http://www-ksl.stanford.edu/email-archives/srkb.messages/27.html
The paper below is an earlier draft of the one in:
Proc 6th Banff Knowledge Acquisition and KBS Workshop (1991) p33.1-33.20,
Dept CS, Univ Calgary, Canada. Also: MIT AI Memo 1183 (1989).
This work became Andew Trice's MIT PhD thesis (1991). It was finally published as:
Trice A, Davis R, "Heuristics for reconciling independent knowledge bases"
in: Information Systems Research, 4 (3), pp262-288, Sept 1993.

Consensus Knowledge Acquisition[1]

Andrew Trice
(Andrew_Trice@mtsg.ubc.ca)
Faculty of Commerce
2053 Main Mall
University of British Columbia Vancouver, BC Canada V6M 2G2

Randall Davis
(davis@ai.mit.edu)
MIT Artificial Intelligence Lab
545 Technology Sq., Room 801
Cambridge, MA 02139
(617) 253-5879

Abstract

We have developed a method and prototype program, called CARTER, that assists two experts in agreeing on what knowledge should go into a single, consensus knowledge base, i.e., a knowledge base that reflects the best judgment of each of them. We show that consensus building can be effectively facilitated by a debugging approach that identifies, explains, and resolves differences in the knowledge of the two experts. CARTER's expertise is in the task of consensus building; its knowledge base is a collection of some 35 entries, each made up of a discrepancy detection procedure and a corresponding resolution procedure. Examples of the use of this knowledge are illustrated with descriptions of the program in operation, assisting with the reconciliation of two independently developed knowledge based systems.

A significant part the program's task lies in finding the places where the two knowledge bases match (e.g., two concepts that may be labeled differently but actually mean the same thing), where they differ, and where they overlap. The program displays an interesting degree of success in using circumstantial evidence in the knowledge bases themselves to help infer the degree of overlap in the intended meaning of concepts. We also show that the same fundamental technique is necessary for matching any two independently constructed representations, and that the richer the set of primitives provided by the representation in use, the more confidence we can place in the circumstantial evidence that is available.

[1]This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defence under Office of Naval Research contract N00014-85-K-0124. Additional support for this research was provided by the International Financial Services Research Center of the MIT School of Management, Digital Equipment Corporation, McDonnell Douglas Space Systems, General Dynamics Corp, and the Faculty of Commerce at the University of British Columbia.

1 INTRODUCTION

There is a curious contradiction in the current state of practice of knowledge acquisition: At a time when the view is widely shared that valuable knowledge is often distributed among multiple experts, common practice in knowledge acquisition still focuses on acquiring the knowledge of a single individual. At best, system builders may work with multiple experts by appointing one of them as "knowledge czar" and giving him the final word in any dispute, or they may simply require the experts to achieve consensus on their own, without providing any substantive assistance.

Multi-expert acquisition techniques that have been proposed to date have significant shortcomings. Work based on mathematical formulations [6] tends to be very restrictive; adaptations of established group decision-making techniques [9] offer little insight on the issue of consensus; methods like [2,18] focus on simply using knowledge from multiple sources rather than finding and resolving the inconsistencies; while methods like [7] provide only very basic assistance in detecting potential inconsistencies.

Our research is focused on developing ideas and tools to facilitate the process by which multiple experts construct a single, consensus knowledge base, the process of consensus knowledge acquisition (CKA). We have developed a system called CARTER that is capable of aiding pairs of experts in this process, by systematically finding, explaining, and suggesting resolutions for discrepancies in their knowledge.

2 HOW CAN WE HANDLE MULTIPLE EXPERTS?

The problem of reconciling multiple points of view has been an issue of study for some time in a number of different areas. Unlike previous work that has focused either on combining decision outcomes [15,1] or supporting the process of logical argumentation [5,13,21,23], our concentration is on getting agreement on the knowledge itself.

We do not want to focus on outcome alone, because we believe that the fundamental task is to reach consensus on the knowledge itself: differences in outcome may simply be symptoms of a disagreement about what to know. We choose not to focus on argumentation in the belief that the knowledge representation in use provides an appropriate framework for structuring the discussion. Instead we seek to assist the experts in detecting, deliberating over and reconciling discrepancies in their knowledge. Our approach is normative and focused on the underlying knowledge used by each expert: we want to understand how experts ought to come to agreement and we want that agreement to be about the thing we consider to be fundamental to this undertaking—the knowledge used to make the decisions.

3 BOUNDING THE PROBLEM

Several assumptions help bound the task. First, we assume the expertise to be reconciled is homogeneous in the sense that both experts are capable of solving the entire problem. This enables us to focus on resolving discrepancies rather than combining knowledge from disparate fields.

Second, we assume the experts have constructed individual knowledge bases (KBs) prior to the start of the process. This ensures that the experts can explain the reasoning they used to arrive at their answers and that that reasoning can be adequately captured by a known reasoning process. This in turn allows us to focus on knowledge debugging rather than knowledge acquisition.

This assumption presents some limits: knowledge bases are difficult enough to build that duplicates are infrequently encountered. They do, however, occur, hence the assumption is plausible. The assumption also gives us a place to start work on a fundamentally difficult problem. In addition, as we have speculated [22] and other work has demonstrated [14], the process of knowledge acquisition with a single expert can itself be viewed as establishing consensus: in this case consensus between the expert's view of the domain and the view currently embodied in the knowledge base.

Third, we assume that the experts already have some shared frame of reference, some basic set of terms and assumptions in common. Without that, determining where they agree and disagree would be difficult, not only for our system, but for any human attempting the task. Note, however, that we do not know in advance which terms those are; we presume only that there is some overlap, up to us to discover. Note, too, that the only information available to our system about the meaning of a term is to be found in the knowledge bases themselves; the system must use that information to infer whether two terms were intended to mean the same thing.

Finally, a hypothesis fundamental to our approach is that a consensus KB can perform better than an individual expert's KB. Our belief is that unearthing and resolving differences between two experts will be fundamentally synergistic, removing limitations and defects in both of their knowledge. This is plausible but of course not guaranteed: some consensus knowledge bases may not be as good as either of the originals.

4 CARTER[1]

Our prototype system plays the role of a non-binding arbitrator mediating between two experts, each of whom has constructed a knowledge base. CARTER examines each knowledge base, looking for matches and discrepancies between them, decides which discrepancy to try to resolve, and suggests possible resolutions. The two experts discuss the suggested resolutions and can choose to update their individual KBs as suggested, update them in some other manner, or not update them at all.[3] The individual KBs are then analyzed anew, until no new discrepancies can be found.

As one of its first tests, we used the program to assist two statisticians (Bruce and John) in reconciling two knowledge-based application systems they had independently developed. They had each developed a rule-based system designed to assist novice statisticians in detecting and repairing problems in applying linear regression models.[4] While the modeling process involves straightforward arithmetic formulae, there is considerable expertise in knowing the conditions on use of those formulae, determining when the data do not satisfy those conditions, and selecting possible corrective actions. The statistician's assistant system built by each expert captured some part of this expertise. The systems test for validity of the

[2]Conflict AnalyzeR for Targeted Expert Resolution

[3]That is, we allow them to "agree to disagree."

[4]Linear regressions model the relation between two variables as a linear function and indicate how well the model fits the (possibly noisy) data.

assumptions of linear regressions (e.g., the independent variables are not correlated) to prevent spurious interpretations of data, and they suggest alternatives for improving the model (e.g., model transforms, removing an outlying point from the dataset).

Bruce's knowledge base contained 54 rules, John's 32. The rule language was essentially identical to MYCIN, with rules constructed from attribute, object, value triples (e.g., if the `Defect` of the `Model` is `Multicollinearity` ...), plus a degree of certainty. The attributes, objects, and values are the primitive vocabulary of each system; Bruce's knowledge base had a total of 108 such terms, John's had 70.

As a MYCIN equivalent, the rule language was of the complexity used to build many real-world application systems and was sufficiently complex to provide an interesting challenge when attempting to match concepts. With a few dozen rules, the knowledge bases were of modest size, but each was substantive enough to provide good advice about a non-trivial domain. Hence the underlying application systems were modest but real examples of knowledge based systems.

Rule based systems were chosen as a testbed for several reasons. First, the wide use of rules meant that real examples of knowledge bases were available, providing realistic data about the kinds of discrepancies that arise.[5] Second, wide use of rules also means that the specific techniques developed in this work could be widely applicable. Finally, since (as we demonstrate below) the fundamental issue of matching representations arises everywhere, all of the difficult problems arise in working with rules, even though they are a relatively modest representation mechanism.

In analyzing the two knowledge bases CARTER was able to identify 49 out of the total of 76 of discrepancies that existed between them. When working with the experts to help them reach agreement, CARTER was able to assist them in repairing 15 of these discrepancies; this lower figure was determined primarily by the time available from the experts. We illustrate the system's knowledge and performance with examples drawn from this application, simplified slightly for the sake of presentation.

4.1 Carter's Task and Knowledge

CARTER's fundamental task is to determine how and where the knowledge bases match and where they differ. While the knowledge bases may be meaningful to the experts, to CARTER of course they are simply elaborate directed graphs. The system's task is interesting precisely because a purely syntactic match of the graphs is of relatively little value, hence this is not a standard graph matching problem. To provide useful assistance CARTER must be able to infer that two concepts that happen to be named differently in the two KBs might in fact mean the same thing, or infer that two concepts that happened to be named identically are actually being used differently in the two knowledge bases.

Strictly speaking, the task is impossible for CARTER to accomplish alone, because only the experts themselves can know for sure what they actually meant. The best CARTER can hope to do is make educated and informed guesses. The system approaches the task with

[5] John's knowledge base had been built as an MS thesis in 1986; we provided Bruce with a high level description of the problem solved by John's system and asked Bruce to independently create a knowledge base on the same topic.

this mindset, trying to find places of most plausible match and most likely discrepancy, then presenting the human experts with its findings and asking for confirmation. The system's utility is thus in asking a small set of well chosen questions, presenting the two experts with the right agenda of things to discuss (possible discrepancies and potential repairs), then making it easy to effect those repairs. The system's effectiveness is determined by how well it can focus that discussion down from the complete space (the crossproduct of concepts from the two KBs) to a smaller, more informed set of choices.

To do this CARTER relies on three forms of knowledge: a knowledge base about the kinds of discrepancies that can arise, a strategy for attacking those discrepancies in a particular order, and a body of circumstantial evidence it accumulates to infer whether two apparently different concepts in fact mean the same thing.

4.1.1 Cataloging Discrepancy Knowledge

CARTER's knowledge is organized in a catalog currently containing 35 entries, each of which in turn consists of a discrepancy detection procedure and a corresponding resolution procedure. This simple detection-resolution organization of the catalog facilitates adding new entries as we gain more experience with the task.

Among the types of discrepancies that CARTER can recognize and repair are:

1. Differences in the *character of the result*: one system is content to classify the problem (e.g., specify the nature of a defect in the data, like `Heteroscedasticity`), while the other both classifies the problem and then goes on to suggest what to do about it (e.g., do a `Log-Transform`).
2. Differences in *vocabularies*: one expert refers to the `Defects` of a regression model, while the other refers to its `Problems`, but they are referring to the same thing. Other forms of vocabulary discrepancy the system knows about include differences in representation choice (e.g., one expert represents a concept as an attribute, while the other represents it as a value) and missing terms (e.g., one KB contains values missing from the other).
3. Differences in *pattern of inference*: the experts agree on the overall vocabulary, but interconnect the terms differently, as in the case where one expert uses only an `F-test` statistic to judge the quality of a model, while the other relies on both an `F-test` and a `S-squared` statistic.
4. Differences in the *rules*: the experts agree on the vocabulary and pattern of interconnection between terms, but write different rules. For example, one expert had a rule that an `F-test` result below a specified threshold indicated that the `Quality` of the `Model` is `Poor`, while the other reasoned from the same evidence that the `Quality` of the `model` is `Fair`. Both reason from the value of the `F-test` to the `Quality` of the `Model`, but they use different rules.

4.1.2 Ordering Strategy

CARTER's overall strategy is to attack these in the order given, beginning by trying to establish agreement on the nature of the outcome. This is motivated by both the computational and negotiation character of the task. Since the computational task is one of matching two KBs that are directed graphs, any useful guidance about where to start the matching process will vastly improve the system's chances of making intelligent suggestions. Expressed in these terms, we anchor the search at the end(s) of the graph, trying first to match the outcomes, relying on the heuristic that two KBs about the same topic are likely to have the same concept(s) as their goal(s).[6] This also makes sense from the negotiation point of view: it is difficult to imagine an effective discussion about the details if the two knowledge bases are trying to arrive at different kinds of conclusions.

Establishing a discrepancy (or correspondence) in the nature of the outcome(s) typically requires dealing with the second task, discrepancies in vocabulary, because the experts may simply have named the goal(s) differently.

The process then works backward a level at a time, attempting to match the nodes (the vocabulary) directly connected to the outcome. Once this is accomplished the system can focus on the patterns of inference that link these nodes with the goal. This in turn enables comparison of the specific rules that establish the goal in each KB.

The process continues from there, moving back one level at a time through the knowledge base, attempting to establish correspondence. The system is also opportunistic in the expected sense, working from any islands of agreement it discovers.

This overall approach, working from goal node to vocabulary, to topology, to specific rules, has proved to be a powerful way to focus the attention of both our system and the human experts working with it, on a task that might otherwise present an overwhelming amount of detail.

4.2 An Example

The examples below show CARTER's first few steps in finding and resolving discrepancies in the two KBs, and illustrates 3 of the 35 types of discrepancies it can detect and resolve.[7]

4.2.1 Matching Vocabularies: Circumstantial Evidence of Semantic Match

In the example at hand, CARTER identifies the goal of Bruce's KB as `Solution` and John's as `Defect`. The concepts are named differently, but perhaps the two experts meant the same thing and simply happened to choose different names? CARTER's most fundamental task is to determine whether this is the case. To do this it accumulates four kinds of circumstantial evidence, available from the knowledge base itself:

- Are the concept labels the same? In this case they are not (`Solution` vs. `Defect`), but this can of course be an artifact of name choice or (in other circumstances) variations in spelling or abbreviation. Conversely, a match in labels is useful evidence but no guarantee of match in meaning.
- In the case of attributes, are they associated with the same object? The answer here is no, which is reasonably strong evidence of mismatch (a positive answer is considered

[6] If the KBs have more than one outcome, there is an n^2 problem in matching the outcomes, but this n involves only a very few of the total set of concepts in the KBs.

[7] The dialog CARTER presents is quite straightforward, constructed from simple text templates that ask multiple choice questions, and is omitted here to save space. A detailed example can be found in [22].

only weak evidence of match).

- In the case of attributes, are the values associated with them the same? Here the answer is no (e.g., `Solution` has as values terms like `Log-transform, Deflate-by-x,` etc., while `Defect` has as values terms like `Heteroscedasticity, Multicollinearity` etc.).
- Are they inferred from the same concepts and are they in turn used to infer the same concepts? That is, do the occupy similar places in the local topology of the knowledge base?[8] Again in this case the answer is no, accumulating substantial circumstantial evidence that `Solution` and `Defect` are labels for genuinely distinct concepts.

Weighing the evidence in the case at hand, CARTER concludes (correctly) that `Solution` and `Defect` are different concepts, hence there is a difference in the character of the result provided by each system. This matching process is conceptually quite simple, but, as we argue below, an unavoidable part of the task. CARTER's expertise is in the 35 varieties of discrepancies and repairs it knows, not the simple matching process that accumulates evidence of the sort noted above.

The system next attempts to classify this discrepancy in more detail. As noted, CARTER knows that one of the ways in which this discrepancy can occur is if the reasoning chain in one system is longer than the other, as for example if one system only classifies the problem, while the other both classifies it and goes on to suggest a solution. To determine whether this is the case CARTER searches along the reasoning chain in each KB to see whether it can find the goal attribute of one KB on the route to the goal attribute in the other KB. That is, since the two endpoints in the graphs do not match, perhaps the endpoint in one matches one of the intermediate points (conclusions) in the other.

CARTER tries to match `Solution` (the endpoint of Bruce's KB) with the intermediate conclusions in John's KB, using the same criteria of label, value, and topological correspondence. When this fails, CARTER attempts it the other way around, trying to match `Defect` of John's KB with the attributes on the route to `Solution` in Bruce's KB. This second version succeeds when CARTER finds evidence that `Defect` seems to mean the same thing as `Problem` in Bruce's KB. Although they are named differently, they are attributes that happen to share two different values (`Heteroscedasticity` and `Multicollinearity`), and they are in turn inferred from attributes that themselves appear to match (`Determinant` and `IV-Determinant`). This is substantial evidence that these two differently named concepts—`Problem` and `Defect`—are in fact identical.

This apparent match enables CARTER to propose a specific diagnosis about the discrepancy between the two KBs. Since the goal attribute of John's KB (`Defect`), seems to match a concept on the route to the goal of Bruce's KB (`Problem`), CARTER concludes that `Solution` is a concept that reflects an additional inference that only Bruce's KB performs (indicated schematically in Fig. 1).

[8]Here "local" means exactly one step away from the concept in either direction, both because of the weakness of the evidence provided by more distant matches and the expense of complete graph matching.

```
John's  Knowledge Base:  ...  ⇒ ...  ⇒  DEFECT

Bruce's Knowledge Base:  ...  ⇒ ...  ⇒  PROBLEM  ⇒  SOLUTION
```

Fig. 1: Aligning the two KBs.

The generic problem now facing CARTER is that one KB contains a term missing from the other. In response the system offers the experts the obvious choice: either add the missing attribute to the first KB or delete the extra attribute from the second.

The experts decide to add the missing attribute (`Solution`) to John's KB. CARTER then indicates the additional work that is necessary: since `Defect` and `Problem` are the same underlying concept, they must be reconciled (e.g., a common name chosen). John must also provide his version of the rules that take the additional inference step, determining `Solution` from `Defect`. Finally, those rules must in turn be compared with Bruce's and any discrepancy resolved.

4.2.2 Matching Rules

After reaching agreement on the ultimate goal for each KB, the system moves one step back in the inference chain in each, to see which attributes are used to determine the goal in each KB. It then uses the techniques just illustrated to get agreement on those attributes.

Once agreement on them is reached, the system begins to detect and remove incompleteness and inconsistency in the rules that link them. An example of detecting two inconsistent rules arises in examining the rules that each expert used to determine the `Quality` of the model:

```
       BRUCE's KB                             JOHN's KB
IF F-TEST less-than THRESHOLD         IF F-TEST less-than THRESHOLD
THEN QUALITY of MODEL is POOR         THEN QUALITY of MODEL is FAIR
```

Fig. 2: Two inconsistent rules

CARTER knows six ways in which this sort of discrepancy—conflicting rules—can arise:

1. There is a misunderstanding about the vocabulary: `Poor` and `Fair` could be synonyms, hence the rules are actually identical. This is ruled out here because the experts have already agreed on the vocabulary (in a part of the interaction not shown).
2. There is not really a mismatch because both rules should be in both knowledge bases (each expert forgot one rule that the other remembered).[9]
3. One of the rules is incorrect and should be removed.

[9] If `QUALITY` were known to be single-valued, this too could be ruled out, but this version of CARTER didn't have that information.

4. There is not sufficient precision in the vocabulary terms to differentiate between the two outcomes, e.g., perhaps we need more than a single threshold.
5. There is chain of inference between `F-TEST` and `QUALITY` that has not been made explicit in the current KBs and the discrepancy lies somewhere along that chain.
6. Both rules are over-generalized as stated: they are both missing an attribute whose value constitutes an important unstated assumption that the experts know, but forgot to make explicit.

After some discussion, the experts agree that the last possibility is correct, and that they omitted information about the `R-squared` statistic. In response they elaborate their rules to include this, and discovere that they had different underlying assumptions about the value of the `R-statistic`. The two revised rules turn out to be acceptable to both experts.

BRUCE's KB	JOHN's KB
`IF F-TEST less-than CRITICAL-VALUE`	`IF F-TEST less-than CRITICAL-VALUE`
`& R-SQUARED is NOT-SIGNIFICANT`	`& R-SQUARED is SIGNIFICANT`
`THEN QUALITY is of MODEL is POOR`	`THEN QUALITY of MODEL is FAIR`

Fig. 3: The modified rules.

CARTER then guides the experts in resolving remaining details about the new attribute `R-squared`, and continues to look for other places where the two knowledge bases differ. The techniques described above allowed CARTER to find roughly 65% of the known discrepancies in the two KBs and allowed it to guide the two experts to consensus on a subset of rules concerned with the problem of multicollinearity.

5 Matching Representations: Circumstantial Evidence Is Inevitable

Rules provide a good way to express inferences, but are not specifically designed to capture and represent meaning. Hence CARTER must attempt to infer the overlap in meaning between two concepts by collecting circumstantial evidence from the knowledge bases themselves, e.g., comparing how these concepts are used in the rules and how they relate to other vocabulary terms. The difficulty here is in using circumstantial evidence to infer whether two concepts were intended to have the same or different meanings.

Why not then just have definitions for each concept in the knowledge base and compare the definitions? That is, build the original domain knowledge bases so that they contain not only rules built from attribute-object-value triples, but also have formal definitions for each concept (i.e., each attribute, object, and value), perhaps definitions of the sort created with predicate calculus, or other formal representation language (e.g., [3]). Given formal definitions of the concepts, it would appear that we could compare those definitions directly and avoid the need for circumstantial evidence.

But the problem is unavoidable. If the domain knowledge bases have formal definitions for their terms, this immediately brings up the question of where those definitions came from. If they had already been agreed upon by the experts, then we are dealing with a situation in which there is a consensus vocabulary already established. CARTER would have to deal only with rule differences, and a substantial part of the consensus task would already have been done.

While this is a plausible situation, it is one in which most of the consensus problem has been solved before we get there.

If, on the other hand, each knowledge base has definitions for its concepts, but these were *created independently* (and have not yet been jointly agreed on), the same fundamental problem arises: given two concepts represented in independently developed KBs, the only grounds on which to determine whether they are likely to be the same is the circumstantial evidence arising from comparing their representations.

The key observation is that two experts working independently will construct ontologies (ways of looking at the world) that differ in some respects simply because they were constructed independently. Given two independently constructed ontologies, the only grounds on which to decide whether two concepts are the same is the circumstantial evidence available from comparing their representations.

A specific example from a common sense domain will help illustrate the point. Assume our first expert defines the concept of a `satisfied-parent` as "a parent with at least two children, all of whom are professionals." A KL-ONE style representation of this would look something like:

```
satisfied-parent    a-k-o       parent
            role    children    (value-restriction professional)
                                (number-restriction >= 2)
```

A parent would be further defined as `a-k-o human`, etc.

The second expert, working independently, decides to define what we as onlookers happen know is the same concept, but phrases it slightly differently: a `happy-parent` is "a parent with at least two offspring, all of whom have professional degrees:"

```
happy-parent    a-k-o        parent
        role    offspring    (value-restriction professional-degreed)
                             (number-restriction >= 2)
```

Now suppose CARTER attempted to match up `satisfied-parent` in one KB with `happy-parent` in the other, in an effort to determine whether they actually meant the same thing (i.e., the experts would agree that they had the same concept in mind). The only form of evidence available to the machine is the circumstantial evidence arising from matches (and mismatches) in the representation structure.

For example, both concepts are `a-k-o parent`, lending some support to the notion that they are the same, but certainly not proving so. Note that even this evidence is in turn dependent on believing that `parent` is intended to capture the same concept in both knowledge bases (likely, but not a foregone conclusion). The only way to determine that in turn would be by matching the definition structures for `parent`, etc., applying the same matching process on up the `a-k-o` chain.[10]

In matching the rest of the definition of these two terms, the machine would discover that one of them has a role called `children`, while the other has a role called `offspring`. The process would once again have to subgoal on gathering evidence about whether those two were in turn the same.

Hence the fundamental task involved in matching two independently constructed representations is unavoidably one of gathering circumstantial evidence from the representation structures themselves.

5.1 The Utility of Formal Definitions

While inferring meaning from syntactic evidence is inevitable, there is some benefit in using a representation that provides a facility for definitions of terms. One obvious reason for this is that when we want to compare meanings, the task is easier if we have a representation designed to capture meaning, rather than one designed primarily to capture inference (i.e., rules)

A more subtle form of benefit arises because representation languages designed to express formal definitions often provide a carefully chosen set of representational primitives (e.g., `a-k-o`, `role`, `value-restriction`, `number-restriction`). To see how these can help, consider two different reasons why two experts might construct two different representations of the same concept. First of course, they may think about the world differently, perhaps creating different `a-k-o` hierarchies.

Second and somewhat more subtly, they may each have a slightly *different understanding* of what `a-k-o` is supposed to mean. That is, the consensus problem recurs here in a subtle way because the experts might not have a shared understanding of *the representation language itself*, and may write different representations of the same concept even when their "internal" ontologies match.

A carefully chosen set of representation primitives can raise the likelihood that the meanings of those primitives are themselves shared between the experts, thereby reducing the chance for error in the translation from mental construct to written representation structure. This also means that a system like CARTER can justifiably put more emphasis on the overlaps that it finds when working with those representations.

Hence using representations that provide formal definitions of meaning may improve the system's ability to match concepts. But the fundamental problem is inescapable: when working with two independently created knowledge bases, we inevitably have to compare meaning by matching representation structures and accumulating circumstantial evidence, no matter what representation language is used.

6 RELATED WORK

Several efforts share some similarity in general spirit with ours. The Delphi technique ([8], [9]) for instance, uses debugging to achieve consensus among a group of experts on a specific issue. While such methods can prove useful, we believe it is premature to combine the results of two

[10]Note the similarity to tracing back along the inference chain when matching concepts in the rule-based system.

reasoning processes before even attempting to achieve consensus on the underlying knowledge used to arrive at those results. Exploring that knowledge may reveal key differences in reasoning, vocabulary, or problem assumptions which, once reconciled, remove the outcome discrepancy entirely.

We have also drawn on the work of Brown [4] and Kuper [12] in conceiving of the problem in terms of detecting bugs in knowledge and associating a repair procedure with each bug.

Work in knowledge acquisition based on the repertory grid notion [2, 6, 18] focuses on using knowledge of multiple experts, but offers only modest guidance for reaching consensus, providing little knowledge about either detection or repair.

Work by Reboh [19] suggests using the explanatory facilities of rule-based systems to pinpoint areas of disagreement between two systems, but leaves it to the user to decide which point of view to accept. Work by Mittal [16] explores the utility of interviewing multiple experts as a way of understanding a task better, but does not deal with combining their knowledge.

Some of the work on the CONSUL system [14] addressed the issue of finding good matches and discrepancies in concept definitions, and shared the intuition that the system might determine whether two concepts were closely related by comparing their representations. In this case the discrepancy problem typically arose when adding new concepts to the knowledge base, and the two participants in the negotiation were the existing knowledge base and the human expert who supplied the new concept.

A recent study [11] addresses the issue of resolving conflicting design specifications. Although their typology of conflicts is similar in some respects to our discrepancy catalog, their primary focus is on reconciling the designs themselves rather than design knowledge.

More distantly related is work on argumentation methods, aimed at assisting the process of discussion by helping people specify the logical structure of their positions. As such they introduce an element of rigor into the deliberation process, but offer little guidance in resolving differences between the experts. These ideas have recently been embodied in computer-based tools (e.g., [13], [16], [20], [21]), but they are almost entirely process oriented: they assist experts in the process of deliberating and debating, but, importantly, do not suggest resolutions to inconsistencies.

7 CONTRIBUTIONS

One primary contribution of this work is the catalog of 35 varieties of discrepancy, a store of detailed information we have codified for facilitating CKA. It represents a small but growing and relatively systematic expression of knowledge about how to detect and resolve disagreements in knowledge.

A second contribution arises from the surprisingly effective degree of bootstrapping displayed by CARTER. The system must make its best guess about the meaning of a term from the way it is used in a knowledge base, it can gather only circumstantial evidence of the sort reviewed above, and it must, paradoxically, gather that evidence from the very same knowledge bases it is attempting to modify to reach consensus. It it thus quite interesting how effective the system's heuristics are at guiding it. CARTER manages to make plausible judgments about which concepts match, so that even when it cannot be sure and has to ask the experts, the questions are for the most part sensible and well chosen.

A third contribution is the extensibility of our discrepancy catalog, illustrated during a session with the two statisticians that revealed the need for a new category of entry: differences in problem solving strategy. Where one expert looked for all possible problems, then suggested a solution, the other searched for and corrected one problem at a time. This new kind of discrepancy turns out to be detectable in terms of a difference in the topology of the two KBs (specifically, a difference in the number of goals); the detection/resolution organization of the catalog allowed us to add the appropriate new entries with minimal effort.

A final contribution is the potential of our work as a general approach to constructing systems that detect and resolve knowledge-level discrepancies. While our current system removes discrepancies in knowledge expressed in rules and attribute-object-value triples, we believe debugging and repair strategies can equally well be organized around other kinds of knowledge representations. The related work done using KL-ONE [14] strongly suggests the plausibility of this approach. The fundamental process involves specifying the components of the representation, developing a taxonomy of how the representations can differ across these components, and prescribing possible resolutions for each of these discrepancies. We anticipate testing this on other forms of representation.

8 FUTURE WORK

One of the most important areas for future research is the question of discrepancy resolution strategy. While our strategy of starting at the outcome and working backward is useful, it is one of several possibilities. One problem is that it may be a bit too myopic to be effective in a large scale knowledge base. The system in effect immediately dives into the details and it needs a better sense of the larger picture. Our next task is thus to generate a number of strategies and evaluate them in terms of the efficiency and effectiveness with which they increase the degree of consensus, and the naturalness and coherence of the dialogues they produce.

Alternative strategies include working forward from inputs (because the two KBs are likely to start reasoning from the same basic information) and beginning at an intermediate point of agreement and expanding in both directions (emphasizing what the experts already agree on and building from there).

9 CONCLUSION

We have described a novel approach to and prototype system for facilitating consensus knowledge acquisition. The key contributions of this work include the development of a detailed store of knowledge for detecting and resolving discrepancies in rule-based systems, a general procedure for developing similar systems for other representations, and the empirical result that circumstantial evidence in the knowledge base itself can provide a useful degree of power in determining concept meaning. We expect the next advance in this area to come from implementing improved discrepancy resolution strategies. This work will serve as the starting point for understanding more generally how experts reach consensus and how we can best support them in their efforts to do so.

REFERENCES

[1] Aczel, J., and Saaty, T. Procedure for Synthesizing Ratio Judgments,*Journal of Mathematical Psychology*, Volume 27, Number 1, March 1983, 93–102.

[2] Boose, J. *Expertise Transfer for Expert Systems Design.* Amsterdam: Elsevier,1986.

[3] Brachman, R. et. al. "KL-ONE Reference Manual," BBN Report No. 3848, July 1978.

[4] Brown, J., and Burton, R. Diagnostic Models for Procedural Bugs in Basic Mathematical Skills, *Cognitive Science*, Volume 2, Number 2, April–June 1978,pp. 155–192.

[5] Fogelin, R.*Understanding arguments: An introduction to informal logic.* New York: Harcourt, Brace, Javanovich, 1982.

[6] Gaglio, S. et al. Multiperson Decision Aspects in the Construction of Expert Systems, *IEEE Trans. on Sys., Man, and Cybernetics*, Volume 15, Number 4,July–August 1985, pp. 536–539.

[7] Gaines B and Shaw M, Comparing the conceptual systems of experts, *Proc IJCAI-89*, pp. 633–638.

[8] Helmer, O., and Rescher, N. On the Epistemology of the Inexact Sciences, *Management Science*, Volume 6, Number 1, October 1959, pp. 25–52.

[9] Jagannathan, V., and Elmaghraby, A. MEDKAT: Multiple Expert Delphi-Based Knowledge Acquisition Tool, Engineering Mathematics and Computer Science Department Technical Report, Univ. of Louisville, 1985.

[10] Kelly, G. *The Psychology of Personal Constructs.* Boston: Norton, 1955.

[11] Klein, M., et al. Towards a Theory of Conflict Resolution in Cooperative Design, Proceedings of the 9th Workshop on Distributed Artificial Intelligence, 1989, pp. 329-349.

[12] Kuper, R., Dependency-Directed Localization of Software Bugs, MIT-AI-TR-1053, MIT AI Lab, May, 1989.

[13] Lowe, D. Co-operative Structuring of Information: The Representation of reasoning and debate, *Int. J. Man-Machine Studies*, Volume 23, Number 2, August 1985, pp. 97–111.

[14] Mark, W., Rule-based inferences in large knowledge bases, *Proc AAAI-80*, pp. 190–194.

[15] Miner, F. Group vs. Individual Decision Making: An Investigation of Performances Measures, Decision Strategies, and Process Losses/Gains, *Organizational Behavior and Human Decision Processes*, Volume 33, Number 1, February 1984, pp. 112–125.

[16] Mittal, S., and Dym, C.L. Knowledge Acquisition from Multiple Experts, *The AI Magazine*, Volume 6, Number 2, Summer 1985, pp. 32–36.

[17] Nunnamaker, J., et al. Computer-aided Deliberation: Model Management and Group Decision Support, *Operations Research*, Volume 36, Number 6, November–December 1988, pp. 826–848.

[18] Plaza, E. et al., Consensus and Knowledge Acquisition, in *Uncertainty in Knowledge–Based Systems*. Bouchon, B., and Yager, R., eds. Berlin: Springer-Verlag, 1987. pp. 294–306.

[19] Reboh R. Extracting useful advice form conflicting expertise, Proc. 8th IJCAI, Karlsruhe, West Germany, August 1983, pp. 145-150.

[20] Smolensky, P. et al., Constraint-Based Hypertext for Argumentation, Dept. of Comp Sci. and Linguistics, University of Colorado WP #CU-CS-358-87, 1987.

[21] Stefik M, et al., Beyond the chalkboard, *CACM*, **30**:1, Jan 1987, pp. 32–47.

[22] Trice A, Davis R, Consensus Knowledge Acquisition, MIT AI Lab Memo 1183, December 1989.

[23] Toulmin, S. *The Uses of Argument*. Cambridge, England: Cambridge University Press, 1958.