Reasoning About the Functionality of Tools and Physical Artifacts

Ergun Bicici and Robert St. Amant
Department of Computer Science
North Carolina State University
Raleigh, NC 27695

Abstract. Tool use is an important characteristic of intelligent human behavior. Representing, classifying and recognizing tools by their functionality can provide us new opportunities for understanding and eventually improving an agent's interaction with the physical world. Techniques have been developed in a wide range of areas within artificial intelligence to represent and automatically reason about the functionality of tools. This article surveys past approaches to reasoning about functionality in the literature and gives an extended example to illustrate the application of the techniques.

Keywords: Tool use, reasoning about function

1. Introduction

In popular thinking, tool use rivals natural language as the defining characteristic of intelligent behavior. Mazlish writes (1993):

> When humans first appear, they are already holding tools. Whatever the evolutionary steps leading to this development, our fossil remains are of human and tool together. Freed from pawing the ground, the released human hand can now hold a stone axe, that is, shaped stone, which obviously gives an adaptive edge. The first reason for tools, then, is that they are part of the process of natural selection, giving humans an advantage in their evolutionary struggle.

From an artificial intelligence viewpoint, gaining an understanding of tools has important implications. The development of habile (tool-using) agents has been identified by Nils Nilsson as one of the key challenges in the future of AI. Representing, classifying and recognizing tools by their functionality can provide us new opportunities for understanding and eventually improving an agent's interaction with the physical world. In *The Society of Mind*, Marvin Minsky describes "bridge definitions," as the best ideas that can bridge between two different worlds (Minsky, 1986). We believe that work on the recognition of tools for specific uses will lead to bridge definitions to facilitate researchers' efforts in bringing robots into the world of human interactions.

Defining tools as physical and functional objects is not as straightforward as it might seem. If our definition is too specific, we may need to include a large number of exceptions; if it is too general, we may end up including many things that we do not want. In the AI literature the example of a chair is often used to illustrate these problems. One purposeful definition for a chair is “something that you can sit on.” However, because you can sit on almost anything, this definition is too general, including such things as floor, food, and other people. At the same time, a structural definition such as “a chair has a sit-able structure that is held between a backing structure and a legged-support structure not much taller than the legs of a human” is too specific. It excludes physical objects we might like to include, such as overturned pails and appropriately shaped rocks. Sowa has described this difficulty in terms of an “egg-yolk theory of word meaning” (Sowa, 2000). The basic idea, related to prototype-based theories in cognitive modelling and linguistics (Lakoff, 1987), is that objects most central to a given concept will be found in the yolk of the egg, while objects that are less similar will be in the white part [Figure 1].

We find this a very interesting approach that can help us learn starting from the functional representations of sample objects in a domain. We can state the basic approach as follows, using Sowa’s example from the domain of chairs:

I– Start with a set of example objects (e.g. chairs).

II– Based on some level of granularity, find their common characteristics and apply these to other objects to find similar ones with the same functionalities.

III– If you run out of objects, search deeper into the objects space by ignoring one or more defining characteristics.

IV– Filter these newly found objects according to the updated functionalities and update all objects accordingly until saturation. Total number of objects may decrease or increase.

The goals or purpose of the learner are important in answering questions about the functional similarities of objects. To continue with the example of a chair, we might ask the learner, which objects are most like a chair? We are implicitly assuming that the functionality of the chair is uppermost in the mind of the learner. However, if the goal is different (perhaps we want a chair that can be used to wedge a door closed), then the learner may not be able to make the relevant distinctions because the yolk of the egg is partly defined by the goals of the learner.

Similarity, however, is difficult to characterize precisely. In the example above, identifying an object as chair can depend on the goals of the observer, the visual similarity of the object to other chairs, the ability of the observer to generalize to past experience with chairs, and so forth.

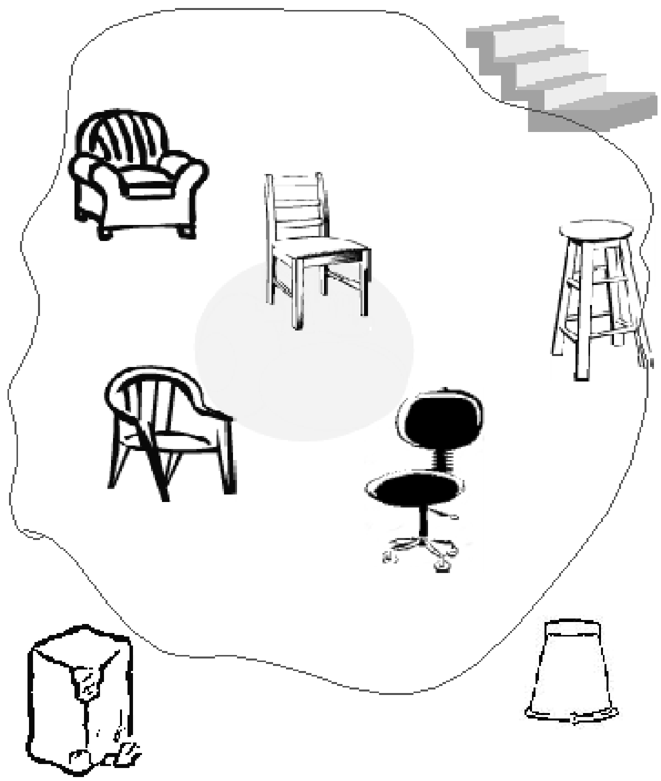

Figure 1. Egg-yolk theory of the meaning of chair

As a starting point in organizing the potentially vast amount of information that might be brought to bear on the interpretation of physical objects as tools, we turn to a definition from Beck (Beck, 1980): “Thus tool use is the external employment of an unattached environmental object to alter more efficiently the form, position or condition of another object, another organism, or the user itself when the user holds or carries the tool during or just prior to use and is responsible for the proper and effective orientation of the tool.” This definition comes from the literature on non-human primate tool use but reflects human tool use as well. Definitions such as this raise a number of important issues for artificial intelligence research; tool use and, generally speaking, reasoning about the functionalities of physical artifacts depends on the following factors and senses:

– *Shape:* For many tools, shape is a decisive factor in their effectiveness. For example, screwdrivers are often sold in sets, in which individual tools vary in length, thickness, and the shape of the driver head. Phillips head or slotted screws much be matched by screwdrivers with particular shapes.

- *Planning:* Appropriate sequences of actions are key to tool use. The function of a tool usually makes it obvious what kinds of plans it takes part in. For example, a mechanic needs to choose the right tool for the job and plan ahead which tools he will need and which tools he can use in the absence of some others. Also, proper usage of tools often involves appropriate application of forces in suitable amounts.

- *Physics:* For reasoning about a tool's interactions with other objects and measuring how it affects other physical artifacts, we need to have a basic understanding of the naive physical rules that govern the objects. We can classify many tools according to the principles of leverage and lever types. For instance, hammer claws function as a type-1 lever, where the pivot (fulcrum) is between the effort and the load and therefore the direction of the force changes.

- *Dynamics:* The motion and the dynamic relationships between the parts of tools and between the tools and their targets provide cues for proper usage. In the case of a hammer, for effective use, we need to swing it with a proper angle and velocity towards the target. By building systems that observe and learn from this type of experience, we can find proper and effective usage of physical artifacts.

- *Causality:* Causal relationships between the parts of tools and their corresponding effects on other physical objects help us understand how we can use them and why they are efficient. For example, in the case of a hammer, it has a graspable portion and a striking surface that, when used, may cause a distortion on the objects that it hits. We think that if the striking surface stays intact after the hit and can be "re-usable" afterwards, then we can use that tool as a hammer.

- *Work space environment:* A tool needs enough work space to be effectively applied. A hammer needs swinging room, a screwdriver needs space for twisting. Finding enough room for a particular tool is closely related to spatial planning and reasoning.

- *Design requirements:* Using a tool to achieve a known task requires close interaction with the general design goal and requirements of the specific task. For example, if we want to nail a carpet to the floor, we may use a hammer, but we might instead design a system that involves three hammers which enable us to nail in one third of the time.

Table I. Approaches to understanding tool use by functionality

Non-interactive approaches: These models do not interact with the objects to realize their functionalities. Most are applications in computer vision.
Functionality = Shape (Solina and Bajcsy, 1986; Vaina and Jaulent, 1991; Zlateva and Vaina, 1992; Rivlin et al., 1995; Kim and Nevatia, 1998; Li and Lee, 2002)
Functionality = Shape + Causality (Winston, 1975)
Functionality = Shape + Planning (DiManzo et al., 1989)
Functionality = Shape + Dynamics (Duric et al., 1996)
Functionality = Shape + Physics + Causality (Brady et al., 1985; Connell and Brady, 1987; Hodges, 1992; Hodges, 1995; Brand, 1997)
Functionality = Common sense theories (Hayes, 1978; Davis, 1990; Davis, 1993; Davis, 1998)
Interactive approaches: These models interact with the objects to realize their functionalities. Most are applications in robotics.
Functionality = Shape (Allen, 1990; Stansfield, 1992)
Functionality = Shape + Work Space (Wilson, 1996a)
Functionality = Shape + Physics (Far, 1992; Bogoni and Bajcsy, 1993; Krotkov, 1994; Green et al., 1994; Bogoni, 1995; Stark and Bowyer, 1996)
Functionality = Shape + Physics + Causality (Cooper et al., 1995)
Abstract approaches: These approaches try to model the functionality of objects in general terms, at a level of abstraction above manipulation and perception.
Reasoning about design requirements (Freeman and Newell, 1971)
Ecological reasoning (St. Amant, 2002; Amant and Horton, 2002)
Common sense reasoning (Lenat, 1996)

– *Common sense:* A good understanding of physical objects needs commonsense knowledge about how to use them and how to match tools with objects that are available in the environment.

This list suggests that reasoning about the functionality of tools, as well as recognizing and using tools according to their functionalities, requires a cross-disciplinary investigation ranging from recognition techniques used in computer vision and robotics to reasoning, representation, and learning methods in artificial intelligence.

We can structure previous work on approaches relevant to tool use and reasoning about functionality into two main categories: systems that interact with tools and environments, and systems that do not. We further subdivide these categories according to the dimensions of functionality they consider and the complexity of the techniques they use. Table 1 summarizes combinations that have appeared in the AI literature. Some of these methods do not necessarily aim to recognize the functionality of an object; sometimes their sole aim is to recognize objects according to their functionalities or recognize functionalities according to the objects.

Over the years, reasoning about functionality has attracted attention in many disciplines, including (but not limited to) robotics, computer vision, psychology, and artificial intelligence; with work originating in image recognition and understanding, (spatial) reasoning, representation and learning. In the remainder of this article we review a number of past approaches to reasoning about functionality and to intelligent use of physical tools from the literature following the basic organization given in Table 1. We examine these approaches in categories of interactive and non-interactive systems and later group them according to the dimensions of functionality and the complexity of the techniques they credit in increasing the sophistication of their modelling capability. Also, we discuss the difficulties that emerge and the issues that need to be addressed.

We end with the application of a selection of these techniques to a few representative examples of tools en route to building a tool-using robot-arm. We believe that work in the recognition of tools for specific uses will lead to bridge definitions that will enable researchers to bring robots into the world of human interactions.

2. Non-interactive approaches

Many approaches to tool use, mainly those in the field of computer vision, do not interact with objects and are limited to the non-contact perceptions to realize the functionalities of objects. This vastly constrains the experiments that can be done with them, since they are only observers that cannot have any effect on the environment. Krotkov (1994) describes methods that are limited to non-contact perception as superficial, in that they are sensitive only to the surface of the object. Since they cannot directly measure properties like density or friction, they are also indeterminate. Nevertheless there is strong intuitive appeal to a non-interactive approach; experienced tool users can often recognize the capabilities of a tool simply by inspection.

2.1. FUNCTIONALITY = SHAPE

Models in this category use only the shape of an object to recognize its functionality, with the idea that the shape of an object specifies its functionality. For example, a hammer can be defined as a T-shaped object with geometric constraints like the (surface normal of the) head is nearly perpendicular to the (surface normal of the) handle, and the handle is positioned near the center of the head.

Solina and Bajcsy (Solina and Bajcsy, 1986) represent generic objects by parts, which are modelled by super-quadric volumetric primitives. Parts are prototypes in that changes in structure and deformation in the shape of objects are allowed. Each part has a set of features that are used for selecting models from a model database. The selected models are then matched with the part data geometrically. The recognition process deforms each part of the models to match the corresponding object part and selects the model that achieves the best match. This system relies on the assumption that the shapes of basic object parts correspond to the function of the artifact.

Vaina and Jaulent (1991) recognize function by using shape and concept representations, object categories, and requirements of actions. They propose a conceptual model of compatibility between objects and their usage in hand actions, based on pattern matching. The level of conceptual or structural description determines the relationship between the object structure and function.

Zlateva and Vaina (1991) provide mathematical support for the formalization and computation of the shape structure and its representation for deriving the possible functions of objects. They discuss axis- and boundary-based methods for defining the parts and subparts of objects. Their method of describing functionality is based on a theorem from differential geometry, which claims that any regular surface can be approximated in a finite environment to some given accuracy by a paraboloid. Based on this observation, they represent convex parts of objects using polyhedra, cylinders, ellipsoids, and generalized cones.

Zlateva and Vaina attach example functionality-to-structure feature mappings by using the decomposition of the object into largest locally convex surface patches (LCP). For instance, the functionality of stability and support needs to have at least three points that define a sufficiently planar surface that includes the projection of the center of gravity. The functionality of an action capability such as “can pound” is recognized by a structure that has an accessible part with a sufficiently flat surface patch; “can be rolled” requires that the shape representation at the highest level is cylindrical.

The LCP method applies to 3D objects such as differently shaped wrenches, different types of screws and bolts, and various hacksaws (Zlateva and Vaina, 1992). They note that the decomposed parts relate to specific affordances of the object (see Section 4.2 for further discussion of this concept): a handle to hold, an opening to grasp the bolt, a head to provide support for the case of a wrench. They also claim that in order to know the use of an object, we need to infer the proper position of the hands, the direction of the action, and the pressure to be applied. These cannot be learned without spatial relations between parts and subparts, which implies that the parts and subparts directly relate to affordances of an object.

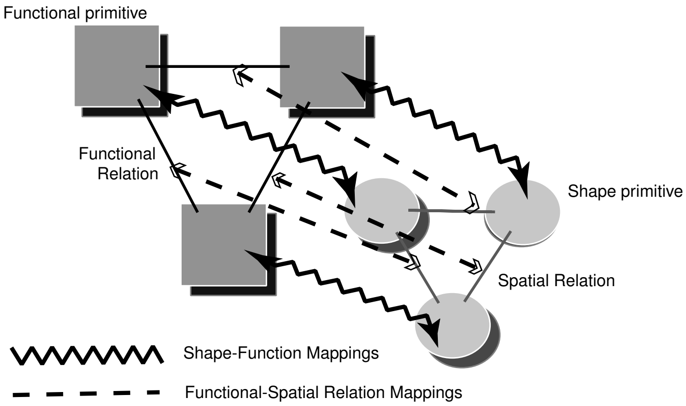

Figure 2. Object representation according to functionality

Rivlin, Dickinson, and Rosenfeld (Rivlin et al., 1995) extend "recognition by parts" shape recognition framework to "recognition by functional parts" by matching functional primitives and their relations with volumetric shape primitives and their relations. They aim to offer an object representation that integrates function and shape, and address the problem of recovering shape and function data from either 2D or 3D images. The representation of object functionality and the matching scheme between two layers of primitives (functional and shape) can be seen in figure 2.

In the shape layer, objects are constructed by using volumetric primitives with spatial relations between them. In the functional layer, objects are represented in terms of functional primitives and relations. The shape primitives are mapped to a set of functional primitives and the spatial relations are mapped to a set of functional relations.

The shape representation Rivlin et al. used models objects using four classes of volumetric shapes: sticks, strips, plates, and blobs. Their relative dimensions distinguish these from each other. If a1, a2, and a3 represent length, width and height respectively, these four classes can be defined as follows:

Stick: $a_1 \simeq a_2 < a_3 \vee a_1 \simeq a_3 < a_2 \vee a_2 \simeq a_3 < a_2$
Strip: $a_1 \neq a_2 \wedge a_2 \neq a_3 \wedge a_1 \neq a_3$
Plate: $a_1 \simeq a_2 > a_3 \vee a_1 \simeq a_3 > a_2 \vee a_2 \simeq a_3 > a_2$
Blob: $a_1 \simeq a_2 \simeq a_3$

The functional representation assumes a set of pre-defined functional primitives such as an end-effector and a handle in the case of a manipulation task and a particular way that these primitives should be joined together.

Although there may be many shape primitives matching a functional primitive (a many-to-one relationship) as in the case of chair legs to chair base, for simplicity, this approach is restricted to object models with one-to-one mappings. Also, by modelling objects by super-quadrics that support the recovery of occluded parts, the approach supports reasoning about the functionality of objects that are only partially visible.

The function-based object recognition procedure supports both bottom-up and top-down recognition. In top-down fashion, the system looks for a given object by mapping its functional parts to the image, whereas in the bottom-up approach, the system recognizes the object according to the given functional parts. This means that when working bottom-up, the object recognized can be unexpected or unknown beforehand but while working top-down, we know in advance what kind of object we are trying to recognize.

In Rivlin et al.'s approach, functionality is defined only in terms of the object's coarse volumetric parts found through region segmentation. Although they claim that segmentation gives them the granularity needed for focusing on local object features, this creates under-segmentation or over-segmentation problems and it relies on opaque object surface textures. Even after realizing that the relation between function and structure is many-to-one, [1] limiting a system to one-to-one matching seems inadequate.

Kim and Nevatia (Kim and Nevatia, 1998) conduct generic object recognition experiments of desks and doors on real scenes for robot navigation. The recognition of significant surfaces was achieved by using the orientation, range of heights, shape, and size of edges in a real intensity scene image. Their functional representations characterize objects by their significant surfaces and name the objects that help the system observe the functional role of another object as "functional evidence." For example, the functional evidence of a door consists of objects that are seen through when it is open. Algorithms for detecting a door frame and the legs of a desk are also given based on the assumptions that surfaces are planar and objects are in a standard pose.

Li and Lee (Li and Lee, 2002) use accumulative Hopfield matching (AHM) in automatic object recognition and learning for articulated object models based on a small number of images. They accept model-based object recognition as the most effective method for rigid objects but note that if the object is articulated [2], its appearance may change for different perspectives. They claim that recognition methods based on difference between the actual image and the model encounter problems for articulated images since the structure changes with changing viewpoints.

Li et al. use many-to-one (homomorphic) attributed relational graph matching for recognizing both the shape and the structure of objects in images. The angle at the breakpoints of sub-images and the distance between breakpoints

[1] Actually, the relationship is many-to-many since many functions can map to different structures as well; for example, hammering functionality can map to the structure of a hammer, a screwdriver, or even a shoe.

[2] Object consists of rigid components

are used as features for the attributed relational graph representation. Their method randomly partitions the input image into many sub-images where Hopfield networks are used to derive the isomorphism mappings between sub-images and models. These results are later accumulated by further iteration until a stable matching is reached. Li et. al. experimented with various hand tools and keys and were able to find both the objects and the poses they appear in the images. They claim that their method is extendable to 3D images as well. Li et. al.'s technique recognizes isolated, recurring, or occluded images invariant to translation, rotation, scale, or distortion.

2.2. FUNCTIONALITY = SHAPE + CAUSALITY

These models use only the shape of an object as input but rely on causal relationships and learn these relationships to develop a model of functionality. An early example is Winston's work (1975) on structural concept learning in the blocks-world domain. To construct representations of the definitions of concepts in the blocks world, Winston used semantic nets. It was one of the first systems that learns a concept from examples, learns by imitation, and learns by being told.

In this learning process, the system starts with a structural description of one known instance of the concept, calling it the concept definition. Through the learning process, this initial definition is amplified according to positive and negative examples encountered. This definition is thus called the evolving model. It is generalized by including descriptions of other instances of the concept and specialized by excluding descriptions of near misses or negative examples. A near miss is an example which is very similar to instances of the concept but in fact it is not an instance.

The ANALOGY program (Evans, 1968) of Winston et al. learned the relation between form and function by using semantic nets (Winston et al., 1983). He also used Brook's object modelling system based on generalized cylinders, ACRONYM (Brooks, 1981), for physical representations. The goal was to use functional definitions to identify physical properties and provide an example system that can learn physical models using these functional definitions (Winston et al., 1983).

The recognition process of this system involved different steps. At first, the object is described in functional terms, which is translated into semantic net links. For example, a cup's functional description is something like "a cup is a kind of object and open-vessel and it is stable and lift-able." Semantic knowledge of "stable", "lift-able", and "open-vessel" are then linked to a cup through causal links. Then the system is given a physical description of an input object in English, which is sent to the ACRONYM system for generating the physical model based on generalized cylinders. This model is later extended with the addition of material properties such as weight and joint locations. These additional data are physical properties that are impossible to obtain from a vision system. The system then tries to show that the functional requirements are still met by the enhanced physical description and identify the object. These functionally recognized objects' physical models are later learned in the form of if-then rules. Once these if-then based physical models are learned, the system does not need functional requirements for recognizing any new examples of the concept.

2.3. Functionality = Shape + Planning

DiManzo et al. (1989) regard functional reasoning as the ability to integrate shape and function with the help of planning. They describe the difficulty of separating the function of a tool from the plan it takes part in, since plans and tools evolve together and differentiate with time. Their reasoning system is based on a hierarchy of levels that interact with each other. At the top level, they have a task and plan representation that uses semantic functional descriptors (SFD) and functional experts (FE) for planning based on functionality of objects. The object representation level uses FE's and geometric primitives to describe objects. The next level carries out function modelling by describing some basic functions in terms of geometric primitives, and the last level performs geometric reasoning by defining geometric constraints.

2.4. Functionality = Shape + Dynamics

These models use the shape, kinematic and dynamic properties of an object (e.g. motion) to recognize its functionality while the system observes the action that is performed with the object.

Duric, Fayman, Rivlin (1996) attempt to derive the function of an object from its motion given a sequence of images of a known object performing some action. The motion analysis results in several motion primitives and these are compared with previously known motion-to-function mappings. They use both the motion and shape of an object because many objects display similar motion characteristics in their use.

They constrain the many-to-many mappings between function and form with the help of motion. Optical flow measurements are used to derive motion information for different objects. The relevant motion is in object's coordinate system and its relation to the object it acts on (the actee). This relation is important for establishing the mapping and creating a frame of reference. Thus, the motion is derived independently of the place of action; whether bread is cut on a table or on a wall, for example, does not affect the motion. Duric et al.'s experiments deal with three cutting actions: jabbing, stabbing, and chopping. They also consider two different functionalities of the same object: scooping and hitting with a shovel and hammering and tightening with a wrench.

Duric et al.'s approach gives a promising path for learning through observing the motion of objects. A robot capable of seeing and reasoning about the function of an object serving in an action can later recognize and apply other tools that can handle the same function better than the observed one.

2.5. FUNCTIONALITY = SHAPE + PHYSICS + CAUSALITY

Approaches in this category are some of the most comprehensive in attempting to model the functionality of tools. They incorporate all of the factors discussed up to this point: shape, physics, and causality.

Brady et al.'s system (Brady et al., 1985), "Mechanic's Mate", is intended to assist a handyman in generic construction and assembly work and to reason about tools. They investigate the interaction among planning and reasoning, geometric representation of the shapes, and qualitative and quantitative representations of the dynamics in the tool world. According to them, robots need detailed geometric models while dealing with the real world, so understanding of geometry needs to be connected with the understanding of naive physics of forces and causation. Also, by focusing on a higher order geometrical representation and their functional interpretation, they obtain a computationally more tractable system.

One of the planning tasks a mechanic needs is *choosing the right tool for the job*. The generic concept of a tool and functional and geometric variations helps us distinguish one tool from another. If we want to drive tacks into soft wood and if we have only a sledgehammer, then we might search for another object with a flat section that can be used as a striking surface like the handle of a screwdriver. This is very similar to finding the optimal solution for a task with the given functionalities of objects in the environment. If we cannot find the optimal tool for the task, we pick the second best tool that can handle the same job.

Changing the direction of forces, torques, and impulses (lever and fulcrum, pulley, cam) and devising plans to transmit forces between parts (links, gears, lead screws) are two main problems that arise in Mechanic's Mate. To solve these, Brady et al. give the general description of sample tools and try to apply them to the problem of peg-out-of-hole. They later give some of the naive structural regularities of objects' shape in the physical world and give some generic knowledge about their usage such as "a saw blade is moved in the direction of its edge." With these heuristics, they also identify ways to use these tools properly or broaden their applicability.

Connell and Brady's system (Connell and Brady, 1987) learns shape models from two-dimensional objects by using a substantially modified version of the ANALOGY program (Evans, 1968) that Winston et al. used (1983). ANALOGY learns the relation between form and function by using semantic nets that learned the generalized structural description from a sequence of positive examples by using 2D images. The system uses the technique of ablation and learned concepts from disjunctions. Their primary motivation is to understand the connection between planning and reasoning about tools and the representations of the objects' shapes or, in other words, the relation between form and function.

Connell and Brady try to find innovative solutions to construction problems by using tools that were designed for other purposes in a novel way. Instead of learning that a specific geometric structure is a hammer, their system infers that something with a graspable portion and a striking surface can be used as a hammer. They define these two functional concepts geometrically in terms of shape descriptions. Such as, a *graspable portion* is something that has a spine that is straight and elongated, and sides that are only slightly curved and a *striking surface* is an end of a sub-shape that is blunt and that is parallel to the spine of the handle. Connell and Brady taught the functionality of a hammer by defining the grasping and striking requirements accordingly, and then showing it examples of graspable objects that has striking surfaces. The program is able to improvise by taking advantage of having a functional description of a hammer (*functional improvisation*). Thus, given a hammering task without a hammer, they were able to match the functional description of a hammer to any other available tool. A close match to the geometric form of another tool implies that it can be used as a hammer by grasping the handle matched to the graspable portion and striking the matched striking surface. Also, Connell and Brady admit that with the structural recognition system they have, the descriptions of even simple shapes typically comprise between fifty and three hundred assertions (Connell and Brady, 1987). In the example given, they represent a tack hammer with 51 associative triples.

Hodges developed EDISON system (Hodges, 1992) in an effort to imitate the human device-using process: match context and object applicability, experiment to see if the object will work, recognize behavior through these experiments, and use experience to predict the function and behavior of new objects. The system' s goal was to represent and manipulate problem-solving situations that require mechanical device use by applying behavioral, functional, and intentional reasoning. EDISON supported mechanical improvisation by applying the notion of functional equivalence from mechanical primitives (MP) of devices in different situations.

Later, he explored the relationship between the physical properties of an object, its functional representation, and its use in problem solving with his Functional Ontology for Naive Mechanics (FONM) model (Hodges, 1995). FONM representation theory identifies causal relationships between device structure, behavior, function, and use with its interdependent abstraction layers. Device statics representation describes the device at rest with states (geometric, material, and kinematic properties), regions (object shape, size, and location), relationships, and processes. Device dynamics explain what would happen when the device is perturbed with behavioral primitives (motion, restrain, transform, store, and deform), and device pragmatics layer describes how and why the device is used with device use plans. Hodges claimed that using MP-equivalence and appropriate contextual knowledge might solve the problem of mapping attributes to function with the vision research on object recognition.

Brand (Brand, 1997) built a system using causal and functional knowledge to see, understand, and manipulate scenes. Understanding a scene's causal physics demonstrates how scene elements interact and respond to forces and shows the scene's potential for action. Brand asserts that systems that use inferences based on connectivity and free space to model a scene's causal structure display desirable properties such as intelligent control of the focus of attention and understanding of the scene's potential for action and manipulation.

2.6. Functionality = Common sense theories

These systems try to model physical objects and their functionalities by using the common sense knowledge of shape, physics, and causality together with naive physical information. With naive physics, we mean the formalization effort that Pat Hayes's naive physics manifesto (Hayes, 1978) anticipated and the efforts towards implementing physical reasoning at the common sense level. This type of naive, commonsense physical knowledge that Hayes talks about is needed to build practical systems that are able to reason and interact with the everyday world around them. Also, DiManzo et al. (1989) mention that the relation between shape and function is dependent on the dynamic representation of the world, which can be given in terms of naive physics models.

Davis has done considerable amount of work (Davis, 1990; Davis, 1993; Davis, 1998; Davis, 2000) towards formalizing the physical world of objects through commonsense naive physical knowledge and has asked an instance of daily physical reasoning problems that led to solutions (Lifschitz, 1998; Morgenstern, 2001; Shanahan, 1998) for his famous problem of egg-cracking.

One of Davis's efforts deals with formalizing the kinematics of cutting solid objects (Davis, 1993). He shows the geometric aspects of various cutting operations: slicing an object in half, cutting a notch into an object, stabbing a hole through an object, and carving away the surface of an object. He also gives a list of geometric relations between the shapes and motions of the blades and targets. For example, he suggests that a blade needs to be sufficiently thin and hard but he does not discuss its elasticity or sharpness. In one representation, Davis (Davis, 1993) views the object as gradually changing its shape until it is split; when the original object no longer exists and two (or more) new objects form. The alternative representation focuses on chunks of material of the overall object. Until a piece from it is cut away, a chunk exists and preserves its shape. Davis also shows that these two theories are sufficient to support some simple commonsense inferences and algorithms.

Davis (Davis, 1998) claims that understanding the relation between the shape of an object and its functions through physical reasoning depends on spatial knowledge and spatial reasoning, which is difficult to express. For example, even if we know the shape of a screw and understand the relation between its shape and its functions, it is not easy to describe or explain these without using a technical vocabulary that is incomprehensible for most people.

As Davis suggests (Davis, 2000), real-time correct reasoning about physical systems is most of the time unnecessary because physical objects go through a series of unimportant mode transitions. He gives predicting the exact behavior of a rigid block falling down from a table as an instance. Instead, he proposes a commonsense reasoner that is concise and close to the mode of transitions in between physical states.

Davis's work pulls together many of the separate ideas in the systems discussed above in an attempt to impose a useful conceptual framework on work in this area.

3. Interactive approaches

Non-interactive methods are very helpful in recognizing candidate objects and disambiguating others. However, the resulting representations are not entirely trustworthy, since the proper usage of an object is usually highly dependent on interaction. The models described in this section, most of which are applications in robotics, interact with objects to recognize their functionalities. Haptic exploration, grasp planning, and physical perception through observing changes in objects that are physically distorted are some of the techniques used in this area.

3.1. FUNCTIONALITY = SHAPE

These models use only the shape of an object to recognize its functionality. The shape of an object can be represented in different ways using different knowledge such as the geometry of the object and the spatial data about it.

Allen's work (Allen, 1990) tries to determine the attributes of 3D objects, especially shape, through haptic exploratory procedures (EPs). He built an intelligent robotic system that can recognize shape from touch sensing and supported it with a vision algorithm for autonomous shape recovery. The system uses previously found EPs that can reach a success rate of 96-99%

in identifying object properties using haptic exploration. Allen used grasping by containment, lateral extent and contour follower perception techniques to obtain super-quadric surface representation, face-edge-vertex model and generalized cylinders of objects correspondingly. He interpreted each representation acquired from EPs as a constraint system that can be used to understand the input scenes. Allen identifies the usage of multiple representations for shape as a key component of any working system.

Stansfield (Stansfield, 1992) presented a model and an implementation of a robotic haptic system based on human haptic exploration and information processing. They used the exploratory procedures (EP) that were studied in previous psychological studies of human haptics such as using pressure to grasp hardness, static contact for perceiving temperature, and unsupported holding for measuring the weight. Furthermore, the robot contained structured-lighting vision and an expert reasoning system performing object categorization and grasp generation. The interaction and manipulation procedures added to their robotic system enhanced the perception capabilities of a robot.

3.2. FUNCTIONALITY = SHAPE + WORK SPACE

These models use the shape of an object with the workspace it requires for working properly to recognize its functionality.

To apply a given tool, Wilson (1996a; 1996b) measured geometric accessibility constraints in the placement volume relative to the other objects where the tool operates. He found out that determining whether a tool can be applied in a given assembly state is an instance of the FINDSPACE [3] problem (Lozano-Pérez, 1983). This spatial planning problem can be more formally defined as:

> Determine where an object A can be placed, inside some specified region R, so that it does not collide with any of the objects B_j already placed there.

For an object that is represented as a single point in configuration space, the configurations forbidden to it due to other objects can be specified as regions, which are called configuration space obstacles (Lozano-Pérez, 1983).

Use volume is the minimum free space needed for a subassembly to apply the tool and placement constraints determine where the volume needs to be placed relative to the reference point, which is at the position of required tool use. The placement of the use volume according to the placement constraints is an instance of the FINDSPACE problem (Lozano-Pérez, 1983).

Through this work, Wilson (Wilson, 1996a) tries to answer questions of the form, "Is there space for this tool to be used?" He also mentions that

[3] Wilson names this problem as the FINDPLACE problem.

in a real-world usage of a tool, there will be more issues that needs to be addressed, such as finding the space required for a human or robot arm to grasp the tool, choosing the best tool among feasible ones, finding an optimal tool-level plan, designing new tools, and dealing with changes that might allow a tool to be used.

3.3. Functionality = Shape + Physics

These models use the shape of an object plus the rules of physics that govern their interactions with each other and the environment to recognize its functionality.

Krotkov (1994) tries to perceive material properties by actively contacting and probing them and later sensing the resulting forces, displacements, sounds. This kind of perception ability is essential for a robotic system to understand not only where the objects are and how they look like but also what they are made of.

The senses of a robot are divided into two groups: non-contact and contact based sensing. Krotkov claims that although there are many non-contact sensing methods available (such as surface luminance for finding coefficient of friction or using thermal images for estimating the granularity of objects), determining the material composition of an object in a reliable way requires contact with it. Similarly, humans practice this physical exploration by pressing on, poking, tapping on, hefting, squeezing, shaking, rubbing or striking on the objects.

Krotkov observes that non-contact methods are superficial because they are sensitive only to the surface of the object and indeterminate since they cannot directly measure properties like density or friction. He extended the acquisition of material properties by procedures like “whack and watch”, “step and feel” and “hit and listen.”

According to Krotkov, perception of material properties will benefit reasoning about object functionality and also other potential applications. With the material properties that can be added to the reasoning process, we can recognize that a hard-heeled shoe could be used as a hammer.

Stark and Bowyer’s GRUFF (Stark and Bowyer, 1996) is a function-based object recognition system that recognizes objects by classifying them into categories that describe the functionality they might serve. It stands for “Generic Representation Using Form and Function” and uses boundary surface descriptions to derive previously defined knowledge primitives such as relative orientation, dimension, stability, proximity, clearance, and enclosure.

The system is based on computer vision techniques for recognizing functionality, and tries to achieve interactive recognition ability by observing the deformations that happen on objects. In the last section of their book (Stark and Bowyer, 1996), Stark and Bowyer demonstrate how to acquire physical and shape properties by analyzing the simulations of object interaction using an object dynamics modelling system named ThingWorld (Pentland et al., 1990). The interaction was achieved by observing the deformation of objects made of rubber or oak from chair category while forces were applied. In a later work (Stark et al., 1996), they give the sequence of steps involved in function verification through planned interactions. First, they change the orientation of the object and check for stability. If the object passes this test, they apply force and then test again. Further tests are done by applying force or changing the orientation and checking for deformation afterwards until the association measure for the object shape stabilizes below (object failed the test) or above (functionality is verified) a threshold.

The system uses ThingWorld (Pentland et al., 1990) to model the dynamics of the objects and to generate planned interactions to verify the suggested functionality of objects. Function-based recognition is used to recognize object categories and their functional requirements. This provides both a high level abstraction for representation and an association of function to the structure.

GRUFF's knowledge primitives are based on geometrical, causal and physical constraints such as a chair should be able to maintain functional orientation after being seated. To acquire these properties, they use simple operators such as apply force and observe deformations, which results in a *pseudo-interactive* system. However, these type of physical constraints exclude chairs that change their shape whenever they are seated such as a beanbag chair. But still, this simple operator can provide as an example of how we need to recognize the functionalities of physical objects in their physical world.

Functional properties are defined in terms of knowledge primitives. For the functionality of "provides X handle", testing the dimensions and the clearance near the object is needed. Green et al. take a comparable approach (Green et al., 1994), in which kinematic properties are investigated where the corresponding functional representation for scissors and chairs is given.

Bogoni (Bogoni, 1995) adds contextual information to the previous efforts. He defines functionality as the application of an object in a specific context for the accomplishment of a particular purpose. Thus, he considers the modality of the operation, which is reflected by the task description and context of application. The modality is the result of using different sensors for the recovery of material and functional properties, where uncertainty and noise can be added from sensors. In his work, models try to reason bottom-up by acquiring the properties of the objects that are investigated and by extracting the functional relations between parts. This decreases the need to make assumptions about object properties. Also, by focusing on the acquisition of basic properties from analyzing functionality, Bogoni aims to create a repertoire of primitive functional procedures.

Bogoni and Bajcsy (1993) implemented a robotic system that recovers shape and material properties and observes the interactions, to establish the functionality of a tool. In the system, there is a compliant wrist that explores tools based on their features. The description of the task is formalized using a discrete event system (Bogoni and Bajcsy, 1994). There are two sensors used: force and end-effector position sensor. Later Bogoni and Bajcsy (1995) introduce a formalization of a representation for functionality that is recovered through classes of force profiles identifying the dynamics of the interaction. They did not use the shape of the object itself for the recognition of the object prior to interaction. They investigated manipulatory interactions that emphasize the verification and recovery of the material properties of an object, using exploration techniques. One of those interactions, piercing, was tested to reveal if the object is capable of piercing.

They claim that generality of the functionality is dependent on the properties assumed. Therefore, inclusion of various properties in object representation both benefits the acquisition of properties and addresses the aspects of functional recognition and representation. Although their approach is limited to tools employed in simple manipulatory interactions, they are able to extend the functionality research by (1) using different sensor modalities for the acquisition of properties, (2) incorporating various material properties as part of the representation, (3) using interaction for verifying, acquiring, and describing the functionality of an object, and (4) extracting functional features for future interactions and functional recognition.

Far (Far, 1992) introduced a functional reasoning technique called Qualitative Function Formation (QFF) that viewed system structures as an organization of finite number of interacting component pairs and derived the function from qualitative behavior. QFF assumes that at least a pair of components is required to interact functionally (functionality in item pair) and interprets a function either as persistence or as an order in the sequence of qualitative states (functionality in state transition). The technique extends some qualitative models by including temporal constraints and physical interaction.

3.4. FUNCTIONALITY = SHAPE + PHYSICS + CAUSALITY

Models in this category use representations of dynamic physical relationships and shape to recognize the functionality of tools. The recognition process is enhanced by the consideration of causal relationships between objects, such as the predictable or observable effect on some target object by carrying out an action with a tool.

Cooper et al. (Cooper et al., 1995) describes a set of programs that attempt to construct causal explanations of scenes by focusing on *why* the scene is the way it is and *how* an agent can interact with it. This causal explanation later forms a basis for functional description of scene elements. They focus mainly on the causality of support, the causality of objects in static equilibrium. They also show how causal descriptions can be exploited to physically interact with the scene. The solutions they offer can be applied to many other problems including occlusion, focus of attention, and grasp planning.

They see function as *a match between tool and intention* and believe that function arises when the physical configuration of an object permits the object to be used to satisfy a goal. Causal reasoning can evaluate this match by understanding both physical and intentional relationships.

They created three different systems. BUSTER (Blocks UnderStander That Explains its Reasoning) explores and explains blocks world and Fido "sees around" occlusions by using the knowledge of static stability and segmenting scenes of link-and-junction objects. The MugShot system understands how and why to interact with and pick up objects with handles.

3.5. DISCUSSION

Generally speaking, relying on only the shape of an object for reasoning about its functionality is limited. The interaction between tools and humans is affected by how we use them, and the proper usage of an object is usually highly dependent on interaction. Objects can have different potential functionalities and we can only be sure about which one they are using by observing their behavior. Stahovich, Davis, and Shrobe (1993) see this problem and attempt to come up with a large-scale, fundamental ontology for mechanical devices that is organized around behavior, not structure. Even if they give the structural definition of a lever as "a rigid bar with a pivot that can rotate," unless the bar is used to amplify the force, they accept it as a beam, not a lever. They also claim that a causal explanation is needed for differentiating between the actual and the possible behavior of a tool.

It is not clever to try to recognize tool functionalities by just looking at tools; since we do not use them by looking (except in the case of a mirror). An agent that is interested in learning how a tool can be used either needs to look for the changes it can achieve in the physical world by using the tool or be aware of the rules governing the creation of those tools. This way, tools are no longer named specifically as hammer but as *a-tool-that-can-increase-my-abilities-of-striking-objects- by-using-the-governing-rule-number-X*.

Therefore, we need to search for where these tools come from and what is the underlying functionality that we achieve while using them. For example, we can think of a hammer as a tool that changes the direction of the force and the momentum applied to it and we can figure out that its functionality is based on the basic functionality of a lever [4]. The human body has itself many

[4] (from www.m-w.com): a rigid piece that transmits and modifies force or motion when forces are applied at two points and it turns about a third; specifically : a rigid bar used to exert

levers; for this reason classifying tools according to their lever types seems appealing.

The criterion of success we are going to accept for a system is also a question of concern. Given that we have a system that can reason about functionality, how do we know that it is functionally aware enough? What are the adequacy constraints? Can we say that a system that can use a screwdriver as a hammer is functionally more intelligent than a system that suggests the usage of a towel instead of a wrist pad? Since there may be various dimensions along which some reasoning technique that a system is based on, becomes limited, and since one can always prefer one system to another given better performance along a dimension, defining a precise notion of the degree of functional intelligence for different systems may be difficult.

4. Abstract approaches

Finally some models try to realize the functionality of objects without any preference towards interactive or non-interactive systems. These tend to take a more generalized view of the problem, abstracting above the level of perception and motor action, while still attempting to represent the core aspects of the tool-using process.

4.1. Reasoning about design requirements

According to Freeman and Newell (1971), humans ubiquitously tend to reason in terms of functions. We name things according to their functions: a machine for washing clothes is called a "washing machine." In their paper, they are not really interested in recognizing objects in terms of their functionalities but designing objects and abstract systems like computer programs that have the desired functions. They give a qualitative model for the task of designing in terms of functions.

In the given model, they assume a set of propositions for the set of structures and a set of functions of a design task environment. They talk about functional connections that occur between structures that provide functionality to each other and how a new structure can be constructed from a set of structures. The propositions they make can also serve as a model for reasoning about functionality of object parts and how structures can be combined into new structures.

They try to answer the generic design problem:

> Given a set of structures and their functional specifications, construct a structure with desired functional properties.

---

a pressure or sustain a weight at one point of its length by the application of a force at a second and turning at a third on a fulcrum.

They examine the aspects and the framework of automated design systems with an example of qualitative design: a symbol table in computer systems. The design methods that can be used can be summed in two different groups: top- down or bottom-up methods. In this context, top-down methods start with the desired functions and try to find the structures that provide them, binding the design as little as possible. Bottom-up methods start with the structures available and construct larger structures until the desired functionality is reached. Freeman and Newell's work (1971) is the first system that attempts to explore the field of *automated functional reasoning*.

4.2. Ecological reasoning

This work aims to model physical objects and their functionalities by using the common sense knowledge of shape, physics, and causality together with naive physical information. In addition to that, they try to interact with objects with the belief that interaction is an important part of functionality.

St. Amant's ecological perspective (St. Amant, 2002; Amant and Horton, 2002) and their efforts of building a robotic system that can reason about the functionality of tool use is the only example in this area that we know of. St. Amant describes an explicitly ecological approach to understanding the nature of tool use. He cites the research in non-human primate cognition that emphasizes behavior in defining tool use:

Tool use involves *direct action*. A striking action with a stone, with the goal of cracking open a nut, is an example of tool use. Tool use often *amplifies existing behavior*. Using a stick to extend one's reach is a common aspect of tool use in experimental settings and in the wild. Tool use is *goal-directed activity*. Sometimes desirable ends are achieved through the incidental or even accidental use of an object, which is not considered a tool in that case. Tool use involves *effective behavior*.

St. Amant also gives a taxonomy of tools according to their intended usage:

– *Effective* tools produce a persistent effect on materials or the environment, such as hammers, saws, screwdrivers after tool use is terminated.

– *Instruments* provide information about materials or the environment. Instruments include measuring tapes, calipers, microscopes and magnifying glasses.

– *Constraining* tools constrain or stabilize materials or the environment for the further application of effective tools. Examples are clamps and rulers.

– *Delimiting/demarcating* tools demarcate the environment or materials, as when a carpenter uses a pencil to mark a piece of wood, or when a designer uses pushpins or labels on a drafting table.

Many tools fit into different categories at the same time. A pair of pliers, for example, constrains the material it grips, but also can be used as an effective tool, to pull on or twist the material.

St. Amant later gives a taxonomy of tools according to their ecological nature: Tool use can be *opportunistic*. Tools can be used for purposes not intended by their designers and conversely, an object can be used as a tool even if it was not designed as a tool initially. Tools *provide rich cues about their appropriate use*. The affordances of a tool become obvious in its use. Tool use *involves establishing and exploiting constraints* (between the user and the tool, the user and the environment, and the tool and the environment).

One might wave a saw or a hammer in the air, for example, or twist a screwdriver randomly, as a young child might do. Effective use, however, requires the establishment of a constrained relationship between the tool and the material it acts on.

Tools have *affordances*: designed relationships between their physical/dynamic properties and the properties/abilities of their intended users. Physical affordances, closely related to constraints, are mutual relationships that involve both the agent and the artifacts it manipulates (and the environment it operates).

The constraints that are relevant in the use of a tool fall into different categories, which would include the following: *Spatial* constraints describe the spatial relationships associated with a tool and its use in an environment. For example, to use a hammer one needs enough room to swing it. *Physical* constraints describe physical relationships in the use of the tool, such as weight or size. *Dynamic* constraints describe movement- or force-related properties of tool use. For example, one needs to swing a hammer with appropriate speed in its use.

4.3. Commonsense reasoning

Commonsense reasoning is a promising technique that concentrates on formalizing and finding computational models of how humans reason and think in a sensible way. Minsky (Minsky, 2000) believes that users have powerful "commonsense" knowledge that helps them correctly predict the behavior of functional objects on the screen. He claims that:

> "The secret of what X means to us lies in how our representations of X connects to the other things we know."

He also mentions the need for classifying objects according to what they can be used for or which goals they can help us achieve (Minsky, 1991).

CYC (Lenat, 1996) is a very large, long-term effort to formally represent commonsense knowledge we have in almost anything. The knowledge stored in this *expert system in commonsense world* is shallow and wide and does not go into many physical details. So, its definitions can be categorized as high-level purposeful definitions.

4.3.1. *Representation in CYCL*

CYC's high-level purposeful definitions are organized around micro-theories (Mt) that bundle a set of assertions based on (1) -a shared set of assumptions on which the truth of the assertions depends, or (2) -a shared topic. Each *Mt* is a set of abstract concept definitions and assertions for representing a domain [5] in CYC. Specialized micro-theories depend on more general micro-theories from which they inherit assertions.

OpenCYC [6] is the open source version of CYC technology. The knowledge base available is very limited. A specialized micro-theory of human activities is the only context that the use of some tools is mentioned. The knowledge is hardcoded, and thus the abilities of tools are limited to what they are supposed to. The "HumanActivitiesMt" assumes that the people are rational but not innovative in using tools; tools are used for their intended purpose and functional improvisation such as using a credit card to unlock a door is not represented.

In the current CYC system, CycL (Lenat, 1996) is used as the representation language. This is an extended version of the language of first-order predicate calculus (FOPC). We implemented the representation of tool use in CycL according the ecological perspective described in Section 4.2 as follows:

```
(and
  (requiresForRole ?TU ?A deviceUsed)
  (or (isa ?A PurposefulAction) ;goal-directed activity
      (isa ?A ActionOnObject))  ;direct action
  ;amplifies existing behavior
  (requiresForRole ?TU ImprovementEvent deviceUsed)
  (isa ?Tuser Animal)
    (thereExists ?Tuser (beneficiary ?TU ?Tuser))
)
```

We created a ToolUse micro-theory by using the OpenCYC system that is running on a Linux OS. However, the system's inferential abilities are restricted in the current version, which prohibits us from deriving conclusions

[5] Technical sense of context
[6] At www.opencyc.org.

that involve multiple micro-theories and their physical constraints. For example, if we are to achieve functional improvisation by using CYC, we need to be able to infer that an object that has a graspable portion and a striking surface can serve as a hammer.

Even though inferential problems are solved in the newer releases of OpenCYC, its compatibility with a robotic system that will interact with a physical environment as well as its efficiency in transmitting and executing those inferences in a timely manner is questionable. Also, the inability to make any additions to the inference engine is another concern.

There are still a couple of problems that face a scientist using CYC. To represent any concept in OpenCYC, we first need to find the micro-theory that it belongs to. Therefore, we need to have an idea of what each micro-theory is about, what they contain and what are the conceptual relationships between them. If you consider the amount of knowledge encoded, it becomes more obvious that you need considerable amount of time to realize where your concept belongs to.

After this first step, you may conclude that the system does not have enough knowledge to represent your concept (either because it is really not encoded yet or because you have not found the possible [7] micro-theory) and end up creating your own; just as we did in the case of ToolUse. Another problem occurs when we try to define each of the ecological requirements of tool use since they contain subjective concepts like “beneficial.”

In addition to that, the micro-theories that we use (either for the whole concept or its subparts) may be either more general or more specific than what we want to cover. So, to overcome the mismatch in the semantic granularity of these definitions, you end up creating your own micro-theories by using more and more basic ones. It is very likely that you are either forced to use cyclic definitions or resolve to infinitely deep chain of micro-theory creations.

In the end, our previous egg-yolk theory of the meanings about the functionalities of objects end up being vague and the only way to know that you cover the objects in your micro-theories is by creating them as instances in an appropriate (“impossible”) micro-theory. So, you end up doing “armchair engineering” rather than conducting empirical experiments in your physical world.

[7] Not correct since we believe that it is nearly impossible to find the correct micro-theory for your concept. This is because the concepts already in the system and the one you want to represent do not match each other. Even if they do match and it is represented the way you want to use it, it may be implemented or interpreted in a different way. Also, since the knowledge base of OpenCYC is not complete and may have discrepancies with the original CYC, which is proprietary, the trust in it is questionable.

5. Issues

There are other issues that still needs to be addressed about function based reasoning. We will try to address these as much as we can here.

5.1. Function-based Reasoning: Constraint Programming or Planning or Common Sense Reasoning?

Function-based reasoning can be seen as a constraint satisfaction problem where functional descriptions constrain structure or structure constrains functional possibilities. The mappings available between form and function are actually many-to-many and recovering an object by matching previously recognized ones' functionalities experience combinatorial growth. Model-based recognition has been thought as a solution.

Another view can consider reasoning about functionality as a planning module that is composed of helper procedures for recognition. In this view, the functional description is done at a higher level, discarding the complete representation. A complete representation of physical world could attempt to represent the forces governing the universe and reach from gravitational forces between planets to forces between chemical compounds and atoms.

Freeman and Newell (1971) claim that the uniformity of functional reasoning across all domains results in a *model-independent* reasoning technique that adapts according to the needs of the reasoner, not the domain. Humans find many ways to represent problems and knowledge so that if one method fails, they have the ability of switching between them. Minsky (2000) accepts commonsense reasoning as a domain-independent, adaptable scheme that switches between representations instead of looking for the best. In this sense, function-based reasoning is similar to commonsense reasoning.

As figure 3 implies, reaching the optimum tool use may sometimes be like finding a needle in a haystack. Selecting the most effective reasoning technique in tool use, or relying on a combination of previous techniques, is one of the issues that needs further investigation.

5.2. Top-down or Bottom-up?

The functional recognition process can be broken down into two types of methods: top-down and bottom-up. This division assumes that the problem we are facing is a search problem. Given the functional classifications, top-down methods start with the desired functions and try to find the structures that provide them, constraining the model as little as possible. Bottom-up methods start with the structures available and construct larger structures until the desired functionality is reached. That's why in the bottom-up methods, the object recognized is unexpected or unknown beforehand but in the top-down approach, we expect what kind of object we are trying to recognize.

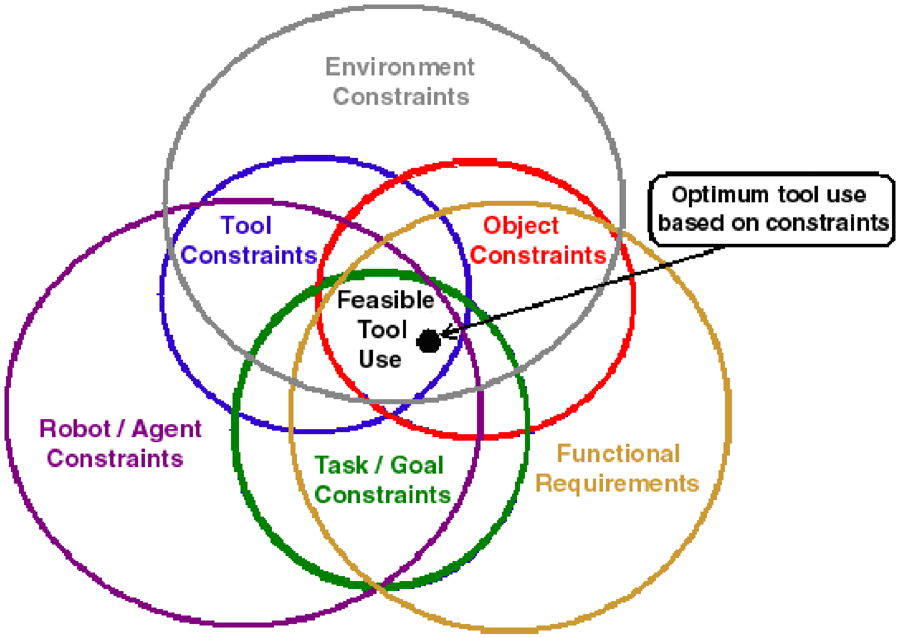

Figure 3. Needle in a haystack

Tools as well as objects in general can be recognized and reasoned about in a top-down fashion, based on conventions for their construction and designers' intentions for their use. For example, if an amateur mechanic is working under his car and has no appropriate tools for a hammering task within reach, the nearest object to hand can become a hammer. The assessment that an object can be used as a hammer is based on its heft, its grasp-ability, its solidity, and so forth, rather than on the imposition of an external categorization.

Clearly tools can be recognized and reasoned about in a bottom-up fashion, based on their physical properties and potentially on the observation of their use. For example, hammers conventionally have a long handle and a protruding head with a flat surface. A carpenter in search of a hammer on his workbench does not need to reason about the functionality of all the objects within sight, but instead can simply search for objects that have the general visual appearance of a hammer.

Consequently, the dilemma boils down to the question of: Are we trying to match some known models that are supposed to give some known functionalities to our objects, or are we trying to find those models that may support some functionalities we seek? How much spatial reasoning do we need to come up with to satisfy the criterion of functionality? As is reflected in much of the preceding work, effective reasoning about tool use must rely on a combination of top-down and bottom-up reasoning. How to combine them effectively also remains for future research.

5.3. Gestaltian "demand character" versus Gibsonian "affordance"

Psychologists have two different perspectives about the ecological existence of objects. On one side, they claim that the use of objects can be directly perceived as it is supported by the idea of demand character of Gestaltians. Objects have a demand character that demands an action and rejects others. On the other hand, they believe that the properties of an object can indirectly determine how it can be used with the idea of affordances described by Gibson (1979). The affordances of an object are the properties that it offers and, the values and meanings of these to the environment can be perceived by looking at what it affords.

The difference between these two approaches can be more obvious with an example. A Gestaltian will think that a hammer is meant for hammering whereas a Gibsonian will claim that with its striking surface and its graspable handle, a hammer affords hammering.

5.4. Frame Axioms and the Frame Problem

Dynamic state of the real world forces a robot to model all of the actions that can modify its own or its environment's state. This is called the frame problem since we are monitoring the environment through the window or frame of all of the actions we think that will effect the conditions (relevant actions), which will result in a reaction in the robot's sense-plan-act cycle.

Frame axioms describe how the world stays the same rather than how the world changes. Each predicate that may change its value over time needs a successor-state axiom, which lists all of the possible ways the predicate can become true or false. However, it may be hard to explicitly state all of the possible ways a predicate will hold true. In real world, it is difficult to define the circumstances under which an action is guaranteed to work (*the qualification problem* (McCarthy, 1977)). For example, grasping the handle of a tool may fail if it is slippery or electrified or too hot or nailed to the table. If we fail to include all of these possible situations, the robot can generate false beliefs.

The ramification problem can also occur, when we cannot predict the exact consequences of actions. For example, lifting a tool from where it was staying can result in an unbalanced weight distribution that can cause a collapse of the structure that was supporting the tool before.

To overcome this type of complications, we can assume that we are in a closed world and all of the things that are not explicitly changed will stay the same (and all of the things we do not know about are false).

5.5. Representing Functionality of Objects

An unbiased view of functionality needs to be a domain-independent solution to many recognition problems. This leads to the question of how functional representation should be. Davis et. al. (Davis et al., 1993) listed five important and distinct virtues of knowledge representation that sometimes conflict in their goals. We will reorder them here to match their importance in terms of functional representation (FR). The original ordering in Davis et. al. is as follows: 3-2-1-5-4.

1. *FR as fragmentary theory of intelligent reasoning:*

2. *FR as a set of ontological commitments:*

3. *FR as a surrogate:*

4. *FR as a medium of human expression:*

5. *FR as a medium for pragmatically efficient computation:*

6. Summary

Representing, classifying and recognizing tools by their functionality can provide us new opportunities for understanding and eventually improving an agent's interaction with the physical world. Throughout this paper, we have seen examples of approaches to functional reasoning in a general framework of tool use. We have collected the work in this area in categories of interactive and non-interactive approaches and later grouped them according to the dimensions of functionality and the complexity of the techniques they credit in increasing the sophistication of their modelling capability.

Non-interactive systems are observers that do not interact with the physical artifacts or the environment to realize object functionalities. A number of these systems use only the shape and structure and some extend these with learning and planning; a few use motion in addition to shape; several add physical and causal rules and some consider commonsense knowledge for recognizing objects with their functionality.

On the other hand, interactive systems interact with the objects and observe changes using techniques like haptic exploration, grasp planning, and physical perception to realize object functionalities. Similarly, a number of these approaches use only the shape and structure and some extend these by adding work space requirements; several use physics in addition to shape and some add causal relations.

There have been a couple of general approaches that do not specify any preference towards interactive or non-interactive approaches. Freeman and Newell look at the problem from a design standpoint and try to construct a structure with desired functionalities starting from a set of structures and their functional specifications. St. Amant describes an explicitly ecological approach to understanding the nature of tool use and CYC attempts an expert system in the commonsense world that has shallow knowledge lacking physical details.

We note and discuss the difficulties that emerge and the issues that need to be addressed for reasoning about functionality. We apply a selection of these techniques toward a few representative examples of tools en route to building a tool-using robot-arm. We believe that our work in the recognition of tools for specific uses will lead to bridge definitions that will enable researchers to bring robots into the world of human interactions. We envision an emerging need for applications using functionally aware robots and systems in the future.

References

Allen, P. K.: 1990, 'Mapping Haptic Exploratory Procedures to Multiple Shape Representations'. In: *IEEE International Conference on Robotics and Automation*. pp. 1679–1684.

Amant, R. S. and T. E. Horton: 2002, 'A tool-based interactive drawing environment'. In: *ACM Conference on Human Factors in Computing Systems (CHI) Extended Abstracts*.

Beck, B. B.: 1980, *Animal Tool Behavior: The Use and Manufacture of Tools*. New York,NY: Garland Press.

Bogoni, L.: 1995, 'Identification of Functional Features through Observations and Interactions'. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

Bogoni, L. and R. Bajcsy: 1993, 'An Active Approach to Characterization and Recognition of Functionality and Functional Properties'. In: *AAAI Workshop on Reasoning about Function*. pp. 9–16.

Bogoni, L. and R. Bajcsy: 1994, 'Functionality Investigation using a Discrete Event System Approach'. *Journal of Robotics and Autonomous Systems* **13**(3), 173–196.

Bogoni, L. and R. Bajcsy: 1995, 'Interactive Recognition and Representation of Functionality'. *Computer Vision and Image Understanding* **62**(2), 194–214.

Brady, M., P. E. Agre, D. J. Braunegg, and J. H. Connell: 1985, 'The Mechanic's Mate'. In: *Advances in Artificial Intelligence: European Conference of Artificial Intelligence*, Vol. 1. pp. 79–94.

Brand, M.: 1997, 'Physics-Based Visual Understanding'. *Computer Vision and Image Understanding: CVIU* **65**(2), 192–205.

Brooks, R. A.: 1981, 'Symbolic reasoning among 3-D models and 2-D images'. *Artificial Intelligence* **17**, 285–348.

Connell, J. and M. Brady: 1987, 'Generating and Generalizing Model of Visual Objects'. *Artificial Intelligence* **31**, 159–183.

Cooper, P. R., L. A. Birnbaum, and M. E. Brand: 1995, 'Causal scene understanding'. *Computer Vision and Image Understanding: CVIU* **62**(2), 215–231.

Davis, E.: 1990, *Representations in Commonsense Knowledge*. San Mateo, CA: Morgan Kaufmann Publishers.

Davis, E.: 1993, ‘The Kinematics of Cutting Solid Objects’. *Annals of Mathematics and Artificial Intelligence* **9**(3/4), 253–305.

Davis, E.: 1998, ‘The Naive Physics Perplex’. *AI Magazine* **19**(4), 51–79.

Davis, E.: 2000, ‘Commonsense Reasoning about Loosely Constrained Systems of Rigid Solid Objects’.

Davis, R., H. Shrobe, and P. Szolovits: 1993, ‘What is a Knowledge Representation’. *AI Magazine* **14**(1), 17–33.

DiManzo, M., E. Trucco, F. Giunchiglia, and F. Ricci: 1989, ‘FUR: Understanding FUnctional Reasoning’. *International Journal of Intelligent Systems* **4**, 431–457.

Duric, Z., J. Fayman, and E. Rivlin: 1996, ‘Function from Motion’. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**, 579–591.

Evans, T. G.: 1968, *A Heuristic Program to Solve Geometric Analogy Problems*. Cambridge, MA: The MIT Press.

Far, B.: 1992, ‘Functional Reasoning, Explanation and Analysis: Qualitative Function Formation Technique’. In: *Proc. 6th Annual Conference of JSAI*. Tokyo, Japan, pp. 229–232.

Freeman, P. and A. Newell: 1971, ‘A Model for Functional Reasoning in Design’. In: *Proceedings of IJCAI*. London, England, pp. 621–633.

Gibson, J. J.: 1979, *The Ecological Approach to Visual Perception*. Boston: Houghton Mifflin.

Green, K., D. Eggert, L. Stark, and K. Bowyer: 1994, ‘Generic Recognition of Articulated Objects by Reasoning about Functionality’. In: *AAAI Workshop on Representing and Reasoning about Function*.

Hayes, P.: 1978, ‘The naive physics manifesto’. In: D. Ritchie (ed.): *Expert Systems in the Microeletronics Age*. Edinburgh University Press, pp. 242–270.

Hodges, J.: 1992, ‘Naive Mechanics: A Computational Model of Device Use and Function in Design Improvisation’. *IEEE Expert* **7**(1), 14–27.

Hodges, J.: 1995, ‘Functional and Physical Object Characteristics and Object Recognition in Improvisation’. *Computer Vision and Image Understanding: CVIU* **62**(2), 147–163.

Kim, D. and R. Nevatia: 1998, ‘Recognition and localization of generic object for indoor navigation using functionality’. *Image and Vision Computing* **16**(11), 729–743.

Krotkov, E.: 1994, ‘Perception of Material by Robotic Probing: Preliminary Investigation’. In: *CVPR Workshop: The Role of Functionality in Object Recognition*.

Lakoff, G.: 1987, *Women, Fire, and Dangerous Things*. Chicago, Mi: Chicago University Press.

Lenat, D. B.: 1996, ‘From 2001 to 2001: Common Sense and the Mind of HAL’. In: D. G. Stork (ed.): *HAL’s Legacy: 2001’s Computer as Dream and Reality*. MIT Press.

Li, W.-J. and T. Lee: 2002, ‘Object recognition and articulated object learning by accumulative Hopfield matching’. *Pattern Recognition* **35**(9), 1933–1948.

Lifschitz, V.: 1998, ‘Cracking an egg: An exercise in formalizing commonsense reasoning’.

Lozano-Pérez, T.: 1983, ‘Spatial planning: A configuration space approach’. *IEEE Transactions on Computers* **32**(2), 108–120.

Mazlish, B.: 1993, *The Fourth Discontinuity: The Co-Evolution of Humans and Machines*. Yale University Press.

McCarthy, J.: 1977, ‘Epistemological problems in artificial intelligence’. In: *Proceedings of IJCAI-77, 5th International Joint Conference on Artificial Intelligence*. Cambridge, US, pp. 1038–1044.

Minsky, M.: 1986, *Society of Mind*. New York, NY: Simon and Schuster, Inc.

Minsky, M.: 1991, ‘Logical versus Analogical or Symbolic versus Connectionist or Neat versus Scruffy’. *AI Magazine* **12**(2), 34–51.

Minsky, M.: 2000, ‘Deep issues: commonsense-based interfaces’. *Communications of the ACM* **43**(8), 66–73.

Morgenstern, L.: 2001, 'Mid-Sized Axiomatizations of Commonsense Problems: A Case Study in Egg Cracking'. *Studia Logica* **67**, 333–384.

Pentland, A., I. Essa, M. Friedman, B. Horowitz, S. Sclaroff, and T. Starner: 1990, 'The Thing-World Modeling System'. Technical Report 145, MIT Media Lab Vision and Modeling Group.

Rivlin, E., S. Dickinson, and A. Rosenfeld: 1995, 'Recognition by Functional Parts'. *Computer Vision and Image Understanding, special issue on Function-based Object Recognition* **62**(2), 164–176.

Shanahan, M.: 1998, 'A logical formalization of Ernie Davis's egg cracking problem'.

Solina, F. and R. Bajcsy: 1986, 'Shape and function'. In: *Proceedings of SPIE: Intelligent Robots and Computer Vision XI*, Vol. 726 of *5*. pp. 284–291.

Sowa, J. F.: 2000, *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA: Brooks Cole Publishing.

St. Amant, R.: 2002, 'A preliminary discussion of tools and tool use'. Technical Report TR-2002-06, North Carolina State University.

Stahovich, T., R. Davis, and H. Shrobe: 1993, 'An Ontology of Mechanical Devices'. In: *AAAI Workshop on Reasoning about Function*. pp. 137–140.

Stansfield, S. A.: 1992, 'Haptic perception with an articulated, sensate robot hand'. *Robotica* **10**.

Stark, L. and K. Bowyer: 1996, *Generic Object Recognition using Form and Function*, Vol. 10 of *Series in Machine Perception and Artificial Intelligence*. World Scientific Press.

Stark, L., K. W. Bowyer, A. W. Hoover, and D. B. Goldgof: 1996, 'Recognizing Object Function Through Reasoning About Partial Shape Descriptions and Dynamic Physical Properties'. *Proceedings of the IEEE* **84**(11), 1640–1658.

Vaina, L. and M. Jaulent: 1991, 'Object structure and action requirements: A compatibility model for functional recognition'. *International Journal of Intelligent Systems* **6**, 313–336.

Wilson, R. H.: 1996a, 'A Framework for Geometric Reasoning About Tools in Assembly'. In: *International Conference on Robotics and Automation*. pp. 1837–1844.

Wilson, R. H.: 1996b, 'Geometric Reasoning About Assembly Tools'. Technical Report SAND95-2423, Sandia National Laboratories.

Winston, P. H.: 1975, 'Learning Structural Descriptions from Examples'. In: *The Psychology of Computer Vision*. McGraw Hill, pp. 157–209.

Winston, P. H., T. O. Binford, B. Katz, and M. R. Lowry: 1983, 'Learning Physical Descriptions from Functional Definitions, Examples, and Precedents'. In: *National Conference on Artificial Intelligence*. Washington, D. C.

Zlateva, S. D. and L. M. Vaina: 1991, 'From object structure to object function'. In: *Proceedings of SPIE: Applications of Artificial Intelligence IX*, Vol. 1468. pp. 379–393.

Zlateva, S. D. and L. M. Vaina: 1992, 'Three-dimensional object representation based on the largest convex patches method'. In: *Proceedings of SPIE: Intelligent Robots and Computer Vision X: Neural, Biological, and 3-D Methods*, Vol. 1608. pp. 72–80.