Designing or Planning? – Cognitive foundations for design aiding

A. O. Kazakçı, A. Tsoukiàs

LAMSADE - CNRS, Université Paris Dauphine
75775 Paris Cedex 16, France

{kazakci,tsoukias}@lamsade.dauphine.fr

Abstract

This paper considers the cognitive aspects of designing. We compare these aspects to those of the (hierarchical) planning process; a process which is of common interest for Cognitive Psychology (CP) and Artificial Intelligence (AI). We show that both processes can be analyzed using similar key notions and they both have the same essential characteristics. Therefore, we establish a *cognitive equivalence* between the two processes. We discuss the implications of such an equivalence, emphasizing three main points. First, the equivalence provides a framework in which to join research efforts coming from design research, AI and CP, to the benefit of all the three fields. Second, it offers the possibility to propose a model of the design process based on the Hierarchical Task Networks (HTN) Planning formalism of AI. As discussed in the paper, such a model would have solid theoretical background and can provide sound foundations for design aiding tools. Third, such a model would facilitate, together with the results established in this paper, interactions between design research and Decision Aiding Sciences.

Key words: Design process, planning, design cognition, design aiding, decision aiding.

1 Introduction

Designing is a complex activity involving different dimensions and can be examined from various perspectives. Accordingly, design research was developed within various research fields: innovation (Le Masson and Weil, 1999), (Chapel, 1997), (Perrin, 2001), project management (Midler, 1993), engineering (Pahl and Beitz, 1984), artificial intelligence (Kannapan and Marshek, 1996), (Gupta et al., 1996), (Gero, 1998), knowledge representation (Coyne et al., 1990), (Gero, 1990), creativity (Gero, 1996), (Logan and Smithers, 1993), etc. These and many other research efforts attempted to determine the place of designing within the organization, to understand its nature and to provide practical methods to support it. The main dimensions studied by these efforts are coordination, collaboration and cognition. This paper considers essentially the latter dimension. We consider designing as a cognitive activity and we are concerned with the natural intelligence of design or *'designerly' ways of thinking* (Cross, 1999).

Our aim is to point out that there exist strong parallels between the cognitive aspects of the design process and the (hierarchical) planning process; a process which is of common interest for Cognitive Psychology (CP) and Artificial Intelligence (AI). Both CP and AI have strong relations with the design research; relations that are, to our opinion, determining factors in giving sound foundations to design aiding tools (section 2). We then analyze the characteristics of the design process (section 3) and the planning process (section 4) from a cognitive perspective. In section 5, we show that both processes can be studied using similar key notions and they both have the same essential characteristics. We, therefore, establish a *cognitive equivalence* between the two processes. In the last section, we discuss the implications of such an equivalence

2 Design Research, Cognitive Psychology and Artificial Intelligence

Although much insight is gained by research on the nature of the design process itself and the way designers do their business, we are yet to uncover fully the cognitive mechanisms that designers use. To deal with such a concern, a 'psychological' look should be fruitful as it is emphasized in (Pahl et al., 1999). Accordingly, we shall use some key notions and ideas that originates from Cognitive Psychology (CP) in our analysis.

A related discipline with design research – but also, with CP – is Artificial Intelligence (AI). AI-in-design aims to support designing. This support may take the form of a collaborative system (i.e., the designers interact with the system) or an automatic system (i.e., the system produces the output given the input). Many techniques and tools of AI have been proposed in the literature (see e.g., (Kannapan and Marshek, 1996), (Gupta et al., 1996), (Gero, 1998), (Coyne et al., 1990)) to assist the design process. However, (Cross, 1999) suggest that this is not the only role of AI in design research: "AI-in-design should attempt to tell us something about how designers think" and that "we can hope to learn some things about the nature of human design cognition through looking at design from the computational perspective."

Here, we take a slightly different stance. We should draw on the wealth of concepts, models and theory of CP to better analyze and understand how designers think. It is only then that we would be able to develop AI-in-design support tools that are firmly based on solid theoretical foundations. We think it is by proceeding this way that designers should “be able to use them (these tools) in ways that are cognitively comfortable” (Cross, 1999).

Therefore, we need a framework that will enable us to better analyze and understand the human design cognition and that will provide us foundations for support tools. A framework that will better highlight the relations between design research, AI and CP and strengthen them. To this end, we investigate, in the following, the nature of the design and planning processes, to show that, from a cognitive perspective, there exist an equivalence between the natures of the design and planning processes.

3 The Design Process

3.1 Design Terminology

As remarked in (Love, 2000) the multiplicity of concepts in design research and the different meanings they are used with can become an embarrassing source of confusion. Accordingly, it would be useful to clarify some terminology we use. Many descriptions of the ‘design process’ exist in the research literature (see (Perrin, 2001), (Evbuomwan et al., 1996) for reviews). We adapt the following general definition for the design process based on the common core of these definitions: “design process is a set of activities and processes which begins with the acknowledgment of needs and the intention to propose a solution responding to those needs where the initial problem and (the associated solution idea) is continuously transformed, refined and detailed by creating and/or using knowledge to provide the necessary information for the implementation of that solution.” Here, solution may be a product, a service or a process; a more or less concrete entity: a car, a building, a chemical process, software, a schedule, etc. ‘Designing’ is an activity where the person who undertakes it (the designer) has to elaborate a solution description from an initial problem description by using his (mental or physical) capacities. A ‘design’ is a solution description, although its level of detail (or abstraction) may vary. From a general point of view this description may be seen as a set of properties (or elements) and the relations existing between these properties. We also use the notion of ‘artefact’ to refer to an external representation of the entity being designed (Simon, 1969).

Design process begins when an actor realizes a problem and takes an action with the intention to propose a solution. It is a problem solving activity where, initially, the problem can not be precisely stated. This, in turn, implies that *there can be no prefixed set of solution alternatives.* The purpose of design, therefore, is to elaborate a solution description from an initially not-so-precise problem definition.

Because of the difficulty in stating the problem precisely, the design problem is often formulated in terms of goals and constraints (Darses and Falzon, 1996). The unclearness of the nature of the problem is then translated as a rough definition of these goals and constraints. As a consequence, initially, their interrelations may not be clear and they might be in conflict. They are subject to change during the process: they "may be extensively revised, or even abandoned altogether" (Lawson, 1980). Furthermore, there is no straightforward process leading to a solution (Darses and Falzon, 1996), (Perrin, 2001). For these reasons, the design problem is often qualified as ill-structured.

Here, 'ill-structured' refers to two points. The first is that, usually, there are lots of problem elements with multiple aspects to consider and their interrelations are too numerous to process easily. Thus, the design problem is large and complex (Coyne et al., 1990). In practice, this implies a group of designers (rather than a single one) with different backgrounds, experiences and skills. The second is that, at the beginning of a problem solving process, a problem is ill-structured for the ones who attempt to solve it, even if it has a clear and well defined structure for an observer (Simon, 1973). Hence, the design problem lacks initially a clear definition and it is precisely the task of the designers to "construct" it. *The construction of the problem definition is progressive.* Designers depart from a generic problem definition - and they explore the potential solutions. By exploring, they increase their understanding of the context in which they operate, the problem situation and the tradeoffs between "what is required" and "what is possible". Here, requirements refer to the goals and constraints and possibilities reflect the set of potential designs that designers can realize. Often, what the designers find out to be possible, considering their resources (especially knowledge) and the context, implies modifications on the problem definition. Thus, the considered solutions contribute to the restructuring of the problem, which, in turn, characterizes the solutions that should be considered. Hence, *problem definition and the potential solutions are progressively co-constructed by exploration of possibilities.* Thus, at a given moment during the process, a problem definition is an intermediary description of an intended final solution.

During this co-evolutionary process, the problem definition (which corresponds to the current solution description) is enriched by rearranging requirements and/or by further specifying existing elements. Rearranging requirements means to adopt or abandon some goals or constraints, that is, adding or removing some of the properties of the problem description, while further specification refers to the decomposition of problem elements to form subproblems. This refinement gives a hierarchical nature to the design process. *The problem definition at a given level of the hierarchy is refined by adding or removing some properties to the problem description and/or by expliciting the subproblems and their interrelations to obtain the next level of hierarchy.* Thus, the description of the artefact evolves hierarchically by its successive refinements until appropriate detail level for the implementation is obtained.

3.3 Designing and Knowledge

During the enrichment of the problem description, the main difficulty lies of course in finding satisfactory refinements. In all evidence, this depends on the current problem structure, but also, on the knowledge of designers. As stated in (Coyne et al., 1990), "design process is a cognitive activity heavily reliant on the application of knowledge". Knowledge, then, is one of the main resources used in design. We should mention at this point that, such a resource is not always available; then it must be looked for or even created. Therefore, *the refinement process inherent to the design necessitates 'knowledge use and creation' to be an essential part of the design activity.* One resulting corollary is that, *at intermediary stages, a complete refinement of the problem is simply to difficult, if not impossible*, as all the necessary knowledge is most often not immediately available. Another important characteristic that follows is that *designers learn inevitably from the design experience.* In fact, some studies indicates ((Bowen et al., 1994) in (Perrin, 2001)) that the most successful design teams are the ones who consider as the most important output of the design process, the "resulting capability" gained via learning, and not the final design. The importance of learning is widely recognized in the design literature and this has given rise to an interest in explicitly recording the design rationale, that is, the knowledge used (such as the available alternatives, the choices made, the reasons behind) for possible later reuses (Chandrasekaran et al., 1993).

During the process, a partial description of the problem delimits the possible admissible designs, but at the same time, leaves open a very large number of possibilities, designs that can be realized and that will fit into the limits imposed by that definition. In other terms, an intermediary description reflects the properties found to be relevant for the final artefact and determines a class of designs which share these properties. This kind of partial design descriptions are sometimes referred to as 'generic designs' (Coyne et al., 1990). We can postulate that an essential part of the 'design knowledge' of designers are about generic designs that correspond to different design descriptions (at different abstraction levels) they learned in their past design experiences as well as how they have been elaborated.

3.4 Designing and Searching

Facing a problem, designers activate their design knowledge, on the one hand, to find the similar generic designs and how they had been manipulated in past design experiences, on the other hand, to select the most appropriate solution elaboration strategies (von der Weth, 1999) for the task at hand. Generic designs 'remembered' as such, may be adapted to the current context (by combining them in various ways or simply by adding/deleting some properties to/from some or all of them). Also, their existing properties may be further specified (or decomposed) to get different 'instances' (Coyne et al., 1990). Remark that, adding/deleting the properties of a generic design corresponds to an horizontal move where another generic design is taken under consideration, whereas, further specifying how to achieve an existing property is a vertical move where an instance of the previous one is obtained. In both case, new generic designs may be obtained. In the former case, by a consideration of a previously unused (unknown) combination of properties (e.g., a phone which is 'mobile'); in the latter case, by applying a new decomposition (e.g., using a solar energy source for a car). Furthermore, there exists a hierarchy between generic designs. By instantiating a generic design we obtain its *partitions*, on the contrary, by making abstraction of it, we have its *type*. A formalization of the "generic designs" notion exists in design literature as "design prototypes" (Gero, 1990).

During the process, different solution elaboration strategies may be adopted at different moments. These strategies have an heuristic nature and they provide a control mechanism on how to explore the partitions/types hierarchy. They are proper to individuals and may change from one designer to an other. Moreover, the selection of strategies is governed by other heuristic knowledge, called "stratagems" in (von der Weth, 1999). We see that *design is a search process, where knowledge about generic designs and search strategies are used in order to elaborate new solutions*, that is, new knowledge about new generic designs. At each intermediary step of the refinement process, the use of knowledge about the past design experiences may permit the designers to generate alternative solutions from the current problem definition. If no alternative can be generated or if all the generated alternatives are judged to be unsatisfactory, then higher level generic designs should be revisited. In other words, the problem has to be restated. At any rate, all of the possibilities can not be explored, as there is a very large number of them and as the solution generation is necessarily time constrained. *During a design process, only a limited portion of what is possible can be explored.* However, the generation of several alternatives rather than a single one is desirable. Both the size and the quality of the set of generated alternatives, are important. As a consequence, the designers have to push their limits to find the most promising designs, a sample that should represent at best the partitions of the considered type, as quickly as possible.

3.5 Designing and Decision

We distinguish two essential kind of decision problem involved in the alternative solution generation by refinement; the *feasibility* and the *preferability.*

3.5.1 Feasibility

Assuring the feasibility of an alternative solution involves management of requirements and interactions between different subproblems. Decisions concerning the properties to be abandoned, introduced or detailed must be taken in a way that allows the requirements to be fulfilled and that avoid any conflict between different subproblems. This is not straightforward, however, as the implications of the decisions taken may not be immediately apparent and conflicts may only be revealed later on, as further refinement decisions are made. In practice, it is necessary to use technics ranging from sketching to computer based simulations, in order to discover the potential effects of decisions taken. These technics allow to consider lower levels of the refinement hierarchy without necessarily taking any definitive refinement decisions; discovering, thus, information that might be relevant for the refinement.

3.5.2 Preferability

Most often, a single feasible alternative is not satisfactory, and further attempts are made to generate several. One reason is that every generation attempt bring into light new information that will help to (re)structure the problem definition. As the co-evolution of the problem/solution description lies at the heart of the design process, amplifying this kind of information entry is important. Yet, one other reason is that this new information will help also to (re)structure the preferences of the designers about abstract artefact being developed.

Design is an evolutionary process where information about some entity which does not even exist is manipulated and where it is hard to predict towards what it will converge exactly. Designers are the ones who *conduct* this evolution by the choices they make. But, during the process, their preferences about the direction this evolution should take are, at best, partially constructed. Every solution generation attempt will help a designer to elaborate his/her preferences, to shape his/her own convictions, by bringing into his/her attention previously unknown or unconsidered information. The success of the process depending on their choices, an effort to generate a diversity of alternatives will be helpful for the designers to better apprehend different aspects involved in the process and the direction towards which the evolution must be conducted.

Under such a perspective, the refinement process can be seen as a search for feasibility by managing requirements and relations between subproblems and for preferability to orient the evolutionary design process. Therefore, another characteristic of the design process is that *the design process is a collection of overlapping and/or interrelated decision processes, where the preferences of the designer(s) about the artefact being designed evolve(s) dynamically with the problem/solution description.*

In this section we have summarized the essential characteristics of a design process. We discuss, in the following, the characteristics of the planning process to show that there is an interesting one-to-one correspondence between the characteristics of the two processes.

4 The Planning Process

4.1 Planning Terminology

Planning is a rather general concept which concerns many different research areas such as economics, urbanism, social welfare, manufacturing, artificial intelligence, cognitive psychology, etc. However, there seems to be no real consensus on what is planning. In this paper, we consider planning from a cognitive perspective and adopt the following definition from (Hoc, 1992). A ‘plan’ is a schematized and/or hierarchical representation, elaborated in order to guide the activity to accomplish a given task. ‘Planning’ is the elaboration of such a representation. Remark that this definition is a rather general one, encompassing the activity to be undertaken in many different problem situations (Ostanello and Tsoukiàs, 1993). We use equivalently ‘representation’ or ‘abstract plan’ when further elaboration is needed to allow an execution.

When a cognitive agent (CA) - human or artificial- is given a task to accomplish, without having an immediate executable procedure, a problem solving procedure begins. The CA begins constructing a representation of the problem which will serve as a basis to reason about the objectives of the task and the available actions. When the CA's representation of the problem matches to an already constructed plan for an already solved problem or when it is simple enough to allow an immediate transition to the execution, no planning activity really occurs. On the contrary, when the size of the problem is large (according to the CA's processing capacity) and/or the task is unfamiliar to the CA then a need for planning arises. In this case, as the CA's initial representation of the task does not match any of the existing representations in its memory, the planning problem can not be precisely stated (in a detailed manner, allowing immediate execution). As a consequence, *the set of potential plans susceptible to solve the problem can not be characterized in advance.* Then, the representation must be more fully elaborated; a detailed construction is needed.

This construction implies an interpretation of the task by the CA, which depends on the 'knowledge structure' that the CA has on the 'domain of tasks' (see below) considered. The construction of the problem representation is based on two important mechanisms that depend on this knowledge structure: to anticipate and to schematize (Hoc, 1992). Schematizing suppose an abstraction of the task, retaining only the details immediately relevant to the elaboration of the representation. Missing elements of this schematized representation are anticipated, based on the knowledge structure that the CA has constructed on the domain of task. The two mechanisms function using the same means: schemas (Bartlett, 1932) (see also (Schank and Abelson, 1977), (Hoc, 1992)). [1] Schemas are frameworks for organizing knowledge in memory. They encapsulate knowledge *elements* about concepts or contexts. Extensive use of the notion has been made in literature and many knowledge representation structures corresponding to it have been proposed; see, for example, "scripts" (Schank and Abelson, 1977). Schemas are closely related to the construction and use of the knowledge structures on different domains of tasks and hence, to the construction and use of the representations of tasks.

A domain of tasks is a structured set of objects, descriptors of the properties of and operations on these objects (Hoc, 1992). For example, an engineer planning on a 'transport' domain may consider objects such as trucks, loads,

[1] It is interesting to note that, in (Gero, 1990) and (Coyne et al., 1990), this very same notion of schema of (Bartlett, 1932) is cited to refer to the knowledge structures of designers.

roads, etc; descriptors are, then, size capacity of a truck, weight capacity of a truck, fuel consumption, load size, load weight, length of a road, etc; operations are assigning a load to a truck, changing destination of a truck, choosing an itinerary (a set of connected roads) for a truck, etc. There exist different levels of description for a domain of tasks, i.e., a domain of tasks has a hierarchical structure. The more abstract levels contain details necessary only for the determination of a general strategy to elaborate a procedure to accomplish the task. When we move to lower levels, details necessary for the execution of the procedure are introduced progressively (Hoc, 1992). Speaking of 'loading of a truck', for instance, we may refer to 'loading by a robot' or 'loading by a human'. For each of such tasks, we need more details for the execution, e.g. the programming of a robot for maximum use of truck capacity. The knowledge structure on a domain of tasks results from the interiorization of it by the CA. It is constituted by the schemas related to that domain of tasks previously constructed (or learnt) by the CA. Remark that a knowledge structure may contain partial information, misinformation, or even inconsistencies about a domain of tasks. Also, a knowledge structure is proper to a CA, although there may be similarities with other CA's knowledge structures constructed on the same domain (Hoc, 1992). The knowledge structure is hierarchical, reflecting the nature of the domain of tasks.

When given a task, the CA schematises, i.e. makes an abstraction of the task in order to extract the characteristics it deems most relevant. This abstraction provides an easier processing and storage. The obtained representation serves as a basis to infer previously encoded schemas, to recall similar ones. Using the elements of the retrieved schemas, the CA complete further its understanding of the task, either by introducing/abandoning some elements to/from the plan being constructed, or by replacing an element with a corresponding schema which details how the replaced element should be achieved.
The recalled schemas allow thus anticipating the missing elements of the representation being constructed. Remark that, this way, new schemas (at different abstraction levels) can be created using the previously known schemas. As a consequence, *during a planning process the cognitive agent learns as its knowledge structures are enriched by new schemas and existing schemas are updated.*

Therefore, *planning is a process which relies on the use of knowledge* and *where new knowledge is created.* The schemas created in this way at any moment during the planning process are stored in short or long term memory, for immediate or later use. During the encoding or the retrieval of the schema, the CA may change the abstraction level (by adding or removing the details of an element) to allow a more effective storage, inference or processing. In other words, different levels of the knowledge structure hierarchy are considered.

The comprehension, that is, the construction of the representation by the CA is heavily dependent on its knowledge structure and on its information processing capacity. Thus, when the problem is large and complex (according to the CA's processing capacity) and/or the task is unfamiliar to the CA, the construction of a schematic representation might prove difficult, the schema may not match any of the existing schemas in the CA's memory or the anticipations may become less credible or mistaken. There are two possibilities: the representation of the task is either incomplete, or inadequate (Hoc, 1992). In the former case, the representation does not contain all of the relevant properties of the task (for the level of detail considered). In the latter, CA has attributed to the task some properties that it has not, in reality. In both cases, the CA does not comprehend the task.

In fact, the comprehension (i.e. the construction of the representation) does not happen readily; *the construction of the plan is progressive*: During the problem solving process, the problem representation is continuously updated and "the general or essential properties of a solution precede its specific properties" (Duncker, 1945) in (Hoc, 1992). This restructuring is due to the interaction of the comprehension of the problem with the elaboration of its representation. The confrontation of a plan to the situation where the plan is supposed to guide the activity may show that in its current state the plan is not well adapted to that situation, because of its inadequateness or incompleteness (Hoc, 1992). Thus, following this confrontation, a better comprehension of the nature of the problem occurs and accordingly, a restructuring of the representation is undertaken to obtain a more adequate and complete plan. In this way, the envisaged plan contributes to the restructuring of the problem representation, which in turn, characterizes the plans to be considered. Hence, *problem representation and the potential plans are progressively co-constructed.*

When the plan is found to be incomplete, it must be further refined by adding elements and details for existing elements in a coherent way. When it is found to be inadequate, then a revision of the plan should be realized. More schematic (less detailed) versions must be reconsidered, to find a detail level where the undesired properties are not introduced yet and where the refinement can restart. Within this framework, *planning consists of moving between different levels of an abstract plan hierarchy, to continuously refine the problem representation by adding or removing properties and by detailing existing properties, until a complete and adequate plan to guide the activity can be constructed.*

The above-mentionned movements can be of two kinds -ascending or descending- and both exploit schemas of knowledge. They may be combined in different ways according to the search strategy adopted. Let us present different possible instances of these generic movements and relate them to the previously introduced ideas about the construction and use of knowledge structures.

Ascending movements, such as evocation of plans (from indices or analogies), abstraction of plans and revision of plans, can be used to obtain plans from the details of a situation (Hoc, 1992). As we have said before, the CS schematize a given task by making abstraction of it to form a mental representation. Some elements of this representation -the indices- may be used to infer in the memory to recall similar plans. We may think of a student passing an exam. Having solved many questions of many different types on the subject (therefore, having learned the corresponding schemas) prior to the exam, (s)he can recognize the type of a question at the exam, and remember the corresponding schemas necessary to solve the problem. When the subject has no sufficient knowledge in the domain of tasks considered, plans constructed for different domains may be adapted (at a sufficiently higher level of abstraction) to the new problem if there exists analogies between the two problem solving situations. Learning to program in Pascal, may facilitate learning to program in C. It is also possible to construct new plans by making abstraction of situations. The simplest case for the abstraction of plans is where examples of resolution of similar problems lead to a generalization of the solution principle for that kind of problems. A more complicated case is the reflective abstraction (Piaget, 1977), (Hoc, 1992). Reflective abstraction goes one step ahead of simple abstraction, as the subject learns not only the solution principle, but also what is it that made this principle work by reflecting about the reasons of the success. The reflective abstraction is one of the prerequisites to develop an expertise on a domain of tasks. The revision of plans happens when a difference between the environment and the CA's representation of it is detected. This amounts to say that, a revision of the plan can arise from two reasons. Firstly, the environment may have evolved in such a way that the plan being elaborated is no longer feasible. Secondly, the CA may realize that its internal representation of the task is mistaken and does not match the real task. In both case, the plan is either inadequate or incomplete and the plan must undergo a revision.

Descending movements are to refine the plan by adding details (further specify the subproblem elements, clarifying relationships between them), necessary for the execution of the plan. Descending movements involves decomposition of a plan to subplans, instantiation of a plan, and management of interferences between subplans (Kutluhan, 1995), (Hoc, 1992). For example, when planning for 'a night out', this initial plan may be decomposed to a set of subplans as going to a restaurant, then to a movie. Instantiation of the subplans implies choosing the restaurant, for the first, choosing a movie for the second. Still, there may be conflicts between subplans which have to be managed. For example, after the restaurant one must have still enough money for the movie, by going to a cheap restaurant. Each of the subplans in the above example also requires to be decomposed by using appropriate schemas available to the CA. Dining at a restaurant involves, let's say, going there, entering the restaurant, choosing a place to sit, reading the menu, giving the command, and so on. This is often the case in planning situations; there exist many available schemas hierarchically related and more than one can be used for decomposing each different element of a schema that need to be decomposed. Also, new decompositions may be created, using these available schemas (for example combining them in some way). This may require possibly using other schemas about other domains of tasks as well. Whether there is a need to create new schemas or not, *the construction of a complete and adequate plan is never immediate* as the comprehension is progressive; the plan may need to be revised for one reason or another, or the knowledge needed for the refinement may not be immediately available, or else, there may be a very large amount of possibilities for refinement and the processing may take long.

During this search, the ascending and descending movements may be articulated in different ways, using many solution elaboration strategies (such as, trial-error, means and ends, hypothesis testing, least commitment, use of analogies, etc). Remark that a solution elaboration strategy is independent of the domain of tasks under consideration, has a heuristic nature and uses meta-operations (Hoc, 1992). Different solution elaboration strategies may be adopted during the problem solving process as the problem representation changes. In fact, the choice of a strategy is closely related to the meta-knowledge, that is the knowledge about the knowledge structures, of the CA on a particular domain of tasks: planning. Deciding which strategy, heuristic, meta-operations to use, when to use them (considering the properties of the task and the environment), prioritizing the meta-goals, selecting the ones to achieve, in short, planning how to plan is referred to as meta-planning. Hence, *planning is searching for a task representation, containing enough detail to be immediately executable, using knowledge about domains of tasks and in particular, knowledge about planning.*

However, all the possible refinements can not be searched for, and at times, even the immediately available refinements can not be examined exhaustively. Remembering or creating schemas, deciding how to refine the plan representation, constructing an executable plan are subject to many constraints, the primary being the time (think of, for example, a student passing an exam, or a basketball player organizing the game ten seconds before the end of a match, or an engineer preparing a project about a transport system). *The* *search during the planning process is limited*, even if, *a priori*, there exists a very large number of plans that could achieve the given task.

4.5 Decision and Planning

"Every comprehension activity", suggests (Hoc, 1992), "implies an evaluation of the representation that is evoked or elaborated, from two points of view, *coherence* and *purpose*." Taking decisions that will maintain the coherence is what we have referred to as the management of interferences between subplans. Constructing a plan that will meet a given purpose depends on the decomposition and instantiation decisions taken during the planning process. Obtaining coherence is not always easy as some interactions between subplans might be hard to detect prior to the actual occurrence of a conflict, when further refinement decisions are made. To seize in advance the potential and/or hidden interferences, a CA may use different techniques such as constraint propagation, simulation, critics and strategies such as least commitment, fewest alternative first (Kutluhan, 1995), (Hoc, 1992). The second perspective from which a plan must be evaluated is the purpose, that is, for what use the plan is being constructed. But, for a CA, the purpose of a plan is somewhat evasive during the planning process. Said in other terms, before a complete and adequate representation is constructed, there is always some degree of liberty in the choices done and the preferences of the CA about the way the task should be achieved can (and most probably will) change. At times, the change in the preferences may be so radical that the CA may decide that the task for which the planning is undertaken is not the right task to be planned for. Eating at restaurant then going to a cinema can become less interesting when it starts raining or when a worth-to-see movie can not be found; in which case, one can stay at home, order a pizza and rent a movie. Not only the preferences may change, but they may even be unestablished yet. Even if the CA has some preestablished preferences about similar planning situations, simply because the current task is a new one, those preferences may not apply. As the planning process (and more generally, the CA's knowledge use) is dynamic (i.e., new information obtention, change of knowledge structures), it is also possible that the preestablished preferences be no longer valid for the current case. Then preferences for the current planning situation must be *constructed*, and this must be so in the light of the currently available knowledge. After all, what are preferences but parts of the knowledge structures.

Thus, preferences are constructed dynamically in parallel with the comprehension/representation of the problem. These preferences are applied, again dynamically, to the available decomposition and instantiation possibilities. The way these preferences are applied may vary greatly depending on the task structure and environment, knowledge available about 'evaluation' per se, and also the nature of the planner. For a human planner, most often, evaluation knowledge that can be qualified 'heuristic' is used to determine the schema to use (Todd and Gigerenzer, 2000), (Payne et al., 1993), (Hoc, 1992). Cognitive limits combined with the time constraints lead to an adaptive behavior of a human CA on the choice of evaluation heuristic; accuracy and necessary effort for the implementation of a decision heuristic is considered according to the characteristics of the task and the environment to choose an evaluation method (Payne et al., 1993) (see also (Todd and Gigerenzer, 2000)). For an artificial system, various (formal) evaluation models can be used as proposed in (Moraitis and Tsoukiàs, 2002). At any rate, there exist an evaluation procedure where the preferences are applied to available schemas to decompose further the plan being constructed and the decision to be taken may interact with other decisions (already taken or to come) about an existing or newly created decomposition, due to the possible interferences. With this regard, we can consider that *planning is a process formed by interacting and overlapping decision processes where preferences are constructed dynamically in parallel with the comprehension/representation of the problem* and where a conflict free plan is searched by managing interferences between different subplans.

In this section, we have highlighted the main characteristics of a planning process. In the next paragraph, we shall argue about the equivalence of the planning and design processes as we have presented them.

5 Designing versus Planning

From what we presented so far, the reader should have already remarked the multiple resemblances between the cognitive aspects of design and planning activities. As a matter of fact, the equivalence between the design and planning problem solving processes is often considered (explicitly or implicitly) in the corresponding literatures, as illustrated in the following paragraph.

5.1 Arguments from the literature

Designing implies planning...

"The size and the complexity of the problems, as well as the absence of the preexisting solution elaboration procedures oblige the designers to formulate the problems in terms of goals to reach. This involves a decomposition of the solution to sub-goals (which remains however incomplete). The designers are led to work out solution elaboration strategies, in particular using the planning activities during which schematic and abstract representations are formulated" (Darses and Falzon, 1996). In fact, "human designers form their individual design experiences into generalized concepts or group of concepts at many different level of abstraction - that is, they schematize their knowledge" (Gero, 1990). These schemas "evoked and used constantly during the design process, [...] allow inference on data structures and functions in order to execute and to solve parts of the problem" (Darses and Falzon, 1996). Within, this framework, a design process can be seen as a planning process where a problem is solved by exploring the problem space that has a tree-like structure. "The nodes of problem space are problem descriptions (plans) of various precision level. The arcs represent planning relations. A node is a plan for the nodes which follow it if its attributes can be interpreted as constraints on the attributes of those nodes" (Hoc, 1992).

Planning implies designing...

"The [planning] process can be seen as the continual refining of the specifications of the plan" (Georgeff, 1990). The refining of the plan implies exploring a search space where "each node (...) corresponds to some possibly partial plan of action to achieve the given goal" (Georgeff, 1990). "Tate defines a plan as a specialized type of design where 'a design for some artefact is a set of constraints on the relationships between the entities involved in the artefact' (Tate, 1996). A plan constricts this definition by specifying that the entities are agents, their purposes, and their behavior. Planning can then be considered to be a specialized type of design activity. Designs or plans are created by an agent or group of agents placing constraints on the developing artifact. We can think of these activities as repeatedly making design decisions that continually transform the artifact until it embodies the requirements necessary to enact the solution" (Polyak, 1998).
"Applied work in AI planning has typically favored approaches based on hierarchical decomposition rather than causal chaining. In particular, most successful planners for practical applications have used hierarchical task network (HTN) planning ((Sacerdoti, 1974); (Tate, 1990); (Currie and Tate, 1991); (Wilkins, 1990)), an AI planning methodology that creates plans by task decomposition. This is a process in which the planning system decomposes tasks into smaller and smaller subtasks, until primitive tasks are found that can be performed directly. HTN planning systems have knowledge bases containing methods (also called schemas by some researchers). Each method includes (1) a prescription for how to decompose some task into a set of subtasks, (2) various restrictions that must be satisfied in order for the method to be applicable, and (3) various constraints on the subtasks and the relationships

among them. Given a task to accomplish, the planner chooses an applicable method, instantiates it to decompose the task into subtasks, and then chooses and instantiates other methods to decompose the subtasks even further. If the constraints on the subtasks or the interactions among them prevent the plan from being feasible, the planner will backtrack and try other methods." (Tsuneto et al., 1998).

Considering these similarities, we formulate the following proposition.

Proposition From a cognitive perspective, design and planning processes can be seen as equivalent.

During the rest of this section we shall argue for the equivalence between the design and the planning processes, first, by pointing out the similarities in the underlying key notions and by stating the one-to-one correspondence between the characteristics of the two process.

5.2 Equivalent key notions

Establishing correspondences between some key notions presented in sections 3 and 4 is rather intuitive: a design can be seen as a plan. Remark that, in some cases, the distinction between the two concepts become hollow. We would rather say "to design" a car, a phone or software but "to plan" a production schedule, marketing campaign or a night out. But what about, for example, an urban transport system? Indeed, this is a complex problem, where many (abstract or not) scenarios have to be *designed* and evaluated to select a satisfying scenario. The resulting descriptions of this scenario is what else but a plan?

Following the same line of reasoning, a generic design is equivalent to an abstract plan or a schema. The use of stratagems is equivalent to meta-planning. A design rational seems equivalent to a reflective abstraction as both are intended to give an account of the reasons behind the success (or eventually, failures) of the elaborated solution and their relation with the choices made and the knowledge used during the process. The types/partitions hierarchy corresponds to the plan hierarchy. Hence, designing by exploring the types/partitions hierarchy can be seen as planning by moving between different abstraction levels of the plan hierarchy. Table 1 shows a summary of the equivalent key notions.

Designing	Planning
An (abstract) Design	An (abstract) Plan
Generic design (Coyne *et al.*, 1990)	Schema (Barlett, 1932)
Design prototype (Gero, 1990)	Script (Shank, Abelson, 1977)
Use of stratagems	Meta-planning
Types/Partitions Hierarchy	Plan Hierarchy
Design Rationale	Reflective Abstraction

Fig. 1. **Similar key notions in Design and Planning Researches**

5.3 Common characteristics

As our discussion about the nature of the design and the planning revealed, both processes share the same essential characteristics. For these two processes, we can recapitulate these characteristics as follows.

(1) **No Prefixed Set of Solutions** No characterization of the set of admissible solutions is possible prior to the end of the process, as there is no complete problem description before. The agent(s) that must solve the problem, has (have) to "construct" such a characterization. We should immediately mention that this property may be common to a great variety of processes. We consider them all design (or planning) processes, as illustrates our general definition of design process.

(2) **Incremental Problem Definition** Initially, the problem is ill-defined. As the problem solving process advances, there is a progressive transition from this ill-defined state to a more precise and satisfying definition of the problem. In other words, at each stage of the process continuous attempts are made to update and enrich the definition of the problem.

(3) **Co-construction of Problem and its Solution** At intermediary stages, every given problem description is the solution of the previous stage and the problem to be solved for the next stage. Thus, the problem and the solution are co-constructed by successive refinements of the description of the problem. Hence, a complete solution to the problem appears only at the end of the process. That is precisely because the complete definition of the problem to solve does not exists prior to the end of the process.

(4) **Hierarchical Refinement** The problem description at a given stage is refined by adding or removing properties and/or by detailing the existing properties. Thus, the description of the artefact to be designed evolves hierarchically by its successive refinements until appropriate detail level for the implementation is obtained. The hierarchical refinement of the problem create a tree-like structure where nodes corresponds to differ-

ent problem descriptions and arcs to instantiation/abstraction relations. At some stage during the process, if no further decomposition is possible (due to feasibility, lack of knowledge, etc.), a backtracking through the arborescence occurs and the process restarts with another (usually similar) problem.

(5) **Knowledge Dependency** The aim of the process itself is to *describe* an artefact which *did not exist* before. Thus, the primary resource being used in the process is knowledge and it is used to create new knowledge. We should mention that the knowledge created may not be so in an absolute scale, but only with respect to the knowledge of the agent(s) who assumed the problem solving task.

(6) **Learning** The agent(s) undertaking the task learns from the experience either by updating her (their) existing knowledge structures by using it, or by integrating to these structures newly created knowledge.

(7) **Search** As an admissible solution does not exist during the process (otherwise there would be no problem solving process at all) it must be looked for. Then, the process is a search process where possibilities offered by the knowledge structures are explored and knowledge structures are updated, in return, following the direction the search takes.

(8) **Limited In-depth Exploration** At a given intermediary stage during the process, a complete refinement of the problem is not possible. The current knowledge level on the variables, constraints, requirements and their interrelations, as well as on how to further decompose the problem to its sub-problems is limited. Thus, the in-depth exploration cannot exceed a certain limit.

(9) **Limited In-Breadth Exploration** The decisions taken on the refinement of the problem is crucial for the success. So the generation of a sufficiently great and diversified subset of the possible refinements space is important to ensure a satisfying representativity level. On the other hand, the refinement process is subject to time constraints, and *a priori* there is a very large number of alternatives to consider at a given level of refinement. Hence, only a limited number of worth-to-consider alternatives can be explored and evaluated.

(10) **Interacting Decision Processes** At each given detail level of the process, refinement decisions are taken to pursue the elaboration of the problem description. The decisions taken at later stages is heavily dependent on the decisions taken in the early stages. Also, a refinement decision concerning a subproblem may create conflict due to interactions with other subproblems. Finally, different parts of the problem may be refined in parallel. Thus, the process can be seen as a collection of interacting and/or overlapping decision processes.

(11) **Dynamically Evolving Preferences** During the process, every refinement attempt bring into light new information (that has been unknown or simply unconsidered). This new information may (and most probably will) (re)structure the preferences of the agent(s) who undertake(s) the

task about the purpose of the task and the way it should be achieved. Thus, preferences are constructed dynamically in parallel with the comprehension/representation of the problem.

We have seen that both the design and the planning processes have these characteristics, therefore they should be considered as equivalent.

5.4 Equivalent Processes, Different Outputs

We observe by the respective explanations in §4.3 that the design and the planning processes have similar purposes and progress in a similar way (§2 and §3). Also, similar notions are used to describe their nature, as illustrated in §4.1. Furthermore, they share the same main characteristics (§4.2). But is there no difference? After all, a plan computed by a robot is rather a plan and not a design, and the design of a car is not executable in the real sense of the term. Then what is the difference?

The difference lies in the nature of the description produced by the two processes. In what we usually call 'planning' the description obtained is a procedural one, whereas, in 'design' the final description is a declarative one. But, then again, what if we want to 'design' a procedural 'plan' that, let's say, a robot arm will use thousands of times to accomplish a task on a production line.

As far as the two problem solving processes aim to produce a 'description' of some solution for a somewhat new problem situation, we may consider that this difference is not essential. At least, not when trying to determine the correct principles for devising adequate support tools. The nature of the outputs changes, but the cognitive process by which these are obtained remains similar. Hence, to our opinion, the equivalence between the two process holds.

Thus far we have tried to point out the equivalence between the two processes. Let us now discuss the importance of this result and how it can be exploited in the following last section.

6 Implications and Research Directions

Through out this paper, we stressed that planning and design activities share some essential characteristics as problem solving processes and from a cognitive point of view they can be considered as equivalent. Although some interactions between the corresponding research fields exist (see e.g. (Nau et al., 2000), (Polyak, 1998), (Gupta et al., 1996)), this equivalence has not been fully exploited yet. To our opinion, three important potential benefits arise. First, the equivalence provides a framework in which the joint research efforts of design research, AI and CP can be concentrated, to the benefit of all the three fields. Second, it offers the possibility of proposing a model of the design process based on the Hierarchical Task Networks (HTN) Planning formalism of AI. Third, such a model would facilitate, together with the results established in this paper, interactions of design research with Decision Aiding Sciences.

6.1 Design and Cognitive Psychology

Design activities are where the human intellectual capacities co-exist in their richest forms. Among those are learning, reasoning, decision-making, creativity, knowledge use (storage, retrieval, processing), etc. Obviously, the above mentioned capacities are within the set of mental activities studied by cognitive psychology. It is therefore natural to think that design research might benefit from the rich concepts, models and theories of cognitive psychology to better comprehend the nature of the cognitive activities of designers. Going in the reverse direction, design activities must surely offer an important field of validation and experimentation for cognitive psychology. The mutual benefits of interactions between design and psychology is also emphasized in (Pahl et al., 1999). We believe that the equivalence we have established is an illustration of that. The key notions and ideas of cognitive psychology of planning *offer* the possibility to improve our understanding of the 'designerly ways of thinking' (Cross, 1999).

6.2 Design and Artificial Intelligence

A wide variety of tools emanated from the AI paradigm to assist design activity in different manners. Presumably, the most dominant trend is the use of knowledge-based design support systems (KBDSS). This seems natural as most of the existing tools (such as database exploration techniques, generation of alternatives, etc.) may be integrated in such systems.

Many successful implementations are reported in the literature, but usually, theoretical foundations are not considered in depth. However, to understand the limits of these managerial tools and to improve them, such foundations are necessary.

We believe that the HTN AI Planning formalisms developed in the AI Planning field (independently from the AI-in-design) can form the basis for the needed foundations, considering the equivalence between planning and design processes. In fact, the research in planning (in the sense that we have defined it above) lies within the intersection of artificial intelligence and cognitive psychology. The theories and models of the cognitive psychology have some formal counterparts in HTN AI Planning (although not necessarily because of an interaction). Therefore, we think that HTN Planning should enable us to define formal design support models whose underlying principles reflect the essential cognitive aspects of the design process in conformity with the findings in cognitive psychology planning (CPP).

In other words, such models would be explicative vis-à-vis the cognitive aspects of the design process as it is deemed necessary by (Cross, 1999), since they will have their roots in cognitive psychology. To rephrase (Cross, 1999) again, designers should "be able to use them in ways that are cognitively comfortable".

Applying the techniques, models and theories of CPP and AIP to a complex activity such as design is not straightforward however, since there exists some gap between the two disciplines as well. Once again, we should expect that AIP and CPP will extensively benefit from such an undertaking.

6.3 Where does 'Decision Aiding Sciences' fit?

We have identified some important characteristics of the design/planning processes about the absence of an initially fixed set of solutions, the dynamic construction of preferences in parallel with the construction of the problem/solution and the interactions of the decision processes involved. What should be the implications for design decision support?

To our opinion, these characteristics, together with the others mentioned before, characterize the design/planning process as a particular decision process. Yet, to our knowledge, decision theory has seldom paid any attention to such decision processes. The emphasis has been mostly on the evaluation procedures, assuming that the decision maker has well-shaped, preestablished preferences and/or the set of alternatives are given. Even if constructing the set of alternatives has been frequently acknowledged as a part of the decision aiding process, what kind of tools, methods or methodologies may support this part has not really been considered.

We believe in the necessity of developing decision aiding approaches that will support not only the evaluation of alternatives or the elaboration of preferences or the construction of a set of alternatives, but all of these in parallel.

We should mention at this point that a particular school of thought, often referred to as 'European School' in decision aiding sciences, has adopted a set of principles that fits particularly well to the approach needed for design/planning processes with respect to the characteristics we have enunciated (Roy and Vanderpooten, 1996), (Roy, 1993): “The main objective is to construct or create something (e.g., a value or utility function, a crisp or fuzzy outranking relation, the conviction that a certain alternative is the best, etc.) which by definition does not completely pre-exist. This entity to be constructed or created is viewed as likely to help an actor taking part in the decision process either to shape and/or argue and/or transform his preferences or to make a decision in conformity with his goals” (Roy and Vanderpooten, 1996). This “constructivist” decision aiding approach’s main motivation is to provide the decision maker with recommendations based on some knowledge obtained from the use of some decision aiding tool in order to assist the decision maker in elaborating his preferences as well as in clarifying his/her problem-solution pair.

We think that a model of the design process based on the Hierarchical Task Networks (HTN) that will integrate the dynamic construction and application of preferences would strengthen, together with the results established in this paper, the relation of design research with Decision Aiding Sciences, allowing a more effective transfer of the concepts and methods of the constructivist decision aiding approach to the field of design. It would do so by providing a better understanding of and technical (computational) basis for the design/planning processes. The construction of such a model forms one of our main research directions.

References

Bartlett, F. C., 1932. Remembering: a study in experimental and social psychology. Cambridge University Press.

Bowen, H. K., Clark, K. B., Holloway, C. A., Leonard-Barton, D., Wheelwright, S., sept-oct. 1994. Regaining the lead in manufacturing. Harward Business Review , 108–120.

Chandrasekaran, B., Goel, A. K., Iwasaki, Y., 1993. Functional representation as design rationale. IEEE Computer 26 (1), 48–56.

Chapel, V., 1997. La croissance par l’innovation intensive: De la dynamique d’apprentissage à la révélation d’un modèle industriel, le cas TEFAL. Ph.D. thesis, Ecole des Mines de Paris.

Coyne, R. D., Rosenman, M. A., Radford, A. D., Balachandran, M., Gero, J. S., 1990. Knowledge-Based Design Systems. Addison-Wesley.

Cross, N., 1999. Natural intelligence in design. Design Studies 20, 25–39.

Currie, K., Tate, A., 1991. O-plan: The open planning architecture. Artificial Intelligence 52 (1), 49–86.

Darses, F., Falzon, P., 1996. La conception collective: une approche de l'ergonomie cognitive. In: De Tersac, G., Friedberg, E. (Eds.), Coopération et Conception. Octares Editions, Toulouse.

Evbuomwan, N. F., Sivaloganathan, S., Jebb, A., 1996. A survey of design philosophies, models, methods and systems. Journal of Engineering Manufacture 210, 301–320.

Georgeff, M., 1990. An introduction to planning. In: Allen, J., Hendler, J., Tate, A. (Eds.), Readings in Planning. Morgan Kaufman, pp. 5–25.

Gero, J. S., 1990. Design prototypes: a knowledge representation schema for design. AI Magazine 11 (4), 26–36.

Gero, J. S., 1996. Creativity, emergence and evolution in design: concepts and framework. Knowledge-Based Systems 9 (7), 435–448.

Gero, J. S., 1998. Design tools that learn. Advances in engineering software 29 (10), 755–761.

Gupta, S., Das, D., Nau, D., 1996. Generating redesign suggestions to reduce setup cost: A step towards automated redesign. Computer Aided Design 28 (10), 763–782.

Hoc, J. M., 1992. Psycologie Cognitive de la Planification. Sciences et Technologie de la Connaissance. Presses Universitaires de Grenoble.

Kannapan, S., Marshek, K. M., 1996. A comparative analysis of techniques in engineering design. In: Waldron, M. B., Waldron, K. J. (Eds.), Mechanical Design: Theory & Methodology. Springer Verlag.

Kutluhan, E., 1995. Hierarchical task network planning: Formalization, analysis and implementation. Ph.D. thesis, University of Maryland.

Lawson, B., 1980. How Designers Think. Architectural Press.

Le Masson, P., Weil, B., 1999. Nature de l'innovation et pilotage de la recherche industrielle. Cahiers des Recherches, Ecole des Mines de Paris - Centre de Gestion Scientifique (16).

Logan, B., Smithers, T., 1993. Creativity and design as exploration. In: Gero, J., Maher, M. L. (Eds.), Modeling Creativity and Knowledge-Based Creative Design. Lawrence Earlbaum, pp. 139–175.

Love, T., 2000. Philosophy of design: a meta-theoretical structure for design theory. Design Studies 21 (3), 293–313.

Midler, C., 1993. L'auto qui n'existait pas: Management des Projets et Transformations de l'Entreprise. InterEditions.

Moraitis, P., Tsoukiàs, A., 2002. Multiple criteria evaluation of actions in hierarchical decomposition, Working Paper.

Nau, D., Ball, M., Baras, J., Chowdhury, A., Lin, E., Meyer, J., Rajamani, R., Splain, J., Trichur, V., 2000. Generating and evaluating designs and plans for microwave modules. AI in Engineering Design and Manufacturing 14, 289–304.

Ostanello, A., Tsoukiàs, A., 1993. An explicative model of public interorganizational interactions. European Journal of Operational Research 70, 67–82.

Pahl, G., Beitz, W., 1984. Engineering Design. The Design Council.

Pahl, G., Frankenberger, E., Badke-Schaub, P., 1999. Historical background and aims of interdisciplinary research between Bamberg, Darmstadt and

Munich. Design Studies 20 (5), 401–406.

Payne, J. W., Bettman, J. R., Johnson, E. J., 1993. The Adaptive Decision Maker. Cambridge University Press, Cambridge, England.

Perrin, J., 2001. Concevoir l'innovation industrielle, Méthodologie de conception de l'innovation. CNRS Editions, Paris.

Piaget, J., 1977. Recherches sur l'Abstraction Réfléchissante. Presses Universitaires de France.

Polyak, S., 1998. Applying design space analysis to planning. In Workshop on Knowledge Engineering and Acquisition for Planning: Bridging Theory and Practice, pages 40-47, Pittsburgh, June 1998.

Roy, B., 1993. Decision science or decision-aid science? European Journal of Operational Research 66 (2), 184–203.

Roy, B., Vanderpooten, D., 1996. The european school of MCDA: Emergence, basic features and current works. Journal of Multi-Criteria Decision Analysis 5, 22–38.

Sacerdoti, E. D., 1974. Planning in a hierarchy of abstraction spaces. Artificial Intelligence 5 (2), 115–135.

Schank, R. C., Abelson, R., 1977. Scripts, plans, goals and understanding. Hillsdale, Erlbaum.

Simon, H. A., 1969. The Sciences of the artificial. MIT Press.

Simon, H. A., 1973. The structure of ill structured problems. Artificial Intelligence 4, 181–201.

Tate, A., 1990. Generating project networks. In: Allen, J., Hendler, J., Tate, A. (Eds.), Readings in Planning. Morgan Kaufman, pp. 291–296.

Tate, A., 1996. Towards a plan ontology. Journal of the Italian AI Association (AIIA) 9 (1), 19–26.

Todd, P. M., Gigerenzer, G., 2000. Précis of simple heuristics that makes us smart. Bahavioral and Brain Sciences 23, 727–741.

Tsuneto, R., Hendler, J., Nau, D., 1998. Analyzing the external conditions to improve the efficiency of HTN planning". In *proceedings of* AAAI-98.

von der Weth, R., 1999. Design instinct? - the development of individual strategies. Design Studies 20, 453–463.

Wilkins, D., 1990. Domain–independent planning: Representation and plan generation. In: Allen, J., Hendler, J., Tate, A. (Eds.), Readings in Planning. Morgan Kaufman, pp. 319–335.