Dillenbourg, P. (1996) Some technical implications the distributed cognition approach on the design of interactive learning environments. *Journal of Artificial Intelligence in Education,* 7 (2), pp. 161-180.

Some technical implications of distributed cognition on the design on interactive learning environments

Pierre Dillenbourg
School of Education and Psychology
University of Geneva
(pdillen@divsun.unige.ch)

Edited transcript of an invited talk at the World Conference on Artificial Intelligence in Education, Washington, DC, August 1995

INTRODUCTION

Good morning mesdames, good morning mesdemoiselles and good morning messieurs. The question I want to address is the following one: If cognition is really situated or distributed (I will come to that later) what do we have to change in our systems? What are the technical implications of distributed cognition? Well, actually, when people ask this question, the real question is not about the implications; the real question is "What does situated cognition mean?", or " I will understand situated cognition the day when you tell me exactly what I have to change in my systems". It's one of the strengths of this community not to discuss theories at a theoretical level but through the design of educational systems.

A few weeks ago, we had a workshop in Sweden about this very topic - what are the implications of situated cognition? At the end of the workshop a lady came up with the following answer: if cognition is really situated, we don't have to change systems at all because the users will adapt anyway!. We can continue today with our boring, old-fashioned, traditional ITSs (Intelligent Tutoring Systems) because that is simply a context to which the user will adapt. That is partly true, but of course not completely true, otherwise we would not be able to develop bad systems... and in my life I have been quite good at developing bad systems!

I have been working on various systems where there was some collaboration between a human agent and a machine agent. These systems did not work very well. I changed the systems a little bit; I changed the inference engine; I changed the parameters; I changed the rules; I did the experiment again; the collaboration was not going well; I changed the systems again... Collaboration was still not working as well as a human-human collaboration. So I went several times through this loop of designing a system, experimenting and so on. After some time, I thought that I should stop this loop and reflect more systematically on how to integrate an explicit model of collaboration into my system.

DISTRIBUTED COGNITION

If you want to have a computational model of something, you need a theory behind it. There are not many theories about collaborative learning: you have basically the Piagetian theory, which has been revisited by the socio-constructivist people, and you have the social-cultural theory, the Russian theory, which has now been revisited by the situated cognition people. Since I work in Geneva I should have jumped onto the social

constructivist theory, the Geneva one. However, when I learned that I was going to Washington DC, so I asked a friend "What does DC mean?" and he said DC means "distributed cognition". So I have decided that I will be developing this model on the basis of distributed cognition!

The bad point of distributed cognition is that there is no computational model of this situated or distributed theory. Well, almost no: there are some computational models but they are not exactly at the level we want. You have what they call situated robots, which are basically some kind of insect, based neural nets, which learn to follow a wall until they find their food. That's very nice research, by Rod Brooks, Luc Steels, Rolf Pfeiffer, Pattie Maes and so on, but it's not exactly what we need to design intelligent learning environments. When you meet someone who's following the wall all day, you don't consider him as especially intelligent. That's nice work but it's not exactly what we need.

My point is that situated cognition theories suffer from some kind of schizophrenia. On the one hand we have the neuron level and the other hand we have the society level, and there is almost nothing in the middle (vertical axis on Figure 1). On the society level, you have sociologists and anthropologists (people like Jean Lave, Lucy Suchman, Etienne Wenger, and so on) doing very nice work about people integrating into communities of practice and so on. My problem is that there is nothing there in the middle. I believe that there should be something in the middle, something like an 'agent level' or maybe a 'knowledge level'. There are a few people, like Bill Clancey, who try to merge the society level and the neuron level but there is too much distance between the two.

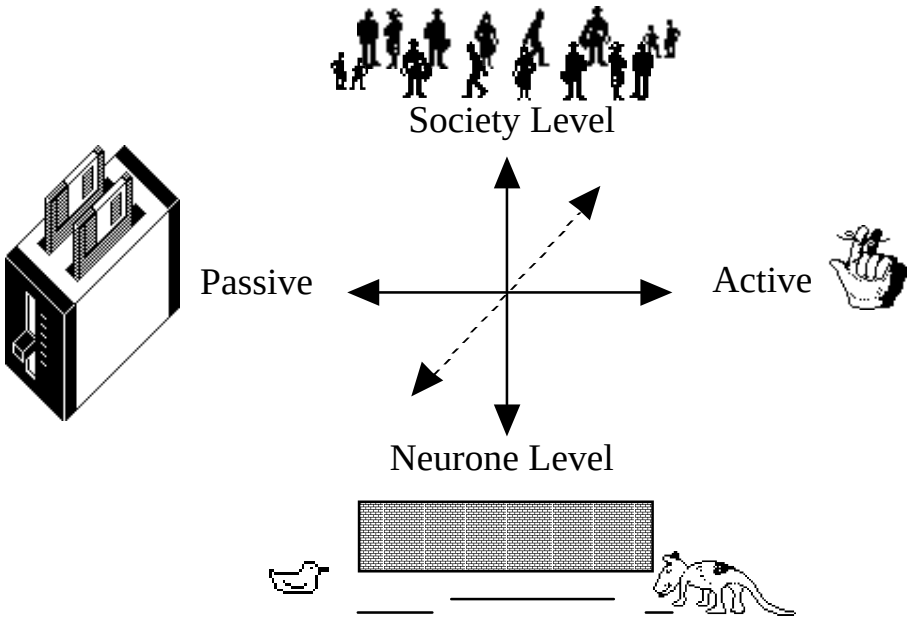

Figure 1: My map of the situated/distributed cognition theories

Let me complete my map of this theory of situated cognition, by putting this second axis here (horizontal axis on Figure 1). If one looks at various theories, one can perceive two views of human beings. In one view, the human is reacting to the context. It's a rather passive view of humans. For instance, according to the concept of affordance - one of the key concepts of this theory -, you see some tool and the very design of the tool triggers your behavior for using this tool. They emphasize that knowledge is context-specific: you go to the swimming pool, you meet the vice-chancellor in the pool but you don't recognize him because you have never seen the vice-chancellor in that context. You have seen that this is a toaster (in Figure 1), but perhaps some of you at the back have not seen that there are two computer discs in the

toaster, and not two slices of bread, because the context induces some expectations about the content that were stronger than the perception of the image.

Another view, which is for me more interesting, is that human beings actively off-load their cognition on the world, like the guy with a string on his finger (in Figure 1) so that he remembers to pick up his child from school. This string on his finger is some kind of external memory, a tool to help him remember. A nice example of this kind of view is given by Roy Pea. It is a story of a forest ranger who had to measure the diameter of a tree to decide whether she would cut the tree or not. What she had to do, of course, was to measure the circumference of the tree and divide it by π. This is not easy to divide mentally by 3.14. So she took a tape and put a mark every π so that, when she put the new tape around the tree, she could directly read the diameter on the tape. So she did not perform any more the computation in her head, the tool was doing the computation for her.

This distinction between reactive and active is not a very solid one from a theoretical point of view. The reason why I made this distinction is that I think that in Europe we see too much of the "reactive' view,. We say there is nothing new in situated cognition because, of course, there are many studies which show that knowledge is context dependent, that transfer is difficult, and so on. We react too much to this part of the situated cognition theories, while for me the active view is much more interesting. Actually the two axes are not perpendicular, but should be rotated as indicated by the doted line: The 'neuron level' people are working more on the context effects, and the 'social level' people are more interested in various tools, not only the concrete tools but also the symbolic tools, and the most powerful among them, the language. My interest is really on the society-active side. For me, the neuron-reactive view is some new behaviorism and I am not very excited by that.

Some of you may have noticed that there was a workshop that was scheduled for Tuesday on computational mathetics and that this workshop has been canceled. Since Tuesday John has been crying every day because his workshop has been canceled, so I have decided to do some computational mathetics in my talk. I hope that you can cope with the complex mathematical notation:

$$1 + 1 = 2$$
$$1 + 1 > 2$$
$$1 + 1 = 1$$

Distributed cognition does not simply mean division of labor where you have a task, you split the task, you give different sub-tasks to different people, they go in different rooms, and when they have finished they assemble the results. That's cooperation (1+1=2). The idea of distributed cognition is something more - it’s the idea that the whole is more than the sum of the parts. Why is the whole greater than the sum of the parts? The traditional distinction, between cooperation and collaboration, is that in cooperation people do their sub-tasks independently of each other and just assemble the partial results. In collaboration people really work together, side by side - they don’t split the tasks into sub-tasks, they really do it together. Actually, this definition is not very solid because when you observe people collaborating, there is some kind of spontaneous division of labor between them. For instance, if there are two people working on the computer, very often it is the one who takes the mouse who does the low level things, and spontaneously the other one will step back a little bit, and take charge of more strategical aspects of collaboration. So even in collaboration, there is some division of labor. The main difference is that in collaboration there is some interaction during the task. So the guy has to do two things: he has to do the sub-task which has been allocated to him, but at the same time he has to interact about it. The cognitive effects of collaboration are precisely due to the fact that in addition to doing

what you have to do, you have to interact about what you are doing. All the mechanisms which have been proposed to explain the effects of collaborative learning (conflict resolution, negotiation, argumentation, internalization, appropriation, mutual regulation,...) concern the interaction that is going on in addition to the task. That's what's meant by the second formula (1+1>2).

The hundreds of experimental studies about the effects of collaborative learning are very complex because there are many independent variables which influence the effectiveness of collaborative learning. Actually, if I would summarize these studies in a single sentence I would say that collaborative learning is efficient if the two guys succeed in building a shared representation of the problem, or in other words, if the two guys together succeed in forming a joint, unique, single cognitive system, at a higher level (1+1=1). I want to stress that the word 'distributed' is a bit misleading. The key point is not that it is distributed, the key point is that it succeeds. Collaborative learning is effective if, *despite* the fact that the task is distributed the quality of interaction is such that people succeed in forming a joint cognitive system, i.e. in building and maintaining a shared understanding of the problem. By the way, some people do not talk about distributed cognition but about shared cognition, which is probably more correct.

IMPLICATIONS FOR THE DESIGN OF EDUCATIONAL SYSTEMS

Several people have been making suggestions regarding the implications of situated cognition for the design of educational systems:

- Clancey said that we should do participatory design, i.e. that we should involve the end user in the design of educational software. Of course, this idea is not knew, people in Sweden have been doing that long before the situated cognition approach.

- The idea of viewing educational software as a cognitive tool, which is the idea of Suzanne Lajoie and others, fits very well with the distributed cognition approach, since the idea of distributed cognition is precisely to think about the role of tools in human cognition, and on how these tools can be progressively internalized by the users.

- Jeremy Roschelle suggested that the computers should not simply support interaction with the user, but also interaction between two users ("Design for conversation"). This suggestion, based on empirical work, makes sense within the distributed cognition framework.

- Another implication at the pedagogical level is to apply this metaphor called 'cognitive apprenticeship'. Just a remark here: This metaphor is a good one for us because it is one where we can keep the things we have been doing before. In the cognitive apprenticeship metaphor there is still an expert of the domain and a learner. So we can roughly keep the architecture we've had so far, which is probably not the case for the cognitive tools approach.

- Of course, there is a lot of emphasis on group work and 'authenticity' (which Louis Gomez talked about). For instance, we have the people at Vanderbilt University who take 500 grams of groupwork, 600 grams of authenticity, shake it a lot, put a lot of video on the top, shake it again, and that makes what they called 'anchored instruction'. It is a shame that John Bransford did not make it here.

- Some implications concern the evaluation of systems. From the situated point of view a pre-test, post-test approach makes no sense, because just making a post-test in a context where the skills will not have to be practiced is not a measure of efficiency. The only measure of efficiency within a situated cognition approach is to see if the guy becomes really better in the tasks he has to do every.

- An implication from the distributed point of view is the idea that what counts is the complete cognitive system, the human user and the tools he uses. We should not do what we often do, i.e. the learner play with the system for some time and then one evaluates his knowledge alone. We should instead make the evaluation when the guy is still using the tools. This is what Perkins calls the 'person-plus hypothesis'.

So, in summary, many people have been writing about the conceptual implications of distributed cognition. What is missing is the technical implications of situated cognition. Why do we need to worry about technical implications? I believe that if there are no technical implications, the 'distributed cognition' idea is not something which has to be discussed in this community. That's a discussion for the educational technology conferences. Let me explain this. A lot of people have criticized AI in education by saying ITSs are not much used. Nevertheless, this community has achieved one major point: before the use of AI in education, the main focus of educational software was on answers, and now, even for people who don't use the AI techniques, the focus is on the reasoning process of the learner. This is a major improvement in educational software and this improvement is something we can claim we have achieved. We can achieve a similar evolution now with respect to distributed cognition theories? Can we develop techniques which would lead designers to view learning not as something which occurs completely inside the learner's head, but which distributed over various artifacts and agents? Actually, we achieved the previous move (focus on processes) because the cognitive science theories included computational models. The problem is, as I said, that distributed cognition theory does not include -as far as I know - any computational model that can be used in ILEs. If there is no computational model, between the theory and the system design, this discussion is not a matter specific to this AI in education community, it belongs to the larger educational technology circle. This is the community of people who build models between theories and systems.

So, what are the technical implications of situated cognition or distributed cognition? Here is my one-sentence summary of the implications: <u>the main functions in a learning environment are intrinsically collaborative</u>. I am not saying that we should use collaborative learning. That's another point. No, what I am saying is that - technically speaking - diagnosis is a collaborative process, explanation is a collaborative process, tutoring is a collaborative process. Problem solving is not intrinsically a collaborative process. It can be collaborative, but you can also solve problems alone. You cannot do diagnosis alone. The mechanisms of diagnosis, explanation, and tutoring are intrinsically collaborative, and I will explain that now.

<u>Diagnosis is mutual</u>

Here are some examples of interactions that we recorded with one of our systems, called Memolab. The guy said, talking about a machine:

> *"I don't know why he puts that at the bottom because there is no need to wait for a while"*

Or another guy said:

> *"Well, there it is ok. I roughly know, I know what he did".*

So what we can see there, which is not surprising, is simply that the user makes some diagnosis of the system - he has some ideas about what the system believes. Now, what is more interesting is when we found a sentence like this:

> *"The system supposes that I want the subjects to do something during 40 seconds, but I wanted the subjects to do nothing".*

Here the guy is doing a diagnosis of the diagnosis done by the system - that is some kind of second order diagnosis, the diagnosis of the diagnosis. Well, you might say "this Belgian is completely crazy, he has drunk too much of his beer (that's not possible): we have so much difficulty doing first level diagnosis, and now he suggests that we do a second level diagnosis!". Actually, in everyday life we do a lot of second level diagnoses. Here are some examples.

me:	*"I saw a football game last week in Geneva"*
Jim:	*"Oh, do they play football in Switzerland?"*
me:	*"I am sorry I mean soccer".*

What is going on there? When I listen to this question from Jim, "do they play football in Switzerland", I think that is a stupid question - of course they play football in Switzerland... If he asks this question... Actually, he often asks stupid questions so I should be used to it with Jim. So, let's imagine it was with Gordon. If Gordon asks this question, I understand that there must be something wrong because I know that he knows that they play football in Switzerland. So why does he ask me this question? I think maybe he has misunderstood what I said, and so I repair what I just said. In linguistics, these mechanisms are called 'social grounding', that is the mechanisms by which you check that your partner has understood what you have said, at least to some extent. Of course, we never understand each other perfectly of course, but we understand enough to carry on the conversation.

The previous example described grounding of the content level. Here is some grounding of the speech act level.

Val:	*"Do you know who is coming tonight?"*
Suzanne:	*"Who?"*
Val:	*"That was a question"*
Suzanne:	*"Oh, Mary and Paul".*

What Suzanne did not understand was the speech act status, the pragmatic value of the question, and then she asked another question, from which Val deduced that she misunderstood the first question. So this mechanism is frequent and easy in everyday conversation. We have many ways to detect and repair misunderstanding, like pointing and so on.

So, the idea here is that the diagnosis is something which is collaborative. Actually, when you go to see your doctor, the quality of the diagnosis made by the doctor will depend to a large extent on your own ability to express what are your symptoms, and if you don't express yourself very well, there is a good chance that the diagnosis will be wrong. This idea that diagnosis is mutual is not actually a new idea - it is quite an old idea in this community. (Anyway, there is always some Greek who has said it before you). If you look at the history of AI in education, very early, Burton came up with the idea of interactive diagnosis. Later on, John Self suggested that we should make the learner model open. Then Michael Twidale made a system where he actually invited the subjects to express intermediate steps in the problem solving to help the machine to make the diagnosis. Then we come to the work of Susan Bull, Helen Pain and Paul Brna, where the learner can negotiate the content of the learner model. In parallel we had the formal approaches to student modelling. You might believe that they are not at all connected to situated cognition. I think that we need formal approaches, because if the machine has to repair a misdiagnosis, it has to reflect on the mechanisms which produced this misdiagnosis.

At the beginning, diagnosis was viewed as a neutral mechanism: you take a picture of something, but you don't change that something. However, as soon as diagnosis becomes interactive, it is not a neutral mechanism anymore. As soon as you say to

someone ‘do you really believe that?’, of course he will change his mind. This is not a negative aspect. For instance, Denis Newman in Amsterdam in 1989 had this idea of replacing diagnosis by appropriation. Appropriation is one of the key mechanisms in the social-cultural approach. It is the idea that when you say something, you may change your understanding of what you said according to all the other guys' interpretation of what you said. Newman suggested that we use this in interactive learning environments. Another kind of positive side-effect of interactive diagnosis is the idea of reflection: a good point in showing the student model to the student is to force him to reflect on what he really knows. So, once again, mutual diagnosis is not a brand new idea, it is just the natural follow up to work which has been done before.

Explanation is collaborative

Second point, explanation is mutual. I will not go too much into details about that point because some other people have been saying that for the last few years and it is not a very new point. At the beginning of expert systems, an explanation was viewed as something that one agent builds and gives to somebody else. Then progressively, we observed that we need to adapt the explanation to the people who receives it. Then scholars like Martial Vivet observed that if you really want to adapt the explanation, you have to adapt the reasoning itself because the explanation is the output of the reasoning, and so on... until you arrive at the work of Michael Baker which really views the explanation as something which is built together by two people, the explainer and the explainee.

Tutoring is collaborative

You will say, ok Pierre, diagnosis is mutual and explanation is, but not tutoring: tutoring is really one guy who is telling things to another one. No. There is some evidence that tutoring itself is a mutual process. The first evidence comes form the work of Sarah Douglas. She observed that tutors make many mistakes when they teach. Some of these mistakes are due to misdiagnosis, some are due to just slips. The learners spontaneously repair some of these mistakes. When the learner makes such a repair, the teacher uses it to generate the next question or the next exercise. These interactions are not just some noise which we repair and then it is over: they change the curriculum of the lesson.

Another example comes from the work of Barbara Fox. In the same vein, she observed that when the students are working with the tutor most of the time they think aloud, just to make sure that the tutor is following them. The tutor is doing something like hmm, hmm, hmm, and as soon as the tutor doesn’t do that for half a second, the learner understands that there is something going wrong. He maybe re-phrases the sentence, or re-formulate an affirmations as a question. She goes through the analysis of many protocols and she really describes tutoring as something which is the achievement of two people who try to understand each other all of the time.

RE-DESIGNING OUR SYSTEMS

That was my point: diagnosis is mutual, explanation is mutual, tutoring is mutual. So, what do we have to change in our systems? One answer would be that we need a better interface model, some natural language processing facilities. Of course, those would improve our systems. But if we just do that, we will miss the point. It would just be an engineering answer to our difficulties. We would do again a mistake which we have been doing for many years: We have always been separating two boxes, one box for communication and one box for reasoning, as if it were two different mechanisms (Figure 2). Basically we have a communication model, which receives inputs and sends them to the real cognitive model which reasons, induces, deduces, make analogies and so on. When the reasoning is over, it sends some output to the dialogue model which

sends it outside. If we continue with this approach, we will never be able to model the distributed cognition ideas, which is precisely to understand why dialogue has an impact on cognition.

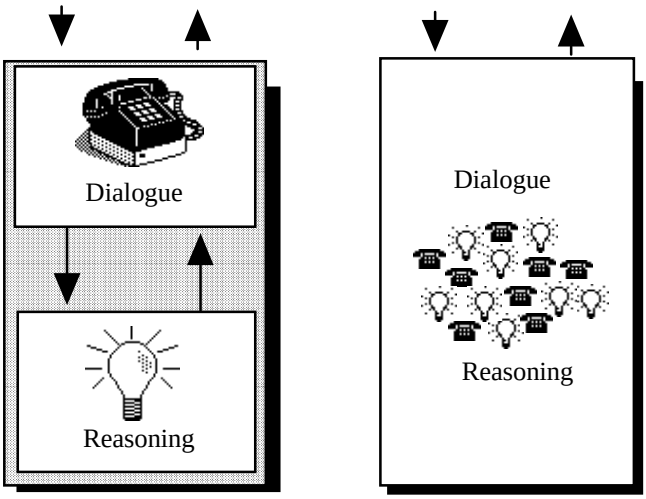

Figure 2: Do we dissociate or not reasoning and cognition?

So my suggestion is that instead of doing this (Figure 2, left) we should do that (Figure 2, right). It means having the same cognitive model for both dialogue and reasoning and trying to model why these two processes are so much going together. You might think that this is completely ridiculous, because from the end of the room, these two boxes are identical. It is just a matter of granularity. Therefore, I will point out two formal differences: (1) in the left hand side model, the two boxes use different operators, while in the right hand side model, dialogue and reasoning use the same operators; (2) in the left hand side model, the dialogue operators can only apply when the reasoning box has come out with some conclusions, while, in the right hand side, dialogue operators intervene inside the reasoning: the dialogue operators are used below the rule level. I will explain what this means.

Using the same operators

First, I will explain is what I mean when I say that the dialogue and the reasoning could use the same operators. Here is an example (Figure 3). Two or more guys go on the mountains and one says “let’s go to the top, the view will be great”. Then the other one says “there are some clouds so we might have no view”, so he refutes his point. This refutation is refuted by another guy (who may be the first) who says, “well, the top of the mountains is higher than the clouds so we will have a good view anyway”. Somebody continues this argument, “well, the day is cold so we will have no risk of avalanches when we go down” and this is refuted by another guy who says “well, we are quite late so there will be a risk of avalanches”. This refutation is refuted by another guy who says, “We won’t risk avalanches because we will walk on the ridge”. So what is the point here? The point is that this tree of arguments can model the dialogue between two people or three people or four people... but it can also model the reasoning of the guy alone. If I am going alone up the mountains I can say “I will go to the top that will be great, oh yes but there are the clouds and the avalanches, and so on”. I can replay the same dialogue, the same set of arguments, if I am reasoning alone. Of course, if there is no snow on the new mountain, or no clouds, I won't reuse the arguments which are not relevant. I is not exactly a 'replay' mechanisms, but some adaptive reuse of dialogue patterns.

This reuse of dialogue patterns has been implemented in a system called People Power. The machine agent uses the same procedure for both reasoning and dialogue. The dialogue procedure uses two arguments, one for the sender and one for the receiver.

When it was used for monologue, the two arguments are the same agent. The same operators were used both for reasoning and for dialogue. In People Power it was extremely simple because we basically had two operators, agree or disagree. This is of course an extreme simplification of the work on dialogue models. We need a much richer set of dialogue moves or speech acts, including partial refinement, rephrasing, things like that. But the principle is there: we can use the same operators for both reasoning and interacting with people.

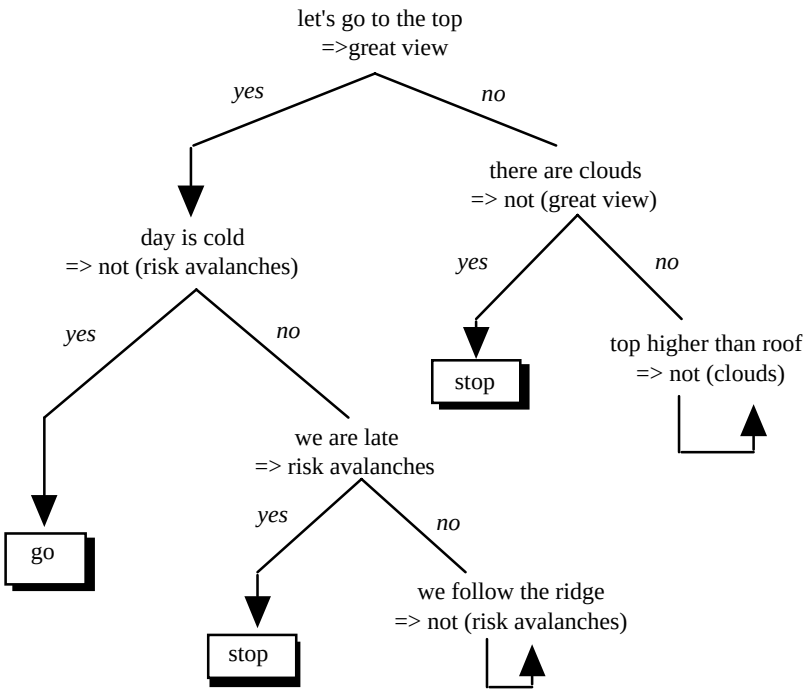

Figure 3: The structure of a simple argument

Below the rule level

In People Power, these operators applied at the rule level: each of these arguments was a rule and you could agree or disagree with the rule. I suggested that these operators should apply *below* the rule level. To explain that will take me a few minutes.

First I have to explain another system I have been working on, where we wanted to have collaboration between a human and a machine. Just imagine that you want to have two expert systems which collaborate step by step. One way to do it is for the two expert systems to have the same set of facts, in such a way that any new fact created by one agent, will be put into the shared set of facts and can trigger a rule from either of the two rule bases. So, if the first agent deduces a new fact, this new fact may trigger a rule from the second expert, and so on. That's if we have two expert systems. In the Memolab project we had one expert and one human. So what we can do is put this shared set of facts on the screen, not as a list of facts of course, but as a concrete representation of the problem. (In this case the problem was to build an experiment in psychology). Any action of the expert changed the state of the problem on the screen, and any action of the learner changed the state of the problem on the screen, and the learner could always see what the expert has done.

The rules of the expert were only sensitive to the things which were on the screen. How did we achieve that? By a very simple trick: we built an object-oriented production system, in which each variable is instantiated with respect to a class:

```
      if ?variable-1 is class-n with slot-n = slot-value-n ... then
...
```

?variable-1 can receive any example of class-n. We have a hash-table with all these examples, and all the examples of the sub-classes, and so on. To have a rule which only reads the objects on the screen, we do the following: When an object is put on the screen by the user or the expert, we simply create an instance of a sub-class which is called a displayed-class. Here ?variable-2 can only be instantiated by an object which has been displayed on the screen.

```
if ?variable-2 is displayed-class-n with ... then ...
```

The expert in Memolab has rules like

```
if ?M is a material and ... then ...
```

The variable ?M can be instantiated by any instance of the class material, which has four sub-classes: words, non-words, digits and letters. Most of the material is a list of words. We have already a few examples of the list of words in the system. Let's look at a protocol between two (human) subjects:

1	Boris	the material... Which material do you want?
2	*Bill*	*You mean, the words*
3	Boris	Yes, you can choose directly in the list of materials which are already available.
4	*Bill*	*I choose between 10 and 20?*
5	Boris	Except if you want to create again a new material... but this is not the idea here.
6	*Bill I*	*take 10*

The two first interactions can be described as narrowing down the scope of a variable from the class 'material' to the sub-class 'words'. The class 'words' include various predefined instances, discussed in utterances 3 and 4. The learner is also allowed to create new instances of it (utterance 5). One learner finally decided to select the particular instance for this variable (utterance 6). These two subjects do not know that the system has are rules and variables and classes, of course. What I am saying is that the kind of dialogue they have could be modeled as a negotiation of the scope of this variable ?M. We have everything to do that since we already have this hierarchy of classes. It would not be too complex to model this negotiation of the value of ?M. This is what I mean by using operators *below the rule level*. Negotiation does not simply occurs when the rule has been instantiated, but during the instantiation process.

Here is a second example of what I mean by negotiating below the rule level. In the rule:

```
If ?M1 is a material and ?M2 is a material and ?M1 is equivalent
to ?M2 then ...
```

what does "equivalent" mean? It can mean exactly the same objects, or two objects with the same words inside the material, or two lists of words which have the same average number of syllables. If this average number of syllabs is respectively 2.22 and 2.23, are the two materials equivalent or not? The value of this predicate is something which is fixed in the system, inside the inference engine. But the users seem to negotiate the value of such a predicate:

1	Boris	It should be the same words
2	*Bill*	*Yes, I think... no! the same number of syllables*

I suggest to model this dialogue between two people as a negotiation of the scope of the variable, the value of the predicate, and things like that. I have not done that, I am

suggesting it as direction for research. I have started to do some research in that direction.

CURRENT RESEARCH

If you want to have this mutual diagnosis, mutual explanation and so on, what is really important is to have a model of grounding mechanisms - all those mechanisms by which we manage to understand each other. To model this grounding mechanism, we should apply dialogue operators below the rule level. I've started a new project on that. One of the problems is that when we do some grounding in everyday life, not everything is verbal: there are a lot of things in the eyes, and there are a lot of gestures and drawings. If you have to solve a problem with a friend very quickly, you take a piece of paper and draw something and then during all the discussion you will point and draw little things. If you are in a restaurant and have to solve some problem together, very quickly you will draw something on the napkin. In social grounding, there is a lot of gestures, pointing and drawing..

SUMMARY

The goal of my work is to improve interactive learning environments. The conclusion from the ILEs I have been testing is that if I really want to make some progress in that direction, I should stop for a while doing research on education and try to improve the techniques for human-computer collaboration. Such an improvement is not just a technical matter: it should be designed/understood within a theoretical framework, namely the distributed cognition theories. To be more concrete about what could be a model of collaboration, I suggested to integrate dialogue operators inside the mechanisms of our inference engine (rule instantiation, predicate evaluation, ...).

There is something strange in this community. I have been around for about a decade and the techniques have not changed a lot. Basically, we are still roughly using the same knowledge-based systems as ten years ago. I don’t believe that we have all the techniques we need for designing effective educational software. I believe there is a need for new technical improvement. There are many directions for improvement. I have just pointed here to one of them.

Three final remarks:

- First, about multi-agent systems. Distributed cognition does not mean multi-agent: there is something more. It is not simply agents who send messages to each other. The challenge is higher than that. It is about agents which build mutual understanding (whatever it means when you are talking about computational agents mutually understanding, which it is a bit bizarre because machines do not understand anything). There is something more ambitious about interaction there.

- The second point is about AI. I think that there is some evolution of AI, which is not any more modelling problem-solving *as* human beings but is modelling problem-solving *with* human beings.

- Last one. In the previous conference (maybe not so much in this conference), there was a lot of emphasis on pragmatics, dialogue models, speech act theory and so on. I see a great potential there. But the potential is not simply to improve the interface. The interesting part is to understand how social interaction influences individual cognition.

Questions

I am a little concerned that if we need all these things from natural language, and other fields, then won't the field of AI in Education become 'AI-complete'? Won't we need everything?

I am not saying that we should start new fundamental research in pragmatics and so on - what I am saying is that there has been a lot of research into pragmatics, and if we want to benefit from this work in AI in education, here are some ways to use this work. My new colleague has developed some computational models of social grounding. I hope we can merge the work which has been done in pragmatics into the more cognitive-orientated work. But of course we will not solve everything like that.

I admire how you argue politely about the situated approach but there is one point where I really want to have a different view, particularly when you aim for technical solutions. My impression is that your view is based on a symmetry between the system and the human as agents, and this will always end up in linguistics. But this is not necessary. Maybe not theoretically but practically, systems and humans have different communication channels. An artificial agent can copy something as many times as you want, send it everywhere you want, collect the results easily - things which are not possible with humans. So the symmetry assumption is not a very adequate one. There may be theoretical reasons but the semantics inside the machine will always be of a different nature than the semantics we have.

Let me answer on these two points. First the symmetry question. I agree that all the people in human-computer interaction say that, since the machine agent and human agent have different features, designers should use the best of each of them. I agree from the practical point of view, but from a theoretical point of view I like the idea of symmetry because there is still this idea of computational models of human cognition behind. About the second point, the fact that there cannot be a real social grounding with a machine because there is no meaning in the machine, this is of course true. However, you can put things the other way round. You can say that the machine is not doing the grounding , but that there is some grounding between the designer and the user. This grounding is asynchronous. There can also be grounding between a user at time n and a user at time m.

I agree that it is important to be able to discuss the contents of your rules but we must not forget that we must discuss also the task and the domain. You will lose a very important part of dialogue if you concentrate on 'below the rules'. So which level are you discussing?

My experience (with People Power) is that people don’t want to talk about knowledge. They don’t want to explain and justify themselves all the time. It's is boring. They firstly want to do the things on the screen. But, of course, through negotiation of action, they may come - and hopefully will come - to negotiate at the knowledge level. Moreover, not all negotiation has to be verbal. Gestures (clicking, pointing, ...) can be translated into speech acts. Some of these speech acts are below the rule level, something that has not been modeled, so I suggest we model that.

If you're right that collaboration only succeeds if two people become of one mind, then that mind is still not completely knowledgeable. So where is that missing knowledge going to come from? We are back in the situation where we have a single learner in a learning environment. Either the environment has to be very carefully designed so that the one mind, in this case, embodied in two people, has the opportunity to discover the knowledge - that's the microworld approach. Or the only other approach that I've seen is that there is an expert someplace who has the knowledge and is responding to the difficulties that this one mind has. So, it seems that by working on collaboration we're going to have some opportunities for learning - namely, when the two people have

unequal and incomplete sources of knowledge, and through collaboration they mesh and form the union of their knowledge. Let's assume that that's still incomplete, as it would be for two normal students, then we're back where we were before.

You have the hypothesis of the union of knowledge, which is not correct. There are many experiments which show that if you put two learners together, even if there is something they don’t know, they may discover it together. It’s not that one has to know it to teach it to the other one. When you put two people together who don’t know something, because of all the interaction which is triggered. they may discover that something. If there is a high quality of interaction, they may move one level up - they will not move twenty levels up. Two people who don’t know something and who interact well with each other (negotiate, explain, ...) may progressively know more about this something. This phenomenon cannot be explained from a 'learning as knowledge transmission' point of view. We need some other mechanisms to explain the fact that the group interaction lead them to discover things that none of them knew at the beginning.