KNOWLEDGE CONSTRUCTION IN ADAPTIVE INTERFACES

Nadine Ozkan[1-2] *and Jean Caelen*[1]

[1] Institut de la Communication Parlée / INPG,
46 ave. Félix Viallet,
38031 Grenoble cedex 1, France

[2] Thomson-CSF
B.P. 10, 92223 Bagneux cedex,
France

ozkan@icp.grenet.fr, caelenj@icp.grenet.fr

KEYWORDS
communication, empirical studies, adaptive interfaces, cooperative task execution

ABSTRACT
Cooperative task execution among humans involves sharing perspectives about how to do the task, and about the task itself. Our research focuses on interfaces which learn and adapt to the user's perspective. This paper presents the theoretical and experimental foundations of our work, as well as a description of the implementation of our concepts. Based on communication theories where the participants jointly build a common perspective, we designed a experiment involving human cooperative task-execution. The corpus analysis yields a taxonomy of the actors' utterances and actions and a dialogue model. Based on these two elements, we propose a representation of the user's perspective, as well as a method for automatic learning of the user's perspective.

INTRODUCTION - COMMUNICATION AS A CONVERGING PROCESS

Recent theories of communication have drifted away from the traditional "transmission" model (Shannon's linear processing of coding, transmitting, receiving and decoding) to adopt a more participatory view, where the actors are jointly involved in the construction of the communicational object. Following this trend, we view communication as a process by which different perspectives converge. Note that convergence does not imply agreement - convergence only means that enough knowledge is shared among the actors to avoid misunderstanding or breakdowns. An efficient and successful exchange both uses knowledge that is common to the actors and adds to this basic knowledge in some way (see [4]).

Applied to the field of HCI, this definition has two important implications:

1. It emphasizes the distinction between the user's perspective and the system (or more accurately the designer's) perspective. The user's perspective is his or her interpretation and appropriation of the task, of task semantics, his or her intention - defined as a high-level goal most often lying outside the scope of the task, etc. The system perspective includes task objects and operations as they are presented through the interface. In other words, the system is viewed as a generic tool, whose use is contextual and takes on different meanings in different situations. As an example, consider a drawing tool (McDraw™ style) used for architectural design, dataflow diagrams, drawing birthday cards etc.

This distinction is not new (see Moran's task model [8] or Norman's Action Theory [9]) and is at the heart of much research in HCI (plan recognition, focusing or adaptive interfaces) as well as in other domains where user-system communication is of prime importance, such as the design of intelligent tutoring systems [11].

2. It highlights the fact that only one of the participants in the human-computer communicational system - namely the user - is involved in the converging process (see Norman's notion of semantic distance, for example). Except in cases of specialized applications tailored specifically to one task, even minimally experienced users know much more about the "semantics" of an application (system perspective), than the application knows about the user's task semantics. In fact, much research today focuses on facilitating the bridging of this gap for the user through sophisticated help and guidance facilities, correction of user misconceptions regarding task domain ([2], [7]).

In our view however, the gap should be bridged by both communication partners: the user by learning the system's perspective, and the system by learning the user's perspective. We define task-oriented cooperative human-computer communication as the bridging of this gap in order to reach a specific goal, known at the outset only by the user. We advocate adaptive interfaces as a means to make this process efficient and successful.

THE CONVERGING PROCESS: AN EXPERIMENTAL ILLUSTRATION OF COOPERATIVE TASK EXECUTION

In order to extract and formalize the characteristics of the converging process, we used an experimental approach: we designed, observed and analysed a restricted but illustrative communicational situation between humans. This situation is described in this section.

Experimental Task and Situation

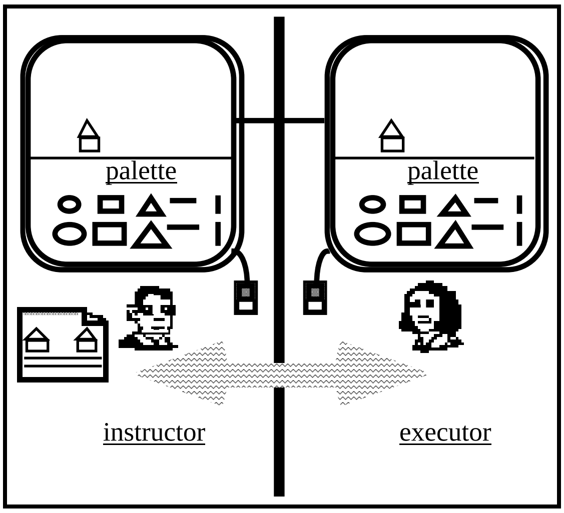

Figure 1: Experimental situation

We designed an experiment involving two humans collaborating around a given task. The setting was the following (see fig.1): two actors were in two different rooms, each equipped with a computer and a drawing software (McDraw™). For simplicity's sake, the only available drawing tool was a home-made palette of fixed-sized geometric objects (small and large triangles, squares, line segments etc.). The actors could communicate vocally through microphones and speakers. The two computer screens were always identical: what was done by one actor on his computer was immediately reproduced on the other actor's screen and vice-versa.

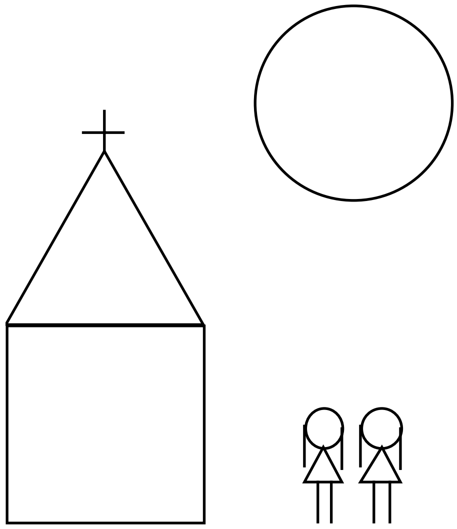

Figure 2: Example of a scene to be reproduced on screen

Their joint task was to reproduce on screen six drawings, each representing a simple figurative scene such as the one illustrated in fig.2. The "expertise" or knowledge required to execute the task was distributed among the two actors: one of them, called the instructor for reasons explained later, was not very familiar with McDraw™ (although familiar enough to use it) and was given the drawings; the other, called the executor, had expert knowledge of the application but no knowledge of what was to be drawn on

2

screen. Because only instructors were given the scenes to be drawn, they had the initiative in the task: they could decide to either draw the scenes alone using the mouse, delegate this to their partner (by conversing through the microphone), or mix both strategies. No instruction was given regarding this.

Let us briefly comment this experimental setting. Note that it is designed so that the instructor's role and knowledge are modelled after that of a user, whereas the executor's role and knowledge are that of a competent and intelligent collaborator (in other words, that of a target-interface). This experimental technique resembles the Wizard of Oz technique, with the important difference that here, the instructors know they are addressing a human and not a machine. This allows them to take full advantage of their partner's collaborating capacities.

Since our interest focuses on the convergence of knowledge, we distinguish on a temporal axis:

- knowledge shared by the the two actors prior to a session, comprised of:
    - basic knowledge of computer and mouse use,
    - the system perspective, that is, the available drawing objects (fixed-size geometric shapes) and operations (essentially, selection and dragging),

    (albeit the executor is more familiar with these)
    - common sense knowledge about the world, objects in the world and how they may represented in a 2-dimensional drawing;
- and the instructor's perspective, that is, knowledge shared at the end of a session (jointly built by the actors as they go along), comprised of:
    - the elements of the scene and the spatial relations among them (for the example in fig.2, these elements would be a church, with a square as its base, a triangle as its roof and two lines segments for the cross, two girls, etc.),
    - the instructor's plan as to the drawing order of the scene elements and task distribution (who draws what when).

Results and Analysis

Out of 11 instructors, 4 decided to mostly draw the scenes themselves, almost never interacting with their partner. The remaining 7 chose to share the task or delegate it entirely to the executors. Among these, two task description strategies were observed: 1 instructor described the task in terms of system-perspective, that is, by referring only to the geometric objects available and the selection and dragging operations (ex: "select a small circle and place it centred above the vertical line and touching it"). The other 6 shared with the executors some of the semantics of the scenes ("Draw the girl's head as a small circle").

The corpus (camera recordings) was analysed from two angles: the instructors' use of the two available modalities (mouse and voice), and their strategies to communicate knowledge about the scenes to the executors in order to delegate task execution. In this paper, only the latter is discussed, and so the analysis presented here pertains to the 6 instructor-executor pairs who shared both the task and knowledge about the scenes.

Taxonomy of an action's function

Our analysis focuses on how, during a drawing session, the instructor's perspective is gradually built, negotiated and agreed upon by the two actors. The actors' actions are classified according to their communicational function. Since we are not concerned here with the issue of multimodality, we use the word "action" here in a broad sense, meaning the minimal communicational unit, and disregarding whether it involves gesture alone (pointing with the mouse), speech alone ("put a triangle above the square, centred") or both speech and gesture ("put the circle here" together with a mouse designation). This notion of action is an extended version of that of a speech act in Speech Act theory [1]. Also note that, in accordance with Grosz [5], a single action may have several functions.

The instructor can:

- <u>do</u>, that is act directly on the task. For example, to mouse-drag a triangle;
- <u>request to do</u>, that is act indirectly on the task by requesting the executor to do so. Ex.: "now can you put a small circle there" plus mouse pointing;
- <u>infirm</u>, that is refuse one of the executor's actions. Ex.: "no, the square goes more to the left";
- <u>state an intention</u>, that is define a high-order goal. Ex.: "The scene represents the sun and two girls beside a church";
- <u>state a plan</u>. Ex.: "First you draw the church, then I will draw the girl on the left.";
- <u>define a bridge</u>, that is, define the building blocks of a scene element. Ex.: "the roof is represented by a large triangle". This type of action is called bridging because it creates a link (in this case, a link of shape identity) between an object belonging to the user's perspective (the roof) and an object belonging to the system perspective (a triangle).

The last three types of action (intentional, planning and bridging) involve explicit knowledge sharing about the instructor's perspective. (The other types of actions may also instruct the executor in some way, but not explicitly.) We group them under the name <u>epistemic actions</u>.

Similarly, the executor can:

- <u>do</u>, that is act directly on the task using the mouse;
- <u>query a request to do</u>, that is require knowledge about a request. Ex.: "Do you want it here?";
- <u>query an intention</u>, that is require knowledge about an intention. Ex.: "What are we drawing here?";
- <u>query a plan</u>, that is require knowledge about the plan. Ex.: "Shouldn't I draw the sun first?";
- <u>query a bridge</u>, that is require knowledge about the links between the system and the user's perspectives. Ex.:"How do I draw a church?".

The last three types are requests for knowledge about the instructor's perspective. We group these under the name epistemic breakdowns. (There are other types of communicational breakdowns - such as mishearing or not knowing a word - with which we are not concerned here.) Epistemic breakdowns occur in two situations: when the executors need to clarify a vague or ambiguous action from the instructors, or when the executors feels that more information would make them more efficient collaborators. Repair of an epistemic breakdown requires an epistemic action from the instructor.

In addition, there is another type of action common to both instructors and executors:

- acknowledgement, which acknowledge, confirm, reformulate or accept the partner's actions. Ex.: "OK", "I understand", or "that's right".

Dialogue model

Another type of analysis concerns the succession of actions between the actors. Because, as we say earlier, the instructors have the initiative in the task, they are unrestricted as to the type of action they take at any point. Regularities in the succession of actions may nonetheless be observed and be used to identify a minimal "well-formed" dialogue. This dialogue model is described here. Note that it does not include acknowledgement type actions, which may occur at any point.

The convention used to define the model is the following:

a := b means a is defined as b
[a] means a is optional
a b means a is followed by b
a | b means either a or b

The actions defined above may be grouped into dialogue phases. A dialogue phase consists of an action, followed by breakdowns and repairs of this action, if any. We have:

```
request_to_do_phase := request_to_do_action
                   [query_on_request_to_do
                     (epistemic_action | request_to_do_phase)]
intentional_phase := intentional_action
                   [query_on_intention  intentional_phase]
planning_phase := planning_action
                   [query_on_plan  planning_phase]
bridging_phase := bridging_action
                   [query_on_bridge  bridging_phase]
do_phase := request_to_do_action  executor_do_action
                   [infirm_action  do_phase]
```

Grosz [5] observed that task-oriented dialogues have the same structure as the underlying task. We assume that the structure of the experimental task is to draw scene elements one after the other. A sub-task then consists of the completion of a scene element. Based on this, the minimal well-formed dialogue for one sub-task (that is, one scene element) is defined as:

```
sub_task_dialogue := instructor_do_action | delegation
delegation := [intentional_phase] [planning_phase]
              [bridging_phase] request_to_do_phase
              [bridging_phase] do_phase
```

Task oriented dialogue is then defined as a succession of sub task dialogue:

```
task_oriented_dialogue := sub_task_dialogue
                          [task_oriented_dialogue]
```

IMPACT FOR HCI DESIGN

One conclusion derived from this analysis is that the user should be able to input what we defined as epistemic knowledge, and that the system should account account for this knowledge throughout the rest of the interaction. This implies the creation of a dynamic knowledge base, which is adjusted and augmented as the user provides epistemic knowledge. Based on the experimental analysis presented above, we are currently working on enhancing an existing home-made multimodal drawing system with such an evolutive knowledge base. We give here a sketchy description of how this is done, illustrated by the experimental setting.

Knowledge known prior to the task by both user and the interface must be explicitly represented in the system. This includes:

- system perspective, that is the 10 types of objects available to the user (small and large triangles, squares, circles, vertical segments and horizontal segments) as well as the three operations (pointing, selection and dragging);
- a dialogue model, which defines the possible succession of actions from both user and system. This is somewhat restrictive for the user, since the dialogue model dictates the types actions which may be taken at one point. However, because our dialogue model is based on observation of natural conversation, this should not be too restrictive. The dialogue model we use is the one described in the previous section.

The user's perspective may be represented as a graph. Fig.3 shows part of this graph for the scene from fig.2. The nodes are the scene elements, and they are connected by of three types:

- is-a links define a class - sub-class - instance relationship,
- part-of links define how scene elements are composed,
- spatial links define how scene elements are to be placed relative to one another
- temporal links define when scene elements are to drawn relative to one another.

4

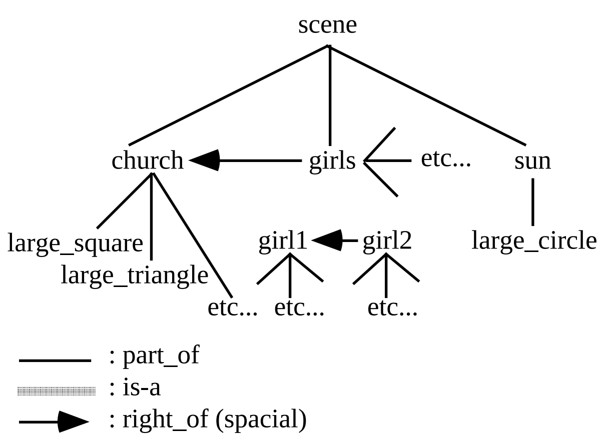

Figure 3: Partial graphical representation of the scene from fig.2

This structure is built (in part or totally) by the interface based on the user's epistemic actions. Each type of epistemic action modifies the structure in the following way. The user's intentional actions create new nodes linked by is-a links and part-of links. Ex.: The action "The scene represents a church and two girls" creates the structure.shown in fig.4.

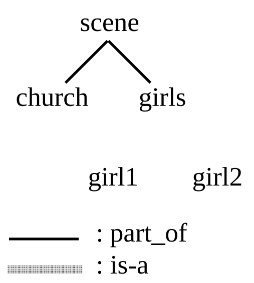

Figure 4: Graphical representation of intentional action

Similarly, planning actions create temporal links and assigns scene elements to be drawn by either the user or the machine; and bridging actions create part-of links. Temporal links, part-of links and assignment of sub-tasks are inherited through is-a links.

CONCLUSION AND FUTURE WORK

Adapting to contextual use currently seems to be a fruitful research paradigm for HCI. We defined our own version of this paradigm, by defining contextual use as the "user's perspective", and we presented both theoretical and experimental evidence to support it. What we advocate is, in fact, to separate the two main functions of an interface: that of a tool for direct task execution, and that of a communication partner for intelligent dialogue about the task.

We have discussed one strategy for learning about the user's perspective: to accept as input the user's knowledge about the specific task. This implies by no means that the "natural language" problem should be resolved. In fact, epistemic actions may be entered using a restricted pseudo-natural language or even a graphics language for defining links and nodes in a graph. Ergonomic studies are needed to evaluate if graphical input of epistemic actions is appropriate.

The model built by the interface using this strategy is nothing more than what the user decides to share. In other words, the model depends on the user's evaluation of the appropriateness of sharing knowledge. This means that an incomplete or even erroneous model may be built. However, the important issue here is not that the model be representative of reality, but of the knowledge communicated by the user. Studies show that, in task-oriented cooperative dialogues, task experts don't communicate all of their knowledge to their collaborator, but a partial and simplified version.

Another strategy for learning about user perspective derived from our analysis is to provoke epistemic breakdowns. We observed that human collaborators feel that, a although costly in time and cognitive load, epistemic breakdowns benefit task execution in the long run. This observation is paradoxical, because breakdowns are usually the manifestation that something went wrong in the communication. It would seem that epistemic breakdowns, on the contrary, are productive in the long run. Transposed to HCI, this means that an interface acting as a communication partner should be able to interrupt task execution in order to acquire more knowledge when it is appropriate and beneficial. For example, if a sub-task resembles another sub-task executed earlier, the interface should ask the user if both sub-tasks are instances of the same generic sub-task, and modify its model of the user's perspective accordingly. The problem with epistemic breakdowns is, of course, their appropriateness. The interface should be able to evaluate when such a breakdown is beneficial. Exploration of epistemic breakdowns as well as dialogue strategies which make use of epistemic knowledge are two of our research avenues for the future.

REFERENCES

1. Austin, J.L. *How to do things with words*. Oxford University Press, Oxford, 1962.

2. Carberry, Sandra. *Plan Recognition in Natural Language Dialogue*. MIT Press, Massachusetts, 1990.

3. Clark, Herbert H. and Wilkes-Gibbs, Deanna. Referring as a collaborative process. *Cognition*. 22 (1986), pp. 1-39.

4. Clark, Herbert H. and Schaefer, Edward F. Contributing to discourse. *Cognitive Science*. 13 (1989), pp. 259-294.

5. Grosz, Barbara and Sidner, Candace. Attention, intention and the structure of discourse. *Computational Linguistics*. 12, 3 (1986), pp. 175-204.

6. Isaacs, Ellen A. and Clark, Herbert H. References in Conversation Between Experts and Novices.

5

Journal of Experimental Psychology. 116, 1 (1987), pp. 26-37.

7. McCoy, Kathleen F. Generating responses to property misconceptions using perspective. In Reilly R.G. (ed.). *Communication Failure in Dialogue and Discourse*. Elsevier Science Publishers, North Holland, 1987, pp. 149-160.

8. Moran, T.P. Getting into a system: External Task Mapping Analysis. *CHI'83 Conference Proceedings*, dec. 12-15, Boston, Mass.

9. Norman, D. A. and Draper, S.W. (eds.). User Centered System Design. Lawrence Erlbaum, Hillsdale, NJ, 1986.

10. Ozkan, Nadine and Caelen, Jean. Vers un modèle de dialogue adaptatif. *Proceedings of the Second International Conference on Interface to Real and Virtual Worlds*. Montpelier, France, march 1993, pp. 77-84.

11. Self, John. Computational Viewpoints. In Moyse, Rod and Elsom-Cook, Mark (eds.). *Knowledge Negotiation*. Academic Press, NewYork, 1992, pp. 21-40.

12. Wahlster, Wolfgang, and Kosba, Alfred. User Models in Dialogue Systems. In Wahlster, Wolfgang, and Kosba, Alfred (eds.). *User Models in Dialogue Systems*. Springer-Verlag, NewYork, 1989, pp. 4-33.

6