

GRANULARITY HIERARCHIES

GORDON MCCALLA, JIM GREER, BRYCE BARRIE AND PAUL POSPISIL

ARIES Laboratory, Department of Computational Science
University of Saskatchewan, Saskatoon, Canada S7N 0W0

Abstract—Many artificial intelligence systems implicitly use notions of granularity in reasoning, but there is very little research into granularity itself. An exception is the work of Hobbs, who outlines several characteristics of granularity. In this paper, we describe an approach to representing granularity which formalizes in computational terms most of Hobbs' notions, often refining and extending them. In particular, two types of granularity have been delineated: aggregation and abstraction. Objects can be described at various grain sizes and connected together into a granularity hierarchy which allows focus shifts along either aggregation or abstraction dimensions.

Granularity hierarchies can be used in recognition. An especially good domain for granularity-based recognition is educational diagnosis. In an intelligent tutoring system, the ability to recognize student behaviour at varying grain sizes is important both for pedagogical reasons (in order to respond to the student at various levels of detail) and for reasons of robustness in diagnosis (obscure student behaviour can be recognized at least at a coarse grain size). We briefly discuss how we have used granularity hierarchies in the recognition of novice LISP programming strategies, and show how this enhances recognition and leads toward planning appropriate feedback for the student.

1. INTRODUCTION

Human perception benefits from the ability to focus attention at various levels of detail and to shift focus from one level to another. The grain size at which people choose to focus affects not only what they can discern but what becomes indistinguishable, thus permitting the mind to ignore confusing detail. However, the explicit study of granularity in artificial intelligence has been essentially overlooked. Hobbs [1] provided a preliminary discussion of granularity when he considered the formalization of certain aspects of granularity, describing some characteristic features and providing a preliminary description of these features.

The apparent relationship between hierarchically organized knowledge, especially semantic networks, and the notion of shifts in grain size, has directed our search toward a semantic network style of representation for granularity. In this paper, we augment the work of Hobbs by refining and extending his notions of granularity, by proposing a coherent and unifying framework that ties together these notions of granularity, and by providing an inference mechanism for using granularity in recognition and diagnosis. We believe that our framework captures many important attributes of granularity in the representation of complex knowledge and provides a sound foundation for model-based diagnosis. We also show how the explicit representation of granularity places new requirements on traditional semantic network concepts such as ISA and Part-Of.

1.1 Semantic Networks Concepts

Semantic networks provide an expressive representation scheme for many kinds of structured knowledge. The early use of semantic networks focussed on representing related concepts in a graphical form using a variety of link types. Unfortunately, all too often the semantics of the links were implicitly embodied in the names of the links themselves. That is, the meaning of traversing

We would like to acknowledge the contributions of all of the members of the ARIES Laboratory, particularly Randy Coulman, Luong Nguyen, and Mike Oliphant who have programmed our granularity-based reasoning system. We would like to thank the two anonymous reviewers for their careful reviews and useful comments. The financial support of the Natural Sciences and Engineering Research Council of Canada and the University of Saskatchewan are also gratefully acknowledged.

Typeset by $\text{\AA}\text{\AA}\text{\AA}\text{-T}\text{E}\text{X}$

some link, such as "number of legs" or "likes," might be subtle and context-dependent; and it was consequently difficult to give precise semantics to such links. For example, although most humans might be attached to the number two by the "number of legs" link, a craftsman with a lathe might be attached to his entire inventory of furniture legs with the same link. One must be careful in precisely defining link semantics to ensure, for example, that a link's meaning is not unduly modified by the objects it connects. The necessity for precise link semantics was argued by, among others, Woods [2] and Cercone and Schubert [3]. Subsequent research has investigated how to incorporate semantics in semantic networks.

Specific kinds of links, especially ISA and Part-Of, have been thoroughly studied with the goal of defining a precise semantics for them [4,5]. ISA links relate specializations to more general classes of objects. Part-Of links relate parts of objects to wholes. A major difficulty with Part-Of relations revolves around the inability to completely characterize any object by its parts alone; the notion that a whole is more than the sum of its parts. The additional ingredient not expressible by the Part-Of links alone is a second-order relation that applies to the links as well as to the parts. Further, different clusters of related parts seem to be able to characterize the same object from varying viewpoints; that is, the same object may consist of a variety of sets of parts. For example a hammer can be viewed as an object consisting of a head and a handle (for driving nails) or as consisting of a claw and a handle (for removing nails).

In order to represent the notion of granularity explicitly, we use *abstraction* and *aggregation* relations, which are somewhat like ISA and Part-Of. Before going into details of our representation of granularity, it is useful first to consider our application area. Our research in model-based diagnosis in intelligent tutoring systems has motivated the development of our theory of granularity. We claim that the generality of our approach for representing and using granularity is not limited to intelligent tutoring systems, but demonstrates that this area poses hard problems whose solutions benefit all of AI.

1.2 An Application for Granularity: Diagnosis in Intelligent Tutoring

Our interest in granularity has arisen through the need for flexible and robust reasoning in intelligent tutoring systems. Granularity is an important part of instruction for two reasons. The first reason involves pedagogy. The level of detail at which a tutor chooses to present a topic, combined with the level at which the student interprets the presentation, will affect the student's success at understanding instruction. Both tutor and student must be "on the same instructional wavelength." Shifting grain size in instruction must proceed smoothly, guided either by tutor or by student.

The second reason concerns diagnosis. It is frequently difficult to precisely diagnose a student's problems. However, it is often possible to understand generally what a student is attempting to do. This knowledge can be used in designing appropriate feedback to the student and in focussing on points of ambiguity and misunderstanding. Thus in educational diagnosis, perhaps unlike other domains, being able to recognize student strategies at coarse grain sizes is often useful. The emphasis in this paper is on how granularity impacts diagnosis. Brecht [6] discusses how it can be used to enhance the pedagogical capabilities of an instructional planner.

In this paper we argue that reasoning based on an explicit representation of granularity is useful in overcoming some of the problems caused by the uncertainty in the interaction between an intelligent tutoring system and a student. Our work rests on the premise that objects (such as strategies for solving problems in complex domains) can be organized in granularity hierarchies. Since detailed recognition is not always possible, it is desirable to be able to account for a student's global behaviour, at least at a general level. In granularity-based diagnosis of students' programming behaviour, strategies can be recognized and explained at various levels of detail. By systematically varying granularity, an intelligent tutoring system can better represent complex problem-solving techniques and can recognize students' problem-solving strategies at various levels.

In order to justify these claims and to address the research questions presented earlier, we have implemented a granularity-based system that recognizes LISP programming strategies and understands student programs. This recognition system forms the core of the student response

analyst in the SCENT-3 architecture [7]. SCENT-3 is the latest incarnation of the multi-faceted SCENT (Student Computing ENvironmentT) project, which is a testbed for research related to building an intelligent programming advisor for novice LISP programmers.

2. THE PREVALENCE OF GRANULARITY

Although systems capable of making shifts in perspective from high level to low level and vice versa are not uncommon, few AI research projects have used any explicit notion of granularity. Clancey, in his work on hierarchical classification [8] and Guidon [9], makes use of abstraction in the recognition of student strategies as they learn to perform medical diagnosis. This type of recognition has not been interpreted in the context of granularity. The hierarchical reasoning used by various planning systems (e.g., [10]) can be interpreted as granularity-based focus shifts. Mackworth and Havens' [11] research into the use of hierarchically structured knowledge to guide computer vision systems is closely related to the kinds of focus shifting we use in granularity-based reasoning. The representation of knowledge in most semantic network schemes (see [12] and [13]) is based on hierarchical relations, which can be interpreted in terms of granularity, but are usually interpreted in terms of organizational and inference procedures such as inheritance.

An exception to the dearth of work on granularity is Hobbs' treatise which attempts to explicitly delineate the nature of granularity and to show how granularity can be used in representation and reasoning [1]. Hobbs describes the following characteristics and properties of granularity:

- **Relevant predicate set (\mathfrak{R}).** Given a view of the world, i.e., a particular situation of interest, only certain selected predicates from the global theory of the world are of interest. These are called "relevant predicates." These must be determined locally since they constitute the perspective from which the world is viewed in a particular situation. For example, predicates relevant for describing a forest might include its size and its shape; predicates about an individual tree in the forest such as type of bark or number of leaves would not be relevant at the grain size of forest.
- **The indistinguishability relation (\sim).** Pairs of objects (interpreted broadly as objects, events, actions, agents, etc.) in the domain of interpretation are considered to be indistinguishable if and only if no relevant predicate can distinguish between them. Thus $x \sim y \iff (\forall p \in \mathfrak{R})(p(x) \equiv p(y))$. Using the forest example again, the individual trees are indistinguishable from one another at the grain size of forest.
- **A simplification mapping (κ).** A detailed view of the world may be collapsed to a more abstract (simpler) view by means of a function κ which maps the objects at one grain size to a simpler set of equivalence classes of objects at a coarser grain size. κ also maps relevant predicates at the finer grain size onto new relevant predicates which make objects within the coarser-grained equivalence classes indistinguishable. Thus, for some equivalence class C in the simpler theory, if $\kappa : v \rightarrow C$ and $\kappa : w \rightarrow C$, then for all predicates $\kappa(p)$ in the simpler theory, v is indistinguishable from w . For example, a grain shift from rain forest to forest allows a simpler view of forest, a view where rain forests are indistinguishable from other kinds of forests such as redwood forests or coniferous forests.
- **Articulation.** Articulation is the translation from a coarse-grained to a finer-grained theory. Although Hobbs only talks in general terms about articulation, such decomposition would presumably be carried out using a mapping like κ^{-1} , which defines the classes of indistinguishable objects at finer grain sizes. It is sometimes useful to shift focus to finer grained levels, from forests to trees or from forests to particular kinds of forests. Articulation turns out to be especially useful in intelligent tutoring systems where students are encouraged to deepen their understanding along many dimensions of granularity.
- **Idealization.** Often the need to differentiate between objects at a coarse grain size forces the imposition of an arbitrary boundary between these objects, a process of idealization that is necessary to preserve the integrity of the coarse-grained classification. For example, if temperatures in the 60's form one such object and temperatures in the 70's form another, a predicate that could distinguish these two objects would need to be capable of distinguishing 69.9 from 70. Hobbs claims that this seems counter-intuitive because a distinguishing predicate involving very fine-grained measurements ($\pm 0.1^\circ$) is used to dis-

tinguish between two classes defined in much coarser terms, but he prefers this to fuzzy or probabilistic approaches.

The theoretical framework for granularity described by Hobbs has been the starting point for our investigations. We have been able to reinterpret Hobbs' notions in a computational formalism which both elaborates and refines the concept of granularity, as will be discussed in the next section.

3. DEFINING A REPRESENTATION FOR GRANULARITY

On closer analysis of granularity, it becomes apparent that there are at least two dimensions along which granularity must be interpreted: *abstraction*, corresponding to shifts in focus from general to specific or vice versa; and *aggregation*, corresponding to shifts in focus through part-whole relationships. We propose a hierarchical representation for granularity in objects, roughly equating granularity with level shifts in a directed graph. Nodes in the graph are thought of as objects (broadly interpreted as in Hobbs), with links representing two distinct granularity relations, abstraction and aggregation.

Formally, a granularity hierarchy, γ , consists of a finite set of objects, \aleph , linked by granularity relations, i.e., the asymmetric binary relations $\overset{A}{C}: \aleph \rightarrow \aleph$ and $\overset{P}{C}: \aleph \rightarrow \aleph$ (corresponding to abstraction and aggregation respectively)

$$\text{for } n_i, n_j \in \aleph \quad n_i \overset{A}{C} n_j \equiv \text{abstr}(n_j, n_i),$$

which may be read n_j is an abstraction of n_i , or alternatively, n_i is a specialization of n_j , and

$$\text{for } n_i, n_j \in \aleph \quad n_i \overset{P}{C} n_j \equiv \text{aggreg}(n_j, n_i),$$

which may be read n_j is an aggregation containing n_i , or alternatively, n_i is a part of n_j . These two relations provide the links for a granularity hierarchy representing objects related by abstraction and/or aggregation. Objects may be maximal aggregations, which by definition are those objects which are not parts of any other objects, i.e., n is a maximal aggregation iff $\nexists n_i$ such that $n \overset{P}{C} n_i$.

A principal abstraction hierarchy, consisting of only these maximal aggregations, is a uniquely-rooted, directed acyclic graph with links corresponding to abstraction relations, $\overset{A}{C}$, connecting the maximal aggregation objects. The principal hierarchy represents the simplest (most aggregate) objects we wish to consider, arranged in terms of relative abstraction. This hierarchy is rooted at the most abstract object, and bottoms out at the most specialized objects. The principal hierarchy is important for recognition, in that it provides a complete description at the coarsest aggregation grain size of all the concepts represented in the granularity hierarchy. In instructional applications, this principal hierarchy is very useful for identifying the most appropriate grain size at which to interact with the student. Figure 1 shows a fragment of a principal abstraction hierarchy for a set of LISP strategies at various grain sizes.

Each maximal aggregation object is the root of another directed acyclic graph, this time linked by aggregation relations $\overset{P}{C}$. Each object in this dimension is a component part of the maximal aggregation object, or a part of one of its parts, etc. Figure 2 shows the aggregation hierarchy rooted at the maximal aggregation "Cdr Recursion" object shown in Figure 1.

Abstraction and aggregation can be thought of as orthogonal dimensions of granularity, relating objects in the granularity hierarchy with one another. The entire granularity hierarchy is connected to the most general object, (*root*), in the principal abstraction hierarchy according to the connectivity axiom,

$$\forall n_i \in \aleph, (n_i \overset{A^*}{C} \text{root}) \vee \exists n_j \in \aleph \text{ such that } (n_i \overset{P^*}{C} n_j) \wedge (n_j \overset{A^*}{C} \text{root}),$$

where $\overset{A^*}{C}$ and $\overset{P^*}{C}$ are the transitive closures of $\overset{A}{C}$ and $\overset{P}{C}$ respectively. This implies that from any object the root can be accessed by shifting focus to more aggregate grain sizes until reaching

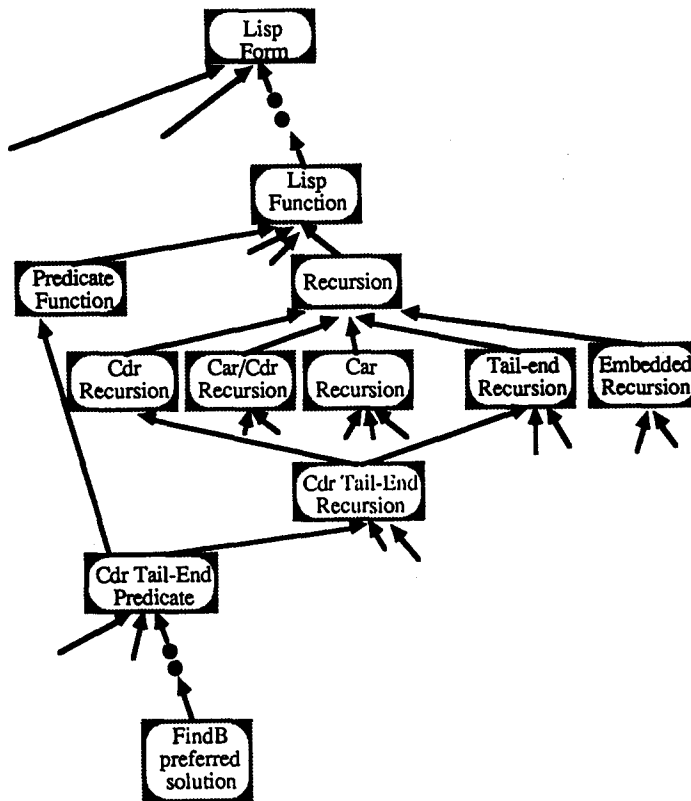


Figure 1. A fragment of the principal abstraction hierarchy.

a maximal aggregation, followed by shifting to more abstract grain sizes along the abstraction dimension. In other words, this axiom guarantees the existence of the principal hierarchy. The root in the above figures is the object "Lisp Form." In addition to these linkages, we permit, but do not require, the existence of abstraction relations between finer-grained objects in the aggregation dimension. This provides for abstraction relationships among parts of finer aggregations. It also allows other paths to the root from a given object to exist besides the path through the principal hierarchy required by the connectivity axiom. Such alternative paths can be useful in enhancing the robustness of recognition. The resulting granularity hierarchy is a partial order of objects characterized by its two orthogonal relations. A more complete description of this formalization of granularity is given in other papers [14,15].

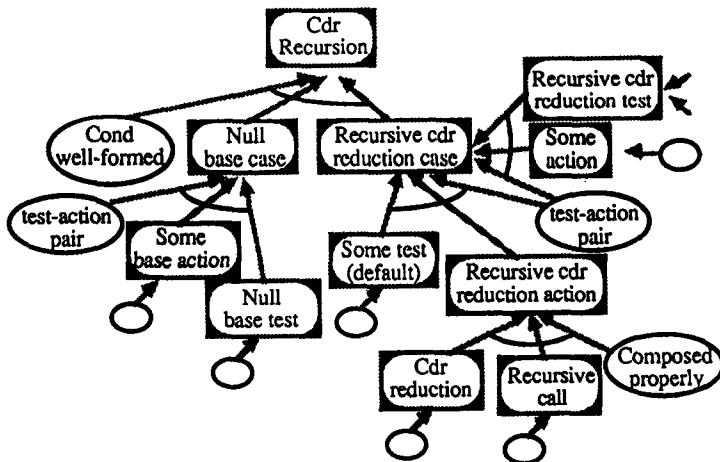


Figure 2. A fragment of the aggregation hierarchy for Cdr Recursion.

Our approach to granularity can be interpreted in terms of Hobbs' characteristics, and in fact often refines and extends his notions. Each characteristic will be considered in turn:

- **Indistinguishability and distinguishability.** Indistinguishability can be defined using the explicit structure of the hierarchy, γ , rather than some set of relevant predicates. The indistinguishability relation is meaningful relative to a particular level of granularity (in abstraction and in aggregation). Objects are considered indistinguishable if they are finer grained than the object under consideration. Hence, there is a notion of indistinguishability with respect to each and every object in the hierarchy (denoted as \sim_n). We define indistinguishability between objects n_i and n_j with respect to an object n as

$$n_i \sim_n n_j \equiv ((n_i \stackrel{P_s}{C} n) \wedge (n_j \stackrel{P_s}{C} n)) \vee ((n_i \stackrel{A_s}{C} n) \wedge (n_j \stackrel{A_s}{C} n)),$$

which indicates that descendants of an object, in either the abstraction or aggregation dimension, are indistinguishable from the point of view of that object.

A related characteristic, not directly discussed by Hobbs, is distinguishability. Objects which can be distinguished with respect to a given object are precisely the other siblings of that object in both the aggregation and abstraction dimensions. Note that both distinguishability and indistinguishability are relations relative to a parent object. This implies that if an object has two parents, say, there will be two sets of distinguishable siblings, one set for each parent. This relation gives an object knowledge of the perspective it embodies relative to local alternative perspectives. Clearly some objects may be neither indistinguishable nor distinguishable relative to some given object. Such objects are simply irrelevant to the given object, at least with respect to granularity considerations. This shows that indistinguishability and distinguishability are not simply duals of one another. Further evidence for this non-duality is that monotonic additions to the granularity hierarchy may affect previous distinguishability relations among objects but will not affect previous indistinguishability relations, while deletions from the hierarchy may affect both.

Looking at Figure 1, from the point of view of "Cdr Recursion," "Car/Cdr Recursion," "Car Recursion," "Tail-end Recursion," and "Embedded Recursion" can all be distinguished from one another (being siblings). Further, "Cdr Tail-End Recursion," "Cdr Tail-End Predicate," etc. down to "FindB Preferred Solution" are all indistinguishable; and "Predicate Function," "Recursion," "Lisp Function," and "Lisp Form" are undefined. Indistinguishability and distinguishability have turned out to be useful in localizing focus shifts (see articulation and simplification below).

- **Relevant predicates.** We refine Hobbs' notion of relevance to incorporate two kinds of predicates: observations and K-clusters. These not only define the aggregation level of granularity of an object, but are utilized in object recognition at a particular level of abstraction granularity, as will be discussed in Section 4.

An observation is a predicate completely determined by evidence obtained directly from the environment through the evaluation of some observer function (ofunction). An observation may be associated with an object for the purposes of distinguishing it from other objects, or to contribute to the recognition of the object on the basis of outside evidence. In fact, at some grain size(s) objects must be recognizable from direct observations alone. Thus aggregation "bottoms out," and every finest grained object along the aggregation dimension ultimately must be recognized by a direct observation.

Frequently an object can be characterized in a number of ways, even at a specific grain size. It may be characterized in terms of its parts and predicates that describe how those parts fit together; it may be characterized by some subset or other set of parts under certain conditions; or it may be characterized by predicates that are quite independent of its parts. For example, a recursive LISP function may be characterized by 1) a combination of recursive function parts (base cases and recursive cases) properly assembled, 2) an infinite tail-end recursion (as in an interpreter), or 3) particular behaviour in traces of function calls and returns. We name each different way of characterizing an object a "K-cluster" (after [16]). A K-cluster is a combination of observations and component parts

that characterize an object under certain conditions. Thus, K-clusters occur along the aggregation dimension of the granularity hierarchy. The K-cluster provides a mechanism for effectively grouping relevant predicates into relevance groups.

In Figure 2 there are a number of K-clusters, indicated by the arc connecting sub-aggregate descendants beneath a node. Observers in the figure are denoted as circles rather than boxes. For example, the object "Recursive cdr reduction case" has two K-clusters, one consisting of the two sub-aggregate objects "Some test (default)" and "Recursive cdr-reduction action" as well as the observation "test-action pair," and the other consisting of the two sub-aggregate objects "Recursive cdr-reduction test" and "Some action" as well as the observation "test-action pair." Each of these K-clusters represents a different way in which a recursive cdr reduction case could be framed in LISP, and hence define alternative groups of relevant "predicates" which must be "true" for the object "Recursive cdr reduction case" to be distinguished. The truth or falsity of such predicate groups is determined by a simplification function described below.

- **Simplification and articulation.** Simplification in granularity hierarchies is accomplished by a focus shift from a particular level of granularity to a coarser grain size. A simplification operator (similar to Hobbs' κ mapping) is required to guide these focus shifts. In the abstraction dimension, simplification amounts to traversing an abstraction relation, which implicitly alters the sets of distinguishable and indistinguishable objects.

In the aggregation dimension, the presence of K-clusters impacts the simplification process. Each object that could be a candidate for aggregation simplification (objects that are not maximal aggregations) is by definition a member of one or more K-clusters. Associated with each K-cluster is a pair of functions which arbitrate the simplification and articulation of K-clusters. The simplification function is essentially a local interpreter for the distinguishable objects (i.e., observations and sub-objects in the K-cluster) which determines how the objects can be put together. This focus shift to a coarser grain size makes these finer-grained parts indistinguishable. For example, in Figure 2, the simplification function associated with the K-cluster of parts of the "Cdr Recursion" object will make sure that the null base case and recursive cdr reduction case are in fact embedded in the observed `cond` function in an appropriate manner (i.e., the null case must precede the reduction case in the `cond`) before it will allow the "Cdr Recursion" object to be distinguished.

The simplification function has proven to be particularly useful for guiding aggregation focus shifts when granularity hierarchies are employed for recognition (as described in Section 4). It is the simplification function that directly captures the notion of "the whole being greater than the sum of the parts"; this function makes sure that the various parts are appropriately constituted in order to form the whole.

Articulation is accomplished by a focus shift from one level of granularity to the next finer grain size. This type of focus shift is the inverse of simplification, but has quite different semantics when applied to recognition tasks. In the abstraction dimension, articulation functions bring information from coarser levels of granularity to finer levels, analogous to inheritance in semantic networks. In the aggregation dimension, articulation functions provide context for recognition to a finer-grained object as to the role it has to play relative to its distinguishable siblings given what has happened so far during recognition. In Figure 2, for example, if a focus shift is desired from "Cdr Recursion" to "Null base case," the articulation function associated with "Cdr Recursion" would make sure that all candidate null base cases were properly situated in relationship to other possible `cond` clauses.

- **Idealization.** The need to impose fine-grained distinctions or boundaries between coarser-grained objects seemed worrisome to Hobbs. This is a non-problem in our framework. In our approach a measurement to distinguish an object from its distinguishable sibling objects must be embedded in a sub-object in the aggregation dimension. The absolute grain size of this measurement is meaningless; the measurement only has meaning relative to the position in the aggregation hierarchy of the sub-object in which it is embedded. Thus, in Figure 2 a measurement taken to recognize a "Null base case" would be of a coarser grain size than a measurement taken to recognize "Null base test," despite possible

similarities from a real world perspective of the precision of the measurements. Thus, the degree of precision of a measurement in the real world and the granularity of the object containing that measurement have no relationship to one another.

Another issue related to idealization is that of specifying a set of objects at some fixed distance from a given object along a granularity dimension as collectively forming a "theory of the world" at that grain size. However, there is really no way to compare objects at a given depth which do not share a parent; the granularity levels are only locally meaningful. Granularity is relative, not absolute. For example in Figure 1 nothing can be said about the relative granularity of "Embedded Recursion" and "Cdr Tail-End Predicate," despite the fact that they are both two levels down the abstraction hierarchy from "Lisp Function." Similarly, in Figure 2 "Recursive cdr reduction action" is not comparable to "Null base test." Thus, it is not possible to specify a complete theory at a particular grain size because objects are mostly incomparable at any particular level. This allows us to avoid the level-preserving intermediate dummy nodes required in Mulder's scene recognition vision system [17]. For a further discussion of our approach to relative granularity see [14].

4. USING GRANULARITY IN LISP PROGRAM STRATEGY RECOGNITION

Our main interest in using granularity has been to recognize the strategies novice students employ when they solve simple recursive LISP programming problems. Many problems arise when a system tries to diagnose errors in a student's programming strategies solely on the basis of the student's code. The space of possible misconceptions is immense; students seem to be infinitely creative in finding new ways to go wrong. There is often uncertainty about whether or not a diagnosis is correct; diagnoses are difficult to validate. The student is constantly learning and adapting, so multiple diagnostic probes often yield contradictory evidence. Human teachers and tutors compensate for these difficulties by making approximate (but as precise as possible) diagnoses of student conceptions and misconceptions.

This work has been carried on in the context of the SCENT project [7], which is investigating issues in the construction of an intelligent tutoring system that dispenses advice to students as they learn to program in LISP. Using granularity, we are able to recognize student programming strategies at various levels of detail. This is useful pedagogically and enhances the robustness of the diagnostic system by allowing at least coarse grained recognition of bizarre student solutions. In fact, in the granularity hierarchy that we have implemented for LISP programming, strategy recognition at some grain size is guaranteed, albeit sometimes at only a coarse grain size.

We have experimented with a variety of approaches to the control of granularity-based recognition. Barrie's pioneering work in this area [18] led to an approach that combined top-down and bottom-up control. Attached to objects were two kinds of methods: weak recognizers and strong recognizers. Weak recognizers were used top-down as heuristics to efficiently select promising recognition paths while strong recognizers were used bottom-up to do the detailed recognition. This approach was very successful, but also quite ad hoc: recognizers had to be designed specifically for each strategy object, and the knowledge engineering task was correspondingly difficult. Further experiments focussed on bottom-up recognition through the principal abstraction hierarchy from task-level strategy objects followed by top-down recognition down the aggregation hierarchy from each principal abstraction object. The recognition algorithm was imposed globally so that there was no need to attach recognizers to each object. This approach worked efficiently and well because the focus was immediately on task-specific strategies, and other supposedly irrelevant objects could be ignored. Unfortunately, in the education context, it is often these supposedly irrelevant objects which are of the most interest! Thus, current experiments are investigating top-down recognition through the hierarchy, only elaborating strategy objects which are completely recognized. Recognition bottoms out either at primitive strategy objects or at coarser-grained strategy objects which are only partially recognized. Heuristic decisions guide the further elaboration of such partially recognized strategy objects according to the needs of the tutoring system.

As our various experiments in recognition control have shown, our approach to granularity is adaptable to various control strategies. We believe that there is no one best control algorithm for recognition, any more than there is one best parsing technique. It all depends on the nature

of the application and the goals of the recognition system. In our discussion of recognition below, issues of control have been ignored; the important point is to see how articulation and simplification occur between objects, and to ground our granularity notions in a concrete example in the tutoring domain.

Consider the following programming task, FindB: write a LISP function that returns T if the atom B exists at the top-level of a list, and returns NIL otherwise. Assume our granularity-based recognition system is asked to recognize the following simple LISP program, submitted as a student's solution to the FindB problem:

```
(defun FindB (Lst)
  (cond ((null Lst) nil)
        ((atom 'B) t)
        (t (cons nil (FindB (cdr Lst))))))
```

This is a flawed solution, in that the task-specific test "(atom 'B)" does not check for a B in the list, and there is no need for the "cons" composition after the recursive call. Most program recognition systems would find such a perturbed solution hard to recognize unless these particular perturbations had been explicitly anticipated. In our granularity-based approach, this program could at least be recognized as "Cdr Recursion" (see Figure 2).

In order to be a cdr recursion, a program must have a null base case, a recursive cdr reduction case, and these must be put together in a well-formed conditional. A null base case, in turn, must have a null base test and some base action, both put together as a test-action pair. Observers looking at the FindB solution above would find that "(null Lst)" is a satisfactory null base test, that "nil" is a suitable base action, and that the two are formed as a test-action pair. Thus, this program has a null base case. Does it have a recursive cdr reduction case? There are two ways of having a recursive cdr reduction case, as shown by the two K-clusters in Figure 2. The relevant K-cluster here is the one requiring some test (possibly a default) and a recursive cdr reduction action combined as a test-action pair. The student's use of "t" in the third clause of the cond can be recognized as a default test, but recognizing a recursive cdr reduction action involves a further aggregation articulation requiring a cdr reduction and a recursive call, properly composed. Observers can recognize "(cdr Lst)" as a cdr reduction, the call to "FindB" as a recursive call, and that these are composed properly (i.e., that the reduction is in the argument to the recursive call). Thus, a recursive cdr reduction action can be recognized. It remains only to observe that the default test "t" and this recursive cdr reduction action form a test-action pair, which means that a recursive cdr reduction case is recognized. The recognition of both the null base case and the recursive cdr reduction case, combined with the observation that the cond is well-formed, means that a cdr reduction has occurred. Since "Cdr Recursion" is an object in the principal hierarchy, further aggregation simplification cannot occur. Despite its perturbations, the student's program has been definitely recognized as a cdr recursion.

The recognition process proceeds as recognition of "Cdr Recursion" propagates through simplification focus shifts through to coarser-grained levels of abstraction, allowing "Recursion," "Lisp Function," and "Lisp Form" all to be recognized as well (see Figure 1). In fact, once an object has been recognized at any level of aggregation, recognition normally propagates upwards through its abstraction ancestors, a fact which often allows rapid recognition of coarser grained sub-aggregate objects without needing to articulate all of their component parts. Using this information, the SCENT system knows what the student is doing at any of 4 levels of abstraction granularity, as well as being able to understand more precisely the various parts of the student's program that have been recognized in the aggregation dimension. This can be very useful for updating the student model, and for framing the discussion with the student by focussing on the parts of his or her program which are correct and well understood.

Pedagogically, it is the unrecognized parts of the student's program which usually form the basis for tutoring, however. The granularity-based recognition system is also quite useful in this regard. In the attempt to recognize the student's program, various objects elsewhere in the granularity hierarchy may have been recognized. These can include sub-aggregate objects of finer grained objects in the principal abstraction hierarchy. In particular, objects like "Cdr Tail-End

Recursion," "Cdr Tail-End Predicate," and "FindB preferred solution" may have some of their sub-parts recognized, while obviously not being able to recognize a complete K-cluster of objects. For example, "FindB preferred solution," which is the finest grained object along the abstraction dimension, would be satisfied with the null base test, but would not accept the task-specific test which should be "(eq (car Lst) 'B)." Further, it would not be satisfied with the existence of a composition step in the recursive action. These unfelicitous parts could form the basis for tutoring, or at least for inquiries of the student or the student model as to problem-solving intentions.

Complete coarse grained recognition and finer grained partial recognition constitute two different styles of qualitative uncertainty. Unlike numerical measures of uncertainty, they suggest the structural nature of the uncertainty, i.e., the particular unfilled slots in a partial recognition are known, the level at which complete recognition bottoms out is specified. This provides the rest of the tutoring system with relevant information for student modelling and pedagogical decision making. We believe that this ability to represent and to recognize objects at multiple grain sizes and with varying degrees of qualitative uncertainty is the key ingredient of a robust knowledge representation system.

The approach to granularity-based recognition discussed here has points of similarity to several other major research projects. In the domain of intelligent tutoring, the PROUST system [19] recognizes student programs at three levels: problem, goal, and plan. Our strategies are most like PROUST's plans, but unlike PROUST, we do not attempt to induce a student's goals in choosing a particular strategy (this is a role envisaged in SCENT for the student modelling component), nor do we formalize the problem description (although our work is currently progressing in this area). Instead, our approach has concentrated on strategy (plan) diagnosis, and in that regard goes considerably beyond PROUST in the formalization and use of granularity, in the delineation of both an aggregation and abstraction dimension to strategy recognition, in robustness, and in the breadth and depth of strategies dealt with.

Knowledge-based vision systems, such as the various Mapsee systems [11,17] and ALVEN and CAA [20], have strong points of similarity with our approach to recognition as well. These systems, which use hierarchies of visual knowledge to guide recognition of scenes, are similar in their organization of knowledge into aggregation and abstraction hierarchies to guide recognition. There are, of course, obvious differences in domain: the Mapsee systems recognize an idealized sketch map, ALVEN looks at medical images, and CAA analyzes electrocardiograms. None of these systems explicitly formulate recognition in terms of granularity, nor are they satisfied, in general, with only achieving coarse-grained recognition. The usefulness in an education domain of coarse grained recognition and partial fine grained recognition may not carry over very well to the scene analysis domain because applications in that domain usually require precise rather than approximate identification. There are also many technical differences between our approach and the knowledge-based vision approaches. Nevertheless, the important point is that the knowledge-based vision systems provide further evidence that the approach taken here may be widely applicable beyond program strategy recognition.

5. RELATED RESEARCH CONTRIBUTIONS

Much work has gone into the creation of a granularity-based recognition system for LISP strategies, and this work continues. One line of development has been to investigate various kinds of control paradigms. As discussed above, we have experimented with top-down, bottom-up, and task-dependent control schemes. Another line of development, explored by Pospisil [21], has been to incorporate buggy strategies into granularity hierarchies, which if recognized allow a definitive diagnosis of the student's misconceptions, and provide even more concrete information for the student modelling and pedagogic components of SCENT.

A third direction of current investigations has been the knowledge engineering of a large system in order to rigorously test this approach to recognition, to prove out its usefulness in a real domain, and to find limitations and/or to explore possible enhancements to the approach. The design of the strategy objects in this large system is based on a repository of actual solutions to several LISP problems collected from 48 novice LISP programmers (see [22] for an analysis of this data). We currently have implemented some 200 objects connected together at ten levels

of abstraction granularity and averaging approximately four levels of aggregation granularity. These objects allow the recognition of a wide variety of the basic recursive strategies used by LISP programmers. We are enhancing the coverage of the hierarchy by adding more strategy objects and by integrating into the system knowledge-based program transformations in order to reduce the observers' dependency on exact code matches [23]. We are currently attempting to reduce the need to store explicit task-dependent strategies at the finer levels of abstraction granularity and instead to generate these task-dependent strategies from the problem description as new tasks are created for students to solve.

Much work has gone into the formalization of granularity as well. We have been able to describe granularity in precise computational terms, have characterized two kinds of granularity, and have shown how this approach to granularity relates to Hobbs' properties of granularity, and in some ways refines and extends his ideas.

6. CONCLUSIONS

There are many implications that can be derived from our description of granularity. Since our work has focussed on diagnosing problem-solving strategies, it seems natural to apply granularity to general fault diagnosis. We claim that understanding a student is more complex and fraught with difficulties than troubleshooting any but the most complex devices, since a student can easily have many misconceptions (faults) which interact and interfere with each other at many levels of detail. Furthermore, it is often inappropriate to make numerous probes or measurements, and most important, it is often impossible to verify the correctness of a proposed diagnosis in an educational setting. In such complex domains, perhaps granularity-based diagnosis can provide a compromise between intractable first principles approaches to diagnosis and empirical diagnostic (expert systems) models. The hierarchical structure of granularity representations can focus analysis on relevant classes of components or on relevant modules. It may be adequate to provide a coarse-grained diagnosis, localizing a fault within some module, when a more detailed diagnosis is impossible or too costly.

There is also an interesting relationship between our approach to granularity and inheritance and logical inference problems in generalization (ISA) hierarchies. In granularity hierarchies, each object interprets the world locally and focus shifts are mediated by simplification and articulation functions. The representational economy that inheritance provides to ISA relations can readily be modelled in granularity hierarchies by a set of articulation functions carrying along relevant predicates to the finer-grained object. For example, one articulation function that relates the object "bird" to "sparrow" could carry along relevant attributes including perhaps feathers and the ability to fly. In contrast, the articulation function between "bird" and "emu" would carry only predicates relevant to emus. In this way, the articulation function can be seen as a second order function that can (among other things) govern inheritance.

The simplification functions in granularity hierarchies mediate the propagation of recognition. Similar to uncertainty propagation in approximate reasoning, the second order local functions for simplification must be capable of propagating just the right sort of recognition to the coarser-grained level. Thus, simplification functions may decide to downplay uncertainty in recognition that occurs at finer-grained levels of the granularity hierarchy as recognition proceeds to coarser-grained objects. In general, granularity hierarchies trade off the benefits accrued by local interpretation and second order functions against the expense of needing to represent explicitly a large amount of knowledge.

An interesting avenue for exploration is the delineation of other kinds of granularity besides aggregation and abstraction. For example, the human ability to carry out goals and sub-goals may suggest the existence of a dimension of granularity involving focus shifts between levels of goals. This may be especially useful if an intelligent tutoring system is to understand student motivation, at least to some degree. In addition, patterns of partial recognition in abstraction-aggregation granularity seem to be the source of new dimensions of granularity where patterns of strategies are finer-grained explanations of deeper conceptions or complex composite tasks.

In our attempts to recognize deep misconceptions in student strategies, it has become apparent that a misconception can be characterized by a recognition pattern of objects at various grain sizes in abstraction and aggregation dimensions. For example, a common student confusion between

iteration and recursion might have a definitive signature described by a particular mix of recursive calls and explicit branches in program control wholly or partially recognized in the aggregation and abstraction hierarchies. Similarly, composite tasks which collect primitive strategies (such as searches or membership tests) into some standard complex strategy (such as a sorting or a transaction update) can be represented as patterns of recognized aggregations and abstractions. Thus, another dimension of granularity among tasks seems plausible, a dimension bottoming out at the abstraction-aggregation hierarchy we have presented here. These possible new dimensions of granularity seem to fit in comfortably with the general computational framework for granularity that we provide.

We would like to explore our approach to granularity further, especially to evaluate the necessity of the principal abstraction hierarchy, to define a notion of relative granularity between coarser and finer grained objects, to investigate how groups of objects at a similar relative grain size can be put together into a coherent "theory" of the world at a particular grain size, and to look at the implications of these variously grain-sized theories for representation and reasoning. Finally, we would like to find other domains where granularity-based recognition would be useful; some possibilities which we are currently studying include recognizing strategies used in software specification and design, and recognizing strategies employed by chess players.

Representing granularity in strategies has significantly improved the expressiveness, flexibility and robustness of our tutoring diagnostic systems. We believe that explicitly investigating granularity will prove to be useful for artificial intelligence more generally, both through theoretical and practical advances; and we are optimistic about our ongoing research into the formalization and use of granularity.

REFERENCES

1. J.R. Hobbs, *Granularity*, Ninth International Joint Conference on Artificial Intelligence, Los Angeles, California, 432-435 (1985).
2. W. Woods, What's in a link: Foundations for semantic networks, In *Representation and Understanding* (D.G. Bobrow and A. Collins, Eds.), Academic Press, New York, 35-82, (1975).
3. N.J. Cercone and L.K. Schubert, Toward a state-based conceptual representation, *Proceedings IJCAI-75*, Tbilisi, USSR, 83-91 (1975).
4. R.J. Brachman, What ISA is and isn't: An analysis of taxonomic links in semantic networks, *IEEE Computer* 16 (10), 30-36 (1983).
5. D.S. Touretsky, Implicit ordering of defaults in inheritance systems, *Proceedings IJCAI-84*, Austin, Texas, 322-325 (1984).
6. B.J. Brecht, Determining the focus of instruction: Content planning for intelligent tutoring systems, Ph.D. Thesis, Department of Computational Science, University of Saskatchewan, (1990).
7. G.I. McCalla, J.E. Greer, and the SCENT Research Team, Intelligent advising in problem solving domains: The SCENT-3 architecture, *Proc. Int'l Conference on Intelligent Tutoring Systems*, Montreal, 124-131 (1988).
8. W.J. Clancey, Heuristic classification, *Artificial Intelligence* 27, North Holland, Amsterdam, 289-350 (1985).
9. W.J. Clancey, *Knowledge-Based Tutoring: The GUIDON Program*, Cambridge, MA, MIT Press, (1987).
10. E.D. Sacerdoti, *A Structure for Plans and Behavior*, Elsevier, New York, (1977).
11. A.K. Mackworth and W.S. Havens, Structuring domain knowledge for visual perception, *Proc. 7th International Joint Conference on Artificial Intelligence*, Vancouver, B.C., 625-627 (1981).
12. N.V. Findler (Ed.), *Associative Networks*, Academic Press, New York, (1979).
13. R.J. Brachman and J.G. Schmolze, An overview of the KL-ONE knowledge representation system, *Cognitive Science*, *Ablex* 9 (2), 171-216 (1985).
14. J.E. Greer and G.I. McCalla, Formalizing granularity for use in recognition, *Applied Mathematics Letters* 1 (4), Pergamon, London, 347-350 (1988).
15. J.E. Greer and G.I. McCalla, A computational framework for granularity and its application to educational diagnosis, *International Joint Conference on Artificial Intelligence (IJCAI)*, Detroit, August, 477-482 (1989).
16. N.J. Nilsson, *Principles of Artificial Intelligence*, Tioga, Menlo Park, (1981).
17. J.A. Mulder, Using discrimination graphs to represent visual knowledge, Ph.D. Thesis, TR 85-14, Dept. of Computer Science, U. of British Columbia, Vancouver, B.C., (1985).
18. J.B. Barrie, Using granularity hierarchies for strategy recognition, M.Sc. Thesis, Dept. of Computational Science, University of Saskatchewan, Saskatoon, Canada, (1988).
19. W.L. Johnson and E. Soloway, Intention-based diagnosis of programming errors, *Proc. 5th Conference of American Association for Artificial Intelligence (AAAI)*, Austin, Texas, 162-168 (1984).

20. J.K. Tsotsos and T. Shibahara, Knowledge organization and its role in temporal and causal signal understanding: The ALVEN and CAA projects, In *The Knowledge Frontier* (N. Cercone and G. McCalla, Eds.), Springer-Verlag, 221-261, (1987).
21. P.R. Pospisil, The diagnosis of strategy errors in the SCENT advisor, M.Sc. Thesis, Dept. of Computational Science, University of Saskatchewan, Saskatoon, Saskatchewan, (1988).
22. J.A. Escott and G.I. McCalla, Problem solving by analogy: A source of errors in novice programming, *Proc. International Conference on Intelligent Tutoring Systems*, Montreal, Quebec, June, 312-319 (1988).
23. X. Huang and G.I. McCalla, A hybrid approach to finding language errors and program equivalence in an automated advisor, *Proc. 7th National Conference of the Canadian Society for Computational Studies of Intelligence (CSCSI/SCEIO)*, Edmonton, Alberta, 161-168 (1988).