattice to reject those word combinations which are syntactically unlikely according to its rules of grammar (e.g. 'Thread ream miss seating') or semantically unlikely (e.g. 'The red dream is eating'). The result is one or more 'parses' of the caller's utterance.

Most existing speech understanding systems stop at this point. A typical prsent day information service system has a 'back-end' program which took the output from the speech recogniser and attempts to provide an answer from a database. The speech system deals with utterances one at a time, with little or no notion of context or of the development of a dialogue over a series of turns.

In contrast, a key component of the Sundial system will be a 'dialogue manager', which will superintend the computer's interaction with the user during the telephone call. The dialogue manager will receive the output from the parser, interact with an external program such as a database to obtain answers to callers' queries, and send instructions to a 'message generator' module. The latter has the task of translating these instructions into natural language for a speech synthesizer, the module which 'talks' to the caller. The dialogue manager has to handle a number of functions as well as interacting with the parser, message generator and database: it manages the conversation as a whole, ensuring that it opens and closes 'properly' and proceeds expeditiously to answer the caller's queries; it determines the questions that the caller should be asked; and it keeps a history of the interaction so far.

Given these objectives for the dialogue manager component, it seems natural to consider whether it would be possible to build some of knowledge of social relations into the design. For instance, it might be helpful if the system had an appreciation of social rôles, such as the status, task and typical knowledge of airline enquiry clerks. The system would clearly need to include some of the tacit and commonsense knowledge which travellers possess, such as that you cannot be in two places at once and that possible delays have to be taken into consideration when planning flights. And it would be an advantage if the system had a fairly sophisticated appreciation of the organisation of conversation, especially telephone conversations. For example, adjacency pairs (such as question - answer, greeting - greeting, offer - acceptance, assessment - agreement and so on) are claimed to be one of the fundamental units of conversational organisation. If so, a dialogue manager able to converse naturally with human users might need to have knowledge of these phenomena.

However, we are now on dangerous ground. Conversation analysts have probably been more explicit about the nature of their work than most sociologists and it is easier to discuss the issues in relation to CA than, for instance, it would be to do so in terms of social rôles, although many of the same arguments would be encountered. The main arguments of those who are sceptical about the use of CA in designing computer technology are that computer systems are inevitably determinate and mechanical, as contrasted with people, who manage with indexicality; that computers operate with rules, while the notion of rule, if applicable to human conduct at all, must be interpreted in an entirely different way; and, thirdly, that human action is essentially different from a machine's behaviour. Let us consider each of these main arguments in turn:

1. Indexicality

One of the most important points which conversation analysts, and more generally ethnomethodologists, made in reacting against 'conventional' approaches was to stress the all-pervasive significance of 'indexicality', the idea that the meaning of an expression is relative to such contextual matters as who says it, to whom it is said, where and on what kind of occasion it is said, the social relations between the speaker and hearer and so forth. In particular, they argued not just against an empiricist view of language which mechanically associated 'names' with phenomena, but also against more sophisticated views which proposed that naming is carried out using sets of rules associating names with specific fixed criteria.

In contrast, conversation analysts assert that no description, however detailed, can exhaust the object to be described. There are always further descriptors which could be applied. Moreover, there is a finite set of descriptors available. Given this, a speaker has on each occasion to select from available descriptions the one which is going to be used. The same descriptors can be used in different contexts to indicate different phenomena, and the same phenomena can be described in different ways. Therefore, there can be no simple correspondence between phenomenon and descriptor, and individual descriptors when considered in abstraction from specific contexts of use seem as a result "muddy" (Heritage, 1978) and indefinite.

If meanings are indexical, varying from occasion to occasion, the hearer is required actively to find the particular meaning intended from the finite and gross description offered, 'filling-in' the detail to the extent that it is needed. Moreover, this search for meaning may have the consequence of re-defining current understandings of the description, the context, or both. This description gives a clue to the way in which a computer system might manage to deal with indexicality. What we require is a technique (an algorithm) for ascertaining the 'meaning' of the words and phrases that make up an utterance which does not merely look them up in a dictionary, but which actively searches out meanings which are consonant with the accumulated context. This algorithm must also be capable of amending those meanings in the light of later input.

In fact, building a small system which will exhibit these characteristics is not particularly difficult (see Gilbert, 1990 for an outline of such a program). The system can generate interpretative hypotheses from an utterance presented as input. For example, we might allow the utterance "It's gone" to be interpreted as a declarative statement of fact, a question, or even an exclamation ("It's gone??!"). On receiving this utterance, the system would choose those interpretations which allow it to extend its current hypotheses. Eventually further input may constrain the possible interpretations by disallowing some of the hypotheses, so that the apparent ambiguity permitted by having several interpretations of the same utterance is reduced or removed by the cumulative effect of other evidence – the "context".

Such context could in principle include the substantive topic of the dialogue, 'real world knowledge' and past experience, as well as prosodic, syntactic, and semantic features of the utterance, although this is not done in the demonstration program. Nevertheless, it has to be admitted that the hypotheses which the program considers for an utterance are always strictly limited to those with which it has been pre-programmed; it cannot 'invent' new interpretations as it goes along. Again, in principle, one could imagine a system which could do so, but this is far beyond the state of the art, involving as it does, machine learning of subtle patterns. Fortunately, in the kind of information provision applications at which the Sundial system is being aimed, it is probably possible to make progress without the system needing to be be able to create or learn new interpretations of words or dialogue items.

2. Rule use

The argument about the fundamental indexicality of descriptors implies that a speech understanding system which is to hold a dialogue with people in a 'human-like' way cannot manage by straightforwardly looking up the meaning of words from their dictionary definitions. Nor can there be a simple rule-like relationship between context and meaning. That is, the relation between meaning and context cannot be as simple as 'in context C1, this expression means X1, and in context C2 it means X2'. Thus, it is argued, a dialogue manager whose understanding and generation is based on rules must be inadequate. Nevertheless, conversation analysts, as a result of their analyses of human interaction, do propose rules, such as the famous turn-taking rule of Sacks, Schegloff and Jefferson (1974) (see Button, 1990 for a consideration of some of the 'rules' proposed by CA and a vigorous argument against the possibility of interpreting them in computational terms; and Schegloff, 1989).

Consider, for example, the "rule" that the first part of an adjacency pair is followed by a second part or, more concretely, that questions are followed by answers. There is clearly some merit in this "rule", but in practice not all questions are answered. Moreover, 'answers' do not come with labels attached identifying them as such; not only does the existence of a question provide the 'context' for identifying a particular utterance as 'an answer', but the presence of an identifiable answer can be used in turn to locate its question as a question (rather than an assertion, a comment, and so on).

It follows that it is not sensible to design a speech understanding system which straightforwardly applies the rule: "whenever you hear a question, generate an answer". Indeed, even if one did design such as system, it would not work, for two reasons: it would have trouble identifying what are questions and there may be no answer to generate. What, then is the status of rules such as this one? What the rule does is to indicate that, given a first part, a second part is to be expected, and to make its absence 'noticeable', that is, the absence is something which might be remarked upon by the participants, referred to later in the conversation, used as a warrant for some inference (e.g. "He wasn't listening to me.") and so forth. It is for this reason that some analysts, such as Button, assert that CA has clearly demonstrated that interaction is not 'rule-governed'. Combining this with an assertion that computers *are* 'rule-governed', the conclusion they draw is that conversational ability cannot be implemented on a computer. The major point being made here is that CA 'rules' may used to guide the *interpretation* of utterances, but that it would be a mistake to see them as determining the *generation* of utterances. CA itself chooses to remain silent on questions of generation, dismissing the issue as one for cognitive psychology.

This is not, however, the end of the matter. CA's agnosticism about issues of generation should rightfully allow any convenient machinery to be pressed into service, including the use of CA rules themselves! This is the approach we are taking with the Sundial dialogue manager, which may logically be divided into two parts: one concerned with the generation of the system's utterances, using whatever knowledge about conversation, langauge and the practical world we are able to build into it; and one part whose task is to interpret *both* sides of the interaction which results. The second, interpretative part monitors the interaction and actively tries to 'make sense' of it, through formulating and testing hypotheses. The output from this part constitutes one of the sources of information used by the generation part to choose suitable things for the system to say.

3. Mechanical behaviour and human action

By virtue of being created by a system designer, computer programs are often regarded as deterministic, predictable and mechanical, in contrast to humans who have none of these characteristics. Furthermore, people are social agents, 'members of society', while computer programs are not. Hence people's utterances are actions, i.e. meaningful behaviour, while the outputs of computer programs are merely phenomena which if they are to have any socially significant consequences have to be interpreted by social actors, i.e. by people.

This line of argument (which has something of the flavour of the old vitalist debate) misses the point that whether an entity is a 'social being' is not a characteristic inherent in the entity, but an ascription by the observer or participant. The possibility of the ascription of 'sociality' to apparently unpromising material is nicely illustrated in Garfinkel's (1967) experiment in which he persuaded students seeking advice to visit a new form of counselling service. The students were told that they could ask only questions expecting a 'Yes' or 'No' answer and that their questions should be addressed to a hidden 'counsellor' sitting behind a curtain. Unknown to the students, the 'counsellor' responded to their questions with answers chosen at random, yet almost without exception the students were able to make sense of the answers, did not doubt the competence of the counsellor and professed themselves well satisfied with the session. A more everyday example is the way that people treat pet cats and dogs (Oldman and Drucker, 1985).

Whether the computer at the end of a telephone line will be treated by callers as a social being, capable of action is a matter which will only be discovered by observation and experiment. My forecast is that it will. We expect to have some evidence on the matter well before the Sundial system is complete because we will shortly be setting up a 'Wizard of Oz' simulation to try to answer this and other similar questions[2]. Wizard of Oz simulations involve a human accomplice talking (in this case over the telephone) to subjects who have been led to believe that they are conversing with a computer (Fraser and Gilbert, 1990). With the help of some audio trickery borrowed from the world of rock music, it is not hard to distort the human voice so that it sounds similar to people's impression of synthesized speech. With this technique, we can find out how people talk to computers without going to the trouble of first building a speech understanding system.

Part 3: Sociology of machines

In the previous sections, we have considered several views on the proper relationship between sociology and computer science. In Part 1, we described what (by analogy with a similar classification made in the sociology of science) might be called the 'weak' programme, that sees sociology as a source of useful ideas. In Part 2, we briefly reviewed some of the arguments for the 'strong' programme, which in its strongest form asserts that machines are not qualitatively different from people and therefore that sociological findings may be applied directly to software design. We also encountered the opposite point of view, which makes a radical distinction between people and machines. What is one to make of this debate?

One answer is that the debate itself requires sociological investigation. This, for example, is the programmatic concern of Woolgar (1985) and the topic of Bloomfield's recent paper (Bloomfield, 1989). This line of inquiry is one which in my opinion is worthwhile and likely to profit from much further work. However, it is unlikely to shed much light on the question of the relationship between sociology and computer science. A more fruitful approach to that question is perhaps to mount empirical investigations of interactions between entities, be they human or machine, with the aim of thus clarifying the notion of agency. Woolgar put it well when he wrote about sociology and AI that

---

[2] The 'Wizard of Oz' technique is so called after the Tin Man in the Wizard of Oz who is eventually revealed to be a person dressed up to look like a robot. Note the paradoxical quality of the set up, which involves a human 'Wizard' simulating a computer which is simulating a human enquiry clerk.

> AI provides the opportunity for reevaluating our preconceptions about behaviour, action, ... agency and, most significantly, our attempts to understand... Are artificially intelligent machines sufficiently like humans to be treated as the subjects of sociological inquiry? (Woolgar, 1985:568)

What could the project of treating machines as the object of sociological inquiry look like in practice? One possibility is to construct artificial societies within a computer and then examine what happens to the members of those societies as (simulated) time passes. In principle the usual apparatus of sociological investigation could be used to quiz the 'artificial' members, although participant observation might be a little troublesome to arrange. More seriously, experiments to set up 'societies' of simulated actors are already under way in a new field of computer science which is variously called distributed artificial intelligence (Bond and Gasser, 1988), multi-actor systems or multi-agent systems.

Multi-agent systems involve a number of autonomous, communicating programmed problem solvers which concurrently collect information from their 'environment', reason with it and act in a shared context. Tools are currently being developed which could enable social scientists to build such systems and investigate their properties. For example, Doran at the University of Essex has implemented a 'testbed' which

> enables an experimenter conveniently to set up a multiple actor system incorporating relatively sophisticated planning and plan execution by the actors (including simultaneous handling of several plans), and structured message passing between actors. The structure of the simulated world in which the actors exist is left entirely to the discretion of the experimenter. With each actor is associated its own data-base of 'beliefs' about actions, the world and other actors, together with a unified hierarchical planning and plan execution system (IPEM – Ambros-Ingerson and Steel, 1988). A simple inter-actor interface is provided whereby actions executed by one actor can access the data-base of any other actor. An additional feature of the testbed is the ability to specify dæmons within individual actors which execute whenever particular 'facts' appear in the actor's data base. This enables actors to be programmed to react directly to a new situation without going through a process of planning. The testbed provides a range of online commands which make it easy to use in experimental mode. (Doran, 1989).

The availability of tools such as this could have some interesting consequences for the development of social theory and even for our ideas of what constitutes social theory. One of its most intriguing aspects is that in order to specify such a simulation in sufficient detail to enable it to be run, the experimenter has to become a social theorist of some sophistication and wrestle with problems, such as the relationship between the 'micro' and 'macro' levels, which previously have only been of interest in abstract discussions of concepts. For example, Doran argues that in order to set up the simulation, the three-way relationship between:

(1) the holistic behaviour of collections of actors, for example, cooperative problem solving behaviour, degree and type of organisation manifested and its adaptability, and

(2) the 'cognitive' or 'information processing' abilities and structures possessed by individual actors, for example, typical actor goals, the structure of an actor's knowledge/belief system, the types of planning used, and the types of communication actors can handle, and

(3) the properties of the task environments within which actors exist, for example, the 'external' capabilities of and limitations on actors, the communication channels available to them, the structure and degree of distribution of tasks to be performed, and the consequences of successful or unsuccessful problem solving

has to be resolved, even if it is done only in the simplest way. With the advent of tools like the Essex testbed, it becomes possible to study the properties of collections of idealised actors, incorporating quite complex cognitive abilities and acting in precisely defined, if abstracted, environments. As an example, Carvajal, one of Doran's students, has studied the trade-off between social model complexity (intuitively, how much an actor knows about other actors and their groupings) and problem solving effectiveness in a community of actors where a pattern of reciprocal exchanges is required for a simple production process. He has explored how the effectiveness of the exchanges varies with variation in the particular actor groupings and group boundaries perceived by the actors when, for example, exchange across a group boundary is more difficult than exchange with a member of one's own group (Carvajal, 1989).

Conclusion

The boundary between disciplines, like most boundaries, is a dangerous, but powerful area, where one needs to tread carefully. In this paper we have sought to explore the boundary between sociology and computer science, somewhat tentatively and mindful of potential minefields, but convinced that the act of exploration will yield insights of *some* kind. Just what these insights might be, and whom they might benefit, is not entirely clear. In Part 1 we provided an example which indicates one way in which a branch of computer science, that concerned with human-computer interaction, might benefit from importing some sociological ideas from over the disciplinary boundary. In Part 2, we began to explore what might happen if we took seriously the notion that computers could have characteristics previously thought to be the preserve of humans, such as intentionality, and what this might mean for sociological ideas about action and interpretation. This served as an example of what investigations in computer science might contribute to sociology. Finally, in Part 3, we argued that there was value in attempting to dissolve the boundary, building societies inside computers, so that there began to be sense in talking about an empirical sociology of computation, or perhaps even a computational study of sociology.

Next, we need to work on ways of getting computers to write research papers for volumes on *Computational Perspectives of Society*...

Acknowledgements

Most of the research described in this paper has been carried out by members of the Social and Computer Sciences Research Group, including Anne Ankrah, Sarah Buckland, Betsy Cordingley, Patrick Dawson, Norman Fraser, David Frohlich, Nigel Gilbert, Michael Hardey, Marina Jirotka, Paul Luff, Scott McGlashan, Peter Robinson, and Rob Wooffitt over the period 1984 to 1989. This chapter has benefitted especially from the comments of Rob Wooffitt and Peter Thomas. The work has been funded by SERC/Alvey, ESRC and ESPRIT.

References

Ainsworth, W.A. (1988) *Speech recognition by machine*, London: IEE.

Ambros-Ingerson, J.A. and Steel, S. (1988) 'Integrating planning, execution and monitoring', Proceedings of *AAAI*, St Paul, Minnesota.

Ankrah A. (1989) Hypercard™ stack on 'Usability', SCS Research Report, University of Surrey.

Bloomfield, Brian P. (1989) 'On speaking about computing' *Sociology* vol. 3, pp. 409-26.

Boguraev, B. (1985) 'User modelling in cooperative natural language front ends', in G.N. Gilbert and C. Heath *Social actions and artificial intelligence*, Aldershot: Gower.

Bond, A.H. and Gasser, L. (1988), *Readings in distributed artificial intelligence*, Morgan Kaufman.

Button, G. (1990) 'Going up a blind alley: conflating conversation analysis and computational modelling', in P. Luff, D. Frohlich and G.N. Gilbert *Computers and Conversation* London: Academic Press.

Carvajal (1989) forthcoming M.Phil dissertation, Department of Computer Science, University of Essex.

Doran, J. (1989) Personal communication.

Foley J.D. and Van Dam A. (1982) *Fundamentals of interactive computer graphics*, Englewood Cliffs, N.J.: Prentice Hall.

Fraser N. and Gilbert, G.N. (1990) 'Simulating speech systems', forthcoming in *Computer Speech and Language*.

Garfinkel, H. (1967) *Studies in ethnomethodology* London: Prentice Hall

Gilbert, G.N., Wooffitt, R.C. and Fraser, N. (1990) 'Organising Computer Talk', in P. Luff, G.N. Gilbert and D.M. Frohlich *Computers and Conversation* London: Academic Press.

Gilbert G.N. and Jirotka M. (1990) 'Planning procedural advice' forthcoming in *Interacting with Computers*.

G.N. Gilbert, Buckland, S., Frohlich, D.M. , Jirotka, M. and Luff, P. (1990) 'Providing advice through dialogue', *Proceedings of ECAI '90*, Stockholm, London: Pitman.

Heritage, J. (1978) 'Aspects of the flexibilities of natural language use', *Sociology* vol. 12, pp. 79-104.

Holmes, J.N. (1988) *Speech synthesis and recognition* Van Nostrand Reinhold

Luff P. and Frohlich, D.M. (1989). 'Mixed initiative interaction', in T. Bench-Capon (ed.) *Knowledge-based systems and legal applications*. London: Academic Press, forthcoming.

Maguire, M. (1982) 'An evaluation of published recommendations on the design of man-computer dialogues', *International Journal of Man-Machine Studies*, vol. 16, pp. 237-261.

Norman D.A. and Draper, S.W. (1986) *User centred system design*, Hillsdale, N.J.: Lawrence Erlbaum.

Norman, M. and Thomas, P. (1990) 'The Very Idea: Informing HCI Design from Conversation Analysis', in P. Luff, G.N. Gilbert and D.M. Frohlich *Computers and Conversation* London: Academic Press.

Oldman, D. and Drucker, C. (1985) 'The non-reducability of ethno-methods: can people and computers form a society?', in G.N. Gilbert and C. Heath *Social actions and artificial intelligence,* Aldershot: Gower.

Sacks, H. , Schegloff E.A. and Jefferson G. (1974) 'A simplest systematics for the organization of turn-taking in conversation', *Language*, vol. 50, pp. 696-735

Shackel , B., Eason, K., Gardener A. and McKenzie, J. (1988) *Human factors guidelines for the design of computer-based systems*, London: HMSO.

Schegloff, E.A. (1989), 'To Searle On Conversation: A Note In Return', prepared for a volume of essays in response to the work of John Searle, edited by E. LePore and R. van Gulick, forthcoming.

Schegloff, E. and Sacks, H. (1973) 'Opening up closings', *Semiotica*, vol. 8, pp. 289-327.

Wright , P. and Monk, A. (1989) *Co-operative evaluation: the York Manual*, University of York.

Woolgar, S. (1985) 'Why not a sociology of machines? The case of Sociology and Artificial Intelligence', *Sociology* vol. 19, pp.557-572.