Learning how to use a command device : a problem solving approach

J.F. Richard

INRIA

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 709

LEARNING HOW TO USE A COMMAND DEVICE : A PROBLEM SOLVING APPROACH

Jean François RICHARD

JUILLET 1987

LEARNING HOW TO USE A COMMAND DEVICE: A PROBLEM SOLVING APPROACH

RESOLUTION DE PROBLEME ET APPRENTISSAGE DE DISPOSITIFS DE COMMANDE

Jean François RICHARD

UNIVERSITE PARIS VIII
ET PROJET DE PSYCHOLOGIE ERGONOMIQUE
POUR L'INFORMATIQUE, INRIA, FRANCE

RESUME

L'apprentissage de l'utilisation de dispositifs de commande (traitement de texte, calculette, gestion de fichiers) peut être vu comme un cas particulier des problèmes de transformation d'états au même titre que des problèmes classiques tels que: Tour de Hanoï, jarres, missionnaires et cannibales. Ces situations en effet ne deviennent des problèmes que parce qu'on introduit des contraintes qui interdisent l'utilisation des procédures connues. De la même façon, apprendre à utiliser un dispositif consiste à trouver comment réaliser à l'aide de ce dernier ce qu'on sait faire à la main. Ici les contraintes sont les opérations de base du dispositif et ses règles de fonctionnement.

L'apport de l'approche résolution de problème se situe à trois niveaux:

- La méthodologie.
  Celle-ci se caractérise par l'utilisation de situations d'apprentissage par l'action, l'observation approfondie des tentatives de solution, l'analyse des protocoles individuels en vue d'identifier les objectifs d'action et les connaissances qui déterminent les comportements, une modélisation utilisant des formalismes tels que les systèmes de productions pour valider cette analyse.

- Les processus d'apprentissage.
  La psychologie de la résolution de problème a fourni des informations sur deux processus fondamentaux: la planification de l'action grâce à la construction de buts intermédiaires, le transfert analogique de procédures qui sont la solution de situations similaires (Hoc, 1986).
  Ces processus permettent:
  . de rechercher et d'expliquer comment se fait la construction des objectifs d'action à partir des connaissances et de la représentation du problème qu'ont les sujets;
  . de fournir une explication de la genèse des erreurs (par non prise en compte des prérequis, des effets multiples d'une commande);
  . de permettre un diagnostic des erreurs et donc d'élaborer des systèmes d'aide à l'apprentissage.

- La problématique de l'analyse de la compatibilité du système de commande avec la représentation de la tâche chez l'opérateur.
  Dans l'utilisation d'un dispositif deux types de connaissances sont nécessaires:
  . des connaissances pour l'utilisation permettant de savoir comment réaliser des objectifs,
  . des connaissances sur le fonctionnement permettant de comprendre ce qui se passe en cas d'incident (le résultat n'est pas celui escompté) et en quoi ce qu'on a fait est inadéquat.

  L'apprentissage consiste à:
  . construire une sémantique du dispositif: apprendre quelles sont les opérations permises, leurs effets et leurs contraintes d'application (prérequis et éventuellement postrequis);
  . établir une relation entre cette sémantique et la sémantique de l'action qui est à la base de la réalisation manuelle de la tâche.

Etant donné ce que l'on sait du transfert analogique, on cherche à avoir la meilleure correspondance possible entre les deux. Les règles de fonctionnement du dispositif limitent évidemment beaucoup la définition des sémantiques possibles pour ce dernier. Par suite, pour l'apprentissage d'un dispositif, il s'agira de définir une sémantique du dispositif qui ait la meilleure compatibilité possible avec la sémantique de l'action et de faire acquérir cette sémantique. Pour la conception de langages de commande il s'agira de définir un dispositif dont le fonctionnement permette de définir une sémantique qui offre une bonne compatibilité avec la sémantique de l'action.

SUMMARY

Learning how to use a command device (text editor, pocket calculator, file manipulation) may be seen as an example of a problem solving task.

Tasks such as Tower of Hanoï, missionaries and cannibals are problems only because constraints are added, which make known procedures not applicable. Similarly learning how to use a device amounts to finding a new way of doing well-known tasks. In this case constraints are basic commands of the device and the way they operate.

The problem solving approach yields a contribution at three levels:

1. A methodology for empirical research

This methodology consists in devising learning by doing situations, studying solution attempts, analyzing individual protocols in order to identify which goals are set up and what knowledge is used, building models such as production systems which provide a simulation of behavior.

2. Hypotheses about the learning process

The information processing approach in psychology has given some insight into two major learning processes: planning of action through elaboration of subgoals and analogical transfer of known solutions.

These processes allow:
- to find out and explain how goals are built from what subjects know and what they understand about the task,
- to explain how errors are generated (prerequisites not taken into account, multiple effects of commands, etc.),
- to make a diagnosis of errors and allow to devise systems able to guide the learning process.

3. A way of analyzing the compatibility between the device model and the user's model

Two types of knowledge are involved: knowledge about how to do in order to complete definite goals and knowledge about how it works in order to understand what happens in case of unexpected events and what ought to be done instead.

Learning consists in:
- building a semantics for the device (operations allowed and prerequisites for their application),
- relating this semantics to the semantics of actions used in familiar manual tasks.

We want the compatibility to be maximal between both semantics. Of course the way the device works limits the possible semantics for the device. As a consequence:
- designing methods for learning amounts to define a semantics for the device which provides maximal compatibility with the user's model of action,
- designing a command language amounts to conceive a device allowing the maximal compatibility with the user's model of action.

SOMMAIRE

INTRODUCTION

The major problems in learning how to use a command devise are problems of representation : representation of the task, representation of the device by the user and problems of compatibility between these representations.

The way a representation of the task is built in a problem solving situation is one of the main topics of research at present time. My argument is that we can gain insight into some difficult problems of learning how to use a command device by considering results which are provided by research in problem solving tasks.

Actually learning how to use a command device may be seen as an example of a problem solving task.

Tasks such asTower of Hanoï, missionnaries and cannibals are problems only because constraints are added such that known procedures are not applicable. Similarly learning how to use a command device amounts to finding a new way of doing well known tasks. Everybody knows how to complete a calculus by hand, how to write a text and make corrections on it. When the task is completed by means of a command device, it becomes a problem because of the constraints of the device, which are due to the way the commands operate. The user has to find out a new way of completing the task which is consistent with the constraints of the device. The way a device is conceived by a user has much to do with the way a user understand instructions in a problem solving task.

1. BASIC QUESTIONS ABOUT THE REPRESENTATION PROCESS

1.1 Learning by doing and learning from text: to what extent is it possible to learn through a guidebook ?

It is well known that people do not use guidebooks, especially people which are not familiar which electronic devices. They apparently prefer to make trials and learn through interaction with the device. This has been noted many times (for example Lewis, Mack 1982).

A first reason may be that guidebooks are inadequate, that they are not written in a proper way and do not contain useful information. But there is perhaps another reason, which is more fundamental. It is possible that it is not the case that people learn from text and apply this knowledge to plan the actions. It is possible that the basic process is learning through action and that what is learned from the text plays only an auxiliary part.

In a study (Friemel, Richard, Silvart, Weil-Barais 1982) we asked students to find out the way to compute the value of the sum of squares of a distribution, using the standard formula:

$$\sum n_i x_i^2 - \frac{(\sum n_i x_i)^2}{n}$$

x_i being each value of the distribution, and n the number of values.

The subjects used a pocket calculator with one cell of memory in which the operations had the same priority as in algebra. Before calculation they had studied a guidebook specially worked out for the experiment and containing all the necessary information.

They were told to find out a solution involving a minimum number of keystrokes. The solution was:

Method	Example
a - entering a number on the key-board	13
b - summing up the number in memory	SUM
c - squaring this number	x^2
d - using operator +	+

and again a - b - c - d for each number to be summed up.

About one hundred students completed the task and no one found this method in these conditions: they computed the sum of squares first and then the sum of numbers. They were then instructed to enter each number only once. About 40% of the subjects discovered the method by trial and error. They tried about every possible combination of commands and observed the result.

What should be emphasized for the present topic is that no one was able to deduce from knowledge about the way the commands were operating that it was possible to apply two operators to the same number: one operating on the memory cell, the other one on the visible register.

It may seem surprising why such knowledge, very basic indeed and very important for a user, is not deduced from the knowledge of the way each command operates. My claim is that a question underlies this simple observation, a question which is basic in device utilization and problem solving. In both cases information is given in one form and has to be transformed into another form in order to be used efficiently.

Let me take as an example the famous 3 monsters problem (Simon Hayes, 1976). A recent experience by Kotovsky, Simon and Hayes (1985) yields a paradoxical result. Subject are given a problem which requires 5 steps and which has a subgoal which may be attained in 2 steps. When told which subgoal has to be completed first, subjects take no advantage of this information: it takes quite a long time to solve the problem (15 to 30 minutes). This is not observed with the Tower of Hanoï problem: giving information about subgoals facilitates solution.

In order to understand why it is so, we are undertaking observations with a problem, the puppets problem, which is isomorphic to the 3 monsters problem and differs from it only in that the presentation is made more similar to the Tower of Hanoï problem (table 1).

PUPPETS

Imagine the following game:
There are 3 puppets: a large one, a medium one, a small one.
Each has a ball. This ball may be large, medium, small.
The large P. has a medium ball.
The medium P. has a small ball.
The small P. has a large ball.

The size of the balls may be changed but it is necessary to follow these rules:

1 - only the size of a ball may be changed at a time
2 - if two balls have the same size it is not allowed to change the size of the ball of the larger puppet
3 - one cannot give to a ball the same size as a ball held by a smaller puppet

The goal is to have:
the large puppet with the large ball
the medium puppet with the medium ball
the small puppet with the small ball

Table 1. The puppets problem

Preliminary observations suggest that the subjects are unable to plan what to do in order to obtain a goal until they have formulated the information in the following way. In order to be able to change the size of the ball held by one puppet, the size of the balls held by the smaller puppet has to be:
- different from the present size of the ball to be changed,
- and different from the new size to be given to this ball.

This formulation tells what are the prerequisites of the action of changing the size of a ball. When one has as a subgoal to change the size of a ball, this formulation allows to define which situation would permit this change: it is the situation which differs from the present one by the prerequisites which are not satisfied. This formulation is equivalent to what is said in the instructions which tell in what cases a change is allowed or not. This transformation usually requires a great deal of trial and error and many violations of instructions.

What is interesting is that, while the second form may be deduced directly from the first, that is not done by subjects: they need to observe the pattern of sizes of the balls which allows to change the size of the ball of the largest puppet from medium to large, that is the situation where the small puppet has a small ball, the medium puppet has a small ball, the large puppet has a medium ball.

In that situation subjects may observe that the balls held by the smaller puppets have a size which is different from the actual size of the ball of the largest puppet and different from its new size. So they may hypothesize that this is a prerequisite of the action of changing the ball of the largest puppet from medium to large. This new formulation makes explicit which prerequisites must be satisfied in this context in order to realize the goal of changing the size of a ball.

So the difficulty of planning moves using a subgoal may be that the subject has to discover what are the prerequisites of a move and that is done by observation of situations where it possible to complete the move.

My hypothesis is that in a similar way a subject learning how to use a command device has to discover which conditions have to be satisfied in order to attain a goal by means of a given command: that is not deduced from reading the guidebook but requires experience.

1.2 Procedural knowledge and declarative knowledge

The second idea is that it is not enough to give procedures, to explain what to do in order to attain some goals. It is necessary to provide informations which can make clear the reasons why the procedure works and gives the expected result.

An usual practice is to present a procedure through an example. This is a general practice in guidebooks explaining how to use pocket calculators. In this way difficult questions can be escaped particularly the fact that in a pocket calculator having the same laws of priority as in algebra the commands corresponding to the signs: **+ - x :** do not behave only as arithmetic operators. They have another property: they cause an operation of the same level of priority or of a higher level to be executed.

Thus **+** or **-** may have a similar effect as **=** . Is it a good pedagogical strategy to mask this information?

Actually users learn very quickly by observation that to get the result of an operation, it is not necessary to press = and that the same effect may be obtained by an operator like **+**.

But as it is equally true that + has another effect (it is an arithmetic operator and is used to delimitate the first operand and the second), it may be inadequate to use a + command in the same way as an = command.

This use leads to accidents such as the following one.

In order to add 3,5 and 4
one may proceed as follows:

3 + 5 + 4 +

Suppose now one wants to pursue the computation: square the sum and divide by 3.

$3 + 5 + 4 + x^2 / 3 \quad =$

$12 \quad + 144/3 \quad = 60$

The result is 60 and not 48 as expected.

Many users have certainly had this experience. I happened to observe a student who had proceeded in this way: he was very puzzled and, although he was very bright, he was unable to understand why such a thing could happen.

Users do not necessarily follow the prescribed procedures: they build new ones which seem to them equivalent but which in fact are so only in some contexts. That leads inevitably to incidents such as the one described.

To prevent these incidents guidebooks sometimes contain prescriptions (when you end a computation use =) or interdictions (within a subprogram never use =, use always parentheses). The reason is that = has a lower level of priority than parentheses and causes the execution of operations beyond the subprogram. The user is usually bothered by such recommandations that he cannot understand.

The combination of accidents and of such mysteries has the effect that users do not feel safe when using calculators or similar devices: they leave the sophisticated procedure and are satisfied with the most simple ones.

1.3 Learning by analogy

The third idea is that when having to learn a new procedure subjects refer to a known procedure whenever it is possible. Let us take as an example an experiment by Hoc (1980, 1981). The task was updating a stock in a shop. In the first phase of the experiment the subject could see three tables: a table presenting the old state of the stock with items numbered in order, a table indicating for each item how many moves had been observed (which sign + or -) a table presenting the new state of the stock.

There was the following set of commands:

ENTER A (which allows to enter into memory register A the content of the first current line of the first table, number of the item and quantity in the stock)

ENTER M (which allows to enter into the memory register B the content of the first current line of the second table, number of the item and quantity added or removed)

ADD (which allows to sum into memory register A the quantity in the memory register A and the quantity in memory register B)

COPY (which allows to copy into the third table, the content of the memory register A)

In the first phase subjects could see the three tables and the content of memory register A and memory register B.

In this first phase the subjects transposed the procedure they had when completing the task with paper and pencil. Items were categorized into three types:

- 1 items with no move
- 2 items with one move
- 3 items with several moves

A special procedure was devised for each case, taking into account the constraints of the device.

1 - ENTER A	COPY		
2 - ENTER A	ENTER M	ADD	COPY
3 - ENTER A	ENTER M	ADD	
	ENTER M	ADD	

	ENTER M	ADD	COPY

The second phase will be described later. In the third phase subject had another command giving the result of the test A = M?, "yes" meaning that the number of the item in the register A is the same as the number of the item in register B, "no" meaning that the numbers are different. But they could not see neither the table neither the contents of memory registers A and B.

Under these new conditions it is necessary to reformulate the problem in the following terms:

either the updating of an item is terminated
or is not terminated

If the result of the test is "no", it means that updating is terminated
so what is to be done is

COPY
ENTER A

if it is "yes" then updating is not terminated, what is to be done is

ADD
ENTER M

In the second phase of the experiment the tables were not visible but the contents of the memory registers A and B were visible.

In these conditions it was possible to maintain the goal structure of the former procedure (3 classes). However this was extremely difficult: it was necessary to identify to what category the item was belonging and to devise the appropriate sequence of commands.

In spite of this, subjects as a rule maintained the categorization into 3 classes. They did not try to reformulate the problem into 2 categories although they had all the information available for that.That shows that subjects maintain a procedure as long as they can, even if this requires to complexify the procedure in order to face the new requirements of the situation. In order to obtain a restructuration of goals, it is necessary to add constraints such that the old procedure cannot be maintained.

We have seen previously another example of this phenomenon: in order to discover that on a pocket calculator two operators could operate on the same number subjects had to be instructed not to enter each number twice.

This is a typical instance of analogical transfer. Another instance is when the effects of commands are interpreted as being analogous to the effects of another known command.

As an example, we have observed children maintaining a very long time an erroneous interpretation:
SUM is like + but operates in memory
RECALL is like = , it makes the result of the operation made in memory to appear in the visible registrer.

The command = used after a sequence of operations done with arithmetic operators is thought not to cause the execution of the operations but to cause the result of these to appear in the visible register. Using SUM as + causes some trouble but RECALL may be interpreted as showing on the visible register the result of the operations done in memory, in a similar way that = is thought to operate.

It is quite easy for children to learn to use a pocket calculator to compute a sequence of additions and substractions. This may be explained because it is possible to have a correspondence between:
+ and adding a quantity
- and removing a quantity
= writing the result

Analogical transfer is a very powerful mechanism in problem solving and this effect has to be taken into account in designing of devices or in teaching how to use a device.

1.4 Which model of the device is to be given to the user?

The question is not: is it useful to give a model to the user or is it enough to give him procedures? Anyway the user builds a model of the device. It seems reasonable to teach a model, but what type of model?

A first possibility is to give the user an analogical model by referring in a systematic way to the knowledge already acquired. This approach has been proposed by Mayer (Mayer, 1981; Mayer, Bayman, 1981) and is based on the ideas developed by Ausubel that learning has to be related to existing structures. Mayer has obtained some positive results in simple programming tasks: the effects of instructions were systematically referred to the effects of actions done when one uses cards arranged in files. Actions considered were exploration, comparison, selection and so on.

Use of analogy has been criticized by Halasz and Moran (1982) on the grounds that analogy works on some aspects but not all. Morever it is not possible to know for which aspects analogy are good and for which aspects analogy are bad, so that using analogy may lead to misconceptions that are difficult to eradicate.

A second possibility is to devise a conceptual automaton and to describe the effects of commands as changes in the state of the automaton. These changes affect the state of information in the registers of the automaton: duplication, transfer, deletion and so on.

I think that a good theoretical argument in favor of such a model is that a given command may be used for different uses which may be quite opposite. To show this, let us take as an example the function STO in a pocket calculator. This function duplicates into memory the number standing in the visible register and deletes the former content of memory.

This function may be used to save a result (analogous to write something on a piece of paper) and to empty memory (analogous to rub out).

It may seem contradictory that the same action may realize opposite goals: fill the memory, empty the memory.

In order to avoid the contradictions it is necessary to give a description of the effects of STO in terms of what changes are produced on the information in the device. So one may understand that if number zero is standing in the visible register and there is a number in memory, STO may be used to empty memory, and that if a number different from number zero is standing on the visible register, STO may be used to save this number into memory.

As a rule it is important to distinguish the descriptions of the objects by their properties and the description of the different uses they may provide (their functionalities).

This is true for the objects we manipulate, it is equally true for systems of information processing. The semantics of how it works have to be distinguished from the semantics of what it may be used for (Richard, 1982).

In that respect the problem of what name to choose to designate a function is important. My position is that many problems are generated for the user when terms appropriate for naming goals of the user are used to designate functions in the device. Let us consider the following example: in a well known text editor (WORD) the function which moves a part of a text from the working register to the buffer is named "delete", the same function may be used to move a part of a text from one place to another: the commands to be used are delete and insert (at the new place).

The result is that almost half of the beginners we have observed do not use delete to transfer a portion of the text to another place. Many of them prefer to use COPY, which copies the text into the buffer, INSERT, which copies the text at the new place, and DELETE which deletes the text from the old place.

It seems contradictory to use "delete" as a subgoal for transfering a text.

2. COMPATIBILITY OF THE MENTAL MODEL OF THE USER WITH THE CONCEPTUAL MODEL OF THE DEVICE

I wish to emphasize that there are 3 terms to be considered:
- the goals of the task
- the operations of the device
- the semantics of elementary actions belonging to the domain of manipulation.

Studies have privileged the relation between two of these terms and have neglected the third.

2.1 Relations between the goals of the task and the operations of the device

Card, Moran and Newell (1983) have developed a model for interaction named GOMS (for Goals, Operations, Methods, Selection).

Given a task (for instance replace a letter by another) given a device, it is possible to define a production system which provides an adequate realization of the task. This system is made of rules which connect goals with operations of the devices and operations with methods and which define in what circumstances a method has to be preferred to another.

Comparing two devices over a set of tasks amounts to comparing the production systems necessary to complete the tasks. The best device is that which corresponds to a production system of minimal complexity.

The complexity of the production system is estimated by the number of rules to be run and by the estimated time necessary to carry out the operations defined in the rules.

Good correlation have been observed between this measure of complexity and the time necessary for experts to complete the task.

A similar approach has been developped by Kieras and Polson (1985). They present a formal analysis of user complexity. This analysis has two aspects:
- representation of the task
- representation of the device

The representation of the task has the form of a production system. This system provides a goal structure (a set of subgoals) to complete the task, and expresses action rules to realize each subgoal.

The goal structure may be represented by a procedural net.

The representation of the device is made through an augmented transition network with recursive properties.This is a way of describing an automaton. I shall not detail this description which is rather specialized. Let me say that this presentation aims at showing the structure of the device. The ATN model is summarized within a procedural net for each function and this function is compared with the procedural net drawn from the representation of the task.

2.2 Direct manipulation

This approach makes a maximal case of the signification conveyed by actions and gestures.

This approach has developped rapidly from the apparition of input devices other than keystrokes such as: mouse, sensitive screens.

A typical example is the command to displace an icon. The steps are:
- using the mouse, place the pointer on the object to be displaced,
- click to "seize the object"
- place the pointer at the place when the object has to be displaced.
- click to "put" the object at the place.

The analogy with the manual action is almost complete.

According to Norman (1986) the direct manipulation approach aims at bridging two gaps:
- the gap between intention and execution,
- the gap between the effects the action has on the device and the interpretation given by the user.

This approach is very interesting and has emphasized the importance of action stereotypes.

The question is: is it possible to make the economy of a model of the display?

The difficulty is that in an information processing system it is necessary to express significations which are expressed quite clearly in the meaning of words but quite poorly by analogical tools such as gestures.

For instance how can we represent in a simple way the difference between "move" and "copy"?
My position is that the basic components of signification we have to consider are those which are associated with gestures and those which are associated with the words which designate actions.

2.3 Some suggestions

I think it is necessary to build a language to describe devices.This language has to be distinct from the language of action but it must keep a correspondence with the basic meanings in the semantics of action.The first question to answer in order to do this is: what semantics of action consist of?

2.3.1 Semantics of action

Some ideas about this question have been presented in a recent paper (Richard, 1986).

The argument is that there are three components in our knowledge of action: the result of the action (the resulting state of the world) the process (the way or the ways through which this result is obtained), the prerequisites or enabling conditions of the process.

This knowledge may be expressed within schemas similar to those which have been used in "language comprehension".

Let us take as an example a possible schema for MOVE

MOVE X from Y to Z
Result: X is in Z
X is not in Y
Process:

1. take X from Y
   prerequisite: 1. to be close to Y
   2. there is nothing on X

2. lay down X at Z
   prerequisite: 1. to be close to Z
   2. there is place on Z to lay down X

This information is supposed to be processed by a planning process which proceeds depth first.

This process computes what to do according to circumstances depending on the fact that prerequisites are satisfied or not.

If no prerequisite is satisfied, we will have:
go close to Y
remove from X what is on X
take X from Y
go close to Z
make place on Z
lay down X on Z

This knowledge about action is at the semantic level not at the motor level. If we think of the example given for moving an icon with a mouse, the correspondence between steps of the action in the device and the components of the representation in the schema are saved as well as in the case of the motor schema.

2.3.2 The correspondence between the language of action and the language of the device

A correspondence must be maintained at three levels between the language of the device and the language of action: results, process and prerequisites. At the level of the result, the problem is perhaps not too difficult: it is necessary to define states of information in registers which may be interpreted as equivalent to results of action in usual world.

In editing tasks, examples of correspondence between one language of action and the language of the device are:

Language of the device	Language of action
enter information into a register	write
delete information from a register	rub out
transfer	move
duplicate	copy
move a pointer	show an object
substitute a caracter for another	replace

It is important to keep the correspondence at the level of procedure and prerequisites of the procedure.

It is also important to preserve the goals structure of the procedure, that is the list of subgoals in the order. It is not so much important to maintain the way each subgoal is realized.

As we have shown,what is difficult to do is to change the goal structure, it is not so difficult to learn a new way of realizing a subgoal.

To return to our example with MOVE applied to an object an icon, a text: it is important to maintain the two subgoals in order

take

lay down

The way take and lay down are realized as not so important. This may be by using a mouse or a key function: the important thing is that a correspondence may be established with two functions the result of which may be related to "take" and "'lay down".

It is important also that the systems allow to execute the action by using a macro command such as "move" or by using directly more elementary functions corresponding to take and lay down. It is natural in fact to think of an action either by thinking of its result (move) or by thinking of the way it will be realized (take and lay down).

In some case of course it may be difficult to maintain a perfect correspondence: the conditions of execution are not alike in a device and in the usual world.

For instance where it is necessary to operate on an object, one has to proceed to a delimitation of this object, for instance delimitation of a part of a text.

Is it better to select the object first and then indicate what function will be applied to the object or the reverse: define the function first and the object second? A study by Dixon (1982) seems to indicate that the second solution would be better but further research is necessary.

I think that the problem of correspondence between the elementary actions and the functions of the device has to be studied within the framework of analogical transfer.

In that respect the reason why I think that it is important to maintain a correspondence between the goal structure of the process of action and the functions of the device is the fact that one may change drastically the effects of transfer by changing the semantics of the action.

Among the isomorphism of the Tower of Hanoï problems studied by Kobovsky, Simon and Hayes (1985), there was a transfer problem and a change problem.

Globes are transferred from one monster to the other
Globes change size by schrinking or expanding.

A transfer problem is much easier. The reason is probably that transfer is an action which may be decomposed into 2 component actions, take and give, which make the problem similar to the Tower of Hanoï problem. On the other hand "change" cannot be divided into component actions.

An interesting result is given by an experiment in which the same authors had another context for change. There were 3 dishes with one ball in each and behind each dish there was a reserve dish in which there were other balls. For instance if a dish had a ball of median size, the corresponding reserve dish had a large and a small ball. Changing the size of a ball was obtained by exchanging the ball in the dish and one of the balls in the reserve dish.

Under these conditions the difficulty of the problem is similar to the difficulty of the transfer problem. The reason is that the semantics of the change action has been modified: change may be conceived as a list of two component actions: remove the ball, take another ball.

Transfer between isomorphic problem tasks using different contexts seems to be very sensitive to the difference of representation of primitive actions and for this reason it is probably a good method to study the primitive semantic components of action.

Such a study should permit to choose which action to take as reference in order to build the basic functions of a device and the basic language to describe a device.

If it is possible to keep a close correspondence between the operations of the device and these basic actions, a correct model of the device may be easy to learn and there will be little discrepancy between the conceptual model of the device and the mental model of the user.

2.3.3 The correspondence between goals of the user and the functions of the device

The second problem, the relation between the goals of the task and the operation of the device will then be solved rather easily. The problem will be to give a description of goals in terms of functions related with primitive actions. In that way it will be possible to think of subgoals in terms of these primitive actions which are related.

For existing devices it is quite difficult to establish a correspondence which is satisfactory. We may hope that if we are able to state precisely what the basic operation of a device should be, changes in the conception of devices could take place such that the correspondence may be improved.

CONCLUSION

We have to define a basic language to describe devices and to define basic operations of a device. These operations which may become meaningful thanks to their correspondence with basic primitive actions.

My opinion is that this language should be general so that it would be applied to different devices in different tasks: text editor, calculators, question answering systems, and so on.

It is not reasonable to require from the user that he/she changes his/her model of the device when going from one machine to another.

One of the major difficulties at the present time is that the user cannot build a model which can be transferred from a machine to another one. I am convinced that a large set of common functions could be used to describe very different devices if one consider the type of tasks they are used for.

REFERENCES

CARD, S.R., MORAN, T.P., NEWELL, A. (1983) The psychology of human-computer interaction. Hillsdale N.J. : Erlbaum

DIXON, P. (1982) Plans and written directions for complex tasks. Journal of Verbal Learning and Verbal Behavior, 21, 70-84.

FRIEMEL, E., RICHARD, J.F., SILVART, L., WEIL-BARAIS, A. (1982) Elaboration d'une procédure de résolution compatible avec les contraintes d'un dispositif. Rapport de fin de contrat ADI. Université Paris VIII.

HALASZ, F., MORAN, T.P. (1982) Analogy considered harmful. Human Factors in Computer Systems. Proceedings of the ACM Gainsburg, March 1982, 383-386.

HOC, J.M. (1980) Un exemple de dispositif informatique expérimental pour la psycho pédagogie de la programmation. Informatique et Sciences Humaines, 44, 67-86.

HOC, J.M. (1981) Une approche diachronique de la résolution de problème. L'apprentissage de la programmation informatique. Psychologie française, 26, 3-4, 182-192.

KOTOVSKY K., HAYES J.R., SIMON H. A. (1985) Why are some problems hard? Evidence from Tower of Hanoï. Cognitive Psychology, 17, 248-294.

KIERAS, D., POLSON, P.G. (1985) An approach to the formal analysis of user complexity. International Journal of Man Machine Studies, 22, 365-394.

LEWIS, C., MACK, R. (1982) Learning to use a text processing system: evidence from "thinking aloud protocols", Human Factors in Computer Systems. Proceedings of the ACM, Gainsburg March 15-17, 1982, 387-392.

MAYER, R.E. (1981) A psychology of how novices learn computer programming. Computing Survey 13, 1, 121-141.

MAYER, R.E. BAYMAN, P. (1981) Psychology of calculator language: a framework for describing differences in user's knowledge. Communication of the ACM, 24, 8, 511-520.

NORMAN, D.A. (1986) User centered computer design, Hillsdale, N.J.: Erlbaum.

RICHARD, J.F. (1985) Logique du fonctionnement et logique de l'utilisation. (Rapport de recherche n° 202) Rocquencourt: INRIA.

RICHARD, J.F. (1986) Semantics of action: its processing as a function of the task. (Rapport de Recherche n° 542) Rocquencourt: INRIA.

SEBILLOTTE, S. (1983) Etude préliminaire du travail de secrétariat dans un service hospitalier spécialisé. (Rapport de Recherche n° 30) Rocquencourt: INRIA.

SEBILLOTTE, S. (1987) La planification hiérarchique comme méthode d'analyse de la tâche, analyse de tâches de bureau. (Rapport de Recherche n° 599) Rocquencourt: INRIA.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique