Object Structure and Action Requirements: A Compatibility Model for Functional Recognition

Lucia M. Vaina*† and Marie-Christine Jaulent*‡
*Boston University, Intelligent Systems Laboratory, College of Engineering, Boston, Massachusetts 02115 and †Harvard-MIT Division of Health Sciences and Technology, Cambridge, Massachusetts 02139

For an intelligent entity to carry out tasks in the real world, perceived three-dimensional shapes are transformed into objects identified by their functional category which makes explicit the roles or uses of objects in actions. This study describes an approach to the recognition of functions that combines ideas about representations of shapes, concepts, and object categories, with goal requirements of actions. A particular conceptual model of the compatibility between objects and actions is introduced, the outline of the solution is given, and the experimental domain of hand actions and of objects useful in such actions is described; the solution is currently under implementation and computational verifications.[1] The article is organized as follows: first, the computational definition of the problem of functional recognition and a comprehensive theoretical framework for it is given and, second, the relation between primary functions of hand-manipulable objects and auxillary functions is discussed and modeled in the framework of fuzzy sets and possibility theory.

I. INTRODUCTION

One of the main goals of computer vision is the construction of explicit, efficient, and useful descriptions of physical objects. These descriptions are necessary for recognition, manipulation, and reasoning about the physical world. Most of the object recognition systems so far have concentrated on the recovery of three-dimensional structure of objects from the image. On the basis of structural, observable properties the object may be assigned to a category. In order to be able to carry out tasks in the real world, an intelligent recognition system must be able to transform the structural knowledge about objects into functional knowledge, that is, knowledge which makes explicit the usefulness

‡M-CJ is currently at the Service d'Informatique Médicale, Assistance Publique, Hospital Broussais-la charité.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 6, 313–336 (1991)
 CCC 0884-8173/91/030313-24$04.00

of objects in actions. Functional knowledge is an intrinsic part of the concept of an object. But the relation between the structure and the function of objects is often not directly available and thus it must be extracted through reasoning. The objects in Figure 1 have very different shapes but by and large all have the same function of *cutting*.

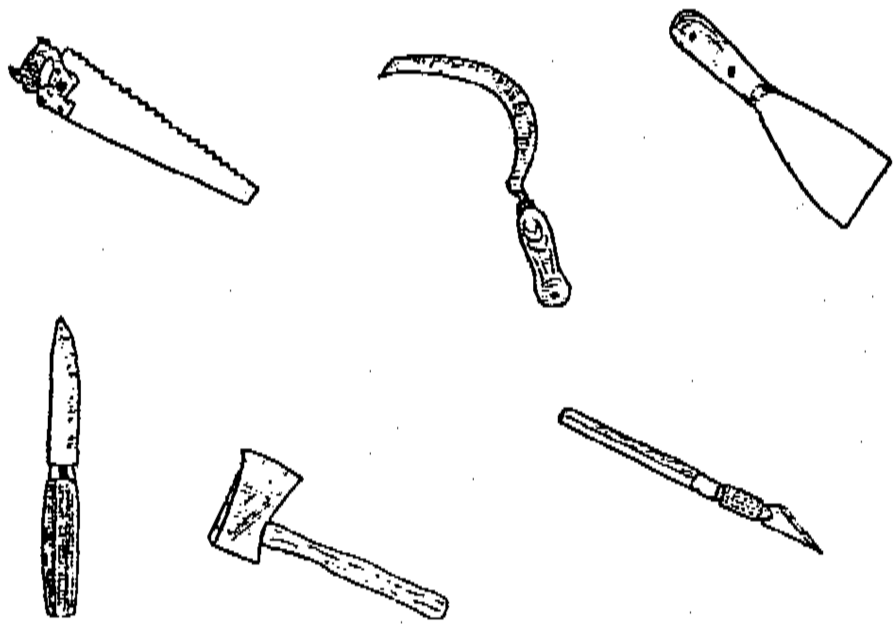

Figure 1

Most man-made objects are designed with a use or a goal in mind: containers are useful for storing material, sharp blades are useful for cutting, hammers are good for pounding, objects which have a flat surface of a certain size are good for sitting on, and so forth.

Thus, naturally, functionality ought to be represented in memory by some schema which indicates how an instance of an object is used. This observation is not new. The Gestalt school has already suggested that objects have a "demand character,"[2] they demand an action while they reject other actions. Later, Gibson[3] boldly states that what is usually perceived of an object are not its individual physical qualities, but its "affordances," that is, its properties which mediate function. There is a fundamental difference between these two considerations of objects' functions. For the Gestaltists the functions are bestowed on the object by the need of the perceiver, while Gibson considers the affordances as intrinsic in the physical characteristics of an object. In the Gibsonian view, the affordances of an object are immediate and they need not be extracted through computations. This is *essentially* different from the approach proposed here, which is computational, and it assumes that the functional information is only implicit in the representation of an object and to make it explicit requires processing.[4]

How may this be done? To understand the problem of functional recognition requires having some idea of which representations to use, and only then can one proceed to analyze the computational problems that arise in obtaining and manipulating each representation. The questions we will ask in this article revolve around the nature of the computational problems that must be solved in order to determine the compatibility of an object with a given action. We will propose a conceptual model of **compatibility**. Our focus is on a development of the compatibility model for which objects and actions will be described in terms of functional and structural features. For an object that has been specifically designed to implement an action, (e.g., bread knives are designed for cutting bread, hammers for driving nails) the compatibility is obviously trivial: the action requirements are automatically matched by the object's functional properties. The compatibility problem *is a problem*, when we try to determine the usefulness of objects for actions other than those which directly embody the designer's goal.

In a subsequent study,[1] we will define and implement an algorithm which takes objects' descriptions and action requirements as inputs and maps them into an output space that describes the functional compatibility between objects and actions.

The organization of this article is as follows: first, we will discuss a general theoretical framework for the functional representation. Second, in order to show that these constraints can be embodied in an algorithm, we will choose examples of appropriate representations for describing a class of manufactured objects and action–requirements for the usefulness of such objects in achieving the purposes of actions. The problems involved in the representation of actions are beyond the scope of this article. Third, a formal model of **functional compatibility** is proposed in the framework of possibility theory.[5]

II. GENERAL FRAMEWORK FOR FUNCTIONAL REPRESENTATION

The goals of correctness, efficiency, and robustness of a functional recognition system place many constraints on its design and performance, especially on the nature of the representations that may be used.

A. Object Prototypes and Functional Prototypes

In the effort to understand the relations between perception and categorization, numerous studies in cognitive psychology,[6,7] have explored the horizontal and vertical structure of the natural categories. The horizontal structure or the within-category organization determines the membership or the degree of membership to a category. The vertical structure, on the other hand, concerns the relationships between categories which often take the form of nested hierarchies extended from the superordinate categories (e.g., tools) to subordinate categories (e.g., hammer, wrench, screwdriver). One particular level is called the "basic level" categorization. *Hammer* is a basic level category, as is *screwdriver;* both are included in the superordinate category *hand tool*. Upholsterer's hammer, the Club hammer, and the Veneer's hammer are all subordinates of the category hammer.

Categories group things that share attributes, and the "basic level" categories represent a level of abstraction in which correlations and discontinuities among objects are most salient. Thus, the information available in the basic level category is the greatest, as in the case of hammers, cars, and chairs. These categories are disjointed thus constituting a preferred level of reference. This sort of information is what people usually know when they know "the categories that are named by nominal expressions in their language."[8] Categories at the basic level are more fundamental psychologically than those at other taxonomic levels,[6] and it has been argued that they are the most general categories whose members have similar shapes and functions.[9]

However, functional or practical knowledge[4,8] by itself is invoked to determine whether an object can be used in an action or whether an object can serve a certain purpose (the goal of the action). Thus, function is represented by some schema that makes explicit how an object can be used.

The problem of the functional recognition of objects is also a categorization problem but this categorization depends on criteria which are specific to the action. Functional categories defined in terms of common functions[10] are proposed as an alternative to the view that perceptual features may form the only basis of categorization.[11] This polarity "perceptual versus functional" conceptualization cannot hold[12] and it has been suggested that it constitutes a "relationship of one large set of perceptual processes that can be subdivided into several subsets."[13] The interest is to learn how the perceptual information may be organized into utilitarian-effective[13] categories, that is into functional information. A useful functional categorization is often defined as *potential means to achieve a certain kind of goal,*[14] and (functionally defined) concepts are formed on the basis of extracting the invariant aspects of objects which can fulfill the same function. This invariant aspect of objects constitutes the "functional prototype."

Our basic hypothesis is that the "functional prototype" is different from the "basic level" or the prototype as defined in object categorization. We submit that the functional classification organizes the information about objects into taxonomies varying in the degree of compatibility between action–requirements and object properties. For example, if one wants to drive a nail into a board, the natural thing to use is a hammer. We may know this from the lexical or categorical knowledge domain at the "basic category" level, for it is often associated with our learning through language. However, we would also know that a rock, a certain part of a pliers or, in some circumstances, the handle of a screwdriver may accomplish the same goal. This sort of knowledge, however, is different in nature from the "basic category" knowledge of the function of the hammer, for it is derived through inference on the physical properties of these objects and the requirements of the action *to pound (a nail)*. The set of physical properties and relations necessary for accomplishing an action, that is the properties and relations which satisfy a specific set of requirements corresponding to the accomplishment of the goal or purpose of an action, define a **functional prototype**.

Consequence 1: Primary functions. It is natural that objects would be put together so as to maximize efficiency of the action. The primary use of an object, the function for which it has been designed, must closely associate the requirements of a specific action and an object's structural part or parts maximally compatible with those requirements. An object designed for cutting, for example, has a physically identifiable part, the blade, which directly implements the action, and another part, the handle, which facilitates the implementation of the action. The handle affords grasping of the object while the blade affords its specific use for cutting. The blade is both necessary and sufficient for this task for it constitutes an instance of the functional prototype of the action cutting.

We submit that the primary function must be made explicit at the basic category level and that there is a direct rule like association to the action.

Consequence 2: Auxiliary functions. Objects may be used in actions other than those for which they were specifically designed.[4] A knife's primary function is cutting but in some cases, it also can be used as a screwdriver. If the tip is structurally a small blade, as in the flared or parallel screwdriver for example, then the knife can play the same functional role. However, this may not be applicable for a phillips or torx type of screwdriver. A knife can also be used to transport a small amount of salt, it can be thrown, or it can be used for stirring minestrone soup. All these are additional or **auxiliary functions** which are only implicit in our knowledge about objects and must be extracted from the physical structural descriptions by an action-guided inference ("naive induction").

Remark. Object recognition and functional recognition often rely on different sources of knowledge. The basic level categories are very useful for object recognition but the only functional information one obtains directly at this level is related to the primary function of the object.

B. Principle of Least Commitment

The principle of least commitment[15] states that one should never do something that may later have to be undone. In the functional recognition task, there may often be several possible interpretations of a particular piece of information but there is not yet sufficient evidence to decide between them. In such cases, one should not be committed to one of these interpretations prematurely since knowledge associated with that possibility and not with the others may be conducive to error.

When several possibilities compete for describing a particular datum, usually constraints or measures of preference will operate among them.

C. The Importance of Parts

Function relating objects and actions for reasoning about objects is closely related to our motor knowledge and this relationship is mediated through object parts and their configuration. The structure and the description of parts form the appropriate level of abstraction for the representation of object functions.

Consequence 1: Structural parts. Representations of objects must take into consideration the importance of parts and provide a direct link between the description of parts as structural components of objects and functional properties, on one hand, and shapes associated with geometrical descriptions, on the other hand.

Consequence 2: Nonstructural parts. Functional parts are not always structural parts, they may be internal or enclosed parts which are not made explicit in the description of the shape of the object (e.g., seeds of a fruit, drawers of a desk, the cover of a box, or the screen of a television). Such parts are described by the relationship to an action (e.g., the drawers of a desk) and by their function (e.g., the television screen).

Consequence 3: Function intrinsic to the object. In a restricted way, function is considered to relate objects to actions of a human agent. However, functionality can be extended to the "working" of the object, independent of its uses. A certain type of a chair, has a seat which affords "sitting," it has a back, which affords "leaning," and it has legs which support the chair seat at a certain height. Although the legs parts are not directly interacting with the user, they do participate in facilitating the action of sitting.

We will restrict this study specifically to the problem posed by structural parts and their functions in different actions and we will not address issues involving the nonstructural parts.

D. Functions of Natural Objects

Man-made objects have always a primary function relative to a human agent, a hammer is designed for pounding, a screwdriver for driving screws. Natural objects have no primary function. Rocks, sticks or plants are not designed by nature with a purpose in human actions; they can still be useful beyond their primary function. A small rock can be used as a hammer, a large flat rock can be a table or a chair, a large shell can be used as a container, a stick can be used to make a hole, and a branch can be a hook. A hammer, a rock or a screwdriver can all be useful for pounding. Thus, the uses of the natural objects in actions are similar to the auxiliary functions of the manufactured objects and, thus, their compatability with actions relies on an inference based procedure.

E. Function in Actions not Bound to an Implement

Functional knowledge refers to the accomplishment of those actions which are not associated with a specific manufactured implement (e.g., throwing, pushing or placing down something) or actions in which the object is a natural, physical object which has not been designed for a specific use by an agent. These are actions for which there is not a direct functional association with some specific basic level object category (or with a prototype). The compatibility between such actions and the functionality of objects is less automatic for, while in the former case the basic level category (the prototype) constitutes a strong bond between the action and the object, in the latter case, perceptual inferences are used for extracting the possible use of an object in the action. In the first case, the association is acquired through learning or experience, and the pair action–object prototype belongs to a conceptual knowledge base. In the second case, the association is between action–requirements and the range of admissible values of the object properties (e.g., size, weight, length).

F. Cognitive Economy

For a useful and efficient functional representation, the number of stored categories (functional categories) is small in comparison to the amount of data encountered. Differentiations occur only when it is necessary for attaining a goal or, as Rosch[7] put it, "it is the organisms's advantage not to differentiate one stimulus from others when the differentiation is irrelevant to purposes at hand."

Cognitive economy promotes efficiency of access of information,[16] which is particularly important for manipulating the functional knowledge and for quick action.

Consequence: Information organization. Cognitive economy is information compression and within a given subset the members are considered equivalent for the current purpose. Thus, a measure of similarity must be defined on the information and, for organizing information about objects functions, the similarity is defined relative to a specific purpose or action.

III. RECOGNITION OF OBJECT FUNCTIONS: A COMPATIBILITY PROBLEM

How can a representation capture the fact that an object has a primary function and other possible or auxiliary functions? Vaina[4] suggested that a functional representation must express the different uses of an object by "answers" to action demands. Essentially an action would be associated with a list of requirements which must be satisfied by any object that is functionally compatible with the goal of that action.

In this article, we are proposing a compatibility model between action requirements and objects properties. Thus, we suggest that there is a rule-like association between the action requirements and the properties of an object which describe its primary function. The association is established between the action requirements and the object prototype or basic category. The relationship between action requirements and object description are function dependent. Thus, the prototype function, or the primary function, is associated with a part in the object prototype. The prototype is inextricable from the action and the bond is expressed in the primary function, and often the specific perceptual characteristics (e.g., geometric, color, texture) may not be made explicit. The auxiliary functions, on the other hand, are associated with descriptions of the object structure.

We define the functional recognition problem as a problem of finding the compatibility between actions requirements and objects properties.

IV. OBJECT AND GOAL REQUIREMENTS

Evaluation of a similarity is information-dependent and thus we must first choose the representations that can be assumed available for describing the objects and the actions whose functional compatibility is searched.

A. Object Representation

The object domain addressed in this study consists of manipulable man-made objects which, by and large, consist of physical parts assembled in a specific way. The current artificial systems for the recognition of such objects have focused mostly on the representation of the structure of the object, that is on "what the object looks like." However, a representation whose goal is to recognize the functions, or uses of objects in actions, would allow one to ask questions such as: "what are objects for?", "what can be done with an object?", and "how do you grasp an object to achieve a specific goal?".

In order to address such questions we suggest a three-module representation of objects.[17,18] These modules are: the object–category, the object–concept, and the object–structure.

The **object–category** module is a hierarchical representation of objects in terms of their semantic categories (the category module in Vaina[17,18]). Thus, for example, a cup is an object, man-made, manipulable, and a container. The role of this representation is to provide classification and taxonomies of objects. Such a representation is an economical organization of the domain of reference (e.g., man-made objects), by generating a succinct representation of knowledge[19] and by allowing quick inference of properties from higher superordinate categories to the subordinate categories.

The **object–concept** module makes explicit parts, their relations and functions. Topological properties are stressed rather than metric properties.[4] The relations are of different types such as *connections* which specify which parts are attached to each other and *positions* which describes where the connections occur on the parts.

The representation of the object prototype as a set of parts and relations also has a hierarchical structure. Each component in this hierarchy is defined by a propositional data base. Parts will have a type which is obtained from the object–structure description. The **type** refers to the shape primitive used to describe the part. Parts which occur at the same level of detail (e.g., the legs of a chair) will be described at the same level on the hierarchy. Similar to the Marr and Nishihara model,[20] the part which has the most connections with other parts will play a more central role (e.g., the seat of the chair). The presence or the absence of a part is often important for distinguishing between similar objects, or between the object prototype and a damaged instance of that object.

In addition to the description of parts and relations, the object–concept level also makes explicity the specific function associated with the object, usually by a direct link between a part and the action requirement that it satisfies. Only the primary function is made explicit in the object concept. Function is the most direct way of relating object to actions. The functional properties are described so as to be consistent with the actions requirements and form the basis of the compatibility model developed in this study.

The third module of the representation addresses the **structure** of the object; it describes the shape of the parts, their geometrical relations, and their visual attributes. The category of manipulable manufactured objects can be described by the combination of a limited vocabulary of primitive shapes and modifiers. For simplicity, we shall call the primitive shapes, **shape–prototypes**. Parts, as shapes, will also be described in terms of shape–prototypes and modifiers. Modifiers may be expressed as values of size, length, relative area and volume, position, protrusion, and as ratios of modifiers' parameters (e.g., height–width). Concavities and holes are expressed in negative values. Although the modifiers are general to most of the shapes in the domain of consideration, their application is different for each shape–prototype. The object–structure is described hierarchically, where the position of each part is defined in the coordinate system of the part on the preceding level.[20] For the class of objects addressed in this study, the shape prototypes may be modeled in terms of generalized cones,[20,21] superquadrics[22,23] or by the RCB model.[24]*

B. Goal and Action Representation

Actions are categorized by goals or consequences. The specific goal of *pounding* can be to drive a nail, to flatten a sheet of metal or to break a walnut, for example. These are different pounding actions which have in common the requirement to apply a certain force to an object. The quality and quantity of the force and the nature of the object will determine *which pounding* is instantiated and the required specific properties of the object whose function *can* be *to pound*. The object can be a hammer, a fist, a rock or any other object which has the appropriate properties to be handled with the goal of impinging enough force: flat surface, certain weight, graspability, and manipulability.

The properties necessary and sufficient for achieving the goal of an action are collectively called **action–requirements**. The action–requirements are associated to actions by a link REQUIRES, or a link FORBIDS, and are represented by triplets of the following types:

1) **Object–requirements.** (Action REQUIRES Object–Attribute:value) and (Action FORBIDS Object–Attribute:value). Examples are: (Hold–liquid REQUIRES Container:value); (Drive–Screw–Manually REQUIRES Blade:value) & (Drive–Screw–Manually REQUIRES

*For parts which cannot be expressed as generalized cones, we are developing a new decomposition of the shape into maximum convex patches and minimum nonconvex patches.[25]

Tip–type:value) & (Drive–Screw–Manually REQUIRES Handle–type:value).

2) **Movement–Requirements.** (Action REQUIRES Object–Movement) as in (Remove–manually–screw REQUIRES Counterclockwise–movement–of tip) or (Drive–manually–screw REQUIRES Clockwise–movement–of–tip).
3) **Grasp–Requirements.** (Action REQUIRES Object–Attributes:Grasp–Type).

The Object–Requirements and Grasp–Requirements are instantiated as object properties and values and the movement–requirements are instantiated as kinematic primitives.[1,26]

In this study, we are discussing the function of objects in actions of the hand and we will concentrate on the compatibility between objects and these types of actions. Specifically, we shall restrict the discussion to the Object–Requirements which generate the patterns for the functional compatibility of objects and actions. It is important to note that the different types of requirements can be decoupled, and that they may have different values. However, the object–requirements must be invariant; there is a set of properties necessary for achieving the goal of the actions. The Movement–Requirements and the Grasp–Requirements are both determined by the Object–Requirements.

For hand movements, for example, two main groups of hand movements have been identified: grasp and nongrasp movements.[27] In the latter, the hand is shaped into a supporting organ, e.g., pushing, or a hook–shape upon which something can be hung. The grasp or prehensile movements performed by the hand are variations of two fundamental but differing types of movements: power grip [Fig. 2(a)] and precision grip [Fig. 2(b)]. Both categories of grip have in common an essential requirement, that is, the stability of the object to achieve manipulation.[26]

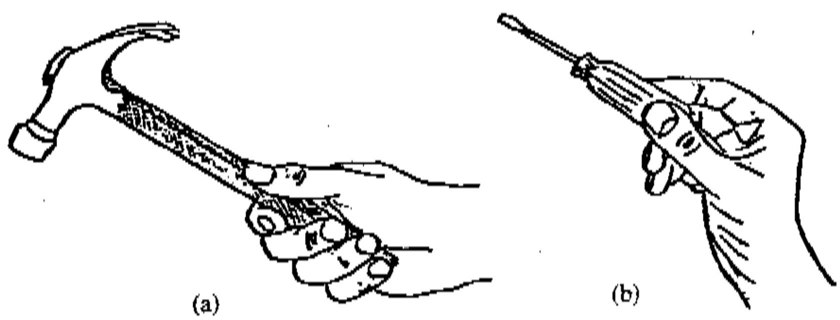

Figure 2

Which grasp is used in a situation is determined primarily by the **action** performed with the object and not by the shape of the object. In other words, it is the intended use of the object, or the **function**, which will determine the grip pattern.[27] This implies that a functional compatibility check must be done at the moment of the planning of the action. prior to initiating it. The functional compatibility evaluates those action–requirements which address the properties of the object.

4) **Object–Requirements.** The specific movement of the hand and arm to perform the action is also determined by the Object–Requirements. Thus, in the action *to screw* the specific movement of the hand is determined by the object used as a screwdriver, the real screwdriver would be manipulated differently from a knife, or coin, although, in some circumstances, all three may be used for achieving the goal of the action *to drive a screw*.

V. FUNCTIONAL COMPATIBILITY

In the simplest case, the functionality is directly associated with the concept of object at the "basic level" and is defined in terms of the structural description of the object part. In this case, the functional prototype of an object is an object which has been manufactured for implementing a specific action and the relationship between object properties and the object–requirements in the action description is an exact match.

Thus, for example, the general purpose hammer is the claw hammer [Figure 3(a)] which is used in general carpentry work and nail pulling. This hammer embodies two functional prototypes: one is for *driving nails* [Figure 3(b)], and the other for *pulling nails* [Figure 3(c)]. It is interesting to note that although the

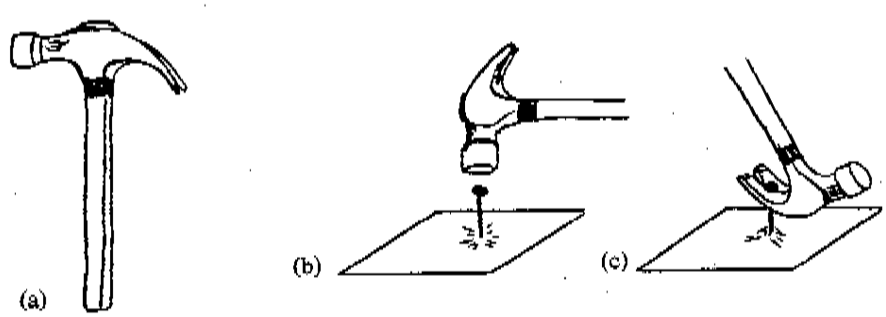

Figure 3

two functional properties are both carried out by the hammerhead at a finer level of structural description, they are embodied by different parts on the same level of the object–structure and object–concept hierarchy. These functional properties can be separated into two different objects, each fulfilling only one action. For example, the medieval carpenter's hammer which is still frequently used in Europe has been manufactured "*specifically*" for driving nails (it cannot be used for pulling nails) and it consists of an iron head of square section with a wedge shaped peen and a (wooden) handle.

In the context of an action, a functional property is an element of the object–requirements and it relates to the goal of the action. In the hammer example, "flat surface" is a functional property, expressed as (ATTRIBUTE (SPACE VALUE)), that is (POUND (SURFACE LARGER–THAN–NAIL–HEAD)). The nature of the values is determined by the descriptors used in the object–structure module. The principle of cognitive economy implies that in the object–concept, specific, exact numerical values must be broken into several conceptual "values" such as deep, shallow, tall, short, sharp, flat, and their meanings correspond to ranges of admissible values. If, for example, capacity is defined by descriptors or modifiers computed on the basis of the height–width ratio then, for a BOWL, ratio < 0.25 would mean shallow while a ratio > 0.75 would mean deep.

Such information is imprecise in the sense that the bounds of the set of values it represents are not well-defined. When does *deep* become *shallow*? What is the boundary? How does one determine the appropriate response, and how is the decision made as to whether a container, for example, is deep enough for using it as a cup? There is an element of uncertainty here which is not caused by measurement error or by chance.

Matching involves quantifiable similarity which evaluates the ressemblance between two entities. This can be expressed by minimizing the dissimilarity or maximizing the similarity,[28] by logically combining matching and mismatching properties,[29] or by using ad hoc metrics.[30] Tversky's **contrast similarity** model[31] is particularly interesting to us here, for the similarity between two objects depends not only on the features that they have in common but also on the features on which they differ. Tversky has demonstrated that categorical similarity is not psychologically symmetric. In all cases, the metrics of a measure of similarity is strongly information-dependent.

The matching techniques include the process of checking a structure with a set of rules which describe constraints[31,32] ("labeling techniques") as well as the process of matching relational structures (semantic nets)[33] ("graph isomorphism techniques").

When the two entities to be matched are similar (have the same nature and structure), the compatibility between the pattern and the data may be expressed by a logical combination of the compatibilities for each component of the pattern.[34,35] For instance, for relational structures, compatibilities are established for the nodes of the structure and the combination is performed on the links.

When the entities are different (e.g., when geometrical data is fitted against conceptual knowledge), the matching process may be interpreted as an inference process since the data are described in a different (more abstract) representation.

In our model, the pattern is given by the object–requirements, the data by the conceptual level of an object and thus the pattern matching operates at the conceptual level. However, when an attribute does not appear in the object–concept description, a different pattern matching is triggered at the structural level between the prototype-part associated with the attribute, such as the physical properties of a blade part for the attribute "cut" and the object–structure description. It is important to note that the data to be matched are always of the same nature as the pattern since we use both object–concept and object–structure descriptions. Here, the inference process just transports the compatibility established at the structural level to the conceptual level.

This view of the pattern matching procedure (using different and cooperative representations of the same data) is attractive for it allows two types of information (conceptual and structural) about an object to account for similarity. It also supports the empirical evidence of the privilege of the **primary function** in the functional recognition. An object which at the basic level has a blade as one of its parts would be correctly identified as something useful for cutting. However, a cup with a sharp rim, is less compatible with the object–requirements associated with the action "cut" than a knife with a thin blade.

In our model the pattern is encoded in a graph structure where the nodes are **functional properties** and the links can be of two types: **type–part** links and **connections**. The links constrain the structure of the pattern and the nodes mediate the partial matching. A type–part link is used when the required functional property has to be located specifically in a part and a connection is used when two parts are required to be physically attached. We show below an example of the object requirements associated with "to chop a log" (Fig. 4).

Example: Action: "to chop a log"

- Object requirements: 1-(cutting (MATERIAL wood))
  2-(graspable (LENGTH long))
- Type-part links: 1 and 2 must be located in two different parts.
- Connection: 1-2 3-(relative-orientation orthogonal)
- associated graph structure:

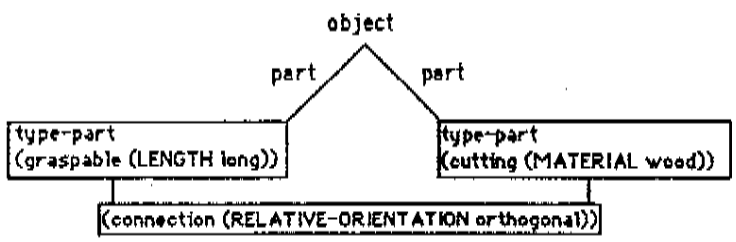

Figure 4

In this study, we will discuss two types of compatibility: **local** (or elementary) **compatibility** and **global compatibility**. The global compatibility between the pattern and the object is expressed by an aggregation of the elementary compatibilities evaluated for each functional property. The aggregation takes into account the structural constraints (type–part and connections constraints) required in the set of functional properties. The estimation of the elementary compatibilities and the modeling of the aggregation operations are based on possibility theory.[5,36]

A. The Notion of Compatibility in the Possibility Theory

The notion of compatibility between two fuzzy sets was introduced by Zadeh.[5] This notion was further used to develop compatibility models for semantic pattern matching procedures.[37,38] In such models, the compatibility refers to the truth of the information relative to a given knowledge and is expressed by two scalar measures: the possibility and the necessity that a fuzzy set $X2$ (the information) is compatible with a fuzzy set $X1$ (the knowledge).

Definitions:

Given two normalized* fuzzy sets $X1$ and $X2$ defined on a referential R by their membership function μ_{X1} and μ_{X2} we define the following:

- $\Pi(X1/X2) = \sup \min(\mu_{X1}(x), \mu_{X2}(x))$ for x in R expresses the POSSIBILITY that $X2$ is compatible with $X1$.
- $N(X1/X2) = \inf \max(\mu_{X1}(x), 1 - \mu_{X2}(x))$ for x in R expresses the NECESSITY that $X2$ is compatible with $X1$. (the necessity is the dual measure of the possibility[5]).

These two scalar measures have relations and properties which depend on the nature of the values $X1$ and $X2$ (fuzzy sets, classical sets, single values). In particular, when $X1$ and $X2$ are two single values, we have: $\Pi(X1/X2) = N(X1/X2) = 1$ for $X1 = X2$, and 0 when $X1$ is different from $X2$.

It is important to observe that while the expression of the possibility measure is symmetrical, the expression of the necessity measure is not. The definition of the necessity function makes it useful for characterizing compatibility because it is sensitive to the asymmetry of similarity: "*a* and *b* are similar to each other" is different from "*a* is similar to *b*".[31] A *knife* may be used to drive in a screw, but a screwdriver cannot be used for cutting (except in very special conditions, such as cutting through a soft cake).

B. Elementary Compatibility

The elementary compatibility is defined between two functional properties. A functional property is given by the expression (attribute (SPACE value)). In the following, we will consider two general functional properties $F1$ and $F2$, where $F1$ is an element in the set of object–requirements and $F2$ is an element in the object–concept.

$F1 = (A1\ (S1\ X1))$

$F2 = (A2\ (S2\ X2))$

The compatibility of $F1$ and $F2$ is evaluated only if $A2$ is equal to $A1$.

If $A2 \neq A1$, the compatibility is not evaluated and the absence of the required functional attribute $A1$ in the object–concept will trigger the pattern matching procedure at the structural level.

When $A1$ is equal to $A2$ the compatibility is expressed by the conjunction of the following:

— $A2$ has a value defined on $S1$ (let say $X2'$)
— $X2'$ is compatible with $X1$.

*A normalized fuzzy set is a set such that it exists a value x in R for which $\mu_A(x) = 1$.

The relation between $S1$ and $S2$ determines the complexity of the process involved in the compatibility evaluation:

Case 1: The simplest case occurs when $S1 = S2$ then we have $X2 = X2'$ and the pair $(\Pi(X1/X2), N(X1/X2))$ provides the compatibility between $F1$ and $F2$.

Case 2: When $S1$ and $S2$ address the same property in a different context, then $S1$ and $S2$ represent the same space but with different dimensions (e.g., capacity for "extinguish a fire," capacity for "drinking"). In this case, $A2$ has a value $X2'$ defined on $S1$ by:

$$\forall x \in S1, \mu_{X2'}(x) = \mu_{X2}(x)$$

and the compatibility is given by $(\Pi(X1/X2'), N(X1/X2'))$.

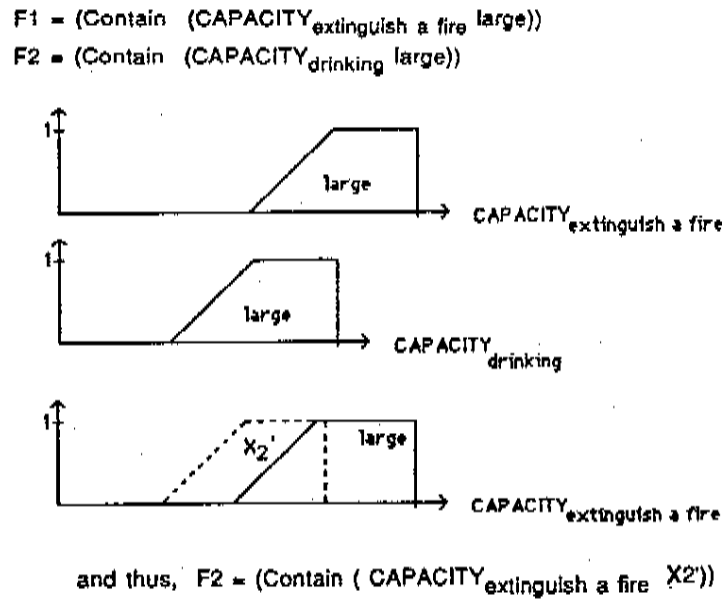

Figure 5

Case 3: The most complex case occurs when $S1$ and $S2$ do not represent the same space. For instance, $F1$ = (Cut (MATERIAL hard)) and $F2$ = (Cut (SHARPNESS sharp)).

In this case, the compatibility between $F1$ and $F2$ depends on the existence of the value $X2'$ which in turn depends on the existence of a correspondence between the two spaces S1 and $S2$.

We assume that several referentials intervene simultaneously in the definition of a complex space. Thus, this space can be defined as the cartesian product of these referentials.

$$S1 = \Omega1_1 \mathrm{x} \Omega1_2 \mathrm{x} \ldots \mathrm{x} \Omega1_N$$

$$S2 = \Omega2_1 \mathrm{x} \Omega2_2 \mathrm{x} \ldots \mathrm{x} \Omega2_M$$

We assume also that the value $X1$ (resp. $X2$) is the cartesian product of the fuzzy sets $X1_1, X1_2, \ldots, X1_N$ (resp. $X2_1, X2_2, \ldots, X2_M$) defined on $\Omega 1_1, \Omega 1_2, \ldots, \Omega 1_N$ (resp. $\Omega 2_1, \Omega 2_2, \ldots, \Omega 2_M$). The membership function of $X1$ (resp. $X2$) is defined to be the intersection of $X1_1, X1_2, \ldots, X1_N$ (resp. $X2_1, X2_2, \ldots, X2_M$) that is (for instance, for $X1$),[39]:

$$\forall\, x = (x_1, x_2, \ldots, x_N)\ \varepsilon\ \Omega 1_1, \Omega 1_2, \ldots, \Omega 1_N,$$

$$\text{and } \mu_{X1}(x_1, x_2, \ldots, x_N) = \min(\mu_{X1_1}(x_1), (\mu_{X1_2}(x_2), \ldots, (\mu_{X1_N}(x_N))$$

We define a correspondence between $S1$ and $S2$ when we have: $\{\Omega 1_1, \Omega 1_2, \ldots, \Omega 1_N\} \cap {}^{-}\Omega 2_1, \Omega 2_2, \ldots, \Omega 2_M\} = \{\Omega p, \ldots, \Omega q\} \neq \emptyset$. Then, $A2$ has a value $X2'$ defined on $S1$ by:

$$V\, x = (x_1, x_2, \ldots, x_N)\ \varepsilon\ S_1,$$

$$\mu_{X2'}(x_1, x_2, \ldots, x_N) = \min(\mu_{X2_p}(xp), \ldots, (\mu_{X2_q}(xq))$$

The compatibility is given by $(\Pi(X1/X2'), N(X1/X2'))$.

Example:

$$F1 = (\text{afford–cut (MATERIAL hard)})$$

$$F2 = (\text{afford–cut (SHARPNESS sharp)))}$$

Let us assume that MATERIAL is a space defined by RIGIDITY × SHARPNESS × DIMENSIONS and that the value "hard" is defined by the cartesian product of "rigid," "sharp," and "large." We have

$$V\, x = (x_1, x_2, x_3)\ \varepsilon\ \text{MATERIAL}$$

$$\mu_{X2'}(x_1, x_2, x_3) = \mu_{\text{sharp}}(x_2)$$

Case 4: Finally, if $S1$ is a discrete space, the value $X1$ is considered as a single value. $\Pi(X1/X2) = N(X1/X2) = 1$ if $X1 = X2$, and is zero in all other cases.

C. Global Compatibility: A Problem of Aggregation

In the structure of the pattern, each atomic pattern can be considered as a partial–goal according to a given criterion. Thus, in the atomic pattern (Contain (CAPACITY deep)), the partial–goal is (CAPACITY deep) along the criterion Contain. The entire pattern represent the "global–goal", and it corresponds to the *Object–Requirements* component of in the representation of the action representation.

For a given object, the global compatibility with the goal is defined by a combination (or aggregation) of all the elementary compatibilities associated with the partial–goals.

This aggregation depends on the salience of the criteria of partial–goals as well as on the nature of the conjunctions between the different criteria which is embodies in the links (connections, type–part links).

D. Criteria Aggregation

For Gi the fuzzy set attached to a partial–goal Ci with a membership function μ_{Gi}, $\mathfrak{D}$ a catalogue of objects, and $\tilde{N}$ a set of objects in $\mathfrak{D}$ compatible with the global pattern (GP), the compatibility with the global–goal requires the existence of an application **h** such as:

$$\forall \text{ ob } \varepsilon \mathfrak{D},\ \mu_{\tilde{N}}(\text{ob}) = \mathbf{h}(\mu_{G1}(\text{ob}), \ldots, \mu_{Gp}(\text{ob}))$$
where p is the number of partial goals.

The operation **h** is a fuzzy operation,[39] which verifies the following natural conditions:

1) $\mathbf{h}(0, \ldots, 0) = 0$; $\mathbf{h}(1, \ldots 1) = 1$
2) $\forall (xi, yi)\ \varepsilon\ [0,1]^2$ with $\forall\ i = 1,p\ xi \geqq yi$ then $\mathbf{h}(x1, \ldots, xp) \geqq \mathbf{h}(y1, \ldots, yp)$

There is a large class of possible operation that can be defined for computing the aggregation.[39] We choose the "min" operation which expresses a simultaneous satisfaction of the partial objectives.*

We define the operation of aggregation as follows:

$$\forall \text{ ob } \varepsilon \mathfrak{D},\ \mu_{\tilde{N}}(\text{ob}) = \min\ (\mu_{G1}(\text{ob}), \ldots, \mu_{Gp}(\text{ob})) \qquad (1)$$

where, for the evaluation of the global compatibility, the min is taken over the whole set with no distinction between the criteria.

In practice, however, one may want to express that some criteria are more important for the achievement of an action–goal. For instance, the required functional property which corresponds to the implement part of an action is obviously more salient than the others. Such an operation can be done by weighting the criteria for importance, and thus the functional compatibility will depend on the particular weights given to the features.

While this approach presents some problems for the definition of category structures,[31] it is well suited for the definition of functional compatibility.

E. Weighted Conjunction[17,36,39]

For $i = 1,p$ and ob $\varepsilon\ \mathfrak{D}$, $\mu_{Gi}(\text{ob})$ is the compatibility degree of an object with the atomic pattern represented by Gi according to the criterion Ci.

Let ai be the weight attached to the criterion Ci, and then $\forall\ i = 1,p$, ai expresses a threshold whose role is to reduce the influence of the criterion Ci on the evaluation of the global compatibility.

We make the following assumptions:

a: $ai = 1 <=>$ the criterion Ci is "essential."

b: $ai = 0 <=>$ the criterion Ci is "negligible."

*The min is an idempotent operator which has the property to be sensitive only to ordering (Prade, personal communication).

and then the partial compatibility with the criterion Ci may be defined as the max $(1 - ai, \mu_{Gi}(\text{ob}))$.

Thus the global compatibility will be given by:

$\min_{i=1,p} (\max (1 - ai, \mu_{Gi}(\text{ob}))$

with $\max_{i=1,p} ai = 1$ meaning that at least one criterion is essential.

In terms of the possibility and necessity measures, we can define the global compatibility as follows:

$$\Pi(\text{GP/ob}) = \min_{i=1,p} (\max (1 - ai, \Pi(Gi/\text{ob}))$$

$$N(\text{GP/ob}) = \min_{i=1,p} (\max (1 - ai, N(Gi/\text{ob}))$$

For determining the compatibility of an object with an action, we submit that the features corresponding to the implement part, that is the object–properties required for the achievement of the goal of the action, must be weighted equally *to* 1. The weights attributed to the other features depend on their roles in the action, and how important their contribution is for attaining the goal of the action. For instance, the *handle,* as a specific part for pounding actions will facilitate the action but is not a necessary part for the goal, the *graspability* of the object is sufficient while the properties associated with the *hammerhead* are necessary (e.g., a "can–of–frozen–orange–juice" is more useful for pounding than a "plastic–knife" held by its handle).

F. A Formulation of the Global Compatibility

We will discuss the global compatibility in the simple case, in which the aggregation operation is not weighted.

The aggregation must respect the structure of the global pattern. This structure includes two types of links: the connection links between two parts and the type–part links. These different kinds of links imply three possible situations for the aggregation of two partial compatibilities. They are called the WITH, APART, and W/A situations.

a) The WITH situation occurs when two functional properties $F1$ and $F2$ have to satisfy the predicate: "The part which affords $F1$ must also afford $F2$ and conversely." For instance, an object for digging is required to have a part which affords simultaneously "cut" and "contain." This situation expresses a conjunction of objectives in the same type–part link and the corresponding operation of aggregation is a min under a labeled part (or a ponderate min).

Example:

$$F1 \text{ WITH } F2 \text{ <==> } (\textit{Parti}\text{: min } (P1, P2))$$

where Parti = part1 = part2

b) The APART situation occurs when two functional properties $F1$ and $F2$ have to satisfy the predicate: "Two different parts must afford respectively

$F1$ and $F2$." For instance, an object for cutting is required to have two distinct parts, the handle and the blade.

This situation expresses a conjunction of objectives in two different type-part links (with eventually a connection link between them) and the corresponding operation of aggregation is a min under two labeled parts (or a weighted min).

Example:

$F1$ APART $F2$ <==> min (part1: $P1$, part2: $P2$, connect: (Part1 Part2))

where "connect: (Part1 Part2)" expresses the compatibility between the connection in the object representation and the required connection. The evaluation of this compatibility is similar to the evaluation of the compatibility between two functional properties since they can be represented in the same way. For instance, the connection (relative–orientation orthogonal) can be represented by:

(connection (RELATIVE–ORIENTATION orthogonal))

Remark: The weight associated with a connection criterion in the weighting operation will always be less than *inferior to 1*.

c) The W/A situation occurs when two functional properties are not constrained in their location with respect to each other (the features can be either in the same part or in different parts). For instance, an object for drinking is required to be graspable but is not necessarily required to have a handle.

In this situation, the conjunction of objectives should be a simple min (or ponderate min). However, if $C1$ and $C2$ are found in two different parts, the connection between these two parts is taken into account. If $C1$ and $C2$ are found in the same part, the conjunction is min ($P1$, $P2$). This two cases form a redundancy which is expressed by a max.

Example:

$F1$ W/A $F2$ <==> max (parti: min ($P1$, $P2$),

min (part1: $P1$, part2: $P2$, connect: (Part1 Part2))

where Parti = part1 = part2

G. An Example of Global Compatibility Using the Weighted Operation Min

In the example, we consider for simplicity that the space attached to the requirement on graspability is the logical space {true, false}, then the functional property with "graspable" as attribute is reduced to: (graspable true) or (graspable false).

The space RELATIVE–ORIENTATION is noted: R–O

Let us consider the action "to screw" (Fig. 6)

—*object–requirements:*

1—(graspable true)

2—(screwing (SHAPE × LENGTH Flared × Short))

—*Type–part links:* 1 W/A 2
—*Connection:* If 1 APART 2 then:
3—(connection (R–O Extension))
—*Ponderation:* $a1 = 0.8$, $a2 = 1$, $a3 = 0.2$

SHAPE is a discrete space composed with the possible shapes of a tip of a screwdriver according to the different screws it is acting on.

SHAPE = {Flared, Phillips, Pozidriv, Reed, Clutch, Robertson, Torx} corresponding respectively to the following screws.

Figure 6

SHAPE being a discrete space for the attribute screwing, we have:

$$\forall\, x,y \text{ in SHAPE}^2,\ \Pi(x/y) = N(x/y) \quad \begin{aligned} &= 1 \text{ if } x = y \\ &= 0 \text{ else} \end{aligned}$$

The space LENGTH is a linear space defined on a subset of $|\Re$, $\{0, \max_{length}\}$ by:

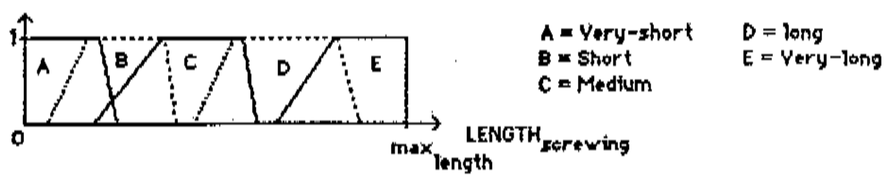

Figure 7

The space R–O is a linear space defined on [0, Π/2]. The values are given by the angle between the direction of the principal axis of each part. (Fig. 8)

R–O = {extension, oblique, orthogonal} such that,

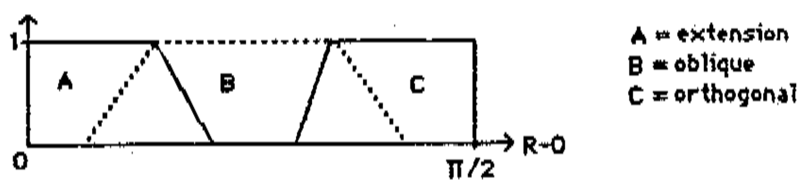

Figure 8

Let $\mathfrak{D}$ be a catalogue of objects and $\mathcal{N}$ be the set of objects in $\mathfrak{D}$ compatible with the global pattern associated with "to screw."

For an object ob in $\mathfrak{D}$, we note ob(*at*) the value of the attribute *at* for the object if it exists.
$\forall$ ob $\varepsilon\mathfrak{D}$,

Π("to screw"/ob)=max (parti:min (max (0.2, Π (true/ob(*graspability*))),
Π (flared × short/ob(*shape*) × ob(*length*)))),
min (partu: max (0.2, Π (true/ob(*graspability*))),
partv: Π (flared × short/ob(*shape*) × ob(*length*)),
max (0.8, Π (Extension/connection partu–partv)))

N("to screw"/ob)=max (parti:min (max (0.2, *N* (true/ob(*graspability*))),
N (flared × short/ob(*shape*) × ob(*length*)))),
min (partu: max (0.2, *N* (true/ob(*graspability*))),

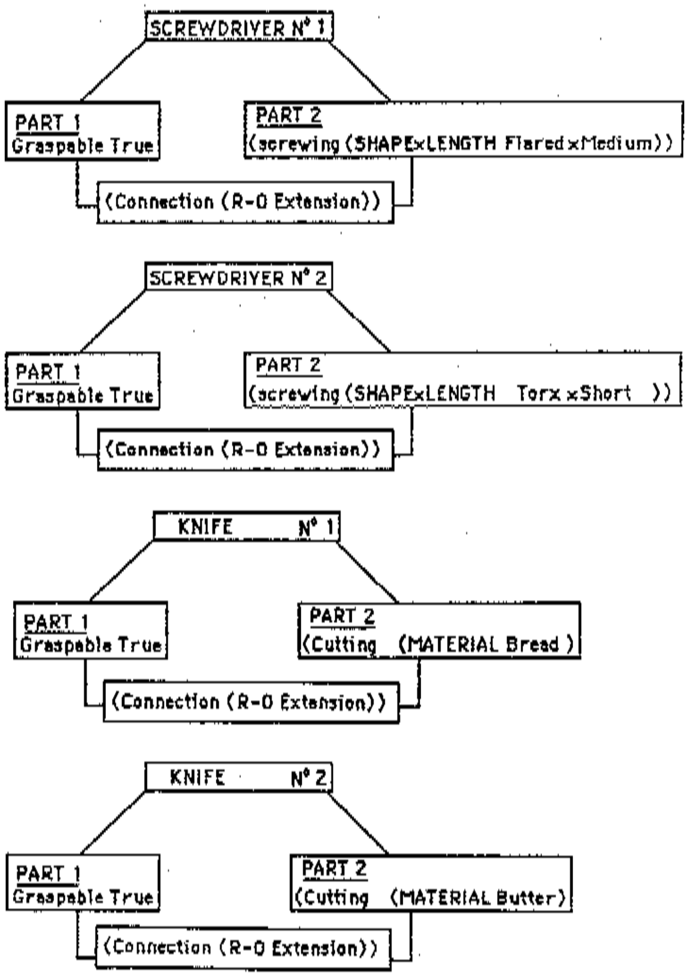

Figure 9

partv: N (flared × short/ob(*shape*) × ob(*length*)),
max (0.8, N (Extension/connection partu–partv)))

Now let us consider more precisely a catalogue composed with four objects: a *screwdriver with a flared tip* (screwdriver no. 1), a *Torx screwdriver* (screwdriver no. 2), a *knife with a pointed end* (knife no. 1) and a *knife with a flat end* (knife no. 2).

Figure 9 shows a coarse object–concept descriptions for these four objects. We voluntarily emphasized in these descriptions the organization which is required in the set of functional properties.

In particular, we have (for the possibility measure for instance):

—*For the screwdriver no. 1:*

Π("to screw"/screwdriver no. 1) = min (max (0.2, 1),
Π(flared × short/flared × medium),
max (0.8, Π(Extension/Extension)))
= min (1, Π(short/medium), 1)
= Π (short/medium)

—*For the screwdriver no. 2:*

Π("to screw"/screwdriver no. 1) = min (max (0.2, 1),
Π(flared × short/torx × short),
max (0.8, Π(Extension/Extension)))
= min (1,0,1)
= 0

—*For the knife no. 1:*

For the knife with a pointed end, the screwing feature is found in the object–structure representation on the edge of the blade and we assume:

Π(flared × short/ob(*shape*) × ob(*length*)) = 1

Thus, the connection between the two parts is transformed in:

(connection (R–O Orthogonal)) and Π (Extension/connection part1–part2) = 0

Π("to screw"/knife no. 1) = min (max (0.2, 1),
1
max (0.8, Π(Extension/Orthogonal)))
= min (1,1,0.8)
= 0.8

—*For the knife no. 2:*

Here, the screwing feature is found at the end of the blade and

Π(Extension/connection part1–part2) = 1

Π("to screw"/knife no. 2) = min (max (0.2, 1),
1
max (0.8, Π(Extension/Orthogonal)))
= min (1,1,1)
= 1

Remark: For the knife no. 2, the global compatibility would have been less than 1 if we would penalize the absence of the criterion screwing in the object–concept description.

VII. CONCLUSION

In this study, we have addressed the problem of recognition of object functions. We have proposed a conceptual model of compatibility between objects and use in hand-actions. The compatibility is computed by a pattern matching. Unlike the classical pattern matching procedures, our model does not assume that the structure of the datum is similar to the structure of the pattern prior to the effective matching. Thus, when required by the pattern, the information is indexed in the object–concept representation by object–parts. The relationship between the object structure and function depends on the level of description: conceptual or structural.

A system that can recognize functions would have many applications. One such application is the ability of an intelligent robot to carry out a task in an unconstrained environment especially as unattended manufacturing will advance.

This report describes research done at the Intelligent Systems Laboratory of the Boston University, College of Engineering. Support for the Laboratory's Artificial Intelligence Research is provided in part by a grant from the National Institute of Health NS06209 and the Giga Moss Systems Corporation. Marie-Christine Jaulent has been supported by a postdoctoral fellowship from the INRIA. We thank Henri Prade for valuable comments on an earlier draft of this article.

References

1. M.Ch Jaulent and L.M. Vaina, in preparation.
2. K. Koffka, *Principles of Gestalt Psychology*, Harcourt, Brace & World, New York, 1935.
3. J. J. Gibson, *The Ecological Approach to Vision Perception*, Houghton Mifflin, Co., 1979.
4. L.M. Vaina, "From shapes and movements to objects and actions: Design constraints on the representation," *Synthese* **54**, 3–36 (1983).
5. L.A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, **1**, 3–28 (1978).
6. E. Rosch, C.G. Mervis, W. Gray, D. Johnson, and P. Boyes-Braem, "Basic Objects in natural categories," *Cognitive Psychology*, **8**, 382–439 (1976).
7. E. Rosch, "On the internal structure of perceptual and semantic categories," In *Cognitive Development and the Acquisition of Language*, Academic Press, New York, 1973.
8. G.A. Miller, *Practical and Lexical Knowledge*, 1978, pp. 305–319.
9. B. Tversky, and K. Hemenway, "Objects, parts, and categories," *Journal of Experimental Psychology: General*, **113**, 169–193 (1984).
10. K. Nelson, "Some evidence for the cognitive primacy of categorization and its functional basis," *Merrill–Palmer Quarterly*, **19**, 21–39 (1973).
11. E.V. Clark, "What in a word? On the child's acquisition of semantics in his first language," In *Cognitive Development and the Acquisition of Language*, T.E. Moore (Ed.), Academic Press, New York, 1973.
12. K. Nelson, "Explorations in the development of a functional semantic system," In *Children's Language and Communication*, W.A. Collins (Ed.), The MInnesota Symposium of Child Psychology, Vol. 12, Lawrence Erlbaum Associates, Hillsdale, NJ, 1979.
13. E. Moerk, "Object, percept, concept, word," In *Sensory Experience, Adaptation, and Perception*, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1984.
14. A.L. Glass and K.J. Holyoak, *Cognition*, Random House, New York, 1986.
15. D. Marr, "Early processing of visual information," *Phil. Trans. R. Soc. Lond. B*, **275**, 483–524 (1976).

16. L. Rendell, "Conceptual knowledge acquisition in search," In *Computational Models of Learning*, L. Bolc (Ed.), Springer-Verlag, Berlin, 1987.
17. L.M. Vaina, "Semiotics of with," *Versus*, 96–112, Bompiani, Milano, Italy, 1978.
18. L.M. Vaina, "Towards a computational theory of semantic memory," In *Cognitive Constraints on Communication*, L. Vaina and J. Hintikka (Eds.), pp. 97–113, 1984.
19. A.M. Collins and M.R. Quillian "Retrieval time from semantic memory," *Journal of Verbal Learning and Verbal Behavior*, **8**, 240–247 (1969).
20. D. Marr and H.K. Nishihara, "Representation and recognition of the spatial organization of three-dimensional shapes," *Proc., R. Soc. Lond. B*, **200**, 269–294 (1978).
21. R. Nevatia and T.O. Binford, "Description and recognition of curved objects," *Artificial Intelligence*, **8**, (1) (1977).
22. A. Pentland, "Perceptual organization and the representation of natural form," *Artificial Intelligence*, **28**, (3) (1986).
23. R. Bajcsy and F. Solina, "Three dimensional object representation revisited," *Proceedings IEEE*, 1987.
24. I. Biederman, "Human image understanding: Recent research and a theory," *Computer Vision, Graphics and Image Processing*, **32**, 29–73 (1985).
25. L.M. Vaina and T. Zlateva, in preparation.
26. L.M. Vaina and C. Krey, in preparation.
27. J. Napier, "The prehensile movements of the hand," *Journal of Bone and Joint Surgery*, **38B**, 4: 902–913 (1956).
28. A. Tversky, "Features of similarity," *Psychological Review*, **84**, 327–352 (1977).
29. D.L. Medin, W.D. Wattenmaker, and S.E. Hampson, "Family ressemblance, conceptual cohesiveness and category construction," *Cognitive Psychology*, **19**, 242–279 (1987).
30. P.H. Winston, "Learning structural descriptions from examples," In *The Psychology of Computer Vision*, McGraw-Hill, New York, 1975.
31. R.M. Haralick and L.G. Shapiro, "The consistent labeling problem," *IEEE Trans., PAMI* **1,2** 173–184, April 1979.
32. L.G. Shapiro and R.M. Haralick, *Structural Description and Inenexact Matching*, Technical Report CS79011–R, Computer Science Dept., Virginia Polytechnic Institute, 1979.
33. J.R. Ullman, "An algorithm for subgraph isomorphism," *J. ACM*, **23/1**, 31–42 (1976).
34. D.A. Waterman and F. Hayes-Roth, "An overview of pattern-directed inference system," In *Pattern-Directed Inference Systems*, Waterman D.A. and F. Hayes-Roth (Eds.), Academic Press, New York, 1978, pp. xx–xx.
35. H. Farreny, *Un Système pour L'expression et la Résolution de Problèmes Orientés Vers le Contrôle de Robots*, Thèse d'état, Université Paul Sabatier, France, 1980.
36. D. Dubois and H. Prade, *Théorie des Possibilités: Applications à le Représentation des Connaissances en Informatique*, Masson, Paris, 1985.
37. M. Cayrol, H. Farreny, and H. Prade, "Fuzzy pattern matching," *Kybernetes* **11**, 137–155 (1982).
38. M.Ch Jaulent, *Un Système Souple pour Identifier des Objects Géométriques Plans Décrits par un Opérateur Humain*, Doctorat INP, Toulouse, France, 1986.
39. R.R. Yager, "Fuzzy decision making including unequal objectives," *Fuzzy Sets and Systems*, **1**, 85–95 (1978).