

6 *Design Strategies*

Can Baykan

Bilkent University, Ankara, Turkey

The aim of the study reported in this paper is to account for the design behaviour of the individual subject. In order to carry it out, we use an analysis of the bike rack design task and a set of hypothesized primitive information processes. These are derived from an analysis of the group design protocol. The decision to use the group protocol for identifying the primitive information processes and doing a task analysis is made before looking at the data, and is based on the expectation that the group protocol would be more complex due to interactions between the group members.

The protocols are first segmented and then encoded in terms of a set of primitive information processes based on the information that is needed. The hypothesized information processes are task dependent. The verbalizations in the protocols indicate the inputs and outputs to the processes which are in short-term memory, rather than the processes themselves.

These processes and information structures identified and used in the analysis are a hierarchy of goals and subgoals, generate and test actions, methods which organize them, constraints and information management and time management. Goals are the most important elements in determining behaviour. They lead to the selection of appropriate problem spaces and methods and information and time management functions. These primitive information processes are sufficient to account for the segments in the individual's protocol. Other aspects of design, such as recognition processes, using intuitive constraints and accessing information from LTM via associations, are not accounted for in this protocol analysis.

1 Background

The first comprehensive work on the assumptions and principles of cognitive science and protocol analysis methods was written by Newell¹. The advances in protocol analysis methods since then are covered in Ericsson and Simon's work², which contains an in-depth exposition of protocol analysis.

The initial application of protocol analysis to design was by Eastman³. In this work he analysed the behaviour of a designer working on a simple spatial layout problem, the configuration of a bathroom. Akin has done extensive cognitive research on different aspects of design, including identification of information processing primitives for architectural design⁴, problem structuring in design⁵ and modelling design⁶. Goel and Pirolli⁷ discuss the characteristics of problem spaces which follow from the nature of design problems and hypothesize that these characteristics are valid across different design domains.

2 Method

In this paper, our aim is to account for the individual designer's protocol. We do a task analysis of the bike rack design problem and formulate hypotheses about a set of primitive information processes by analysing the design team protocol. The primitive information processes are formulated on the basis of an analysis of the heeded information revealed in the protocol in the context of task analysis. We then use these hypotheses to analyse the individual designer's protocol.

Segments are verbalization units that correspond to units of heeded information, pauses and syntactic information. The segments may be sentences, clauses, phrases or words². We segment the whole protocol first and try to encode each segment using the hypothesized processes. Encoding a segment sometimes requires taking its context into account when the information contained in it has references or is fragmentary, but we try to keep the context as narrow as possible.

The other, larger unit of behaviour that we use in the analysis of the protocols is the *episode*, which is a collection of segments. An episode is a succinctly describable portion of behaviour¹. The episodes are identified after encoding the protocol. Looking at the episodes, it may be possible to see the overall structure of behaviour in an aggregate form and to account for individual differences.

The decision to use the design team protocol for doing a task analysis of the design problem used in the experiments and for

formulating primitive information processes was made prior to looking at the data. This was done because we expected that analysing the problem-solving behaviour of the team would also involve looking at other issues, such as group interactions. We use the team protocol for formulating the hypotheses and then use these for analysing the individual designer's data. We segmented, encoded and identified episodes in the team protocol. While doing this we did not try to account for everything in the protocol – but we tried to account for most of it. The process of segmenting, encoding and aggregating into episodes was repeated for the individual designer's protocol.

3 Task Analysis

Task analysis produces a description of the constraints on behaviour that must be satisfied to solve the problem at a given level of intelligence, where intelligence is defined as the ability to use knowledge to solve problems. Certain paths in the task environment are not available as paths to the goal for a given level of intelligence because they are too difficult¹. The definition of intelligence given above supposes that the problem solver has knowledge and can bring it to bear on the problem. Expertise in some domain means having extensive and structured knowledge about the domain that can be applied easily.

One of the difficulties of research on design thinking is that task analysis is not definitive, for example as in chess, cryptarithmic or other puzzles. We don't really know what it takes to solve the task. The same is true of evaluating designs. The solution is to rely on the opinions of experts in that field of design for resolving such issues. Since we are not experts in bike rack design, we use the design team protocol to do a simple task analysis.

The design problem used in the experiments involves finding a position for carrying the backpack on the bike, locations on the bike for mounting the rack, methods for attaching the rack to the bike and the backpack to the rack, and determining dimensions, materials and production methods. Cost, weight, structural strength and stability, ease of use and aesthetics are some of the important dependent variables. The dependencies between different parts of the problem and between the variables are complex. Determining the position to carry the backpack on the bike seems to be an independent decision, which affects mounting the rack to the bike. The method for attaching the backpack to the rack depends on the orientation of the rack. Ease of use and

aesthetics can be evaluated on conceptual designs, but cost, structural strength, stability and weight require the design to be quite complete before they can be determined. The methods that can be applied vary from subtask to subtask. Some aspects of the problem, such as position of the rack and determining the attaching points of the rack, require trial and error. Others, such as checking structural stability or estimating cost, have well-defined procedures that may be used.

In design tasks, the solution is not implicit in task information as in puzzles or other well-defined problems. Solving the problem requires identifying the knowledge that applies unless the problem solver is an expert in that design field. Even then, they may have to access information from outside sources. There is no correct solution to a design problem. Thus, the problem solver stops working on the problem when satisfied with the result, or more often when available resources, the most important of which is time, run out. Designers manage their time and tailor their efforts by taking available time into consideration.

4 Assumptions

The first assumption is that the subjects are information processing systems (IPS)¹. An IPS consists of a processor which carries out some set of primitive information processing operations, a short term memory (STM), an associative long term memory (LTM), and effectors and receptors for interacting with the environment. The capacity of the STM is estimated to be seven plus or minus two chunks¹ or four plus or minus two chunks² at the same time. The capacity of the LTM is practically unlimited, but especially fixing, and to some extent accessing information, is slow. Due to the limited capacity of the STM and the long times required for fixing information in the LTM, an IPS needs to use external memory (EM) as an aid during problem solving. EM is a visual display which consists of the notes of the subject. It is possible to think of the STM as consisting of the STM internal to the IPS plus the portion of EM in the subject's foveal view. The task-independent primitive information processes are reading and writing to EM, accessing and fixing information in LTM and making inferences. The STM is always under the direct attention of the problem solver.

An IPS solves problems in problem space(s) internal to the IPS. The problem space encodes significant information about the objective task environment and organizes problem-solving effort.

The task environment and the characteristics of the human IPS determine the possible structures of problem spaces that will be used in problem solving¹. A problem space consists of symbol structures, operators, an initial state, the problem and knowledge. The selection of problem spaces may imply a very large selection but not much processing, since the operators are built of familiar components¹.

The assumptions about the verbalizations in protocols are that a thought is verbalized as it is heeded. The inputs and outputs to the processes are in STM and are verbalized rather than the process itself. Types of verbalization depend on the time of verbalization and mapping from the heeded to the verbalized information².

5 Hypotheses

The primitive information processes and functionalities given below are formulated as a result of the analysis of the design team protocol, and will be used to analyse and explain the individual's protocol. They are goals, methods, generate and test actions, constraint and information management, time management, and focus of attention functions.

5.1 Goals

Goals come from the task or problem. They are not strictly determined by the problem definition, but have a rational relation to it. Problem decomposition takes place by setting up subgoals. As a result, there may be a hierarchy of goals and subgoals. A goal is capable of controlling behaviour by evoking a pattern of behaviour that has a rational relation to it. When a goal cannot be achieved immediately, i.e. by recognition or by a primitive information process, it requires a problem space to work in. Thus, goals define problem spaces.

The goal statement may be in the form of a question or a statement as seen in the protocol fragment in Figure 6.1. Every line in this figure denotes a separate segment, and every segment

00:06:00

K *what do we need?*

I guess we should look at their existing prototype, huh?

J ...

we could also just sort of like try to quantify the problem because what's your understanding of the problem first of all?

Figure 6.1
Goals and planning
example

happens to be a goal. Some of the goals are attended to immediately. The first segment is the reason for the generation of the rest. The last segment leads to behaviour aimed at defining the problem. Based on whether goals are attended to immediately or not, we can make a distinction between goals and plans. The first and last segments are goals and the second and third are plans.

5.2 Generate Operation

A generate operation is another primitive information process. It proposes an idea or a symbol structure that is a (partial) candidate solution for achieving the goal. The second, third and fourth segments in Figure 6.1 are verbalized as the result of generate operations.

Figure 6.2
Generating candidate
solutions – type g1

00:41:00
 I OK pack to rack Velcro um
 pack to rack gravity
 J gravity er
 K snaps
 straps
 you can have um bungee cords

Each segment in Figure 6.2 contains a possible method for attaching the backpack to the rack. A generated idea or symbol structure can be expressed by a single word, or it may also contain the reason for the structure being generated, as in Figure 6.3.

Figure 6.3
Generating candidate
solutions – type g2

01:52:00
 K I go at steel here for the for the stiffness

The two different types of generate operations, seen in Figures 6.2 and 6.3, will be termed type g1 and type g2 respectively. Type g1 can be formalized by indicating the proposed idea and what it is a solution for, i.e., $g1\{\text{attach backpack to rack} = \text{bungee}\}$, and type g2 by indicating a reason, i.e., $g2\{\text{material of tubing} = \text{steel} ; \text{stiffness}\}$. The reason is usually a constraint, which will be defined below. The formal descriptions are used for detailed encoding of the individual's protocol.

5.3 Test Operation

A test evaluates a proposed partial solution. The test may contain no reason, may contain a reference to a constraint or may

Figure 6.4
Type t1 test example

00:29:00
K *mm mm that'd be good*

Figure 6.5
Type t2 test example

01:58:00
I *OK time to remove less than thirty seconds yeah we'll be able to permanently remove it in less than thirty seconds*

compare two solutions with respect to a constraint. These are denoted as type t1, t2 and t3 tests respectively. A test segment is identified by a verbalized result, such as yes, no, good, etc. An example of a type t1 test is seen in Figure 6.4. It is formalized as $t1\{candidate\ solution\} \rightarrow result$.

The candidate solution that is being evaluated is not always specified, but it is possible to infer from context what it is most of the time. Sometimes it is possible to infer using context what the constraint used in the evaluation is, but at other times the verbalized result is based on an intuitive evaluation and the constraint cannot be specified exactly.

An example of a type 2 test is seen in Figure 6.5. It contains the result of the evaluation as well as the constraint that is used in evaluating a generated solution. Its general format is $t2\{candidate\ solution\ ;\ constraint\} \rightarrow result$.

Figure 6.6
Type t3 test example

00:30:00
K *that's not as bad (inaudible)*

01:52:00
J *and these things could be aluminium too they don't have to be steel they're easier to form if they're aluminum*

A type t3 test evaluates two proposed solutions with respect to a constraint to determine which one is better. Its form is $t3\{candidate\ solution1;\ candidate\ solution2;\ constraint\} \rightarrow result$. Two examples of a type t3 test are seen in Figure 6.6. The result of the evaluations in the protocol are binary, that is, yes or no in the case of t1 and t2, and selecting one of the two structures in the case of t3.

5.4 Methods

A method is composed of primitive information processes. It organizes generate and test actions to achieve the goal. Methods are

Figure 6.7
Method for
calculating
manufactured cost
from MSRP

J is there any sort of um cost specification for we know the sales price is fiftyfive dollars but um

...

X No, I'm asking for the my assistant to say again have to estimate their own ratios OK yep you have to estimate your own

...

J manufactured costs will be one fourth the the MSRP

J OK ... so what's a quarter of fiftyfive bucks er twelve er twelve fourteen bucks

determined by the structure of the problem space¹. Weak methods are general and widely applicable but may take too long, and they do not guarantee the finding of a solution. Examples of weak methods are trial and error, working backwards, working forwards, and means–ends analysis. Strong methods take the structure of the problem space into account. Specific methods for achieving a goal, such as procedures for calculating cost or structural strength, are examples of strong methods.

Figure 6.7 shows the application of a method for calculating the manufactured cost of a product given its manufacturer's suggested retail price (MSRP). In this example, the method is the application of a simple formula, which generates the MSRP from the sales price. It is a strong method. The manufactured cost is used as a constraint later on.

5.5 Constraints and Information Management

The knowledge required for solving a design problem is not implicit in the problem definition as it is in the case of puzzles. The appropriate knowledge must be accessed from external sources or from the designer's LTM using the appropriate associations.

A constraint encodes knowledge about the problem in a form that can be easily applied in a problem space for generating and testing candidate solutions. For example, the constraint used in the protocol fragment in Figure 6.5 can be expressed as follows: it should take less than 30 seconds to remove the backpack from the rack.

Information management is finding the knowledge that is applicable in a problem situation and formulating it in the form of a constraint. Knowledge sources can be existing designs, solutions

00:19:00

K the problem I see right away with that pro with that uh is that these are in that orientation and that's also the orientation of jolting and vibration so it'd be nice if

I oh up and down right

so you have to do it much tighter in order to

K yeah you have to put a lot of tension on to these em bolts 'cos that's the opening is in the direction of the

I right

K loading the force it'd be nice if you could have the forks coming more like that so

Figure 6.8
Using an existing design for finding out about a problem

that have been previously generated by the designer, using simulation and scenarios, and reasoning from general principles. Using designs (existing designs or newly generated ones) for identifying relevant knowledge will be termed a solution based strategy.

New constraints are recognized most often when they are failing in a design³. Thus generating tentative solutions for identifying problems with it is a possible strategy. Existing designs for the same problem or for analogous problems can also be used for the same purpose.

In the protocol fragment seen in Figure 6.8, the designers are looking at the prototype rack that was designed earlier. They discover shortcomings which lead them to formulate a new constraint, that the opening of the forks should not be in the direction of jolts and vibration. There does not seem to be any difference between using previous designs or solutions generated by themselves in this respect. Thus, solution based strategies for identifying constraints operate in similar ways, independent of the origin of the solution that is used.

A scenario is a mental simulation carried out in order to identify relevant knowledge about the problem. It is a means for accessing knowledge from the designer's LTM. It helps in making associations between the design situation and the knowledge in the designer's LTM. The knowledge can be used to identify constraints or to test a proposed design. Figure 6.9 contains a section from the group protocol which shows the use of a scenario for how the bike rack may be used, which results in the identification of two constraints; the backpack has to be firmly attached to the rack, and it should be removable in under 30 seconds. We can conjecture, based on this example and others, that scenarios to

00:18:00

*I do they talk about how the people wanna use it?
they uh do these do the vacations they take long bicycle trips
and then take short feet off uh short trips off by foot*

...

*I em so they use the bike to get where they're going and then do
a little hiking sounds like the bike becomes the*

...

*I it sounds like they oughta really ride the bicycle and just
temporarily go to work or something but you wanna be able
to ride the bicycle*

...

*K ride it through the country and then you get to the base of the
hill and you wanna take your backpack and summit the
mountain or something*

...

*K and it's an off-road bike so you'd need a real rugged rugged
attachment or a rigid attachment*

...

*J so what's a reasonable time to like allow somebody to take
this off their bike?
should it take like under five seconds or under thirty seconds?*

...

K thirty seconds?

I yeah I would say thirty seconds as well

Figure 6.9
The use of a
scenario for
identifying
information

fit every situation do not exist in the minds of designers, but are constructed from general knowledge when needed.

Other means of information management are reasoning from general principles and making simulations (for example, when the designers in the group stuff the backpack to get an idea of how much it will hold). The above strategies of information management deal with identifying the relevant knowledge and information for the purpose of identifying constraints. As design progresses, constraints may be modified or removed based on new knowledge that is gained. This is called constraint management.

5.6 Time Management

Design is open ended, thus designers are required to manage their resources. The most important resource that we see in the group protocol is time. Time management operates at the level of goals. Time management determines how much time one should spend on achieving a goal, when to stop working on a goal to move on to a new one and when to omit a goal altogether. Goals are accompanied by estimates about how long it will take to

00:20:00

...

J OK you you were talking about schedule stuff before
do you wanna just set some time limits for ourselves?

I yeah I think we should uh just figure out how how much we
wanna spend on each thing yeah so we can just move on

J so so this is kinda quantifying the problem or wherever we
are right now (laugh)

...

I right now it's uh we have uh an hour and forty minutes so it's
two it's four forty right now four twenty excuse me

J when do you wanna like stop conceptualizing

I well I think we should stop conceptualizing at uh say forty
minutes from now

Figure 6.10
Time management

achieve them. These are used to decide whether to pursue that goal or not. Since goals define problem spaces, and the structure of the problem space determines the methods that may be used, goals have sufficient information associated with them to enable time management decisions to be made.

The protocol fragment seen in Figure 6.10 is an example of time management. Making a schedule is an explicit goal. The elements being scheduled, i.e., quantifying the problem and conceptualizing, are also goals.

5.7 Organization of Problem Solving

The order in which the goals are to be considered is not strictly determined by the task, and there is some jumping around and switching back and forth during problem solving. The focus of attention of the designer is controlled by the recognition of relevant aspects during problem solving, the knowledge that they access from LTM via associations or the problem specifications that they notice, rather than a predetermined order for attending to the goals. This is called opportunism.

In the example seen in Figure 6.11, the designers notice that weight is an important consideration as the result of comparing two solutions. Even though the current goal is not satisfied, they switch their focus of attention to a new goal, which is determining the weight of the backpack. They return to the goal that they gave up, which is mounting, later on.

Attention to control and focus of attention is indicated in the protocol by verbalizations such as 'let's see', 'where was I?', etc.

0:50:00
 ...
 J *yeah so there there seems like two solution sets
 there's one where the whatever it is stays with the pack
 the other solution says the stuff that stays with the bike*
 K *and that's probably better for hiking*
 J *yeah yeah*
 K *then you don't have to lug around that weight*
 J *don't have to lug that weight so and ...
 there's one of the things on our spec that we didn't have is
 weight*
 ...
 J *do we have any information that would tell us what's a
 reasonable weight for the product?*

Figure 6.11
 Opportunism

6 Analysis of Individual's Protocol

The individual designer's protocol has been segmented and coded using the primitive information-processes and other information processing functionalities hypothesized above. The aim of analysing the individual designer's protocol is to account for as much of the protocol as possible and to understand his strategies for structuring and using these primitives.

There is a difference between the two protocols with respect to the aims of the verbalizations, which is also reflected in the instructions given to the subjects for the experiment. In the design team protocol, the aim of verbalization is communication between the group members so that they can cooperate while doing the task. The members of the design team work for the same product design consultancy firm and are used to working together. They communicate naturally. The individual subject is verbalizing his thoughts as a requirement of the experiment. Possibly as a result of this, there are some differences in the verbalization patterns in the two protocols. The design team protocol contains level 1 and 2 verbalizations, whereas the individual's protocol also contains some level 3 verbalizations. In level 1 verbalizations, the subject is reporting the information in the form it is heeded. In level 2 verbalizations information that is not originally in verbal code is translated into that form, and in level 3 verbalizations there are intermediate scanning processes or

Figure 6.12
 Level 3
 verbalization

00:26:00
 OK so em I've read the User Trials Evaluation em

Figure 6.13
Top level goals
identifying design
episodes

00:16:00

You bet. So I've been given the assignment and I'm reading the assignment

I'll read it out loud I guess in part (reads brief)

00:21:00

OK ... em OK em and the first thing I would do as a quick way of getting started is to ask you for the picture of a prototype and the test report.

00:51:00

let me just sorta get some ideas here

01:06:00

let's just em design a mount for the em the bike em er

inference and generative processes². In the segment seen in Figure 6.12, taken from the protocol of the individual designer, he is interpreting what he has done rather than reporting on the information he is heeding.

The top-level goals of the individual designer are to understand the problem by reading the assignment, to start designing by finding the relevant information, to decide on the location of the backpack on the bike, to determine detailed location and dimensions, design the mount, design backpack to rack connection, and specify materials and details. The segments in the protocol identifying the first four of these episodes are seen in Figure 6.13.

In identifying the episodes, the task analysis of the problem and the behaviour that occurs after the goal statement are taken into account. The goal leads to generate and test actions, to subgoals or to information management. In each episode, we see that generating and testing solutions, i.e., search, alternates with information management. In the second episode, defined by the goal to get started quickly, the designer generates a possible location just to rule it out, as seen at the top in Figure 6.14. It occurs among information management functions. This also indicates that searching for a solution can possibly start without much preparation.

Just as the design problem is divided into subtasks by setting up goals, sometimes subgoals are set up in order to achieve a goal. For example, during the design of the mounting the designer first generates a solution which fails structural con-

Figure 6.14
Generate operation
for ruling out a
possible location
right away

00:22:00

em off bat immediately my impression is hey it's nice to have putting it on the front handlebars because you uh like low inertia on the handlebars

straints because it is a parallelogram. As a result of this, the designer makes a request for drawings showing the front, side and rear views of the bike. Then he looks at a Blackburn rack, after which search continues by generating another solution. The different information management episodes are defined by subgoals, and alternate with search. In this case, a failing design alternative leads to information management; the designer looks at the front and rear views of the Batavus bike and the Blackburn design.

We see that goals and time management always occur together in the individual designer's protocol. There is no explicit scheduling as in the group protocol, seen in Figure 6.10.

Three examples of goals and time management from the individual designer's protocol are seen in Figure 6.15. In the first segment, the designer has been working on locating the backpack on the bike. In this segment the goal is not explicitly mentioned but in order to achieve the goal in a short time he changes the method he is using to exhaustive search. This is the only positive example of time management in his protocol. In the others, examples of which are seen in the next two segments, he considers a goal and then decides not to pursue it. Thus, the last two are examples of planning.

00:56:00

OK I have spent er fortyfive minutes or so now forty minutes doing that er so em ... (mutter) going to check every possible location here

00:58:00

I might actually at this stage of the game actually do some trials of my own go out and er see if I found people who had bike racks and do a little more field testing myself but since I don't have much time I would er not do that.

01:54:00

I have a half hour left I haven't even gotten to some of the calculations but I'm not gonna worry about them right now em

Figure 6.15
Goals and time
management
examples

Figure 6.16
Identifying
constraints from an
existing design

00:22:00

OK and it probably right off the bat says the backpack's too high or something like that and that bicycle stability's an issue ...

00:23:00

em eh interestingly enough em ... let me see ... I see how the backpack interfaces interestingly enough em ... doesn't directly take advantage of the eh frame ... nor the fact that there is a frame

Information management functions make up a large part of the design process in the protocols. All of the information sources identified in the group protocol are also used by the individual designer. In Figure 6.16, he is using the prototype design to identify constraints.

The two segments seen in Figure 6.16 are tests of type t2. In the first one, the location of the backpack which is placed vertically on the rear wheel is determined to be too high, and the constraint that the centre of gravity should be kept low for stability is identified. In the second one, the constraint identified is that mounting the frame to the rack should make use of the backpack's external frame.

In the example seen in Figure 6.17, he uses the same type of test to identify the constraint that triangular forms should be used to make the rack rigid. In this example, he is evaluating the rack design he has generated. This is also an example of reasoning from first principles.

In the two segments seen in Figure 6.18, the constraints are

Figure 6.17
Identifying
constraints from
basic principles

01:07:00

one of the problems with em a bicycle carrier where the frame is mounted out here and it goes to that is is that you end up with a parallelogram bad thing bad thing

Figure 6.18
Identifying
constraints from
simulation

00:53:00

and in fact em one of the things I noticed right off the bat is that em it doesn't really stick out much more than the em than the pedal

00:54:00

mmm so right so my thighs are gonna come at least er two or three inches at least behind there my butt OK so ...

identified by actually placing the backpack on the bike and noticing things. In the first segment, the designer removes the constraint that placing the backpack at the side of the rear wheel causes the backpack to stick out and hit things on narrow trails. This constraint has been identified at 00:26 when the side location was generated and then tested using a scenario. In the second segment, the constraint that is identified is how close the backpack can come to the seat when it is placed on the rear wheel.

As a result of analysing the individual designer's protocol, we can say that the hypothesized primitive information processes are sufficient for encoding the segments. No new primitives are needed for accounting for the protocol.

Generating the hypotheses from the group protocol helped identify some implicit, non-verbalized structures in the protocol of the individual designer. Since the group members have to cooperate on designing and follow the reasoning of each other in order to agree or disagree, they are more explicit in their goals, time management and planning functions, and even in their generate and test operations.

7 Discussion

The subjects do not seem to be familiar with bicycle rack design even though they are bikers. They do not have expertise about the specific design problem they are trying to solve. They are using general design heuristics and identifying and accessing relevant information figures very strongly in their design behaviours. We can speculate that an expert doing the same design task may require less information management.

The main parts of the problem, such as, deciding on the position of the backpack, designing a mount, and designing a backpack to rack connection, may be determined by the given task. Thus we see both subjects attend to the same issues.

The group follows a brainstorming strategy; they try to make analogies, defer judgement and try to generate many alternatives. They go through these issues twice, once during what they call the 'ideation' phase, and then again during what they call the design phase. The individual designer focuses on each issue once, but during this time he goes through what can be called 'ideation' and design.

The individual uses existing designs more extensively, and by making a telephone call finds expert opinion about the positioning of the backpack on the bike. He also reasons from first prin-

principles. His comments indicate that his preferred strategy is to use expert knowledge, failing that to use simulation and scenarios and then search. This is what he does when determining the location of the backpack. In other subtasks such as when mounting the rack to the bike, he starts with search. We can speculate that he has more knowledge about this or that it will be too time consuming to seek expert knowledge.

The constraints used for defining the design problem differ between group and individual. The design group designs a tray with straps instead of taking advantage of the external frame, and removing the backpack off the rack is expected to take less than 30 seconds. The individual designer, on the other hand, attaches the backpack to the rack with clips so that it can be taken off very easily but the clips are expected to carry a pack weighing less than 15 lb. According to available information, the pack can weigh up to 50 lb, but this information was not seen by the individual.

As a conclusion we can say that it is possible to account for some aspects of design using simple structures and diversified knowledge from very different sources. These elements are used very flexibly in different combinations. Information management alternates with search. The most important differences between group and individual are in information management. Other aspects of design, such as recognition processes, using intuitive constraints, aesthetics, and accessing information from LTM via associations, are not accounted for in the protocol data.

8 Conclusion

The aim of this study is to account for the design behaviour of the individual subject. In order to carry it out, we use an analysis of the bike rack design task and a set of hypothesized primitive information processes. These are derived from an analysis of the group design protocol. The protocols are first segmented and then encoded in terms of primitive information processes and the information that is needed. The hypothesized primitive information processes are based on the design task and are task dependent. The verbalizations in the protocols indicate the inputs and outputs to the processes rather than the processes themselves.

These processes and information structures are a hierarchy of goals and subgoals, generate and test operations, methods that organize them, constraints and constraint management, information management and time management. Goals are the most important elements in determining the behaviour. They lead to

the selection of appropriate problem spaces, methods and information and time management functions. The two types of generate operations and the three types of tests are used by both the individual and the group. These primitive information processes are sufficient to account for the segments in the individual's protocol. Other aspects of design, such as recognition processes, using intuitive constraints and accessing information from LTM via associations, are not accounted for.

The similarities between the two protocols can be attributed to the requirements of the design task and to the fact that the subjects are not experts in the task. Also, some of these are due to general design heuristics used by both the individual and the group. The differences may be due to designing in a group versus individually, and to differences in the subjects' backgrounds and personal preferences.

References

- 1 Newell, A. and H.A. Simon, *Human Problem Solving*, Prentice Hall, Englewood Cliffs, NJ (1972)
- 2 Ericsson, K.A. and H.A. Simon, *Protocol Analysis*, MIT Press, Cambridge, MA (1993)
- 3 Eastman, C.M., On the analysis of intuitive design processes, in G.T. Moore (Ed.), *Emerging Methods in Environmental Design and Planning*, MIT Press, Cambridge, MA (1970) pp. 21–37
- 4 Akin, Ö., *Models of architectural knowledge: an information processing view of architectural design*, PhD thesis, Carnegie-Mellon University, Pittsburgh, PA (1979)
- 5 Akin, Ö., A formalism for problem restructuring and resolution in design, *Environment and Planning B: Planning and Design*, **13** (1986) 223–232
- 6 Akin, Ö., Heuristic generation of layouts (HeGel): based on a paradigm for problem structuring, *Environment and Planning B: Planning and Design*, **19** (1992) 33–59
- 7 Goel, V. and P. Pirolli, Motivating the notion of generic design within information processing theory: The design problem space, *AI Magazine*, **10**(1) (Spring 1989) 18–36