May 1992

An Active Approach to Functionality Characterization and Recognition

Luca Bogoni
University of Pennsylvania

Ruzena Bajcsy
University of Pennsylvania

Recommended Citation

Luca Bogoni and Ruzena Bajcsy, "An Active Approach to Functionality Characterization and Recognition", . May 1992.

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-92-37.

An Active Approach to Functionality Characterization and Recognition

Abstract

In this paper we focus on understanding and defining a methodology for object description and recognition both in terms of its geometrical, material and functional specifications. We define functionality in an object as its applicability toward the achievement of a task. We emphasize and develop an interactive and performatory approach to functionality recovery. Furthermore, we introduce the distinction between *Inherent, Intended and Imposed* functionality.

By analyzing interaction and manipulation tasks as goal-oriented recognition processes we propose to identify and characterize functionalities of objects. This interaction is not only a means of verification of the hypothesized presence of functionality in objects but also a way to actively and purposively recognize the object.

In order to accomplish our goal, we introduce a formal model, based on Discrete Event Dynamic System Theory, to define a task for recovering and describing functionality. We extend the recovery process to an algebra of tasks. We describe how a more complex task call be composed from a set of primitive ones. This constructive approach allows a task to be built from simpler ones in an stepwise fashion.

Once the manipulatory task has been described in the formal model, it must be instantiated in a context. In such a context, the behavior of the system in which the interactio between a Manipulator, a Tool and a Target object must be observed. Thus, the description of tasks themselves provide must for means of addressing observability through different sensor modalities. For this purpose, we introduce the notion of *Partial Observability* of a task. This allows the description of a plant in which not all events and the time of their occurrence might he modelled and therefore predictable in advance.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-92-37.

An Active Approach to Functionality Chracterization and Recognition

MS-CIS-92-37
GRASP LAB 315

Luca Bogoni
Ruzena Bajcsy

University of Pennsylvania
School of Engineering and Applied Science
Computer and Information Science Department
Philadelphia, PA 19104-6389

May 1992

An Active Approach to Functionality Characterization and Recognition

Luca Bogoni

and

Ruzena Bajcsy

May 5, 1992

AN ACTIVE APPROACH TO FUNCTIONALITY CHARACTERIZATION AND RECOGNITION

Luca Bogoni and **Ruzena Bajcsy**

Abstract

In this paper we focus on understanding and defining a methodology for object description and recognition both in terms of its geometrical, material and functional specifications. We define functionality in an object as its applicability toward the achievement of a task. We emphasize and develop an interactive and performatory approach to functionality recovery. Furthermore, we introduce the distinction between *Inherent, Intended* and *Imposed* functionality.

By analyzing interaction and manipulation tasks as goal-oriented recognition processes we propose to identify and characterize functionalities of objects. This interaction is not only a means of verification of the hypothesized presence of functionality in objects but also a way to actively and purposively recognize the object.

In order to accomplish our goal, we introduce a formal model, based on Discrete Event Dynamic System Theory, to define a task for recovering and describing functionality. We extend the recovery process to an algebra of tasks. We describe how a more complex task can be composed from a set of primitive ones. This constructive approach allows a task to be built from simpler ones in an stepwise fashion.

Once the manipulatory task has been described in the formal model, it must be instantiated in a context. In such a context, the behavior of the system in which the interaction between a Manipulator, a Tool and a Target object must be observed. Thus, the description of tasks themselves provide must for means of addressing observability through different sensor modalities. For this purpose, we introduce the notion of *Partial Observability* of a task. This allows the description of a plant in which not all events and the time of their occurrence might be modelled and therefore predictable in advance.

Contents

1 Introduction

Interacting with any environment requires knowledge or means of acquiring information and processing it. Specifically, when performing a task we need to obtain information about the physical characteristics of the objects. Some of the data is then interpreted and functional and/or relational labels are attached. This process allows us to classify objects and, in the context of a task, to associate suitable interpretation to them. But how are object's labels associated with a completely unknown object? We analyze and perform inductions, deductions, analogies, etc. . However, what we can formulate are really only hypotheses about an object. Claiming that an object looks like a container attaches a functional description to it; only upon interacting with the object and testing it can we corroborate our hypotheses. Furthermore, if the interaction contradicts some of the hypotheses and yields a completely different result what steps should be taken?

The interactive approach is a technique which is often used even when a person interacts with a known environment. When considering the issue of *support*, for instance, we don't obtain exact measures of the weight of an object and then model and measure the stress that it would import on a supporting surface. When we are unsure simply place the object on the surface progressively releasing our hold and observing the behavior of the supporting surface. While we might have a priori knowledge about the ranges of support, still the interactive methodology remains the *sine-qua-non* for verifying that the surface can or can not "support" the object thus qualifying its functional information.

Objects have geometrical, material, kinematic and functional properties, and their description must capture both aspects. Historically, mankind, in his interaction with the environment, observed the properties of natural objects and their applicability. Some of these natural objects were then modified and adapted to emphasize specific functionalities, see [Oakley, 1976; Willis, 1989; Grace, 1989]. The art of toolmaking marks the dawn of the functional characterization of objects. Specific physical properties were selected to characterize a particular function. While only speculations can be our guide to the initial names attributed to objects, physical and functional properties in tools provide the strongest evidence that the essential constituents of object representation and description had been determined. Later on, the adaptation of natural objects was slowly replaced by materials – clay, wood, iron, etc – in which the geometrical and physical properties could be molded to achieve the same functionalities. Containment could be performed by an amphora rather than a coconut shell and sawing could be done by a serrated knife rather than just a sharp stone. We observe that while some of the properties can be associated with natural objects because of their implicit functionality, tools have become specialized modifications of natural objects to best suit man's manipulatory capabilities and are best used in performing a function.

Thus, we focus on understanding and defining a methodology for object description and recognition both in terms of its geometrical, material and functional specifications. We analyze interaction/manipulation tasks as goal–oriented recognition processes which allow us to identify and characterize functions in objects. The evaluation of the applicability of an object, with respect to a function to be tested, allows us to qualify and quantify the occurrence of a given function in an object.

Such an evaluation process can also extended to develop a methodology of learning of new functionalities. This process of acquiring new knowledge allows an interaction of symbol driven and data driven approaches as a way for explaining new phenomena. Explaining and understanding of new phenomena are key factors to the discovery of new functions, redefinition of previous ones (either by extending, specializing or presenting a different procedural approach), and determining a qualification and quantification procedure to evaluate functional performance.

1.1 Overview

We introduce the process for active recovery and characterization of functionality. While the overall plan is introduced, the main thrust of this paper is that of presenting a formal basis for tasks descriptions. The remaining stages are outlined here and will be dealt with at a later time.

In section 2 we introduce what we mean by functionality. Its characterization in artifacts and in natural objects is presented. Differentiation between *Inherent, Intended* and *Imposed* functionality is introduced. Since functionality is performatory in nature, we examine the importance of having an interactive approach for recovering it. Finally we conclude by investigating the importance of functionality as a component in the object's representation and interpretation. Furthermore, we point out that the description of an object, used as a tool, should not only have a declarative but also a procedural component. The declarative component characterizes the geometrical and material properties and defines how they are structurally related to represent the functionality. The procedural component identifies how the object is to be applied, the type of actuator intended, and possibly the contexts of applicatibility

A review of previous work in recovering functionality is presented in section 3. As we shall see, many of the approaches have been either concerned with attempting to recover all the functionalities in an object, or restricted themselves to high levels of recovery, or concerned with a limited domain. It becomes quite clear, after short reflection, that the topic of functional recovery is marred by an endless list of problems which are the intrinsic problems of not only Computer Vision, but also Control Theory, Cognitive Science, and Psychology. This suggests why this important topic of research has received only limited attention. Notably, none of the previous approaches identify how the nature of the problem should clearly be presented as interactive fashion. This interaction is not only a means to verify hypothesized presence of functionality in objects but also a way to actively and purposively recognize the object.

Section 4 addresses the need for a formalism and proposes properties which it should embody in order to recover functionality. The notion of observability is developed to provide a way to close the interactive and the interpretation loops. The components for the task description, instantiation and recovery are then presented.

The description of the recovery process is topic of section 5. The section is divided into two parts. In the first portion of the section we progressively introduce the requirements, pointed out in section 4, for a formal model to describe tasks. In the second part, we define an algebra for tasks. We identify how, from a set of primitive actions, elementary tasks, more complex actions can be composed. This constructivist approach allows a task to be built from simpler ones in an stepwise fashion.

While it is not the goal of this paper to address instantiation of a task, in section 6, we propose a means both of describing a task abstractly in terms of position, velocity, force, etc. and a way of linking those abstract descriptions to real quantities in a domain. While the full instantiation process require more consideration of the domains in which the task is mapped to, as we have pointed out in section 4, being able to characterize these quantities in a context is essential to the whole process.

We then conclude in section 7 by briefly reviewing what was accomplished in the present investigation.

1.2 Acknowledgements

I would like to thank the members of the GRASP Lab and visitors who have stimulated and supported this work. In particular, I would like to thank my advisor Dr. Ruzena Bajcsy and my office-mate Marcos Salganicoff for the many insightful conversations and support.

This research was supported by: Navy Grant N0014-88-K-0630, AFOSR Grants 88-0244, AFOSR 88-0296; Army/DAAL 03-89-C-0031PRI; NSF Grants CISE/CDA 88-22719, IRI 89-06770, and ASC 91 0813; and Du Pont Corporation

2 Characterization of Functionality

In this section we describe the underlying philosophy of the active approach to function characterization and recognition.

2.1 Functionality in Objects

To address the notion of functionality in objects it is essential to precisely define what is meant by it.

> The **functionality** in an object identifies the applicability of a specific object toward the achievement of a task.

The function of containability in a cup, for instance, points out its applicability when planning to transport some substance, possibly a liquid. The function of piercing in a knife or sharp–pointed tool identifies that it might be applied to perforate some target object. The function of hammering identifies in a tool its applicability to transfer, and possibly amplify, an impact force on some target object(s).

2.1.1 Objects: Natural and Artifacts

In order to represent objects we need to be able to incorporate both factual and functional knowledge about their properties. The factual aspect encapsulates information about measurable attributes of an object; the functional knowledge describes application of a subset of the properties toward an action or goal[1].

In order to establish properties, and functional descriptions of objects, we need to understand what defines and classifies an object[2].

Objects can be partitioned into two major categories, *Natural Objects*[3] and *Artifacts*. This partition is necessary because while in the former case there is no a priori functional description of the object, in the latter case certain functional properties have been specifically assigned to a given object.

The observation of specific shapes and correlation to the functionality may have given origin to the conceptualization and abstraction of the underlying geometrical and material properties.

---

[1] To avoid pointless discussion when addressing the importance of functionality in object descriptions we are going to restrict our attention to non–teleological interpretations of functionality. Thus considerations such as "the function of man", "the function of art", and alike belong to a "meta–functionality" which is not addressed here.

[2] While it is possible to consider the manipulator as one of the objects whose capabilities and descriptions can be determined, we will assume that any system will have knowledge of itself. We will consider this as portion of the a priori domain knowledge.

[3] To clarify, while it is possible to address the function of a plant or an animal, when addressing natural objects we are primarily interested in inanimate objects.

Such properties were then adapted and employed to give shape to tools and utensils, see [Oakley, 1976; Willis, 1989], and become part of the description of such object; specialization then gave them further refinement.

Any artifact has not only the properties associated with it but also has an *intended actuator* for which these properties have been specifically associated with. Thus, we must observe that while there is a function associated with an object, there is an implicit definition of the actuator, for which it was intended for, built in the functional specification of the object. For instance a tool with a handle is meant to be manipulated by a hand–like manipulator.

With these considerations in mind we proceed to outline some of the fundamental issues and as well as to present some definitions with the goal to clarify the objects' properties and, later on, our interactions with them.

Objects possess *Geometrical, Material, Kinematic,*[4] and *Functional* properties.

- **Geometrical** properties identify quantifiable parameters defining shape described in terms of length, width, height, volume, etc.
- **Material** properties also are identifiable by quantifiable measures. Their attributes are defined in terms of units of weight, reflectance, coefficient of friction at the surface, density, etc.
- **Kinematic** properties describe the kinematic behavior of an object or some of its parts.
- **Functional** properties in an object describe sets of physical properties, material and geometrical, which are necessary for a given action to be successfully carried out. – containment, support [5], pounding, etc.

Furthermore, we can characterize functionality as inherent (intrinsic), intended and imposed (extrinsic);

- **Inherent** functionality defines the functional specifications which arise from the physical properties of the object. Namely, if the object has a rather pronounced concavity it is possible that it can contain, liquids or solids pending upon the material it is made out of. That specification arises independently of any actuator interaction.
- **Intended** functionality defines the specifications which were defined as a part of the object at the time of design. Unlike the previous one, such functional specifications are

---

[4] These distinctions were identified and investigated by Lederman and Klatzky in the domain of Psychology and Psychophysics, see [Lederman and Thorne, 1986; Lederman and Klatzky, 1987b; Klatzky *et al.*, 1987a]. Work by Campos and Bajcsy, [Campos, 1992], and Sinha and Bajcsy, [Sinha, 1992], investigated these properties by using experimental procedures, EP's.

[5] Actually a special case of containment. We need to describe the relation of the functional specification so that we are able to understand when support can be interpreted as containment as vice–versa.

not directly deduced or inferred from the physical structure of the object – a hammer or screwdriver for instance.

- **Imposed** functionality defines the ability of using an object for a function for which it is not intended. The use of a bowl for hammering would be such an instance.

To clarify the distinction between intended and imposed we note that a fork is constructed with the intended functionality of piercing and carrying, yet one may impose on it the functional property of cutting.

Interesting examples of imposed functionality in the natural setting are presented in the work of the anthropologist Jane Goodall, [Goodall, 1986]. Goodall documented the use of tool by chimpanzees of the Gombe Stream Reserve in Tanzania. She noted that chimpanzees used straws for fishing of termites from a termite hill as well as using a sponge to gather water from the bottom of a basin. They not only used the most appropriate vines to employ as tool, but also stripped off the leaves, in a form of tool–making.

Amongst the many other uses of natural objects for a specific function the following application of natural objects as tools is illuminating.

> " A novel problem arises in West Africa, where chimps feed on nuts of palm trees. Some of the nuts are too hard to crack by biting; the chimps use a stone as a hammer to smash the nuts against the roots of the tree. When this did not always work, so the chimps resorted to a hammer–and–anvil technique of transporting two stones several hundred yards in anticipation. They knew which tree needed this technique and not others", [Goodall, 1986] [6].

While in the above examples one can't really make a case for the chimpanzees' mental ability to conceptualize objects, it makes a strong support for the attribution of functionality in a natural setting.

Imposed functionality considers the applicability of an object toward a goal without considering physical alteration of its external properties in order to accomplish the function. Namely, an object which *per–se* is not grippable could become "grippable" if a handle were attached to it. This would, however, constitute an alteration of the original object implying an addition or possible deletion of one or more properties or functionalities. It does suggest that a specialization or generalization of the properties and hence of its classification. While this can be perceived as an imposed functionality, its characterization and description goes beyond the scope of this paper, for it addresses a higher level of interaction with an object. Furthermore, it requires a higher degree of reasoning.

---

[6] We find similar approaches to the use of tools in many other animals. Sea–otters, for instance, use a stone for cracking open shells.

2.1.2 Order and Degree of Functionality

Certain objects may have more than one inherently characterizing functionality. Thus we can define the functionality present as

- Simple
- Complex
  - Degree Preeminence in the object
  - Order Relevance to given task to be performed.

If the functionality is complex, it is necessary to address the notion of degree and order of a given functionality.

The term *Degree* refers to the notion that several functionalities might be present in a specific object, however, some of them might be more pronounced than others.

The notion of *Order* for a given functionality becomes a means of imposing a different ordering based on the task to be performed. Such ordering need not reflect the implicit ordering of the intended functionalities.

Thus, a mug may be used for carrying liquid and for hammering. The property of containment would certainly be the one preeminent, unless it were extremely heavy. However, if the task at hand were that of hammering, then the property of containment would be less relevant.

The need for the ordering of functionalities and actually the features which characterize them appears more apparent when discriminating between several object in the same class. For instance two hammers may be made out of different materials, rubber and steel, or when the length of the handle may be quite different. We begin to notice a need for some sort of quantification and qualification of the functions appearing in an object.

2.1.3 Physical and Functional Descriptions

The definitions presented so far allow us to apply our terminology of physical and functional descriptions to objects both in a natural environment as well as in a structured one. Thus we can talk about the functional description of a stick or of a stone in a natural setting and that of a screwdriver and of a brick in a well defined environment.

In artifacts, function is the preeminent characteristic by which they can be described. A cup or a glass, for instance, have the function of containing some liquid. It is, after all, the first mental association that one makes with respect to the function of a glass. Properties of the specific function implicitly suggests the association of geometrical and physical characteristics.

Natural objects, on the other hand, present a bit of a dilemma because some of them seem to have many functionalities associated with them. A rock for instance could be used for support, hammering, containment, etc. but it seems that, in general, none of these functional properties are intrinsic to definition of a rock.

What strongly characterizes some natural objects are some physical properties which are intrinsically defining it. When thinking of a "rock" one does not have a specific function in mind, but rather properties, such as hardness and density. These ***physical properties*** are indeed the constituents for the concept of the natural object. On the other hand, containability is intrinsic to the concept of glass independently of the constituent material.

We have thus observed that there are objects for which the functional description may not provide a means of acquiring information specific to their conceptual abstraction but it might at least provide understanding of their applicability and attribute some labeling in terms of functionality.

2.2 Underlying Motivations for the Use of Functionality

From the previous section, it would appear that functionality is not too crucial to the description of natural objects. They are often described in terms of a taxonomy; one then might consider the possibility of applying such a description methodology to all objects.

In [Winston *et al.*, 1984], it is observed that it is extremely hard to provide a vision system with accurate and meaningful descriptions of objects. It is on the other hand much easier to describe what objects can be used for. Considering this philosophy of object description we outline some of the reasons supporting the use and the importance of functionality and the questions it raises. Here are, then, the most prominent ones with some associated questions they address:

- It suggests a shift of focus in the description of objects based on their applicability in a task rather than only on their factual components.
- It falls under the umbrella of purposive recognition, and objects can be classified based on the minimal amount of information sufficient to fit the functional description.
- It can be used in the recognition process: "recognizing by interaction". It also includes observation and verification. How much information is required before I can identify an object?
- The meaning of the name attributed to an object depends on the physical properties but also on the functional properties associated to it. How is functional semantics initially associated to a given object and how is any given prototype defined?
- The recovery of the association of groups of features which identify a given function. How is it known that concavity suggests containability in an object?

- It addresses important issues in the origin of knowledge and methodologies both for learning of new functions and extending previous objects definitions. What is it that suggests the applicability of an object in a given task? Are the physical properties sufficient or is the dynamic interaction and the expectations necessary to describe it?
- The recovery of prototypical objects identifying the most general exponent in a class. What is the granularity in the representation which is necessary to define a representative in a class?

2.3 Choosing to Recognize by Interaction

Object recognition using a classification based approach requires that features be recognized in objects by clustering them, indexing into a data base. However, that assumes that we are able to:

- recover the relevant features from the object,
- that the features be relatively clean from noise both at sensor level and at object level (irrelevant detail),
- that the recovered features be interpreted after the modeling schemas have performed appropriate data reductions,
- that the recovered set match some model in the data base.
- that the modeling schema chosen be appropriate.

While all of the above restrictions seem logical and sensible, they impose major constraints on the recognition process. Testing for functionality in a given object provides for a considerable step forward in object recognition. It allows us to recognize objects based on their applicability in a context rather than only on specific features. In a task–driven environment, which characterizes many environments, such approach may be possible when recognition entails some object interaction. While it may not be a viable approach when attempting to classify objects in which no interaction is possible, it provides a methodology for qualification of material properties even in a natural setting. In such case, testing for given functionalities, using experimental procedures (EP), will provide the ability of recognizing object properties essential to classification. Support of the application of EP's for the purpose of classifying objects and recovering material properties is found in [Lederman and Klatzky, 1987b; Lederman and Klatzky, 1987a; Klatzky *et al.*, 1987a; Sinha, 1992; Campos, 1992].

Alternatively, to explicitly interacting with an object in order to determine its functionality, simulation could be carried out. Such approach, however, would require a complete model of the object and the result of the interaction.

2.3.1 Perceptual and Performatory Testing

Perceptual testing, characterized by simple EPs, provides us with the possibility of recovering inherent and intended functionalities; however, it is in the performance of a task to verify the applicability of an object that we are able to recover imposed functionality. Bajcsy [Bajcsy *et al.*, 1991] identifies the different type of testing as *Perceptual Tests* and *Performatory Tests.* This distinction further emphasizes the added contribution of the interactive aspect of the recovery process.

2.3.2 Coarse Recognition

Proceeding to recognize an object based on its functionality provides means of performing a coarse recognition allowing to focus on its functional components rather than on all the details.

Winston, in [Winston *et al.*, 1984], notes the relevance of functional description as a way to describe the object in terms of its relevant characteristics rather than trying to recognize an object across all its possible instantiations. Thus, by focusing on the relevant components we might be able to recognize an object as a member of a class and then to key into the details.

What it is suggested here, instead, is to proceed by testing whether the object in consideration has the functionalities which are characteristic of a chair. One would basically want to answer the question "can I sit on this thing?".

One could take two paths to answer this question:

- Observe an interaction: "A human sitting on it" or
- Test for specifics such as:
  - the object has a surface for support and it can support 'x' amount of weight.
  - the supporting surface is at a given height from the ground.
  - it has a back side to it which can be leaned against.

While it may be appealing to be able to recognize human behavior, it requires a great amount of additional representation and processing in order to analyze the scene. If, on the other hand, a system is interacting with an object, it can control the interaction and require less amount of representation and information.

The sequence of tests outlined above can be used to recognize "chair" objects, which with the added addition of support, must have a back and a supporting surface located at a given height from the ground. The last step in the sequence already sets restrictions to what is going to be consider a chair versus what would be considered a stool. However, in both cases, the notion that the object could be used for sitting may be recovered.

If the chair where to be of rococo style then once it was recognized to be a chair, it could be further analyzed for specific features characteristic of chair taxonomy. This approach to recognition could be identified as a refinement process in which tests for functionality and taxonomy could be alternated. In fact, some subcategories could have functional distinctions.

Thus, the virtue of functional recognition is presented here not in antithesis to classification methodologies, rather as an integration to the standard feature based recognition process.

2.4 Interpretation and Semantic Attribution

Any system in which taxonomy alone is sufficient for recovering a functional property, must have some meta–knowledge of the functionality in terms of features and their relations. The associations of the physical relations and the use of an object must have been formulated on the basis of pre-existing knowledge provided to the system.

Fitting the superquadric's parameters which describe a bowl–shaped object allows only to hypothesize its function. It is only by the experimental verification that the function of containment can be associated with the features defining concavity.

As humans we have acquired these relations through experimentations. Some relations have learnt but others may have been discovered at dawn of time. Then they could have been passed on as a collection of examples and respective methodology of investigation.

The name associated with a specific object implicitly determines strict requirements for grouping of the features characterized in the prototype for a class. The criteria determining the necessary and sufficient features must arise with respect to relations between features.

2.4.1 Difference of Interpretation in Active Definition

While the semantics associated with a prototypical object is based on its intended functionality and its geometrical and material properties, its contextual interpretation may vary.

The notion of the functional classification is relative in the sense that labels, defining the presence of a given functionality, may be applied only with respect to a given sensor/actuator. It may be that an object be liftable yet the operation can not be performed with a specific manipulator. Furthermore, the notion of something being liftable has reason to exist only in the context of an manipulator which can perform the action. Thus, small little coins may be pickable *per–se* but do not possess that property if the operator can only use boxing gloves as end effectors.

When imposed functionality instead of inherent functionality is considered, with respect to a new context, then a new interpretation of the object may be derived. For example, a spoon may actually be used as a knife. While this may seem to be a drawback, it is this flexibility of interpretation which defines one of major strengths of functional classification. A system capable of performing this operation is likely to be more robust in an unknown environment or in an environment of which little is known. If the agent were in a situation where the proper tools were not available then it might consider any one object which could fulfill the requirements for the task to be carried out. Furthermore, if there were several tools, then a ranking of the tools could be performed and the most appropriate one selected.

2.4.2 Multiplicity of Representation and Inconsistency

When classifying an object one would like the properties which characterize it to be *well defined* and that the representation be unique. In terms of the properties of any given object, it is important to observe that while the geometrical and physical properties are ***absolute***, they are intrinsic to the object in a given environment[7], its functional properties are ***relative*** in that they depend in part on the actuator which is performing the classification. This may suggest a different procedural interpretation for the object pending on the manipulator which is being used and how it is employed. Considering different manipulators with their idiosyncrasies, it might be impossible to make any sense of similarity of the task being carried out. Thus the specific steps involved in the recovering task for the same functionality when performed by two different manipulators may differ. The instantiated primitives describing the steps involved could yield quite different interpretations.

For instance, a large bar may not be either "grippable" nor "liftable" for some gripper, yet perfectly "grippable" nor "liftable" for some others. The above property is *relative* to the actuator employed. Then, in this relative interpretation scheme, we would be met with an inconsistent conclusion. To avoid inconsistency we can restrict the inability to recover a functionality with as a null reinforcement instead than a negative one. Negative reinforcement for the lack of a given functionality is harder to establish than positive ones. It may be quite impossible in certain instances to prove that a given functionality is not present in an object. It would require testing all possible approaches and all possible handlings. For instance, the inability to grasp something would not guarantee exclusion of the existence some manipulator which has different parameters and for which grasping can be performed. So functional interpretation will be associated with the context in which it was determined.

Multiplicity of representation also arises from the possibility of achieving the goal for a given task using different grasps, different angles of approach of a given tool with respect to an object. Instead of placing all the emphasis of the classification on the parameters characterizing an object, the emphasis can be measured in terms of the feasibility of the goal completion. Clearly, there have to be some common underlying properties for a given function to hold, but these are

---

[7] Physical properties are subject to the environment in which they are perceived. Thus, in order for the descriptions to be meaningful there must be an associated description of the environment in which they were measured. Weight and temperature may be greatly different and may loose of importance if measured at sea level, underwater or in free-fall.

captured by the actual feasibility of performing a given function. Thus, one can classify objects in terms of their applicability for a specific function:

> Two objects are said to be **functionally equivalent** if they can be employed to accomplish the same goal.

Under this description one can classify objects in terms of equivalence classes of functions. This is quite interesting when considering both intended functionality and imposed functionality. Members of a given equivalent class could turn out to be quite different. While, at first, this seems a major drawback it can be used as an advantage. Objects which are radically different, but which have reasons to be associated together, are most informative in the sense that it makes it easy to disambiguated the causation for their association. Since the reason for their association is known, they are functionally equivalent, what can be recovered are actually the necessary features, expressible in terms of physical properties, which characterize the functionality recovered. Specifically, if the objects in questions are two hammers which have minute differences, color and length of handle, one could perhaps infer that neither attributes are essential. However, if the two objects were a hammer and a stone, then "hardness of material" and "manipulability" could be the observed common denominators[8]. These two properties would much more representative of what characterizes the function of "hammering".[9]

2.4.3 Extracting the Concept from the Consensus

The process of extracting the concept from separate experiences is reminiscent of the tale of "Three Blind Men and An Elephant". In this tale the three men, who have never heard of this beast, are brought near the animal. After having had a little time to get acquainted with the animal, they are asked to describe what it was they felt. Three different opinions and different representations were reported. This tale tell us that the process of extracting an interpretation is clearly relativistic and rather myopic. All the acquired knowledge and definition of the prototype is relegated to the experiences that the recognition process was exposed to and based on the model employed. In the case of this tale no common denominators could be discovered. This process suggests the complexity of reaching a consensus without having any common knowledge. The question addressed is slightly different then the one presented to the three blind men of the story. We are asking whether a given function is present in the object and not what the object is. So by asking the same questions and knowing the context of each recovery operation one should be able to determine what are the common denominators and reach some consensus among different recognition systems.

---

[8] This is approach is also known as inductive generalization.

[9] We note that in this example "manipulability" is actually a type of functionality. It supports the observation that object description may involve further functional properties as well as physical properties

In the process that we advocate that recovery can be considered as extracting the consensus for a description at two levels.

- By varying the objects on which the functionality test is performed. One considers the objects which could perform the functionality and compare them on the basis of the common denominators. This should recover what are the invariants amongst the different objects. This is a form of *intra-characterization*
- By considering different operators performing the same functionality test with a given tool and on similar objects (the experiment could not be repeatable). This could result in detecting the invariants in the representations attained. This would be some sort of *inter-characterization.*

We can see immediately that in order to have a better description a combination of both should be used.

Thus, an object can initially be assigned to some class, perhaps being the only element in such class. At a later stage, other objects can be assigned to the same class. The common denominators of the attributions, both physical and functional, will determine the prototype for the class. One can envision a long process by which several objects can be inserted in the class. This process can actually been considered as the one yielding the characterization for the object. It has, furthermore, the additional benefit of suggesting how certain features may be actually be identified as some of the essential descriptors for functionality. After a long process, concavity may be identified as the geometrical prerequisite to containment. The prototype for a given class can then be assigned a symbolic name, a label. Such a label may be quite different from the label assigned by any other recognition system. However, upon comparing the characterizing invariants for a given class, consensus may be reached and a common labels can be applied. The two schemas may be merged into a common one. While the labeling may initially be representative of the underlying functionalities and physical properties, the characterizing taxonomical description associated with the name may eventually loose importance, perhaps being to cumbersome, and a completely artificial label might be attached [10].

---

[10] As an aside observation, it might be that the original name could be really descriptive and then in the use it has lost the initial association and just became a symbol. The name itself now characterizes the object and a search must be performed in order to recover the original discriminating function. Some of these names are onomatopoetics and have preserved in their label the function associated with them. (It would actually be interesting to see if there is name identifying objects which have the function of which is most characteristic in them, same for smell, touch etc.) So should be some others with functions. Television is one of such instances where the name still defines its function. It was actually coined to denote the function. We do in general have a sense that if there is a function that is being performed by an object then for the name of the object in order to be descriptive, we should include a function connotation in its name. There are other objects which have completely lost from their name the semantics

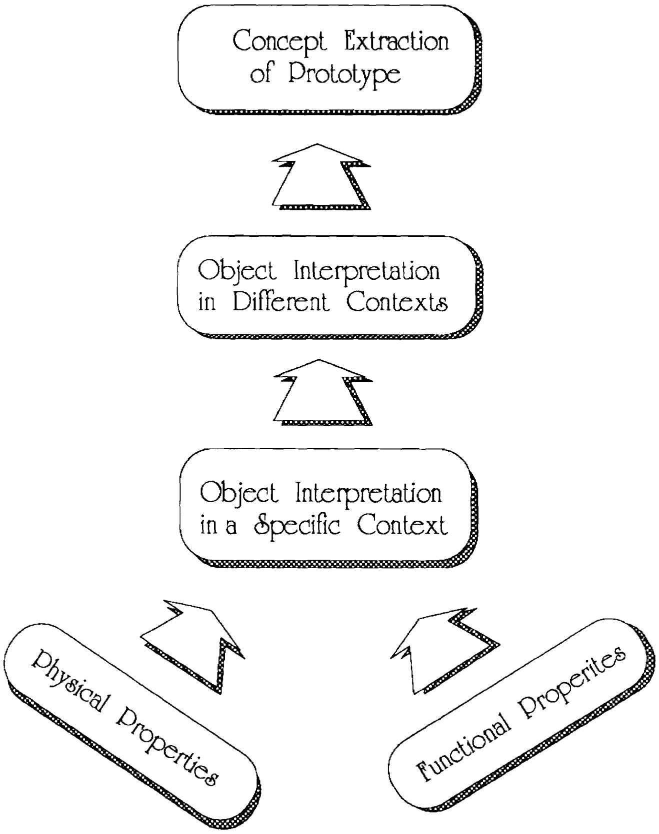

Figure 1: **Object interpretation: from Physical and Functional properties to Concept**

For our purpose, however, the actual label attributed to a given object will not be relevant. The relation with a "dictionary" definition of the object is beyond the discussion developed in this paper.

The process described above can be represented schematically in Figure 1. When considering a tool or and object to be identified, it would be convenient to have some *Feature Association Criteria* which would respond to the features recoverable from the object and determine the corresponding functional attributions. Having obtained the functional properties of the object, or tool, the system could observe the relation of the functional and physical properties of the object and provide and interpretation of the object in the specific context. By recognizing tools in different contexts and observing the variability of the tools in different contexts perhaps extract the prototypical definition of a given tool.

What appears clear in this process of interpretation and what will be the focus of our attention will be the establishing of the association criteria to allow to extract a functional description of the object. We will discuss these later on.

2.4.4 Dealing with a New Object

When we are faced with labeling a "new" object, i.e.. one for which there is no prototypical definition, we observe it and interact with it. Often times the context gives us the interpretation. An object in a hardware store could be some sort of tool, just as an item from the kitchen drawer could be a piece of cutlery. In the case, however, that we were presented with something completely foreign, given without a context, and asked "what is it?" then we would proceed in different directions. Some of the analysis could be done by considering the the material properties and the geometrical components. Analogy and deduction would possibly be additional means used. If, however, the object did not have any resemblance to anything familiar then our description would be based on the underlying shape and physical properties. Biederman, [Biederman, 1987], presents a *do-it-yourself* object. This object does not resemble any familiar object. In his study when people were given pictures of these non-sensical objects. These were immediately singled out as having no description; and, in fact, having only parts description. After a while, analogy and associations lead them to make some temporary association of the object as having characteristics of some class or another even though not being distinctly member of any. Similarly one could extend the comparison to consider objects for which no direct functionality can be associated with. It would be not possible to really qualify it. Thus, an object having no obvious application could be best described in terms of the under-

---

associated with them. This is the case with objects which are addressed by their brand names rather than by their original nomenclature. An example of this is the word "hoover" which identifies the particular brand of vacuum cleaners.

lying geometrical and physical properties. However if it were tested with respect to a function, then one could evaluate its applicability or performance in the given context. Such evaluation would provide the object with functional attributions.

If the object has no obvious application then it could be deemed as an obstacle or a "thing" with some properties.

If there is a sudden surge for the application of such an object with the attributions recognized then the applicability will be what characterizes the new object. In fact, a name might be coined for the the widget in question.

2.5 Procedural and Declarative Components in Object Description

We conclude this section by addressing that both components[11] in an object description *must* be recovered. The description of an object, used as a tool, should not only have a declarative but also a procedural component. The declarative component characterized the geometrical and material properties and how they are structurally related to represent the functionality. The procedural component identifies how the object is to be applied, the type of actuator intended, and possibly the contexts of applicatibility.

Declarative knowledge encompass properties about the object, both material and geometrical. The procedural component tells us how these properties receive meaning in the context of a functional application of the object. The properties concurring in an object description, and in particular its functionality, greatly depends on the underlying criteria used to order and group such properties. In our case, we use the performatory aspect of the recovery process as a means of establishing the criteria under which the properties are to be organized. Thus, as pointed out in [Bajcsy *et al.*, 1991], it is necessary to have both performatory tests as well as perceptual tests to recover both aspects that are characterized in objects.

2.5.1 Top–down and Bottom–up Recovery

In a pure bottom up approach, while the system might be able to recover geometrical properties, [Gupta, 1991], and material properties, [Sinha, 1992; Campos, 1992], the underlying criteria which allow us to group the properties into functional properties are not bottom–up recoverable. The reason is that, as we have seen, the contexts and manner of application lead to a structure which yields the semantics of the grouping of properties defining a given functionality. In a sense we can perceive the bottom–up portion as the process which allows speech recognition processes to identify the different phonemes and group them into individual words. Just recovering the

---

[11] The presence of both Procedural and Declarative Knowledge in object description has been a common tenet of AI for quite some time now.

individual separate words is insufficient to gain any understanding of their function in a sentence. Language structures provide for both syntactic and semantic validity. In our case the top–down component must provide for a structured methodology for recovering the procedural semantics of the object, having recovered the material and geometrical properties.

Thus we require some knowledge a priori. To assume none would imply:

- We are describing the whole learning process from beginning of time. *Trial and error can hardly be considered a methodology especially if nothing is known about what the desired goal is.*
- We are not taking advantage of some of the knowledge already available. If the recognition process is to be able to proceed and to classify interactions, it must be able to recall previous interactions results.
- The recognition process is fully data–driven and no rules on interpretation are present. This is not promising at all when considering functionality and the context in which an object is used.

The methods described in the previous section were based on matching recovered properties to a prototype. Defining such a prototype provides for a frame of reference in which to match the recovered properties. It also eliminates the issue of establishing the relevance of a property vis–a–vis its contribution in capturing some aspect of functionality. Namely, requiring that the prototype for hammer must incorporate hardness, rigidity, prehensibility, etc. within some range of values, requires for the system to be able to observe such parameters. It can safely ignore other properties which are only accidental in the instance being examined.

Attempting to describe a general prototype capable of capturing every possible instantiation is however not possible. Considering all the possible contexts in which a given function may be applicable allows us to recover a prototype which may be too generic to be of any practical purpose. Therefore, a prototype would need to be interpreted into a context so that it may be possible to evaluate the functional attribution of an object in that context.

The alternative to this prototyping is that of learning the relevance of given properties in a functional context. Namely, instead of knowing that hardness is important for a hammer, precede by trying to apply an object as a hammer having the expectation of the interaction. By observing the effects vis–a–vis the initial task expectations, one can determine the relevance of a property in the performance of the operation in a particular context.

2.5.2 Association Criteria

The process of associating properties and attributes with a given object with the goal of identification is based on a set of association criteria. The actual selection of features which should

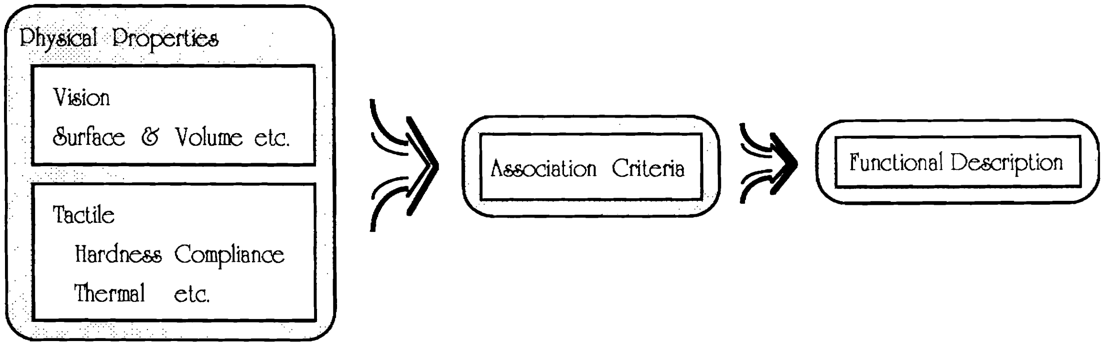

Figure 2: **Association Criteria**: *"what goes with what?"*

be grouped by these criteria and which are sufficient and necessary to represent the object constitutes a serious dilemma.

An approach based on recognizing and clustering of particular properties that can be considered to characterize a function must possess criteria for establishing such grouping, see Figure 2. In a prototype approach, the underlying criteria which describe the set of properties to be matched can be either implicit or explicit. In the first case, we would be interested in just a set of properties occurring; in the latter one would consider the relations that these properties exhibit.

This process would suggest that if it were possible to cleanly identify both the functional and physical properties of an object then we would be able to fully characterize it. While this approach is quite viable in an extremely limited domain, it becomes extremely impractical when the description of all the features can not be cleanly recovered, suggest contradictory interpretation, or the environment is not too well structured.

Some of the problems are due to the fact that the clustered physical properties are interpreted in a limited domain. However, one could argue that a rock, for instance, could also be a hammer as well as a platform for support. This consideration suggests that given objects can have an extremely large set of functions associated with them.

The problems of acquiring the properties can be grouped into two categories: the *A priori Knowledge* and the *Data Acquisition* problems.

The A priori Knowledge problem addresses the question of what knowledge of the object is required in terms of recognizing it. After all we are looking for specific features which should be present. The Data Acquisition problem considers the quality of the data that is acquired and how well it fits into a given model(s).

The process of property recognition, as stated so far, appears to be quite limited and does not incorporate the philosophy of active recognition. It is, therefore, necessary to redefine the process shown in Figure 2 to incorporate such interactive component. As we have seen, such component is an essential requirement since while functionality can be associated with some material and geometrical properties, it is performatory in nature and its interpretation can fully be evaluated in a domain which comprises actions.

2.5.3 Classification and Recognition

We have given several instances about object recognition and classification. The interplay between recognition and classification further exemplify the relation between the performatory tests and the perceptual tests, also known as informatory tests.

We distinguish two levels of tasks in relation to their goals.

- The first one has an action whose primary goal is not that of entirely classifying a given object, but that of *recognizing* whether a given object possesses the suitable requirements, i.e. a given functionality, to be employed for the task at hand.
- The second one has *classification* as major goal. Here physical as well as functional properties are used as parameters in identifying a given object. Such identification is a very complex procedure. This differs from the preceding instance where we simply look as to whether a given object has a well–defined property or function. We now focus on testing whether the object under observation meets the criteria for a given functionality.

This depends on our knowledge of the object, the environment and the action which we are performing. The approach could be directly that of classification if the functional attribution of the object were fully known and if we were able to recognize the unique set of features which characterize it. While it might be clear the given set of features characterizing the physical and geometrical properties, it is not clear that such set can uniquely describe all objects having given the desired functional properties in all contexts. Namely, while there is a finite set of features which we can consider being fundamental to the description of a given functionality, once such functionality is described with respect to a particular context the initial set of features may be insufficient. This observation would lead to consider that perhaps it might be quite impossible to describe all the features to satisfy the functionality requirements with respect to all contexts. Thus, while we might not have the ability to recover all the descriptors a priori that define a particular functionality, we might be able to supply part of the context specific additional constraints which would make the recovery possible.

Further investigation is needed to examine more in detail the relationship between classification and recognition of objects and how to classify functional properties. In particular, we would like to see how the interactive experiences can be used to extract the invariants across different contexts of interaction. Smith, in [Smith, 1990], investigates how concepts and inductions are related. Rosch, in [Rosch, 1973], addresses the relation between the internal structure of perception and the semantic categories which can be formed. The process of investigating several contexts and the use of induction will allow us to try to identify which are the invariants describing the semantics of a given functionality.

3 Other Approaches for Recovering Functionality

Up to this point we have discussed the importance of functionality in the representation and interpretation process, yet we need to address a methodology for recovering functionality in an object or tool. Before doing that, however, let us examine a few of the approaches taken dealing with functionality and compare our approach to them.

The approaches taken recognize the importance of relating the function in one part in the object to some other. It is based on being able to recognize the structural and functional purpose of the components. The importance of structural components was first addressed by Brooks in ACRONYM, [Brooks, 1980]. While not all of the following issues are addressed, this list will provide a guideline for the analysis of the effectiveness of the methods discussed:

- Selection of properties which should be associated with a given component.
- How many such properties are sufficient and how many necessary?
- How well do the perceptual properties need to be recognized in order for the the method to be able to recover the underlying functionality?
- Does a probability measure or a goodness of recognition need to be associated with a set of components?
- How well can we handle situation in which more than one functionality is present and the one that we are trying to recover is not the preeminent one?
- How are structural and functional constraints propagated or imposed?

If, on the other hand, we could *actively* investigate the object then we could take advantage of:

- The active recognition approach.
- Verify that the hypothesized functionality is present,
- Do away with some of the many stringent requirements which must be available to provide for feature recognition.
- Address performance issues which in this context are never brought in question.
- Characterize how much of a given functionality must be recoverable and actually be present before we can recognize it.

While the performatory approach does provide means for verifying the hypotheses it is limited by the problems which are inherent in interactions.

- Uncertainty and unreliability occur in the sensors.
- The observation of the interactions is limited by our ability to model the behaviour of the system, by our viewing position, by our knowledge of the environment.

- The specification of the actions which are to be carried out vary, even slightly, in different contexts.
- Properties other than only geometric must be recovered and hence EP's may be required.

These are but a few of the issues which we must be concerned with in an interactive approach. We will address how they can be brough under control in the sections to come.

3.1 Systems Recovering Functionality

Work in the field of functionality was pioneered by Freeman, [Freeman and Newell, 1971]. This work addresses the importance of functionality as ways of "devising artifacts to accomplish goals". While this paper briefly outlines the structural relations of components in a knife, handle and blade, the focus is that of constructing artifacts for a given functionality, rather than trying to understand the functionality of an object. This approach is that which is often at the core of the work done in constructing integrated circuits and alike. This approach addressing functionality can be paralleled to the difference between the approach taken in Rendering by Computer Graphics and in Recognizing by Computer Vision.

The importance of the relation between structure and functionality is not new and appears as a prominent aspect in the description of the ACRONYM system, [Brooks, 1980]. Lowry, [Lowry, 1982], investigates this relation only at a conceptual level. In [Lowry, 1982] the author emphasizes that function must be represented as a hierachy of kinematic primitives, functional primitives and causal networks. Furthermore, the work points out the relevancy of having both qualitative and quantitative reasoning to determine the relationship between structure and function. The issues investigate focus on very basic questions and problems of this important relation. Furthermore, this very relation between structure and functionality is effectively what we are attempting to recover.

We now look systems which have attempted to recover object functionality rather than construct it in them.

3.1.1 Stark and Bowyer 90 and 91

Stark and Bowyer, [Stark and Bowyer, 1990], apply a CAD approach to function recognition in the "chair" domain. This is an interesting approach for it identifies that attention should be given to recognition of relation between parts of an object and the function that such relations describe.

It has appeal to the recognition and classification process for it seems not too concerned with specific geometrical considerations. The approach, however, uses planar descriptions. Although the idea of recognizing the relations between parts is a viable one, the implementation is handicapped in this application to evaluate relations based on planar models. There are, however, many instances which can't be described in terms of planar patches – a chair with a convexity in the seat and a back with several steaks which do allow back support (a dining room chair for instance).

This observation highlights the fundamental difference between an interactive approach, as we propose here, and one which is based on recognized features. The major differences stems from the ties with the modeling schema chosen as well as the consideration that the approach only really furthers the hypothesis but does not allow for its verification. An object is provided and decomposed into planar patches and relations between the parts is then examined. Assuming for now that the perceptual process provides clean objects which are easily decomposable using the particular modeling schema, the problem of the verification is still open. Is the object in question really endowed with the functionality which we hypothesize? Can it really be used in a given task? Clearly if all physical properties were recoverable from the object and were either indexable or modelable then one could claim that the actual performance of the task would add nothing further the knowledge of the functionality of the given object. However, in practice it is extremely complex to describe models, physical and dynamic, which fully account for large classes of objects.

Furthermore, members are categorized, or subcategorized according to the classical approach to categorization, [Smith and Medin, 1981], i.e. as *member* and *non–member*. It would seem ,at first, illogical to approach the problem by considering certain objects "more" *chair* than others. Yet such evaluation provides more insight when objects which are not fully characterizable by a given functionality may be satisfying some of the requirements. A bed, for instance, can be seen as some sort of chair. Where this interpretation is more interesting is when the object in question does not have intended functionality. Namely any object with a sharp edge could be used as a screwdriver: "How many times have some of us used a dime as a screwdriver ?" Yet when presented with a knife, we immediately decide that the dime would probably be less applicable than the knife might be. So, given a set of objects, the issue of performance evaluation becomes a necessary one, expecially when we are given the option of using different ones.

In the approach presented, a distance from the prototype would have to be introduced. Furthermore, the description of the prototype presented here does not include or address issues of material properties of the object.

In [Stark and Bowyer, 1991] the approach has been extended to consider parameterized geometric models as investigated by Brooks in the ACRONYM system, [Brooks, 1980] .

3.1.2 Brady 83

While the work presented in [Brady, 1983] does not specifically present a function recovery system it sets the stage for the work which was developed in papers discussed below. In particular, it points out some key questions which need to be addressed when recovering a functionality.

The emphasis is set on the reasoning which connects perception to action. In particular the type of things that a robot is required to know and how it is to use its knowledge to perform interactions with the environment. Knowledge of concern addresses geometry, forces, process, space and shape. Its domain involves the usage of tools.

In particular, the authors examine the functional description of objects which either do not greatly differ or which are extremely specific in their application. The concern is the interplay between the use or recognition of a tool and constraints on the use of tools. In one instance, shape variations, 10 kind of hammers, is considered.

When dealing with particular domains and extremely specialized tools, the initial guess may actually lead astray. A tenet proposed here characterizes the strong relation of tools shapes and the functions which they are meant to describe. *"Tools have the shape they do in order to solve some problem which is difficult or impossible to solve with more generally useful forms"*.

In one of the experiment reported the initial hypotheses suggest that the tool should have been some sort of crank. A closer inspection by the vision system revealed that actually, where there should have been a socket to be used to insert the crank, there were actually blades which indicated a screwdriver. The initial hypothesis was actually wrong. In this approach the discriminating features were available to the vision system and hence disambiguation could be performed. This consideration further emphasizes the need for a verification process both to corroborate the initial hypotheses and to fill in information which might otherwise not be available.

They also investigate the relation between certain shapes and the functions that such shapes are meant to absolve. They address the asymmetry in wrenches. They ask very fundamental questions about the relation between shapes and applications. Such investigation, however, involves a high level of cognition that is important for relating aspects of physical laws applications to functional requirements which are desired in tools.

3.1.3 Brady et al. 85

In [Brady *et al.*, 1985] the authors developed a system, Mechanic's Mate, intended to assist a handyman in the generic construction and assembly work. While the focus is that of studying the interplay between planning and reasoning, they introduce a number of higher order structures and assign functional significance to them. Tools form a family tree. One can then choose the right tool for the job by treating this tree as a family tree. Plans and Tools form a family tree in which new plans and tools are derived from old ones by being patched.

The goals inverstigated are: (1) planning and reasoning (2) geometric representation of tools (3) qualitative and quantitative representations of the dynamics of using tools, fasteners and objects

They note the importance of understanding of geometry in connection with understanding of naive physics of forces and causation. "Only so much force can be applied in mating parts without welding or jamming them. Pushing hard on a surface can damage it but not pushing hard enough can be ineffective for inscribing, polishing, or fettling."

The main emphasis is that of investigating high order geometrical structures and their functional interpretation. High order structures are context independent. However, this requires quite an ability of abstracting the components which define these structures.

They conclude pointing out ways in which a tool may be applied to either extend its applicability or to properly apply it. They note the importance of changing directions of forces, torques and impulses and deriving ways of transmitting forces between parts. The techniques for changing directions are physics based: level and fulcrum, pulleys, cams. To change the ways of transmitting forces particular mechanisms such as links and gears must be used.

This investigation, quite consonant with the previous investigation [Brady, 1983], provides quite a degree of insight but takes place at a high level of abstraction given that no mechanism is given for instantiation.

3.1.4 Connell and Brady 87

In [Connell and Brady, 1987] the authors describe a system which learn two dimensional object shapes by using a substantially modified version of Winston's Analogy program. In it the authors consider the applicability of tools in an *innovative* way.

The representation chosen, semantic nets, is very appropriate for describing structural relations and graycoding is introduced as a way to define a metric between semantic and symbolic description. If, however, the examplars used are not fully representative then there might be gaps in the concept.

The primary interest of the work is that of understanding the interplay between planning and reasoning leading to the understanding of the relation between form and function.

Instead of learning that a certain geometric structure is labeled a hammer, they focus on learning that something which has a graspable portion and a striking surface can be used as a hammer. The two functional concepts are defined geometrically in terms of the shapes representations. The graspable portion should have a description like the hammer handle.

Faced with the hammering task but no hammer the system can try matching the hammer shape description to that of any available tool. A close match suggests that the object may be adapted as a hammer. The program would, then, note that a screwdriver would provide a suitable match.

They taught the system the grasping and the striking requirements and provided positive reinforcement for explicitly telling the program what constitutes a graspable and a striking surface. Thus, a prototype incorporating both the structural and functional definition was available.

They note that "the advantage of providing a functional description of a hammer is that the system can improvise". This freedom must then be controllable by having either a strong description of a prototype which characterized the underlying functional structure of the object or a way in which that applicability can be verified.

They conclude by noting that this approach leads to some means of relating function and form but, as the authors state, ".. Plainly there is much to do."

3.2 Addressing Issues in a Theory of Functionality

In this section we discuss the work of [Davis, 1991] and [Allen, 1984]. Both works investigated focus on very abstract description of actions.

3.2.1 Allen 84

Allen, in [Allen, 1984], investigate a General Theory of Action and Time. The formalism, based on temporal logic, allows to characterize different type of events, processes, actions and properties. The characterization investigated addresses actions and intended actions as expressed in English sentences. Such formalism is then developed into a framework for planning in a dynamic world where there are unexpected events and other agents.

The approach is quite interesting since it points out the necessity of focusing on the temporal aspect and sequencing of actions. The attention is developed toward the description of plans in a changing world. The investigation of the dynamic aspect of the world is performed with respect to what is present in the world at the time of the generation of the plan and how to handle changes in the environment. This investigation is common in the AI community and it is quite important. It is necessary to have a current and consistent view of what is the environment in which an agent is to interact with or move about. In the case that changes occur then it is important to be able to adjust to such changes and to take appropriate steps.

While the concern for a changing environment is rather important such investigation is too general for addressing the problems which we must address. On the other hand, it points out that one must account for the possibility of some unexpected event to take place. In our case, unexpected events are not caused by "mischievous" agent playing tricks but rather by our lack of knowledge or inability to fully model the environment in which the interaction is taking place.

3.2.2 Davis 91

The paper studies how the cutting of one solid object by another can be described in a formal theory. Two alternative first–order representations are discussed. The first one views an object as gradually changing its shape until it is split, at which the original object ceases to exist and two new objects come into existence. The second focuses instead on chunks of material which are part of the overall object. A chunk persists with constant shape until some piece of it is cut away. Davis proved that they are sufficient to support simple common sense inferences algorithms.

The two theories can be differentiated as follows:

- First theory: a target being cut by a blade retains its identity, but changes its shape up to the moment that it falls into pieces. Once it comes into two parts, the original target ceases to exist and each piece becomes a new object. This is mutable object theory. It addresses a point "what constitute a change in an object." If the object is cut, change, if the object is filed down to a small object, the identity of the object remains the same. Three kind of changes are identified: the shape of object is cut away, the object comes into existence, and the object ceases to exist. The dynamics of the theory consist primarily of specifications of the circumstances and extent of the changes.
- Second theory: the "immutable chunk", viewing all three types of changes – creation, destruction, and change of shape – as consequence of a singe type of change: destruction of material.

We note that these theories outline the importance of dealing with change in the object. It reflects our concern with the observability of the interaction. The theories characterize the intermediate states that take place during a cutting process and the geometric relations between the shapes and motions of the blades and targets.

The following limitations are pointed out. A cutting operation works by removing and destroying of the material of the target in the path of the blade (pushing it aside) in reality. It deals only with the kinematics of cutting, relations among the positions and the shapes of the objects involved, not with its dynamics, forces and velocities required for cutting. A dynamic theory of rigid objects useful for commonsense inference is still much an open problem (see [Davis, 1990]).

The major restrictions are, however, that the algorithm calculates the result of cutting given complete knowledge of objects and that the object is not subject to alteration or deformations. If this theory is to be of use in the real world, means must be suggested or provided so that non complete objects representations may be handled.

After having observed all of the above, we note that what [Davis, 1991] is actually doing is presenting a higher level description of a process which has been modeled by mechanical engineers for quite a while, see [Dieter, 1961].

This analysis does, however, reflect the need to establish both a symbolic description of the interaction expressing a functionality and the ability to map it to a context.

3.3 Comments

We have incorporated observations in each of the papers examined in this section but we would like to emphasize some of the stringent assumptions:

- All conclusions regarding the object's functionality are based on the geometrical properties of the object. No consideration is ever bought in regarding the other material properties.
- The comparison is based on a prototype approach match. Too often the prototype definition can not accommodate for variation in parameters, non exact matching and is strictly dependent of the existence of a valid prototype. Such prototype recovery is often quite complex, [Smith, 1990].

  The prototype approach brings in implicitly a context of application in the case of [Brady *et al.*, 1985; Connell and Brady, 1987]. While the introduction of the context as the only possible locus of interpretation, no attempt is made to address how and if the prototype could be applicable to a different context. Maybe a functional mapping of the prototype to different contexts is what should be provided.
- The functionality of the object is implicitly specified in the geometrical structure of the prototype. No assertion is made about the way the object is to be applied. [Brady *et al.*, 1985] mentions the possibility of recognizing a screwdriver for the purpose of hammering. However, it is not clear how much one should relax the initial matching constraints in order to classify the screwdriver as a hammer. Furthermore, no mention is made on how sensitive such constraints would be to missclassification.
- The intended user is not specified. It is important that the user be specified in the way that the tool is going to be applied. The way that a tool should be applied needs to be specified so that then it can be translated in the appropriate user application. While this problem does not occur if no interaction is required, the application of a tool is with respect both to an intended user but also the current user. Such context dependency must be always addressed when interacting with the environment.

The performatory component is missing and the prototype is given so that just matching the geometrical properties is considered sufficient by the authors for the matching. This approach brings about a fundamental question about the approaches: "*can an object be recognized to have a particular function only from geometrical structure?*". A fundamenental belief developed here is that geometrical properties, while necessary are not sufficient to determine the functional behavior of an object.

4 Toward a Formalism for Expressing Functionality

We have investigated functional characterization, in section 2 and then identified the components which should appear in an object description addressing functionality. In section 3, then we looked at other approaches to functionality recovery.

From our current investigation it appears clear that it is necessary to describe a task for the recovery of a particular functionality in a manner that observation and verification can be carried out interactively.

The actions which describe the performatory component of a function must have a very well defined order. They must take into account the time in which this interaction is carried out. They must be able to describe the observation of interactions which can not be fully modeled. Hence a the description of the interaction and the actions themselves must be describable in terms of continous, when full modeling is possible, and discrete states, when only transitions between states of the system are observable.

It is necessary that the description of tasks themselves provide for addressing observability through different sensor modalities. They must also handle an environment in which not all events and the time of their occurrence might be modelable and predictable in advance. For the time being, events denote the occurrence of state–variable changes that are significant enough to warrant a new state in the system. Later on we will extend and clarify the notion of event. Existing formalisms express only some of the required components for describing the type of tasks which we are interested in.

Furthermore, a formalism, in order to preserve its expressiveness and its generality, must not be imbued with domain specific constraints. We present a methodology for instantiating and transforming the abstract task into a specific context. We approach the instantiation problem by describing a process which can progressively introduce constraints and transform and bind high–level description of a task to specific values.

Once constraints have been introduced, the actual task, describing the interaction apt to recover and verify the presence of a particular functionality can be carried out.

The approach which is outlined by the recovery process can be described as a task oriented approach in the sense that the task establishes the aspects which should be investigated.

We begin this section by outlining the importance of having a formalism suitable for task description. Then proceed by relating and identifying components in task's description, instantiation and recovery processes.

The process describing the recovery procedure is made even more complex since, as we have noted, functionality is performatory in nature, recovery process itself can not be cleanly separated from the actual functional description. A process for recoverying a 3D description, on the other hand, can be expressed independently from the object which is trying to recover.

The main thrust of this outline is that pointing out why it is necessary to have a powerful formalism which can express the performatory component both of the function description and of the recovery process.

4.1 The Need for an Appropriate Formalism

We need to be able to observe an interaction being carried out, to handle uncertainty both at sensor–modeling level and at object–modeling level. We would like a formalism to describe the interaction irrespective of the manipulator which is utilized, of the type of object modeling schema chosen, and be capable of integrating different type of sensors employed as observers. Furthermore, we would like it to provide a means to describe how a high–level task description, as provided by a planner, can be instantiated into a specific domain. This requires means of translating symbolic descriptions of spatial relations, temporal descriptions, force and accelerations into actual values which can be operated on in a specific environment.

Since an interactive process is dynamic in nature we begin by looking at ways in which the dynamic component of the interaction should be incorporated into a formalism together with the other aspects listed above.

Interactive processes involving a manipulator, a tool and objects have been in the domain of Robotics and Manufacturing for a long time. The behavior of such systems has been modeled using either linear or non–linear dynamic systems according to Classical Control Theory, [Reid, 1983]. That has been possible because the state of the system when the tasks are carried out could have been predicted and the behavior of the system could have be observed throughout the operation.

If the environment, (object, manipulator, and tool) can be fully modeled we could employ the formalisms provided by Continuous Variable Dynamic Systems Theory (CVDS), as pointed out in [Ho, 1989]. In these systems we could model continuous changes in the state of the system by using differential or partial differential equations.

Uncertainty, external observability, unpredictability of behavior and non–determinism rule out the use CVDS to model the process. Furthermore, if changes occur at discrete periods in time and it is not possible to fully model the process(es) which lead to occurrence of events, then a different approach needs to be employed. In such instances, processes are described in terms of events rather than use continuous functions. This allows issues of observability, non–determinism, and uncertainty with respect to occurrence of these events to be addresses. The approach for describing the behavior of a dynamic system falls into the domain of Discrete Events Dynamic Systems theory.

The reason for the existence of the different formalisms for DEDS is that of ensuring the appropriate behavior of a system by means of appropriate control, as pointed out in [Košecká, 1992]. Properties which are investigated are either qualitative, e.g. *logical DEDS*, and quantitative in nature, e.g. *timed or performance DEDS*. The former may address issues of stability, convergence, correct use of resources, correct event ordering, deadlock prevention, liveliness, etc. and the latter issues of performance.

As there is no unified theory for DEDS, [Ho, 1989], supporting all desirable aspects, we base our description for our formalism on the general notion of DEDs and combine the desired aspects from different areas and accrue some them to fit our goal.

In particular we will consider some aspects from DEDS based on the following [Ramadge and Wonham, 1989], [Sobh, 1991], [Ostroff, 1992], [Cameron and Lin, 1991] and [Košecká, 1992], and adopt the terminology of control theory as a way to express the behavior of our system.

- [Ramadge and Wonham, 1989] introduces the basic theory of DEDS
- [Sobh, 1991] addresses issues dealing with uncertainty in DEDS and presents some examples in which probability measures have been introduced for handling the uncertainty.
- [Cameron and Lin, 1991] focuses on the Real–Time Temporal Logic issues of Events and Actions Systems, (RTTL–E/AS). Thus, bringing into view the necessity of binding time constraints and temporal dependencies into DEDS.
- [Ostroff, 1992] and [Košecká, 1992] survey different type of formalisms for DEDS. The former reviews formalism connected to the design of real time systems while the latter analyzes primarily DEDS both of the type presented by [Ramadge and Wonham, 1989] and in the context of RTTL–E/AS.

If we focus on tasks as action compositions, then we can consider the properties that tasks are meant to express rather than the methodology which is used to compose the actions in order to express a meaningful task. In other words, the formalism which we are proposing presupposes the existence of a high level planner which orders the actions. We will further elaborate this point at the very end of the section.

The properties of tasks, which we are interested in, investigate the presence of functionality in different tools. Thus, while the formalism can be employed to express interactions in a manipulator–tool–object environment, we will restrict our attention to those tasks which investigate functionality.

4.1.1 The Role of the Observer and Origin of Non–Determinism

The observation of the system is characterized in terms of discrete events which might not be modelable at every point in time because there might be no clear description of the environment. In the case that it were so perceputal aliasing may occurr; namely, the observer might not be informed at all times of the state of the system because:

- the interaction may not be observable by a particular sensor, since occlusion may occur;
- the sensor may not be able to measure the values, outside the range of its perception;
- the interaction may take place in an area not currently monitored.

Even in a constraint environment, lab or manufacturing environment, certain aspects are best characterized in terms of events. Observation allows independence in monitoring both for feedback purpose as well as for verification of task being carried out. Changing a task may require different means of observability. In the case that the observer is separate from the plant, no procedural change may be required in the architecture of the plant if an observer is added or removed. Clearly the sequence of transformations which could replace, add, or simply remove an observer must not jeopardize the controllability and observability of the plant.

When more than one event can take place at any instance of within a particular interval in time, non–determinism is introduced. Non–determinism is introduced since

- the environment may not be always fully modelable;
- the knowledge of the interaction may not be complete;
- the events which can take place at any given time may be both controllable and uncontrollable.

An abstract description of an interactive task must be mappable to a specific context. In such context we must be able to monitor the progress of the operation and thus control it. Therefore, to be able to carry out a task we need to develop a formalism which is powerful enough to describe observers, model non-determinism, and handle uncertainty.

4.2 Closing the Interaction and the Interpretation Loops

Having identified what kind of components the formalism should include, we can now proceed to describe how a task is actually expressed in terms of the system in which it is to be performed and the observer(s) which are both to provide feedback and to evaluate the performance of the task.

Having noticed the role that the observer performs and we can distinguish an *internal* and an *external* observer. In the former, the purpose is that of monitoring the state variables and provides feedback control to drive the system. In the latter, while the observation of the state of the system could be used as feedback, its main function is that of providing verification of the hypothesized interpretation.

Thus, we can see that observations close two loops, see Figure 3, one for the interpretation and the other for the interaction.

A task then can be thought as being mapped to the **interaction** structure and the **interpretation** structure. The purpose of the interaction structure is characterized mostly by

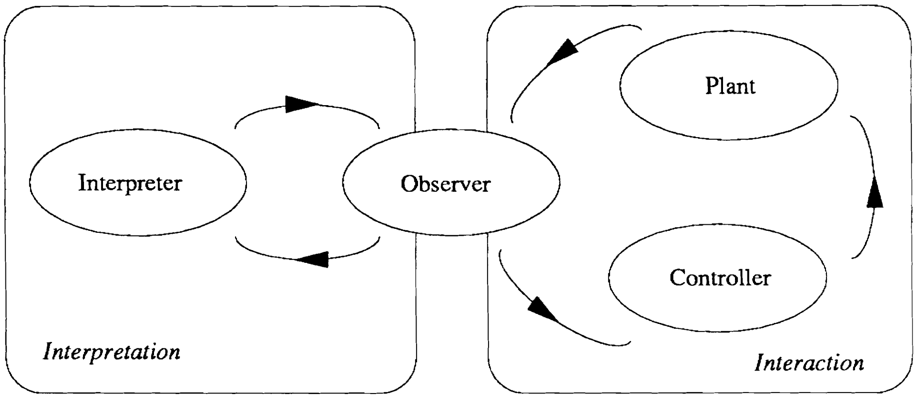

Figure 3: Closing the loop in the Interpretation and in the Interaction.

controlling the behavior of the plant. This structure characterizes the performance of the experiment. The interpretation structure addresses the observation of the behavior of the system with the purpose of verifying the expected results of the interaction.

While the interactive structure might be modeled by a CVDS, we will not propose this approach for the reasons pointed out in the previous section. The interpretation structure, on the other hand, falls clearly under the paradigm of DEDS. [Sobh, 1991] uses the observer to drive the robot's mounted camera to gain a better viewing position. He presents an open–loop architecture. Many other authors have presented a closed–loop architecture but the main purpose was that of controlling the system behavior.

In our case, we consider how, actually, two loops should be employed. It is only by closing the interpretation loop that we can accomplish the verification that the expectation of the interactions can be met.

4.3 Mapping a Task to the Interpretation and the Interaction Structures

Expressing a task both in terms of the interaction and the associated interpretation allows us to further focus on the fundamental difference between the purpose of the two structures.

4.3.1 The Interpretation Structure

The intpretation structure defines the expected results associated with events which are to occur in the system. This process of verification allows the observers to step through a Non–Deterministic Finite Automaton, NDFA, as described by the theory of DEDS. Paths denote the

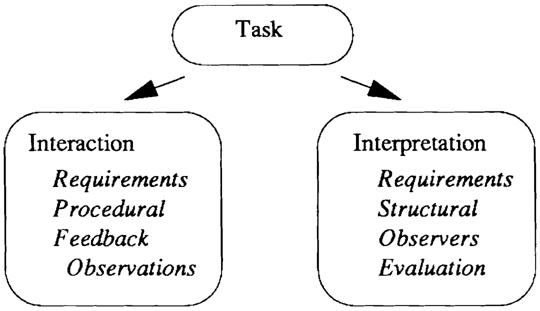

Figure 4: Mapping a Task for testing functionality into an Interpretation and Interaction Structure

development of the task and provide an event by event description of the behavior of the system vis–a–vis the expected behavior. Thus, a task can be thought as of being mapped to a NDFA in which each state has associated the expected state of the system after the event.

Measuring the "distance" between the expected and observed state allows to determine and verify whether the desired behavior of the system is being pursued. Considering the path from the initial to the ending state allows the interpretation and verification of the validity of the initial hypothesis and also attribution of both a qualitative and quantitative measure to the interaction observed.

We can distinguish the following components:

req the REQUIREMENT describes the initial state of the system in terms of expectations.

str the STRUCTURAL defines a net of events, the NDFA, describing the expected flow of the task.

obs the OBSERVERS identifies a set of sensors which can be mapped to the transition in the net. The purpose of these sensors is that of observing the particular events. The mapping function which associates particular sensors to specific transitions in the net is what characterizes the task description in the interpretation structure.

evl the EVALUATION allows for each node in the transition to measure the quality of the match between the observed and the expect state of the system after said transition.

4.3.2 The Interaction Structure

The interaction structure must address the behavior of the system and its controllability. The observations provide feedback information to the controller that can then guide the behavior of the system. We require the following components:

req - the REQUIREMENT for the action to be performed and the assumptions under which the operation is valid.

prc - the PROCEDURAL prescribes a sequence of actions which must be carried out in order to fulfill the operation. It must encapsulate rules and conditions governing the sequencing of the routines present. These rules may allow selection between alternative paths present in the procedural components. Evaluation of said conditions depend of the sensory feedback. In fact, task termination itself is contingent on the realization that conditions can or can not be satisfied.

fob - the FEEDBACK OBSERVABLES identify a set of functions which are used to monitor the events which mark the state transitions. These functions are associated with different steps in the procedural network describing the transitions apt to perform the task.

As we shall see, the events which characterize these transitions are of various origins and types. We will identify events describing: changes in state variables, logical conditions asserted and guarded events (addressing safety conditions of the system).[12]

4.4 The Instantiation Process

Prior to describing the instantiation process we introduce the domains in which the task has to be instantiated to. We identify the following

- Tools/Instruments T: to be the domain of all the objects which are to be used by one of the actuators. Its members, t_i, can be defined in terms of geometrical and physical properties.
- Target Object/Recipient O: to be the domain of all action recipient objects operated on by one of the members of the actuator domain. As in the previous case, its components, o_j, are defined in terms of geometrical and material properties.[13]
- Actuator/Manipulator M: to be the domain of all manipulators which can perform some operation on an object with or without the aid of an instrument. Each of the elements, m_k, in this collection has very well defined characteristics and constraints which are very important when addressing an interaction. We will suggest that each of them should have
  - types of grasps that it provides, i.e. pinch, spherical, hook, etc.
  - relation of grasps and applications of the grasps, i.e., pinch grasp may be used for some minute type of manipulation, etc.

---

[12] These are also known as controlling functions in Control Theory literature. They prevent the system from entering an unsafe or uncontrollable state.

[13] The quality, type of properties recovered depend on the underlying model describing the objects. At this point of the discussion we will not address the source for these properties nor the choice of the model. We will return on this topic at a later stage.

- – ranges of forces and pressures which it can apply.
- – constraints on operability associated with specific manipulations, workspace of operation.

Tasks will then be considered as operations on these domains. In particular we will consider them defined in terms of the following type of triples; (ϵ identifies a missing component).

1. $(\mathsf{M}\times\mathsf{T}\times\mathsf{O})$. This would characterize a situation in which a gripper is applying a tool to a target.
2. $(\mathsf{M}\times\epsilon\times\mathsf{O})$. In this instance the system in question would characterize interactions focussed on determining whether the target may be grasped or lifted.
3. $(\mathsf{M}\times\mathsf{T}\times\epsilon)$. This system would characterize the operation of carrying an tool to a given destination. In a certain sense this is analogous to the previous one.

While we could also denote other type of triplets with more than one component missing, these would either define instances which do not represent an interaction or simply certain stages in the interaction in which we would be focusing only one of the components.

In order to map a task to the interested element in a particular domain[14], there must be models associated with these elements.

4.4.1 Emphasizing Different Components

We take a moment to observe the effect of shifting emphasis on one of the domains and allowing different parameters to vary. Focussing on the individual components in $(\mathsf{M}\times\mathsf{T}\times\mathsf{O})$ allows us to describe the following types of problems:

1. **Target** Problem: addresses the interpretation of an object as it is operated on performing different functionality tests.
   - *fixed* instrument[15] and actuator(s).
   - *observed* the effect of an action on an object.
2. **Instrument** Problem: identifies the most appropriate tool to perform a given function.
   - *fixed*: the manipulator, and target object, we will allow this to be composed by different materials.
   - *observed*: tool behavior with respect to different functionalities tested.

---

[14] In this case, an element will take different meaning based on the domain it pertains to. Namely an element from the object domain will be an object, one from the manipulator domain will represent an instance a class of manipulator, from the class of two fingered manipulators, and so on.

[15] As it was noticed the instrument may or may not be present.

3. **Classification** Problem: classify a tool with respect to different functionalities, N, and with respect with a set of target objects, M (N and M are some positive integer).
    - *instrument*: Solve prob. 2 for N functions.
    - *recipient*: Solve prob. 1 with M instruments.

In the target problem definition the tool is assumed not to be subjected to any deformations or alterations due to the interaction. While this may not always be the case, the statement of the problem in this fashion allows us to considered the effects on the tool as separate problems. Campos, [Campos, 1992], addressed some of the issues in the target problem problem identified above. The one we will focus on in the recovery process is actually the second one.

4.4.2 Instantiating Components

Instantiating a task into a specific context brings in domain knowledge which was not available at the abstract level in which the task is initially defined. In order to do this, the process must provide for

- **Selection** of the models in which to map the task,
- **Represent** these elements into the chosen models,
- **Introduce Constraints** from within a model and propagate them across different models.
- **Bind** the values and ranges of specific attributes so that the hypotheses may be generated.

Selection and Representation of the appropriate models is a decisive step, for it directly influences the type of actions which can be described and the type of information which can be abstracted from the models.[Ikeuchi and Hebert, 1990] attacks the issues by emphasizing the paramount importance of this step and that the task determines the adequacy of the representation. Thus, while we will not investigate of the relation between the different choices of the modeling schemas for similar tasks, we will assume that we can chose an adequate, not necessarily optimal, representation. This enables us to press on and address the other important aspects in the interaction.

Introducing Constraints can be accomplished by bringing in domain and model knowledge. If these constraints were really limiting then an operation such as grasping an object might have a well-defined set of parameters to carry it out. That would require that prehensions constraints relating grasping to object, object model, and object application be very well defined. This would require almost an exact model of the object and of the tool. This instance would restrict the problem of recovering the functionality on an object to the verification and observation stage. While it might be convenient to initiate the process by progressively relaxing the assumptions, at this stage, we notice that in the general case there might be more than one grasping options, more than one possible way to apply the tool to the object, more than one possible effect depending on the material composing the object. Thus, on the one hand there could be a well defined description and optimal representation of the components involved in the interaction; on the other hand, there could be several areas which need to be constrained for the investigation to become feasible.

In our investigation, we will progressively remove some of the assumptions. Some of the assumptions we introduce are actually based on earlier work, [Campos, 1992] and [Sinha, 1992], in which, by applying Experimental Procedures (EP), the material composing the object and some of its physical properties could be identified. Thus, assuming that these properties can be recovered, we will focus on progressively reducing assumptions relating to the grasping and application of the tool. As we have pointed out, with different possibilities for grasping and application, it will be necessary to generate, rank and test hypotheses.

Binding the Values and Ranges of Values for a specific context addresses the combination of values which originates from different levels. There are values which are recovered bottom up and others which are described at a high level but which are quantized and take significance once they are mapped into a specific context.

The bottom up group includes those values that describe parameters obtained by the procedures interpreting objects according to the recovery process associated with a modeling schema. Parameters describing the volumetric extent of an object, as recovered by superquadric, or material properties as recovered by EP's are examples of these.

The high–level group identifies symbolic and qualitative descriptions of quantities that enable the description of a task at a high level but which become defined when they are considered in a specified context. High level descriptions of a task such as "Move a tool near an object very slowly and then decrease the speed until coming to contact," need to be converted to actual measures. As we shall see, these values can not be absolute, but are dependent on the context in which the operation is carried out. The distance identifying nearness in two cars is quite different than the distance describing the nearness of two pencils.

4.4.3 Overview of the Instantiating Process

The components identified above characterize stages which lead to the instantiation of a task. We can describe this process (see Figure 5) by describing how a task is progressively instantiated. Initially it is mapped to the manipulator domain so that the constraints from the manipulator are defined. In the domain, an element a class of manipulator is selected. Subsequently the object is characterized in the object domain and modeled using some representation. Clearly, there are

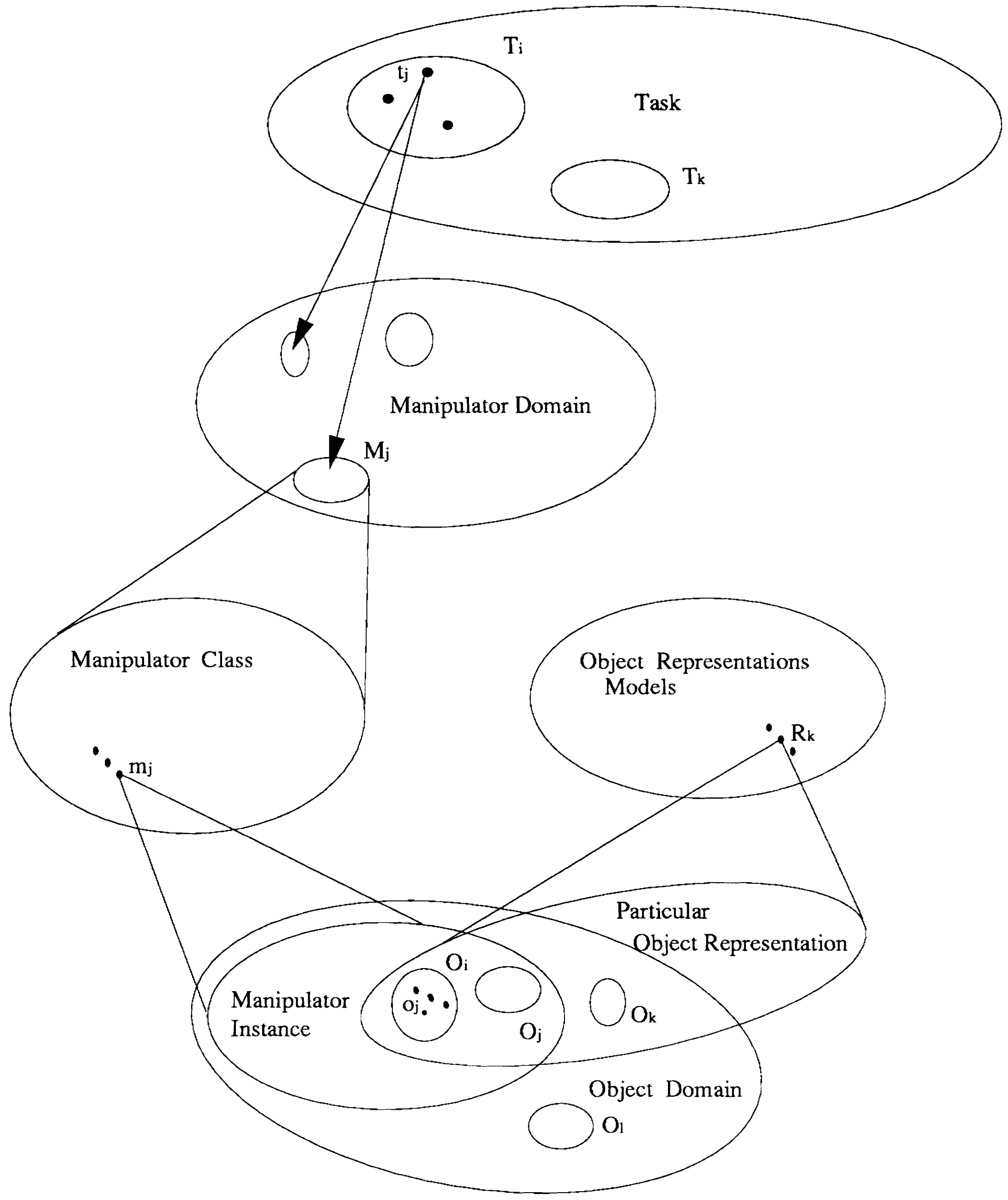

Figure 5: Instantiation Process: task t_j in task class of *Manipulatory tasks* T_i in the Task High Level description is mapped to the specific context. The tasks in the class T_i identify tasks that recover functionality. The context is defined in terms of a particular manipulator m_j in manipulator class M_j, an object o_j in object class O_i represented according to the modeling representation R_k.

elements in the object domain which can not be accessed by the manipulator, perhaps because of size or weight constraints, so that the manipulator choice restricts the set of object which can be operated on. The manipulator instance bring with itself constraining characteristics and domain knowledge which will be employed and which will allow to determine feasible, and then possibly rank, different grasps. Once the modeling schema for the object and the tool, the manipulator instance, with its constraints, rules and operational procedures, are defined, then the task becomes well defined and it can be carried out. Thus the next step will be that of generating the hypotheses to proceed with the recovery.

4.5 Interactive Recovery

This constitutes the last step in the process for the recovery of a functionality in an object. The interplay between the hypotheses generation, as described in the previous section, and the carrying our of a task will enable us to investigate the presence of a given functionality.

By considering a given hypothesis, from the generated hypotheses set, we have now constrained the recovery task to an instance in which the interaction does have a specific meaning and can be evaluated. It is in the generation of the hypothesis, the carrying out of the interaction, and the observation of the interaction with the purpose of verifying the hypotheses which will constitute the focus of the research.

We will discuss this last step more in detail after having presented the formalism and the instantiating process to generate the hypotheses.

We conclude this section by addressing the impact that this approach has on high level task description. As we have pointed out, we will not focus directly topics in Planning and Common Sense Reasoning, much research has been devoted to address issues regarding plans' *goal satisfaction, validity, ordering of subactions*, etc. If task descriptions can be mapped into the formalism presented here, then the instantiation process can provide a way for mapping from symbolic level descriptions into specific instances. Thus values, identifying location, velocities, forces, etc., traditionally described in terms of attributions and qualitatively, can, in this manner, be bound to real world data.

The discussion presented in this section suggests that the recovery process must incorporate the possibility of observing the interaction and verifying that as the task is completed, the expected result of the interaction can be evaluated. Thus, plans that describe interactions should also incorporate and address *observers* with different sensor modalities, a *verification processes*, a *description and evaluation procedures for expectations*, and *uncertainty* both at the sensor level and at model description level.

5 Description of a Manipulatory Task

In this section we present a formalism for expressing a manipulation task and a methodology for enriching the procedural description. In general, one would like to be able to associate learned or a priori information with the performance of a task. It is, in a sense, the baggage of knowledge which is associated with a task, experience, which make us able to decide on the procedural approach we should take. The tasks which we aim to describe provide a basis for an interaction of a particular type. As we will point out, such basis can be extended either by learning or by a priori knowledge.

In the context of our discussion, when referring to a task, we will always be talking about a task expressing interaction for the recovery of a functionality.

Initially a task, such as hitting or cutting, will be described by very basic actions and their respective events. In the action description, we would not want to incorporate any knowledge about an object or a procedural aspect which could affect the generality of the task description. Yet we might want to be able to incorporate learned or a priori procedural specializations. To do this, we associate a situation set with any task. The elements in this adjoined set allow us to incorporate specializations such as object characteristics, associated procedural relations, and expectations for the interaction.

Upon instantiation, the situation set would provide the a priori knowledge, either learned or given. Clearly, the applicability may not be immediate, some initial interaction may be required, or it might not be applicable at all. The association of the basic task and the situation set allows the task description to maintain its level of abstraction and at the same tag to it useful procedural knowledge. For now we will focus on the description of the structure of a basic operational task with an empty situation set. We will investigate the introduction of a priori data from the situation set when addressing the generation of hypotheses.

In the first half of the section we describe the structure of a basic task; in the second half, we will present how the basic tasks, labeled primitive, can be built bottom–up to describe an algebra of tasks.

Any task T can be described as a sequence of actions or subtasks. Any subsequence of these actions may also represent a task; in this case then task T is composed of subtasks. Any of the actions composing a task describe some change in the state of the system. If the task is not composed of subtasks but just of actions, then it will constitute what we will term a *primitive* task; otherwise it will be termed a *complex* task. Recognizing this hierarchy is important, since it allows us to investigate how the more complex tasks are describable in a structured form from primitive tasks; we will discuss this more in the last portion of this section.

5.1 States and Transitions

A **system configuration** is described in terms of system variables. These characterize positions and orientations of the components from the three domains (M, T, O). While there is a continuum of these configurations, their sequence ordering are expressed by actions. Any of these actions, then, identify a path in the system configuration continuum. Thus, an action, in a dynamic system, can be described as a path in the configuration space both as a function of the forces and of the velocities which relate the instantaneous configurations visited along the path.

While an overall path describes a complete action, it may be partitioned into subparts defining phases into which the action can be specified. The initial and final configurations in any such phases are labeled as significant configurations. A **significant configuration** describes, in terms of system variables, how the system should have changed within any given phase of an action. We will designate any of these phase to represent a **state**. Thus, a task can be described as the system evolution through different states[16].

The preceding discussion essentially describes the discritization of a continuous process. In our approach, however, as we have seen in the previous section, we will retain the concept of a state. The abstract description provided will not define a path between two significant configurations, rather exhibit *how* such a path may be constructed. This approach may be compared to a path specification in N–dimensions with could be given as a function traversing a sequence of points or just the sequence of the required points a function is supposed to go through. Clearly the latter is less constrained and suggests that the choice of the function to traverse the points greatly depend of the available function descriptions. Such a situation, however, would provide too many unconstrained parameters. In our case the procedural component of a task provides a guide on how to modulate the system variables between two successive significant configurations.

The description of the sequence of states can not be portrayed simply as a one–dimensional linear order. The dynamic systems we are interested in modeling may allow more than two significant configurations, initial and final, to be associated with any given state.

Depending on the values of the state variables associated with a given state the development of the dynamics of the system may greatly vary. Once the system, in some state, s_i, reaches a significant configuration, it is immediately ushered into a different state, s_j[17]. This instan-

---

[16] In CVDS the *state* **of** *the system* at a time t represents a summary of all the preceding information which would allow one to either to reconstruct the behavior of the system up to and including time t or to predict the future behavior of the system. In DEDS, we talk about *states* **in** *the system* describing different stages of the system. Having pointed that out, we will in general refer to a *state* in the DEDS sense or otherwise explicitly state that we are addressing the over all behavior vis–a–vis a task and hence the behavior *of* the system.

[17] We note that it could be possible for $s_j = s_i$.

taneous ushering from one state into another defines what in DEDS is generally known as an **event**.

The definition of what constitutes an event varies according to which DEDS formalism one embraces. The semantics of an event has so far only been associated with changes in the state of the system; however, we would like to incorporate both logical assertions as well as a way to monitor for values which "guard" values denoting safety or boundary range values. An event may then be properly characterized as being one of the following sets:

- A *change in the state variable*, in which the value of the one more more variables defining the event has reached the value established in one of the final significant configurations for the given state. Δ defines the set of such events.
- The *assertion of a logical expression*, possibly denoting no change in some state variables. We let Λ denote such a set of events describing assertions of logical expressions.
- The *reaching of a guarded value*, for one or more state variables. Let Γ denote the set of events describing guards to values that denote boundary values in a range of variability of a parameter[18].

While this last type of event is actually a special case of the first type, we would like to distinguish it since it expresses a particular type of condition. This distinction is made explicit in real-time system in which the controller for the plant must guard the state variables and prevent them from reaching values which could be critical for the safety and stability of the system behavior.

5.2 The Model

We now describe how an action can be expressed in terms of a NDFA. The notation is adopted from Automata Theory and the Logical Model for DEDS presented in [Ramadge and Wonham, 1989].

The set of the labels of the events is given by

$$\Sigma = \Delta \cup \Lambda \cup \Gamma$$

then Σ^+ defines the set of all finite strings of elements from the set Σ. A string $s = \sigma_1, \sigma_2, \ldots, \sigma_k$ from Σ^+ describes a sequence of events. Such string represents a partial path, partial because it may actually appear as a prefix to some other string from Σ^+. The set of all the strings from Σ^+ may describe paths which do not reflect physically possible sequences of events. Thus we will be interested in describing only on the **admissible** subset of strings from Σ^+ describing

---

[18] These values are often use to guard against maximum force or velocity which could cause structural damage to the robot.

feasible sequences of events. In Automata Theory this subset is referred to as a language L over the alphabet Σ.

In our case we can describe a recognizer M_{t_i} which will accepts the strings from L if such strings describe a sequence of events denoting a task, t_i. In particular we will say that M_{t_i} characterizes the task's procedural description. This approach will allow us, in the next section, to compose some of the recognizers to perform more complex tasks [19].

M, can be described as a NDFA consisting of a set of states, Q, an initial state, q_0, a transition function $\delta : \Sigma \times Q \rightarrow Q$, and a set of final states, F. The set $\Sigma(q_i)$ designates the collection of events, marking outgoing transitions, which are associated with state q_i. The set $\Sigma(q_i)$ is defined as

$$\Sigma(q_i) = \Delta(q_i) \cup \Lambda(q_i) \cup \Gamma(q_i)$$

These transitions occur instantaneously, as stated above, and spontaneously, basically depending on when one of the events is asserted.

While there is no physical meaning for an empty event, ϵ, we will include to our original set to make our model description proper. Now we can define $\Sigma^* = \Sigma^+ \cup \{\epsilon\}$.

The transition function was described over only one event. We can extend it to operate on strings of events, after all, only few tasks can be described by a single event string. The extended definition for δ defines a partial function on $\Sigma^* \times Q$. This can be accomplished by defining

$$\delta(\epsilon, q) = q; \qquad \delta(\omega\sigma, q) = \delta(\sigma, \delta(\omega, q))$$

In this way we can address the behavior of the system relative to a sequence of events. We can thus talk about some language L_T describing interactive tasks and the recognizers $M_{t_i}(L_T)$, where t_i is a task description from T[20].

So far, we have just proposed a relatively standard description of an Automaton to model a task description. This description, however, must be augmented to address whether a given sensor is both capable and actually delivers a given type of observation.

We then define $\xi(q_i, \sigma_j)$ to designate the expected values of the state variables in state q_i which are associated with the description of event σ_j. Where $\sigma_j \in \Sigma(q_i)$, i.e. it is one of the events which may occur when in state q_i.

The set $\xi(q_0, \epsilon)$ define the state of the system at state q_0 and since q_0 is the initial state and ϵ represents no transition, we then say that $\xi(q_0, \epsilon)$ identifies the initial configuration of the system. Then $\xi(q_i, \epsilon)$ describes the value of the state variables upon entering state q_i.

---

[19] For now we will not address issues of observability or uncertainty, we will return to include this topic after the discussion on observability in section 5.3.

[20] The notation we have adopted above and some of the definitions were presented in [Ramadge and Wonham, 1989].

Furthermore, let $\pi(q_i, \sigma_j)$ define the set of state variables which are associated with event σ_j. We can then define an evaluation of the values associated with a given state. The following expresses the evaluation restricted to only the variables describing one of the events which may occur within a specific state.

$$\mathcal{E}(q_i, \sigma_j) = \xi(q_i, \epsilon)/\pi(q_i, \sigma_j) - \xi(q_i, \sigma_j)$$

We can extend the above function to apply to all the events in a state but that in general we are interested in monitoring the changes associated with a particular event.

This evaluation of the observations allows us to note whether a given event will take place. We will discuss the observables in the sections to follow.

5.3 Observability of the Task

So far we have assumed that all events which describe a task may be observable at all times. Observability, however, is contingent on the availability of means of monitoring the different events.

If we consider some task t_i recognized by M_{t_i}, then the strings from W, defined as

$$W = \{w_i \mid w \in \Sigma^* \ni w \in M_{t_i}(L_T)\}$$

should be observable. To state that some task is observable, we need to investigate upon the observability of the strings from W. If we examine, for instance, one such string w_i as composed of events $a_1a_2a_3$, then we can address the observability of the individual events and account about the observability of all string w_i.

In Figure 6 we portray an instance in which some of the events from a string from Σ^* are not observable. Then we can define a projection function which maps events from Σ to the individual sensors. Let $\mathcal{S}$ describe the set of the sensors, S_j, available. Then an event σ_i from Σ can be mapped to some event $e_{ji} \in S_j$ if the given event may be observable by the sensor in question and to the Φ otherwise. This can be stated as

$$P(\sigma_i, S_j) = \begin{cases} e_{ji} & \text{if event } \sigma_i \text{ is observable by sensor } S_j \\ \Phi & \text{o.w.} \end{cases}$$

We have changed the label for the event, upon mapping it to a specific sensor, both to reflect the index association, but more importantly to reflect that the observation provided varies from sensor to sensor.

We can notice in figure 6, that by mapping the individual events in w_i, the first event, a_1, gets mapped in both cases, exhibited above, to the Φ. Unless there is a mapping of a_i, such that the event is observable by one of the sensors $S_j \in \mathcal{S}$, only a portion of the string w_i would be observable.

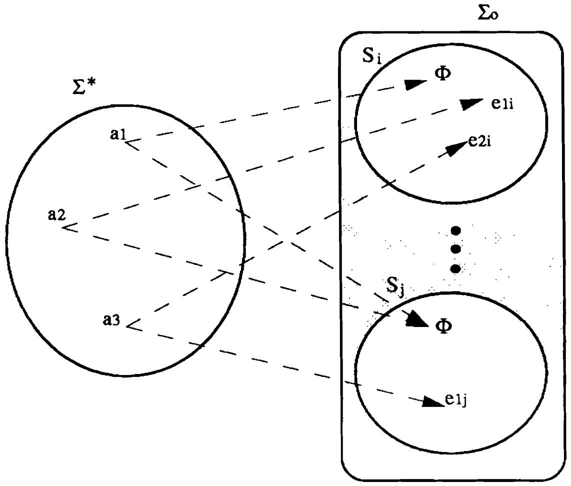

Figure 6: Observability of a String of Events, $w_i = a_1a_2a_3$: Σ^* identifies the set of all strings; Σ_o the set of observable string. Each of the S_i within Σ_o describes a different sensor to which a given event, $a_i \in \Sigma^*$ is mapped to. The Φ in each of the S_i characterizes those events from Σ^* which are not observable by the particular sensor. Effectively the only actions from Σ^* we are interested are those which do not map to Φ in _all_ of the S_i's. (map to at least one S_i)

We would like to ponder on the implications that the above entails. In particular address the following issues: *Full*, *Partial*, and *Piecewise* observability.

5.3.1 Full Observability

It can be stated as follows. Let W, defined as above, describe all the possible strings of events accepted by a recognizer M_{t_i}, which describes the procedural behavior of task t_i. Then task t_i is observable if all of the events in strings from W are observable.

We can express this condition by considering the following. Let $\mathcal{V}$ define an indicator function which will assign a 1 if the projection of some event σ_i in string w_k from W maps to some sensor from $\mathcal{S}$.

$$\mathcal{V}(\sigma_i) = \begin{cases} 1 & \text{iff } P(\sigma_i, S_j) \neq \Phi \ \ \forall S_j \in \mathcal{S} \\ 0 & \text{o.w.} \end{cases}$$

Then we can express full observability of task t_i by the following function describing the boolean product.

$$\mathcal{O}_f(W) = \prod_{w_i \in W} \left(\prod_{\sigma_j \in w_i} \mathcal{V}(\sigma_j) \right)$$

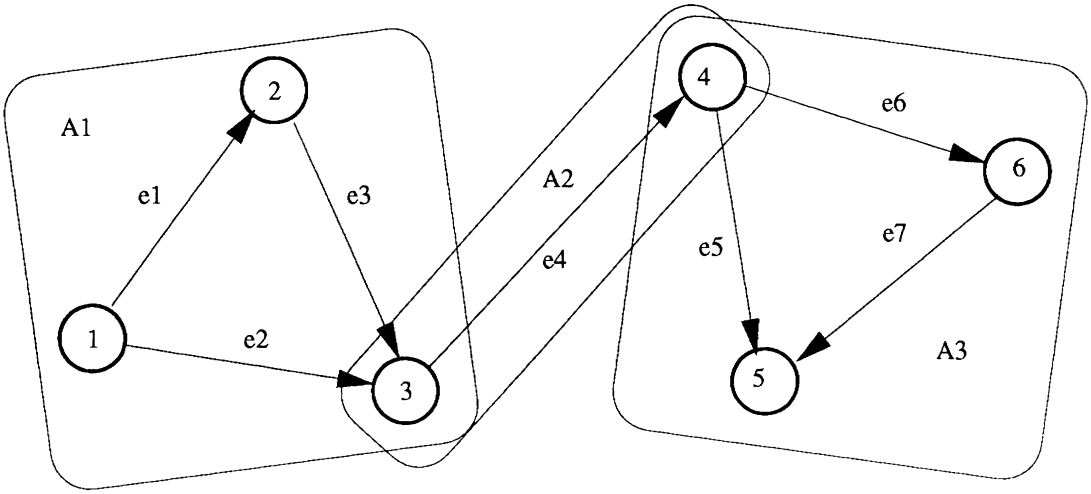

Figure 7: Task t in which subactions A_1, A_2 and A_3 appear as subcomponents.

Where $\mathcal{O}_f(W) = 1$ then describes that all the paths describing the task t_i are observable. The subscript f in $\mathcal{O}$ identifies the full observability.

The above condition, however, appears to be rather stringent for it requires that all possible events be observable. In general that is not possible at all times. In Figure 7, we present some task t described by some subactions A_1, A_2, and A_3. Furthermore we identify an initial state, 1, and a final state, 5. Let's focus on one of the actions A_1. We notice that there are two paths from the initial state, 1, to the final of this action, 3. While it might be possible to observe all of the events, for the purpose of recognizing that actions A_1 has taken place, it is sufficient to observe either of the two paths. There is clearly a semantic difference between the two paths. However, for the action to be observable it is important to realize that it has taken place, namely that some event has occurred to change the state variables from the initial state to the final state.

We define the sequence of events identifying a path from the initial node to a final node as a **critical path**. The task description presented so far has not clarified that what kind of states comprise a final state. In the example described in Figure 7 the final path denoted the completion of the action. However, some event occurring in one of the subactions may not lead to a final state in which the task is denoted as successfully accomplished. This could be an instance in which while carrying an object to a destination, the object slips from the gripper. These states, though not favored represent possible happenings. We denote such states as **dead states**. The connotation underlying the fact that the task "died" in those states.

We are now ready to tackle the notion of partial observability of a task.

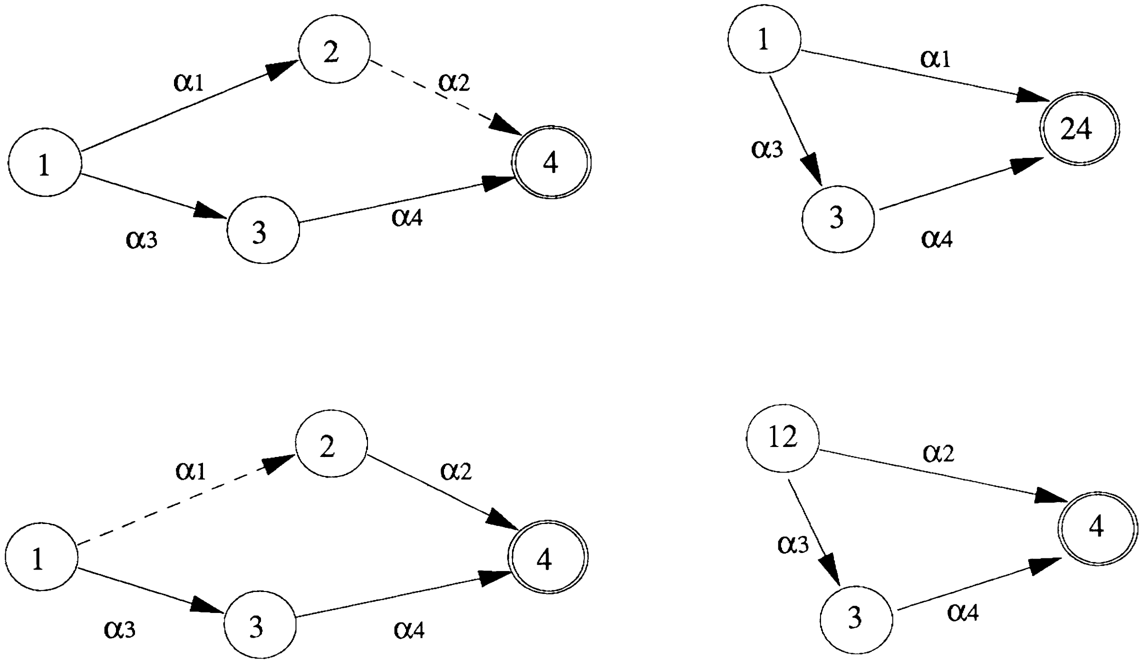

Figure 8: This example illustrates two instances of the effects of Partial Observability of two actions. The dotted lines represent non observable events.

5.3.2 Partial Observability

The projection function, $P(\sigma_i, S_j)$, allowed the description of observability of an event by some sensor S_j. What happens, however, if some of the events which characterize the task are not observable, $\mathcal{V}_f(\sigma_i) = 0$? In this case, some of the states become indistinguishable and we have situations, as presented in Figure 8. We label this effect of projecting one event to the null event and collapsing two states into one as **aliasing**.

It is also possible that, as exhibited in figure 9, partial observability may give rise to ambiguity. In fact, in this case while the original task description exhibits a clear procedural flow from the the initial state to the final state, the partial observation transformation introduces ambiguity. In particular considering the possible paths which could be taken during the evolution of the task, in the first example, in figure 9, it is not clear how important it is that event α_3 should take place. In the other instance it is not clear that the action may be at all observable.

It is clear from the above examples that not all partial observable mapping are desirable. Thus our notion of observability must include some strong conditions on the observability of certain events. This is equivalently expressed by considering the distinguishibility of the states

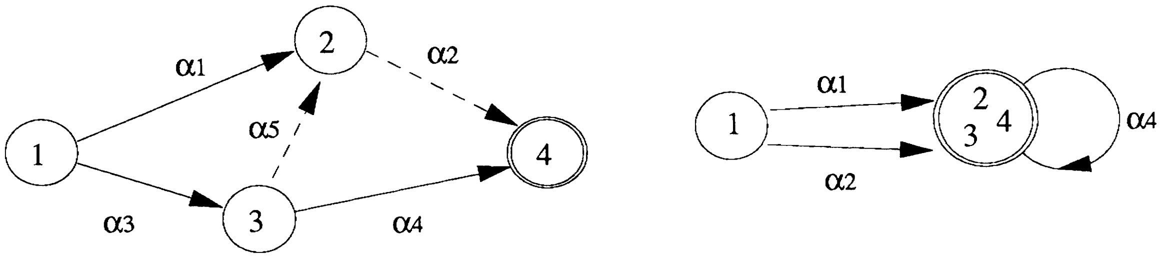

Figure 9: **This example illustrates an instance of the adverse effects of Partial Observability. The dotted lines represent non observable events.**

that these events transfer to. This can be accomplished by requiring that the critical paths include these states.

Thus, we define a set of **distinguished states** to describe those states which, if they appear along a critical path, must not not be aliased with some other distinguishable state. The distinguished states described here define a concept similar to what in DEDS terminology is known as marked states.

This definition does not required that all the distinguishable states be visited, only that they be unambiguously marked. This distinguishable states do not include the initial and final state, nor the set of the dead states.

We are then ready to provide two definitions for partial observability, describing a weak and the other a strong notion observability.

Weak–Partial Observability (WPO): Let w_i define a critical path, then the action which this path describe is weak–partial observable iff its length is at least one event, (requirement for observability), and if it contains distinguishable states these are either aliased between themselves or with the initial or a final state.

The functional description of weak–partial observability is similar to the full observability. The major difference is that events composing W and which we require to be observable. Then $\mathcal{O}_{wp}(W) = 1$ will describe WPO.

Strong–Partial Observability (SPO): Let w_i define a critical path and let substring u_j in w_i contain only two distinguishable states, then the action which this path describes is strong–partial observable iff each of the strings u_j, as defined above, is weak–partial observable. We notice that by the observability constraint, which requires at least the length of one event for two states to be weak–partial observable, than then even though we are considering a string u which contains within itself two distinguishable states, no aliasing will occur.

As in the functional description the observability of WPO, SPO observability is also defined by the same observable function with the set W changed. Then $\mathcal{O}_{sp}(W) = 1$ will describe SPO.

Why distinguish WPO and SPO? The main reason addresses some fundamental differences and requirements which are imposed by the internal and the external observer.

In the case of the internal observer, it is clearly the case that the supervisor monitoring the behavior of the system, would opt for full observability if it were possible. This would identify the best of possible worlds in which all that occurs can be explained. Most of times, however, not all events are observable. Then, in order to guarantee that the behavior of the system can successfully be monitored then SPO should be enforced.

An external observer, however, has little power on the selection on the choice of which events should be observable. In many cases it might be confined to observe whatever it can. In this case, aliasing may occur, but we want to establish whether the external observer could be actually able to observe anything at all. We have codified such requirement in the WPO which guarantees that at least the interaction can be observable.

5.3.3 Piecewise Observability

WPO and SPO were concerned with observability but not so much with the sensors involved in providing the observations. We note here the implications of the different sensor modalities which are to be employed and why one might want to have more than one.

Redundancy in observing an event, performed by more than one sensor, can be employed to corroborate the evaluation of the observations. However, the application of more than one sensor modality goes beyond the issue of corroboration. Unless the interaction is very particular, if only one sensor modality is involved ,there is very little hope of being able to observe all the stages of a task.

The importance of the above statement becomes clearer in the context of an interaction when the only way to express the observability of the different transitions is that of use more than one sensor modality. When applying a tool for piercing, for instance, a vision based sensor can observe the motion which brings the tool to the object surface. However, a force sensor can observe the effects on the object surface with accuracy. It would seem, on the other hand if one were to rely on position control of the end–effector to describe the state of the system, then the overall process would automatically become observable. This is, however, not so because, while it is possible to describe position and orientation transformations of the end–effector, we might not be able to recover other fundamental characteristics of the interaction, such as the force applied.

Any critical path observable by different sensors is **piecewise observable**. Thus, the previous example would clearly require having a critical path which is piecewise observable.

If a piecewise observable path has every event observable by only one sensor modality then we say that the path is **minimally observable**. If, on the other hand, the path is minimally observable independently by each sensor then we say that the path is **maximally observable**.

5.4 Continuity in Observability

The analysis of observables investigated so far brings forth the different sensor modalities involved in the operation. Once more than one sensor is involved, we are immediately drawn to the question of continuity of the observability. It is clear that if we had a critical path which were at all times observable by a single sensor modality, minimally observable, then the issue of continuity of observability would not arise. In that case we would be concerned only with the identification of the events. When more than one sensors are required, however, we need to worry about what happen between the time that task is observable by the two different modalities.

It appears that not only must we address the issue of an event being detectable by assigning some probability, as discussed in the next section, but also note how continuity should be propagated to guarantee overall observability of a task. Motivated by the use of sheaf theory, in [Bajcsy, 1973], we are currently investigating the possibility of applying such theoretical support to address continuity issues.

5.5 Unexpected Events

The observations we have addressed so far do not include any notion of unexpected event. We have only addressed the "slip" of an object from a gripper as an observable event in the sense that it represents a transition in a binary contact sensor. Thus, one would want to consider how such an event can be characterized in the context of observations.

Unexpected events can only be events which can be observed for if they were not observable then they would not be events for they would not be perceived. What is really meant by an unexpected event is really that the probability of its occurrence is quite low. On the other hand, other events are unexpected in the sense that there was no prior knowledge of their existence as a possible occurrence at a given state of the system. We thus distinguish them into **low–probability** and **unknown** events. Low–probability events cause the system to change to into state which can be recognized. Unknown events cause a transition to into a state of the system which was totally unexpected at the current stage.

The slipping of an object while lifting can be considered as a low–probability event. On the other hand, the shattering of an object upon applying pressure with the intent of grasping it could be considered an instance of an unknown event.

We clearly notice a substantial difference in the two kind of events. In the former case, the low–probability event depicts transitions to undesired states. It is, in fact, to avoid this undesired transition that when lifting a body specific contact locations are chosen so as to reduce the possibility of slippage. In the latter case, however, it is totally inconceivable that one might want to take precautions against all possible unknown happenings.

While unknown events are unpredictable if no prior knowledge is available, either in the form of previous experience or laws governing physical interactions, they can provide a source for expanding current knowledge of events.

To avoid all the issue of unknown events, one could put many constraints on the environment so that although low–probability events could occur, all possible event would be known. While assuming that no extreme events can occur, we would like to allow for the possibility of handling the unexpected events and the unknown events without having to describe all of them explicitly.

We can extend our formalism by assigning weights to events. Furthermore, we could allow transitions to unknown events to take place at any given time. That introduces the problem as to the decision for the value to attribute to the unknown event. Alternatively to using the Bayesian Probability descriptions we could use the Dempster–Shafer Theory, see [Pearl, 1988], which allows to label events as unknown as well as not requiring to have priors for describing events.

5.5.1 Uncertainty and Unreliability

Thus far we have addressed sensors as being faithful and reliable informers; however, the reality is that uncertainty in the measurements make the distinguishing of the transition and the stability of measurements much less distinct that we have assumed.

Uncertainty is due to the uncertainty in the behavior of the system while **Unreliability** is due to limitations of the devices.

It is important that we be able to distinguish between noise that a sensor is subjected to in the reading and the uncertainty in the detection that a transition to a different state in the system has occurred. Furthermore, we need to be able to define a measure of our confidence that a given transition has taken place.

In [Sobh, 1991] the transition to a different state is defined as the displacement of an observed feature in the vision system. The probability that a given event has taken place is computed. Based on some empirical threshold on probability, a decision is made as to whether the system has changed to a different state. In the case that it is not possible to determine the current state of the system, the control structure resorts to backtracking to a previous known state and reevaluates the situation from there.

Source of Uncertainty and Unreliability We have pointed out the following sources. The first one addresses reliability the others uncertainty.

- Resolution Noise – this depends on both on the modeling of the sensors and the required resolution of the observations (often a thresholding is adopted to handle this situation)
- Incomplete Observation Position – because of occlusion or inability to have sufficient detailed data, it might be impossible to establish exactly whether a given event has occurred. We need to be able to control the so that at any time the operation is piecewise observable.
- Nature Playing Tricks – this is an instance of the unknown event taking place. In this case in order to derive an explanation higher order of reasoning are required. As stated in the previous section, we would like at least to be able to recognize that some anomaly has occurred.

In order to be able to address uncertainty in the observer we would like to see:

- How it affects the overall observability of an action,
- How can piecewise observability be guaranteed.
- How the uncertainty propagates – perhaps using conditional probabilities or rules from the DS-Theory, see [Pearl, 1988].
- Whether there is a backtracking mechanism which may allow to "test before committing" to a transition. This clearly has to do with a measure of the confidence that we have in the given transition and in the measurement.

Addressing issues of uncertainty in the observability requires considering how specific sensors can handle uncertainty. Thus, it is important to address how the abstractions of the different sensor modalities can be augmented to incorporate uncertainty issues.

- In a vision based sensor, we might consider a "pixel based" measure of uncertainty (spatial information)
- In a force based sensor, some other unit of measure (force based information).
- In some other (possibly a combination of the the force and spatial issue)

5.6 Primitive and Complex Tasks

In the preceding portion of the section we have investigated the actual structure describing a task. As we have pointed out, in the beginning a task may be composed of subtasks. It is only natural to consider how these tasks should be combined in a hierarchical fashion.

In what it is to follow, we describe how from an initial a set of tasks we can construct a more complex set which has subtasks as components. The benefit of developing an algebra of tasks is that any action appearing as a subtask has a well-defined recognizer associated to it. This recognizer is defined in terms of the recognizers for the individual actions in the task. In particular, if some subtask has been previously fully explored, then the overall procedure for carry out the task is greatly improved in performance. This advocates a construction of complex tasks from a set of initial tasks, ***kernel actions***, which have been fully explored. composed by several others.

5.7 Kernel Actions

The tasks which we deem to be fundamental for the constructions of other task are often extremely simple and characterizable in terms of a single action rather than a sequence of actions. We will thus refer to this initial of tasks as the actions associated with them. This also has the effect of allowing a looser definition of what we can actually incorporate in our initial set. In this manner, we can incorporate actions for which the label "task" might resound a bit pompous for their procedural simplicity.

The choice of actions taken to be both necessary and sufficient to describe the components of tasks is a critical one. It may greatly limit the amount of tasks which can be formed from the basic set of actions. We define this set of actions as *Primitive Actions* since they are considered to be constituent of any other actions.

For each of the domains $\mathcal{D}_i$ from $\mathsf{M}{\times}\mathsf{T}{\times}\mathsf{O}$, as defined in the previous section, we define the **kernel** of the domain to be composed of those actions from which all the domain can be generated.

$$ker(\mathcal{D}_i) \;=\; \{a \;|\; a \in \mathcal{D}_i \;\ni\; a \text{ is a Primitive Action}\}$$

As a rule of thumb we may decide to incorporate into this set all those actions which do not require others as a prerequisite in order to be performed. However, the criteria which will guide our choice rely on the purposefulness of the action as a basis to construct others. We would like, however, that the primitives had some well defined properties. We consider the list of criteria, suggested in [Wilks], to qualify actions as primitive.

- *Finitude*: The number of primitives should be finite.
- *Comprehensiveness*: The set should be adequate to distinguish among different actions.
- *Independence*: No primitive should be defined in terms of others.
- *Noncircularity*: No two primitives should be described in terms of each other.
- *Primitiveness*: No subset of primitives should be replaceable by a smaller set.

Further motivations can be gathered from behavioral psychology. In these cases primitive actions can be classified as the initial actions which an infant uses to interact and discover his environment. The rules for constructing more complex actions vary with the age of the child and the basic set of actions have become more complex[21].

The level of complexity of actions in the primitive set is also fundamental for the power of expression that can be accomplished, i.e. the complexity of the tasks which can be described. The initial choice of the members of this set directly addresses the *A Priori Knowledge* issue.

For the time being we can naively sidestep the selection problem of the primitive actions. We will just state that such primitive set exist and can be constructed.

5.8 Interaction Domain $\mathcal{A}$

We define the *kernel* of $\mathcal{A}$ recursively to be the union of all the *unique* primitive actions from each of the $\mathcal{D}_i$. In the following definition we let

$$ker_0(\mathcal{A}\ ) = \ \{\ \Phi\ \},$$

then

$$Ker(\mathcal{A}) = \{ker_{i+1}(\mathcal{A}) = ker_i(\mathcal{A}) \cup \{\ ker(\mathcal{D}_i)\ /\ (ker(\mathcal{D}_i) \cap ker_i(\mathcal{A}))\ \}\}\ \ for\ i = 1, \ldots, 3$$

Namely, the primitive set of actions should contain no redundancy.

To exhibit how the construction of the above set can be accomplished, we need to address the notion of *Action Equality* between the different domains. Establishing equality between tasks is a wide research area and a well know problem in automata theory. Our approach will be based on considering the the components defining each individual action. Then equality could be defined in terms of the prerequisite, the procedural component, or the expected result from the interaction. We chose to define equality on the result of the interaction. Thus two actions, a_i and a_j will be deemed equal if their effect is going to be equal.

Having discussed the primitive actions forming, $Ker(\mathcal{A})$, we now address the construction of more complex actions from the primitive set.

5.9 Operators on Actions

The interaction domain $\mathcal{A}$ can then be defined as the set of actions generated from $Ker(\mathcal{A})$, the generating set for $\mathcal{A}$, by composing actions that already are in the domain.

We distinguish two operators for action construction.

- *Composition* as the sequencing of a list of actions, we express that as $C(a_1, \ldots, a_k)$.

---

[21] *Note*: Actually, this is a very interesting point because it suggests from a developmental stand point how this can also address the notion of learning and becoming an expert in doing something. By considering the *optimality* of the elements in the situation set which allows to be fine tuned to certain manipulations or interactions.

- *Repetition* as the composition of a given action $a_i \in \mathcal{A}$ with itself. It effectively identifies a special case of the previous one above.

The reason for determining the operators over the domain is that of providing syntactic validity for the actions that are being constructed. The repetition operation can be also interpreted as an operator to construct loops.

5.9.1 Composition of actions

We say that two actions can be composed if the final state of the first action can be identified as the initial state of the following actions. In other words, the final state of the first action can constitute the prerequisite for the second action. If the second action has no prerequisite state description then they can be composed.

5.9.2 Repetition of actions

This construction operator finds its motivation because it classifies a set of operation which are characterized by the repetition of a set of basic operations. In the English language these are characterized by verbs which denote the repetition of one or more actions which compose them. The nature of the repetition also depend on the type of objects on which the interaction is being carried out. Badler et al., [Badler *et al.*, 1990], describe some of these verbs and their relation to the repetition of the action. *Saw* represents an iteration of a forward an back motion and so does *shake*. However, *fill* and *load* may describe a repetition depending on the nature of the object they are applied to.

Extending $\mathcal{A}$ The composition and the repetition operators can be employed to generate some new actions. The newly generated actions are included in the action domain $\mathcal{A}$ by extending it to incorporate the new action.

$$\mathcal{A}_e = \mathcal{A} \cup \{ C(a_1, \ldots, a_k)\}$$

The construction of actions from the interaction domain guarantees that if the individual components are piecewise observable the resulting new action will be observable.

5.10 Properties of Actions

Property (Closure): the Interaction Domain $\mathcal{A}$ is closed under the operations constructing it: composition and repetition.
This property is quite an important one for it allows us to describe complex action in terms of primitive ones.

Property (Action Decomposition) Any given action $a_i \in \mathcal{A}$, $a_i \not\ni Ker(\mathcal{A})$, can be expressed in term of compositions and repetitions of primitive actions.

Property (Uniqueness) of the action is guaranteed by the construction methodology applied by the operations from the initial set.

5.11 A Language for Tasks Describing Functionalities

The actions that were selected from M×T×O to form $Ker(\mathcal{A})$ were chosen because either in themselves they represented actions for testing for functionality or they are needed in the composition of complex tasks.

Actions from $\mathcal{A}$ provide elements for the construction of functions. It is important to select these actions in a way which will express a process for describing and recovering a particular functionality. If we let $\mathcal{A}$ define the alphabet of actions, which can be composed, then a process to perform a functionality recovery can be described as an ordered sequence of actions from $\mathcal{A}$. Such ordered sequence is generated by the operators of composition and repetition, as described above. The recovery process is defined by a sequence and hence lineary. Thus, no ambiguities in the sequence can occur. While it is obvious that there is more than one way to carry out a task to recover a given functionality, we assume that once a decision has been made the task for recovering functionality is fixed. Having more than one action possible at any given time may reflect the ability to adapt to a changing environment. However, the price to pay for such adaptability is the introduction of non–determinism in the carrying out of tasks. Thus, we focus only on tasks which can be defined by an ordered sequence.

We can defined a language over the alphabet from $\mathcal{A}$; (We follow the standard notation of Automata theory as defined in [Hopcroft and Ullman, 1979]). Any function which can be constructed by composition and repetition must be in the language

$$L_{func} = \bigcup_{i=1}^{N} L^i$$

with $L^1 = \mathcal{A}$. L_{func} defines the *positive closure*. We do not consider null actions. Thus the minimum length for a task describing a functionality must be of length at least one.

However, not all the strings from L_{func} define functionalities, in fact, only a small subset of the strings define functionalities. The elements in this subset, $\mathcal{F}$, can be generated by some *Context Sensitive Grammar*, (CSG), which expresses the functionality description in terms of individual actions from $\mathcal{A}$.

We note that while the high–level description of a task can be expressed as a string in $\mathcal{F}$, each action in the sequence must also be piecewise observable.

We now present some examples of action compositions that illustrate this approach.

5.12 Examples of Actions and Action Compositions

The primitive actions that we have chosen are those which were deemed sufficient to describe the more complex actions defined below.

5.12.1 Example of Primitive Actions

We attempt to describe functionality tests for piercing and cutting using the primitive action *move, contact, puncture.*

Move a Target Object or Tool This operation describes a generic motion after the object has been contacted by the manipulator, grasped and then lifted from the supporting surface. Parameters identify initial and final position and the transformation between the two end–points. During this maneuver, observability hinges on the ability to either visually track the motion of the manipulator and object or to haptically corroborate that the object is remaining in contact with the manipulator.

Bring to Contact In this case we define an operation of bringing a manipulator in contact with an Object. We assume that there are no obstacles to be avoided. It can be essentially conceived as the moving of the manipulator to a specific location. The difference from the previous action is that now the observation of the interaction changes to reflect the contact. In the case that the tool contacts the object, the tool can be described as a specialization of the end–effector.

Since one can not rely on exact measurements to estimate the position of contact the object, then one would want to monitor the contact sensors and then acknowledge that contact has been obtained. Aspects addressing the reliability of the sensors and the instantiation will have to be addressed when dealing with the implementation of the actual routines which instantiate a given function.

We note that this action is a specialization of the previous one. In particular, the end–position of the transformation is now specified as the locus at which contact will occur. It is important to make this distinction because it emphasizes a switch in sensor modality.

Puncture Identifies an operation in which Position Control of the end–effector has been switched to Force Control to guide and stop the operation. The manipulator in this case must have a way to control the force that is being applied to the surface being operated on.

5.12.2 Examples of Complex Actions from Primitive Actions

In both *Piercing* and *Cutting* we identify three phases, once the tool has been grasped. The tool is moved to obtain contact, contact is established and pressure is applied to accomplish puncturing. Both actions are similar. They differ in the way the tool is being grasped, in the way it is applied, in the type of contact that is required.

The three phases identified above allow us to focus on the piecewise observability of the overall action. The motion portion, can be observed both visually as well as by contact sensors in the manipulator hold the tool. The coming to contact of the tool with the target object, is only coarsely observable by vision. If the action were to be observable only using vision, it would be quite hard to determine the exact moment of contact and control the manipulator. In this operation, it is more important that the observation be carried out by the force sensor. Once the tool is in full contact with the target object, the observation by a vision based sensor is no longer informative. It is, in fact, quite hard to observe deformations which might take place on the surface as pressure is applied with the tool. Feedback observation at the tactile level gives us also corroboration of the implicitly hypothesized material composing the object. Furthermore, once the operation has switched to force control it is unobservable by the vision sensor. To provide and guarantee overall piecewise observability is, as we have discussed quite essential to describe the complete task.

The description of the tasks in this fashion make also intuitive sense. When human perform the action of piercing, for instance, they use both visual and tactile feedback both to control the operation but also to observe how effective the applicability of the tool for the function.

Just by considering these two functionality tests, we notice that the two actions require a more detailed definition involving many aspects of grasping, of coming to contact and of applying forces. These observations uncover a large set of issues which must be addressed in a real world application.

In high–level description of tasks, not enough consideration is given to the manner in which a tool is to be grasped, how it is to be applied, and what kind of contact it is to make with a target object. These specifications are brushed aside as details of implementation, yet it is important to be able to bridge the gap between the high–level specification and the context. *Grasping* is too often erroneously considered a simple operation. Much literature addresses the issue of grasping an object with a specified intent. [Liu *et al.*, 1989; Iberall *et al.*, 1988; Iberall, 1987] consider the types of grasps and the functional properties that they exhibits. Cutkosky, [Cutkosky, 1989], presents a hierarchy of grasping with intended configurations. In [Nguyen and Stephanou, 1990; Nguyen and Stephanou, 1991], the authors describe continuous domain for mapping hand preshaping; they also, as the preceding authors, associated functionalities with the respective hand preshapes. [Klatzky *et al.*, 1987a; Klatzky *et al.*, 1987b; Pellegrino *et al.*, 1989], investigate the different type of prehensions for different objects in human grasping. [Pollard and Lozano-Perez, 1990; Stansfield, 1990] investigate grasping properties connected with particular polyhedral objects. The Handy System, [Lozano-Perez *et al.*, 1987], develops and carry out a complete task of pick–and–place.

These are but a few examples which address the task of grasping in a non–analytic manner. These approaches are particular important in the development of our recovery procedure. They, in fact, reflect our goal of instantiating the task and systematically generating plausible approaches to consider the many parameters that the high–level description leaves unspecified and unconstrained.

6 From Qualification to Quantification

In the previous section we have seen how some of the actions can be described as compositions of already defined actions and the components that characterize an action. The description presented so far, however, is insufficient because even at the abstract level, we need to address the issues of relative location and orientation of objects, the qualification of the force, the velocity which is to be applied, and the time interval in which an operation is to take place.

One can readily observe that the difference between the action of *hitting* and *contacting* a surface depends on the scale of the time frame in which the operation of contacting takes place. It is necessary to incorporate this essential aspect in our formalism to make it more expressive.

We will define a relative notion of time as a qualitative description which becomes instantiated into a specific domain. This introduces problems when actions are compared between different contexts. However, as long as the reference frame of the quantification is known, the comparison can be performed. The qualification of an action depends on the specific context; what is "slow" in one context may be perceived as "extremely fast" in another. While the attribution to some exact values is possible in a specific context, when trying to relate two objects in an abstract manner we need to use spatial relations. These relations will have to be properly defined so that they are meaningful in various contexts.

Reasoning about actions usually takes place in two separate domains, either too high level or too specific. The high–level domain addresses task decomposition, planning structure, temporal ordering of the events, etc.. The literature is quite extensive in this area, we single in [Davis, 1990], [Allen, 1984]. While it is *essential* to be able to describe actions at a high level to guarantee that given sequences of actions are meaningful and can achieve the stated goals. Therefore, it is important that the individual action be mappable to a specific context. Planning involves high level description of motion, velocity and force, but when it comes down to actually implementing the operation in a specific context of manipulation or motion it is necessary to attribute values to the individual actiona. Picking something up and putting it down are sufficiently differentiated in the direction of motion. But, we are interested in being able to describe the difference between operations which differ in the aspect of force, velocity, time in which they are carried out.

In the low–level domain, the operation takes place in very specific domains or small classes of domains in which specific algorithms are applied, or specific classes of manipulators are involved.

As we stated in the introduction, we would like to be able to accomplish a transition between the very abstract level and the specific context in which the action is to be carried out.

As we pointed out, the difference between the two operations, of hitting and contacting, is clearly the speed in which it is carried out. But, while one can describe it at a high level simply as faster, that has really very little meaning in a specific context. One can state that the time interval should be shorter, but how is this expressible? We choose to focus on describing this in terms of time and positions, or conditions as we will see later, which are to be occupied at different time. We will describe it in terms of relative velocity at specific "time labels", as described below. Thus, the notion of both positions and conditions describing monitored events which can demarcate the time intervals. This allows us to describe how velocity and force can vary within intervals. We initially describe how events are defined and how force and velocity can be described in terms of these events. Later we will point out how to determine a quantization for both force and velocity so that high level labels of "slow" and "fast" can actually be attributed appropriate values in different domains, yet still capture the semantics of the actions.

6.1 Size, Position and Orientation

In this section we address the definition of size, position and orientation. These properties are usually used to relate one object to another or a single object with respect to a reference frame.

While position and orientation are generally expressed as a position vector $\vec{p}$ and an orientation vector $\vec{o}$, we would like to allow to have descriptions in which they can be describe both as parameters to a functional description of the action as well as at a higher level.

When addressing size, we are interested in some specific property which would allow us to describe spatial relation between objects. Classifications with respect to size may be useful, but their usefulness depends on the task. Instead of highlighting all possible interesting properties which allow geometric differentiation between two objects, we will focus on those which will allow us to describe spatial relation between them.

We now proceed to determine size relations. We will build on these relations to qualitatively describe position and orientation.

6.1.1 Criteria for determining Size Relations

Size relations are important descriptors in determining spatial relations between objects. In order to simplify the problem of classifying size in objects which are volumetrically defined by different shape, we can consider the convex hull of the object. The convex hull can be either computed with one of the many algorithms from Computational Geometry or can be approximated with a volumetric primitive of choice which would be able to "best contain" the object, according to some chosen objective function.

We will not focus on the choice of the volumetric primitives to best contain and describe the object but rather on the properties which this selected representation should provide.

Properties include

- The **volume** criterion. This criterion is best employed if the objects under comparisons are modeled by the same parametric volumetric primitive.

- The **axis ratio** between the two object, including **minor** and **major** axes.
- The **eccentricity** of each object separately.

This list is by no means comprehensive but it gives starting criteria as a basis for the relations.

An other important point refers to the number of objects that are to be compared. Since the description is given of one object relative to another, it is a simple matter to determine which object should be considered as a reference for comparison. In our case we choose the smaller of the two as a normalizing factor. In the case that there are many objects then the choice could be described, for instance, using the median in the distribution of the sizes. Clearly some other criteria can also be employed.

In our case since we are interested in manipulating a specific object, we will always choose our description to be given with the respect to the object on which the focus of attention is geared. In the case that there were many objects to be considered at once then we could resort again to using the mediam in the set of objects. We will consider relation involving two objects only the description can extended to several objects.

6.1.2 Position Relations

In determining spatial relations, it is important to recognize which relation amongst the set of relations is most significative and best suited for description. [Mukerjee, 1991] discusses an approach to building qualitative models for vision based on the functionally–relevant attributes of each object. Spatial discretization is derived systematically, and is complete under considerations of tangency, no–contact and overlap. While the work identifies the properties which are invariant with respect to the object's location, the Medial Axis Transformation suggested for mapping the relations between object in property space appears to be closely linked by the underlying generalized cylinders description. The underlying implicit criteria for choosing the properties is that they should be independent of the viewer and the description is presented in terms relations between adjacent objects. While the algebra developed is based on point and line relations, the examples provided are limited to object representation by parts. Analysis is provided for separate object or for objects which poorly described by generalized cylinders. On the other hand it points out the importance of preserving relative size and alignment in object part descriptions. We will consider the relevance of alignment at a later stage when talking about models in the object domain.

In order to address relative spatial placement amongst objects, we will describe these relations in terms of the ratio between the actual distance and the minor axis of the smaller of the two objects. We attribute to this value to the spatial relation of *Near*.

Let's define it as follows:

$$P_{near} = min \, \{length(a_i) \mid a_i \text{ is an axis in object } O_j\}$$

This provides us with a basis for establishing the other position relations. These positions could be described either as absolute positions or relative position based on the viewer location. We will address here the absolute position of the objects. However, the relative perspective needs to be addressed when addressing the visual the observer for two objects. In that case, the perceived position may or may not be representative of the absolute position of the two objects. The interpretation and the relative actions which should be taken by an observer, depend greatly on the goal of the observer. Thus, we will address here the absolute position only.

Magnitude Spatial Relation . The following relations are determined only by the magnitude of the distance and are irrespective of other characterizations. Particular spatial relations, such as right, left, etc, which are observer dependent will be discussed below.

The position relations defined below are dependent on constants that are identified at the end of this section when the abstract description is quantified in the domain.

- **remote** $P_{remote} = \alpha_r \; P_{near}$. This position effectively identifies the extreme of the range.
- **far** $P_{far} = \alpha_f \; P_{near}$ with $\alpha_r < \alpha_n$
- **near** P_{near}, as defined above,
- **close** $P_{close} = \alpha_c \; P_{near}$
- **contact** the distance is zero.

Relative Placement

- **above** if O_2's lowest point is above the highest point of O_1.
- **below** the inverse of *above*.
- **same level** neither above nor below.

Real vs. Perceived Distance : The perceived distance between two objects is subject to the location of the viewer. However, if the observer is remote from the pair objects, their sizes will be perceived as smaller and their spatial relation can still be determined.

External Observer Description of O_2 w.r.t. O_1 : The descriptions of obj O_2 w.r.t. O_1 as presented above may be not consonant with the external observer. Unless we assume that the observer can simultaneously occupy the position of the O_2, or is, in fact, the object.

Relative Placement based on External Observer : This position is described in terms of an external observer who is considering the relation of the two objects. This description clearly varies with respect to the viewing position adopted. This description will be defined with respect to the viewing plane of the external observer.

- **left**
- **right**
- **front**
- **behind**

6.1.3 Determining Orientation

In order to determine the relative orientation, it is necessary to choose some way of considering the alignment of the objects. We will address the relative orientation between two objects, O_1 and O_2. For each object let a_1, a_2, a_3 describe the axes of each object with the following magnitude ordering convention: $a_1 \geq a_2 \geq a_3 > 0$. A second subscript will identify the object the axis refers to. Thus, we can compute the angles formed between the respective axes as follows:

$$\vartheta_i = cos^{-1}\left(\frac{a_{i1}^T\, a_{i2}}{||a_{i1}||\, ||a_{i2}||}\right)$$

where $0 \leq \vartheta_i \leq \pi$ and for $i = 1, 2, 3$. We can however, restrict our attention to an interval between $0 \leq \theta_i \leq \pi/2$ by letting

$$\theta_i = \begin{cases} \vartheta_i & \vartheta_i \leq \pi/2 \\ \pi - \vartheta_i & \text{o.w.} \end{cases}$$

since we are interested in the relative orientation of the objects independently of the viewing direction.

If we now consider the space defined by $0 \leq \theta_i \leq \pi/2$ for $i = 1, 2, 3$ then we can observe that low values of θ_i identify similar orientations while high values will identify very dissimilar orientation.

The space that is defined in the above gives each of the axis equal importance, however, we would like to incorporate the relative notion of the axis size in the measure for the orientation. In this manner we would be able to determine the degree of orientation between the two objects. Thus we can define the following three functions:

The *first* one addresses the degree of parallelism present between the two objects:

$$P_{||}(\theta_1, \theta_2, \theta_3) = \left(1 - \frac{\theta_1}{\pi/2}\right) + w_2 \frac{\theta_1}{\pi/2}\left(1 - \frac{\theta_2}{\pi/2}\right) + w_3 \frac{\theta_1}{\pi/2}\frac{\theta_2}{\pi/2}\left(1 - \frac{\theta_3}{\pi/2}\right)$$

The *second* one describes the degree of perpendicularity

$$P_{\perp}(\theta_1,\ \theta_2,\ \theta_3) \ = \ \frac{\theta_1}{\pi/2} \ + \ w_2\left(1 - \frac{\theta_1}{\pi/2}\right)\frac{\theta_2}{\pi/2} \ + \ w_3\left(1 - \frac{\theta_1}{\pi/2}\right)\left(1 - \frac{\theta_2}{\pi/2}\right)\frac{\theta_3}{\pi/2}$$

The *third* one describe the orientation with respect to an angle of $\pi/4$

$$\begin{aligned} P_{\pi/4}(\theta_1,\ \theta_2,\ \theta_3) \ = \ & \left(1 - \frac{\mid \theta_1 - \pi/4 \mid}{\pi/4}\right) \ + \ w_2\left(1 - \frac{\mid \theta_2 - \pi/4 \mid}{\pi/4}\right) \ + \\ & w_3\left(1 - \frac{\mid \theta_3 - \pi/4 \mid}{\pi/4}\right) \end{aligned}$$

where w_2, w_3 represent the weight associated with each of the axis and are defined as:

$$w_2 = \frac{min\ \{\ a_{2,i}\ \}}{max\ \{\ a_{1,i}\ \}}\ ; \quad w_3 = \frac{min\ \{\ a_{3,i}\ \}}{max\ \{\ a_{1,i}\ \}}$$

and the i subscript identifies the objects $i \ = \ 1,\ 2$. In this manner if the object have very different sizes, then the major axis will result more prominent. If two object have approximate size then the other axis will also tell about the orientation.

We notice that the values which both functions above range in the interval $[\ 0\ ,\ 2\ ]$. Clearly, not all of them can have the maximum value at the same time.

Then, by comparing the values obtained for the functions

$$\begin{aligned} V_{\perp} &= P_{\perp}(\theta_1,\ \theta_2,\ \theta_3) \\ V_{\|} &= P_{\|}(\theta_1,\ \theta_2,\ \theta_3) \\ V_{\pi/4} &= P_{\pi/4}(\theta_1,\ \theta_2,\ \theta_3) \end{aligned}$$

we can describe the type of orientation present. However, we also would like to preserve the association of the function which provided the orientation and also the axis which was most prominent in deriving the labeling.

One can therefore describe the following triplet $(l(f), v(f), a(f))$ in which

- $l(f)$ represent a label, $l(f) \in \{\perp, \|, \pi/4\}$
- $v(f)$ identifies the the respective value for the function and
- $a(f)$ describes the axis which is most prominent in determining the particular attribution.

The axis which is most prominent in determining a value can be determined by considering the components. Each one refers to one of the axes of each of the above functions. In the case that more than one component, per function were to yield the same value then it could be picked with respect to importance of axis, as ordered from major to minor.

One can now describe the orientation in the object as the total order between the triplets. The ordering function for the triplet is simply determined by considering the quantitative component in the triplet.

An example of such order could be:

$$(\perp , V_{\perp} , a_1), \quad (\| , V_{\|} , a_3), \quad (\pi/4 , V_{\pi/4} , a_2)$$

Actually, by considering the difference between the values in the triplets, a more qualitative description of the orientation can be defined. These objects, however, are described relative to some frame of reference, be that of the resting surface of that of the endeffector manipulating the object. Thus, we will address, therefore, the relative orientation of an object with respect to the gravity vector. This will allow us to give definition to the pose of an object as well as to describe the relative orientation of a tool with respect to the object.

Location of Gravity Vector [Salganicoff, 1992] describes the pose of objects with respect to the location of the gravity vector. He identifies the orientation of the axis with respect to the orientation of the Gravity Vector. This can be used as an additional descriptor for object direction and eventually for approach.

6.2 Time

Having described position and orientation relation, we begin by identifying the components for properly describing time, force and velocity.

When describing time as the reference frame for operations, we need to consider the temporal sequencing of the events. Thus the time frame will provide a temporal ordering for the events. It is the context in which an operation is carried out that the duration is to be described. Events, as we have seen, describe transitions between states of the system. Such transitions are characterized by having some conditions met. Thus, we can describe the sequencing of the different configurations as the temporal thread which defines time. The relevant instances which determine the elements in the temporal ordering could be described as a ordered sequence of *time labels.*

A time label is associated with by the change in the truth value of conditions:

- *location* element. This depends on a particular spatial location being occupied during motion. T_{near} identifies a label being asserted when an object O_1 reaches the *near* location of object O_2.
- *guarded* element, described below.
- *condition* element. This identifies a generic notion of a condition being asserted. For instance, **Displacement** equals half of D_{max} represents one such general condition and is associated with a spatial relation. Its relevance as an element in the temporal sequencing of events depends on its being asserted.

Thus the *time interval* between two *time labels* can be described as the elapsed time which occurs between two relevant instances. This description, however, does not provide a qualification relating the "wall–clock time" to the sequencing of the events. Going from configuration c_i to next configuration c_{i+1} requires that some observations be performed or conditions be monitored. Furthermore, the actual "wall–clock time" between the same time labels will vary in different contexts.

To better characterize the time labels we can distinguish them into two sets:

- **Basic** or **Coarse** set. It characterizes those labels which are known prior to the beginning of the operation. The order of these labels is well–defined by the task that is to be carried out. They identify states in which the system could be depending on the occurred event.
- **Refined** set. It represents the set of labels which are known procedurally apriory but whose values can only be determined at execution time. Furthermore, the sequencing of these labels can not be exactly predicted a priori. Their sequencing may vary pending on which conditions are asserted. This refined sequence occurs always between two time labels in the basic group.

To make the distinction clear we can consider the following task. Assume that a given tool for piercing, held by a manipulator, is at a distance from an object. Now we can characterize the two sets as being composed by the following events:

In the Basic set we can identify the following:

- E_0 motion is initiated
- E_1 the tool is close to the object
- E_2 the tool is in contact with the object
- E_3 the tool has reached the final state: penetration by distance D_{max} was accomplished or maximum threshold of force has been reached.

When considering the event E_3, we can observe that from the moment that the tool comes into contact with the object, E_2, to the moment that E_3 is asserted, many other things may happen. By examining how the variables change from the moment that E_2 is asserted to the moment that E_3 occurs, we can observe that that transition can take place in many different ways. It could be that, irrespective of the penetration rate the same force is applied, or it could be that if the penetration rate decreases then the force is incremented. It could also be the case that the force is incremented only when the penetration rate falls below a certain value. Since we would like our formalism to provide with the flexibility to express a situation as the last one, we need to be able to recognize the assertion of the conditions which characterize the need to increment force with the possibility to do so in a stepwise fashion. The ordering of these time labels can not be defined a priori unless all is known about the interaction. It depends on the composition of the material and the force applied, but we would like to the able to perform the operation without having to model the dynamic behavior of the material. In order to do this, we can define the procedural relation of the interaction. The events are thus characterized by the conditions which may be asserted while executing the action. We call these the refined set of labels.

Thus in the interval $E_2 < t \leq E_3$ we could define the following refined set

- e_{23_A} if penetration rate has fallen below a desired threshold, then the force should increase.
- e_{23_B} if penetration rate has changed to be above the desired threshold then stop to increasing the force and apply it with the last magnitude obtained.

As we have stated previously the sequencing between e_{23_A} and e_{23_B} depends on several factor such the material, how the force is increased, the surface area determined by the tool at contact point.

6.3 Kinematic and Dynamic Characterization

Having described how we can quantify position and time components, we can now look at actions in terms of their kinematics and dynamic characterizations. Thus, we will present how Force and Velocity can be described in terms of our previous relations on time and space. We have chosen to use velocity as one the descriptors for actions attributions because we both talk about variations in velocity, and hence acceleration, but also differentiate between values of constant velocity. Constant velocity identifies zero acceleration and hence the underlying difference a high and low values of constant velocity could not be expressed.

[Badler *et al.*, 1990] identifies a characterization of actions as *Kinematic-Dynamic*. *Kinematic* describe actions involving motions along an arbitrary path and with an arbitrary velocity, for instance *displace, roll, etc.*. *Dynamics* addresses the force which characterizes a motion, such as *press, hit, pull, push, etc.*. Both *Kinematics* and *Dynamics* describe operation which can be characterized both by motion and force. In a certain sense, very few actions can be described as having only kinematics components and no dynamic component. Thus what is pointed out in the characterization in [Badler *et al.*, 1990] is the predominant of one characterization over the other.

In our approach we will consider the interplay between kinematics and dynamics to describe actions. But prior to describing actions we will look at both descriptions of force and velocity. In the section when we actually proceed to describe primitive actions, we will address the interplay of velocity and force.

6.4 Force and Velocity

We identify the following components:

- **function** it provides a functional description of how the quantity should change over time in the periods defined by duration. The function description could be described as *constant, logarithmic, linear*, etc..
- **duration** it describes the subdivision of the action period accordingly to events which are to take place.

In both cases we are interested in describing the magnitude rather than the path in which these are to be applied. We use the sign in the force to express the direction. The location at which to apply the force and path for the motion can be described as parameters to the action and are very much context dependent. We will allow those components to be determined at action–instantiation time.

6.4.1 Guarded Conditions on Actions

We introduce the notion also of guarded actions. This a popular manner of controlling range operations in robotics. Guards are normally introduce to verify that certain values will not extend over a certain critical value.

Guards are special conditions which define the safe boundaries of operability. While in general, one might presume that when a guard is "alarmed" we would like to stop the operation, we will treat them as exceptions, since we would expect them to occur infrequently. We would want to provide a *default handling* and with the possibility of an *alternative handling*. The effect of specifying an alternative handling would be that of suppressing the default. To allow flexibility of description, while a guard has a default action associated with it, we allow the alternative handle to be either, *null* or specified with a new handling action.

The following monitors the reaching of the threshold value for the force that it is to be applied and upon reaching it initiate a reverse motion.

$$g_1 :: \begin{cases} g & force = F_{max} \\ d & \text{stop action} \\ a & \text{revert direction of motion} \end{cases}$$

This extension allows to express operations more clearly but it implies that the guarded subaction must also be observable, or at least monitorable, for otherwise the system could end up in unknown state.

More than One Guard at One Time : We allow also more than one guard to be active at one time. There could be more than one safety condition which should be monitored. Thus, instead of having only one guard we might want to have a list of them.

6.4.2 Describing Velocity

In order to see how the components defined in the previous paragraph can be employed we consider a simple action involving only motion. Namely, we describe the velocity component of an action which brings a manipulator to *contact* a surface.

$$V(t) = \begin{cases} \mathsf{const}_1 & T_{init} \leq t \leq T_{near} \\ \mathsf{linear\ decrease} :: g_1 & T_{near} \leq t \leq T_{close} \\ \mathsf{const}_2 & T_{close} \leq t < T_{contact} \\ 0 & t = T_{contact} \end{cases}$$

The above expresses the following:

1. Initiate the motion with constant velocity, for some constant const_1 velocity. The duration of this motion is to last until the manipulator is near the object. The notion of *near*, *close*, *remote* are spatial relations that have been previously defined.
2. Upon reaching a near neighborhood of the object, decrease the velocity to const_2 . That should be accomplished upon reaching a closeness to the object. g_1 identifies a guard, described below.
3. The final approach should be performed at the speed reached in the previous section (very slow) until reaching contact with the object. At that point the Velocity of the manipulator should be 0, described by the last condition.

In the description above we have included the guard g_1 in one of the components. Since our time interval is described in terms of position events, it could very well be that the estimation of the location of T_{close} was not correct and that the velocity const_2 was reached before, or even worse that the calculation of the location were quite off or that the coefficient specifying the linear decrease in velocity was too big so that actually the velocity had fallen to Zero or very closed to it. This identifies a very undesirable behavior.

A guard to prevent this from happening can be described as follows:

$$g_1 :: \begin{cases} g & V = \mathsf{const}_2 \\ d & \mathsf{NULL} \\ a & V = \mathsf{const}_2 \end{cases}$$

The above states that upon reaching the guarded value, whether or not the location is T_{close}, the speed should now be constant.

6.4.3 Describing Force

We now consider an operation which primarily characterized by the dynamics of an interaction such as *pressing* and we focus only on the force component.

$$F(t) = \begin{cases} F_{init} & t = T_{init} \\ \mathsf{linear\ increase} & T_{init} < t \leq T = c_0 \\ \mathsf{quadratic\ decrease} & T_{init} = c_1 \leq t \leq T = c_2 \end{cases}$$

where the conditions c_0 and c_1 can be expressed as

$$c_0 = \mathsf{while}\ (\mathsf{force} < F_{max})$$
$$c_1 = \mathsf{while}\ (\mathsf{force} > F_0)$$

The above description suggests that the force is to be applied from an initial value F_{init}, the value depends on the state of the system. It is then to be increased linearly upon reaching maximum force and then decrease quadratically until it is F_0. In this case we have included guards on the action since they are implicit in the time description. The above could also be described using only guards:

$$F(t) = \begin{cases} F_{init} & t = T_{init} \\ \mathsf{linear\ increase} :: g_1 & \\ \mathsf{quadratic\ decrease} :: g_2 & \end{cases}$$

where g_1 and g_2 are specified as

$$g_1 :: \begin{cases} g & force = F_{max} \\ d & \mathsf{NULL} \\ a & F = \mathsf{quadratic\ decrease} \end{cases} \quad ; \quad g_2 :: \begin{cases} g & force = F_0 \\ d & \mathsf{NULL} \\ a & F = F_0 \end{cases}$$

Introducing a partition for a force to be applied, as was described above, requires the expression of the notion of a duration of an operation with respect to a force to be applied. The behavior of the dynamic interaction varies greatly on the goal that is pursued and the material involved. In the case of a "soft" material, when a force is applied, the surface of the material will deform so that a constant force, possibly a small one will be sufficient. If, on the other hand, the material is "hard" then a constant force or an increasing force might not affect the material at all and there might not be any displacement at all. Thus, in the first case there was some displacement, while in the second case there was no displacement at all. However, when the goal is that of grasping an object, then, upon reaching contact, the amount of pressure that it is to be applied to the object depends on the manipulatory operation that it is to be performed, on the type of preshape involved, and on the location of the grasp (stable vs unstable).

6.5 Determining Values for Velocity and Force

In the previous sections we have talked about a starting value for velocity and force. Yet, while it is trivial to attribute the extrema of range as the minimum and maximum velocity or force which can be developed by a system, it is a completely different matter to attempt to attribute some value constant V_0 or some force F_0.

6.5.1 Range Description

The real issue that we are faced with is that of partitioning the continuous range and attribute into significant labels. We suggest that within an absolute range of values, say from 0 to 1, intervals may be identified to describe qualifications. For the time being we will just outline some of these qualifications which should be assigned values in order to provide for an interpretation of task descriptors.

- adjectives: *slow – fast*, *small – big*, *short – long*
- modifiers: *extremely*, *very*
- intermediary: *moderate*

These absolute scale attribution clearly must be defined for velocity, force, and possibly time. The reason that we might be interested in addressing a qualification of time is that at times duration must be made explicit. The time duration of force application might be needed explicitly in the eventuality that no observable state variable may change. In such a situation we might want to be able to point out how long an action should last. This clearly calls for the extension of the formalism which would require temporal descriptors for actions.

6.5.2 Context Qualification

As we have pointed out, context qualification is a necessary condition for the scale attribution. The investigation proposed by [Cahn von Seelen, 1988] addresses exactly this point in the context of verbs addressing motion. He discusses how the notion of a *fast moving car* is directly dependent on context in which the attribute is evaluated. As the road varies in terrain and slope, the notion of "fast" also varies. He formally defines 36 motion verbs according several characteristics. Firstly, 3 types of motion verbs are identified based on the time interval in which they are valid. These are: *Progressive* describing the on going action valid during the whole extent of the interval, e.g. to follow. *Inchoative* denoting the beginning of an action as the moment of validity, e.g. to leave. *Resultative* describing the termination of an action and valid only at the end of an action, e.g. to stop. Then, *monotone transitions* of parameters such as *direction*, *speed*, *distance*. Finally, considering initial and final values of the parameters a *Precondition*, *Monotonic Condition*, *Postcondition* are consider to qualify the motion.

The above investigation further emphasizes the reason for having an abstract description and mapping it to a context in which it may take the "appropriate" values.

6.5.3 Force Application: *How much and for how long?*

When initiating an operation with a system there is a well known notion of the range of the force that it can applied. These specifications are provided by the manufacturing of the device. However, short of that it is important to classify the effect that particular forces have on different materials. Sinha [Sinha, 1992] has approached the problem by modelling the effects of applying a force to a surface in order to determine whether the surface can be stepped on. This type of procedure can be employed to calibrate and quantify the effect of force application. In particular it can provide with a time scale for force application. The ratio of the period of the force application versus the force applied to the surface can tell about properties of the material and therefore, when one should desist in performing the operation.

The duration of time is an important issue in force application. Furthermore, in the case of surface penetration, which may or may not be a desirable effect, depending on the task, the dynamic behavior may change. This effect is often due to the material properties.

7 Conclusion

In this paper we have exposed some of the complexity which lies under the neat and clean label of functionality. We have pointed out that in order to recognize functionality we need an interactive process.

We have examined what is meant by functionality in an object and presented an abstract task description apt to recover it.

We have stressed in this investigation the importance of recovering material properties, of observing and verifying the interaction, of evaluating its performance with an initial set of hypotheses. We have seen that recovering functionality requires the ability to map a manipulatory task into a context.

We have focussed on the description of the performatory component of the task for recovering the functionality and in particular examined the observability aspect. Instantiation of the task and generation of the hypotheses were only outlined and clearly need to receive more attention.

Having understood the importance of functionality in object description we will also need to expand on means for incorporating the result of the recovered evaluation of the interaction in the object description.

References

[Allen, 1984] J.H. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154, 1984.

[Badler *et al.*, 1990] Norman Badler, Bonnie Webber, Jugal Kalita, and Jeffrey Esakov. Animation from instructions. In N. Badler, B. Barsky, and D. Zeltzer, editors, *Making Them Move: Mechanics, Control and Animation of Articulated Figures*, pages 51–93, Morgan-Kaufmann, 1990.

[Bajcsy, 1973] R.K. Bajcsy. Computer identification of visual surfaces. *Computer Graphics and Image Processing*, 2:118–130, 1973.

[Bajcsy *et al.*, 1991] R.K. Bajcsy, R. Klatzky, V. Kumar, D. Sparks, A. Diamond, and S. Lederman. University research initiative program of the office of naval research, proposal: center for haptic object recognition. 1991.

[Biederman, 1987] I. Biederman. Recognition-by-components: a theory of human image understanding. *Psycological Review*, 94(2):115–147, 1987.

[Brady, 1983] M. Brady. *Robotics and Artificial Intelligence*, chapter Artificial Intelligence and Robotics, pages 47–63. Volume II, NATO ASI Series, 1983.

[Brady *et al.*, 1985] M. Brady, P.E. Agre, D.J. Braunegg, and J.H. Connell. The mechanic's mate. In *Advances in Artificial Intelligence: European Conference of Aritificial Intellingece*, pages 79–94, 1985.

[Brooks, 1980] R.A. Brooks. Symbolic reasoning among 3-d models and 2-d images. *Artificial Intelligence*, :285–348, 1980.

[Cahn von Seelen, 1988] U. Cahn von Seelen. *Ein Formalismus zur Beschreibung von Bewegungsverben mit Hilfe von Trajektorien (A formalim for the description of motion verbs with the help of trajectories).* Technical Report, University of Karlsruhe, dept. of Computer Science, 1988.

[Cameron and Lin, 1991] E.J. Cameron and Y.J. Lin. A real-time transition model for analyzing behavioural compatibility of telecommunication services. *ACM*, :101–111, 1991.

[Campos, 1992] M. Campos. *Robotic Exploration of Material and Kinematic Properties of Objects.* PhD thesis, University of Pennsylvania, 1992.

[Connell and Brady, 1987] J.H. Connell and M. Brady. Generating and generalizing model of visual objects. *Artificial Intelligence*, 31:159–183, 1987.

[Cutkosky, 1989] M. R. Cutkosky. On grasp choice, grasp models, and the design of hands for manifacturing tasks. *IEEE Transactions on Robotics and Automation*, 5(3):269–279, 1989.

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

[Davis, 1991] E. Davis. *The Kinematics of Cutting Solid Objects.* Technical Report 541, New York University, Dept. of Computer and Information Science, Courant Institute of Mathematical Sciences, 251 Mercer Street NY, 1991.

[Dieter, 1961] G.E. Dieter. *Mechanical Metallurgy.* McGraw-Hill Book Co. Inc., 1961.

[Freeman and Newell, 1971] P. Freeman and A. Newell. A model for functional reasoning in design. In *Proceedings of IJCAI-71*, pages 621–632, 1971.

[Goodall, 1986] J. Goodall. *The Chimpanzees of Gombe, Patterns of Behaviour*, chapter Object Manipulation, pages 535–564. Belknap Press of Harvard University Press, 1986.

[Grace, 1989] R. Grace. *Interpreting the Function of Stone Tools*, pages 9–16, 106–115. B.A.R. International Series, Oxford, England, 1989.

[Gupta, 1991] A. Gupta. *Surface and Volumetric Segmentation of Complex 3-D Objects Using Parametric Shape Models.* PhD thesis, University of Pennsylvania, 1991. Technical Report MS-CIS-91-45.

[Ho, 1989] Yu-Chi Ho. Dynamics of discrete event systems. *Proceedings of the IEEE*, :3–6, 1989. Introduction to a special issue devoted to DEDS.

[Hopcroft and Ullman, 1979] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages and Compuation.* Addison Wesley, 1979.

[Iberall, 1987] T. Iberall. The nature of human prehension: three dextrous hands in one. In *International Conference on Robotic Research*, pages 396–401, 1987.

[Iberall *et al.*, 1988] T. Iberall, J. Jackson, L. Labbe, and R. Zampano. Knowledge-based prehension: capturing human dexterity. In *International Conference on Robotic Research*, pages 82–87, 1988.

[Ikeuchi and Hebert, 1990] K. Ikeuchi and M. Hebert. Task oriented vision. In *Image Understanding Workshop*, pages 497–507, 1990.

[Klatzky *et al.*, 1987a] R. Klatzky, S. Lederman, and C. Reed. *There's more to touch than meets the eye: The salience of Object's Attributes for haptics with and without vision.* Technical Report, University of California, Cognitive Science, Santa Barbara, CA, 1987.

[Klatzky *et al.*, 1987b] R. Klatzky, B.P. McCloskey, S. Doherty, J.W Pellegrino, and T. Smith. Knowledge about hand shaping and knowledge about objects. *Journal of Motor Behaviour*, 19(2):187–213, 1987.

[Košecká, 1992] J. Košecká. *Control of Discrete Event Systems.* Technical Report, University of Pennsylvania, Dept. of Computer and Information Science, Philadelphia, Pa., 1992.

[Lederman and Klatzky, 1987a] S. Lederman and R. Klatzky. Hand movements: a window into haptic object recogntion. *Cognitive Psycology*, 19():342–368, 1987.

[Lederman and Klatzky, 1987b] S. Lederman and R. Klatzky. Haptic classification of common objects: knowledge-driven exploration. *Cognitive Psycology*, 22():421–459, 1987.

[Lederman and Thorne, 1986] S. Lederman and G. Thorne. Perception of texture by vision and touch: multidimensionality and intersensory integration. *Journal of Experimental Psycology*, 12(2):169–180, 1986.

[Liu *et al.*, 1989] H. Liu, T. Iberall, and G.A. Bekey. The multi-dimensional quality of task requirements for dextrous robot hand control. In *International Conference on Robotic Research*, pages 452–457, 1989.

[Lowry, 1982] M.R. Lowry. Reasoning between structure and function. In *Proceedings fo the Image Understanding Workshop*, pages 260–264, Science Applications In., Mclean VA, 1982.

[Lozano-Perez *et al.*, 1987] T. Lozano-Perez, J.L. Jones, E. Mazer, P.A. O'Donnell, W.E. Grimson, P. Tournassound, and A. Lanusse. Handey: a robot system that recongnizes, plans and manipulates. In *International Conference on Robotic Research*, pages 843–849, 1987.

[Mukerjee, 1991] A. Mukerjee. Accidental alignments - an approach to qualitative vision. In *Proceedings of the 1991 IEEE: International Conference on Robotics and Automation*, pages 1096–1101, 1991.

[Nguyen and Stephanou, 1990] T. N. Nguyen and H. E. Stephanou. A topological algorithm for continous grasp planning. In *International Conference on Robotic Research*, pages 670–675, 1990.

[Nguyen and Stephanou, 1991] T. N. Nguyen and H. E. Stephanou. A computational model of prehensihility and its applications to dextrous manipulation. In *International Conference on Robotic Research*, pages 878–883, 1991.

[Oakley, 1976] K. P. Oakley. *Man the Tool-maker*, pages 1–23. University of Chicago Press, Chicago, Ill., 1976.

[Ostroff, 1992] J.S. Ostroff. *Tutorial on Specification of Time*, chapter Survey of Formal Methods for the Specification and Design of Real-Time Systems. 1992.

[Pearl, 1988] J. Pearl. *Probabilistic Reasoning In Intelligent System: Networks of Plausible Ingerence*, chapter 9 Non-bayesian Formalism For Managing Uncertainty: 9.1 The Dempster-Shafer Theory, pages 416–447. Morgan Kaufmann Publishers Inc., San Mateo, CA, 1988.

[Pellegrino *et al.*, 1989] J.W. Pellegrino, R. Klatzky, and B. P. McCloskey. Timecourse of preshaping for functional responses to objects. *Journal of Motor Behavior*, 21(3):307–316, 1989.

[Pollard and Lozano-Perez, 1990] N. S. Pollard and T. Lozano-Perez. Grasp stability and feasibility for an arm with an articulated hand. In *International Conference on Robotic Research*, pages 1581–1585, 1990.

[Ramadge and Wonham, 1989] P.J.G. Ramadge and W. M. Wonham. The control of discrete event systems. In *Proceedings of the IEEE*, pages 81–97, 1989.

[Reid, 1983] G.J. Reid. *Linear System Fundamentals: Continuous and Discrete, Classic and Modern*. McGraw-Hill, Inc, 1983.

[Rosch, 1973] E. H. Rosch. *Cognitive Development and the Acquisition of Language*, chapter On the Internal Structure of Perceputal and Semantic Categories, pages 111–144. Academic Press, NY, 1973.

[Salganicoff, 1992] M. Salganicoff. *A Robotic System for Learning Visually-Driven Grasp Planning*. Technical Report, University of Pennsylvania, 1992. Ph.D. Dissertation Proposal, In Preparation.

[Sinha, 1992] P. Sinha. *Robotic Exploration Of Surfaces and Its Application to Legged Locomotion*. PhD thesis, University of Pennsylvania, 1992. Technical Report MS-CIS-92-12, GRASP Lab 301.

[Smith, 1990] E. E. Smith. *Fundations of Cognitive Science*, chapter 13 Concepts and Induction, pages 501–526. The MIT Press, Cambridge, Massachusetts, 1990.

[Smith and Medin, 1981] E. E. Smith and D. L. Medin. *Categories and Concepts*. Harvard Unversity Press, Cambridge, Massachusetts, 1981.

[Sobh, 1991] T. Sobh. *Active Observer: A Discrete Event Dynamic System Model for Controlling and Observer under Uncertainty*. PhD thesis, University of Pennsylvania, 1991. Technical Report MS-CIS-91-99.

[Stansfield, 1990] S.A. Stansfield. Knowledge-based robotic grasping. In *International Conference on Robotic Research*, pages 1270–1275, 1990.

[Stark and Bowyer, 1990] L. Stark and K. Bowyer. Achieving generalized object recogntion through reasoning about association of function to structure. *AAAI-90: Workshop on Qualitative Vision*, ():, 1990.

[Stark and Bowyer, 1991] L. Stark and K. Bowyer. Generic recognition through qualitative reasoning about 3-d shape and object function. In *Proceedings of CVPR*, pages 251–256, 1991.

[Wilks] Y.A. Wilks. Methodological questions about AI: Approaches to understanding. In *Journal of Pragmatics*, pages 69–84.

[Willis, 1989] D. Willis. *The Hominid Gang: Behind the Scenes in the Search for Human Origins*, chapter Handy Man, pages 86–110. Viking Penguin, New York, N.Y., 1989.

[Winston *et al.*, 1984] P.H. Winston, T.O Binford, B. Katz, and M. Lowry. *Learning Physical Descriptions from Functional Defintions, Examples, and Precedents*, chapter Connecting Perception to Action, pages 117–135. MIT Press, Cambridge, Ma., 1984.