AUGMENTING A DATABASE KNOWLEDGE REPRESENTATION FOR NATURAL LANGUAGE GENERATION*

Kathleen F. McCoy
Dept. of Computer and Information Science
The Moore School
University of Pennsylvania
Philadelphia, Pa. 19104

ABSTRACT

The knowledge representation is an important factor in natural language generation since it limits the semantic capabilities of the generation system. This paper identifies several information types in a knowledge representation that can be used to generate meaningful responses to questions about database structure. Creating such a knowledge representation, however, is a long and tedious process. A system is presented which uses the contents of the database to form part of this knowledge representation automatically. It employs three types of world knowledge axioms to ensure that the representation formed is meaningful and contains salient information.

1.0 INTRODUCTION

In order for a user to extract meaningful information from a database system, s/he must first understand the system's view of the world what information the system contains and what that information represents. An optimal way of acquiring this knowledge is to interact, in natural language, with the system itself, posing questions to it about the structure of its contents. The TEXT system [McKeown 82] was developed to facilitate this type of interaction.

In order to make use of the TEXT system, a system's knowledge about itself must be rich enough to support the generation of interesting texts about the structure of its contents. As I will demonstrate, standard database models [Chen 76], [Smith & Smith 77] are not sufficient to support this type of generation. Moreover, since time is such an important factor when generating answers, and extensive inferencing is therefore not practical, the system's self knowledge must be immediately available in its knowledge representation. The ENHANCE system, described here, has been developed to augment a database schema with the kind of information necessary for generating informative answers to users' queries. The ENHANCE system creates part of the knowledge representation used by TEXT based on the contents of the database. A set of world knowledge axioms are used to ensure that this knowledge representation reflects both the database contents and the database designer's view of the world.

One important class of questions involves comparing database entities. The system's knowledge representation must therefore contain meaningful information that can be used to make comparisons (analogies) between various entity classes. This paper focuses specifically on those aspects of the knowledge representation generated by ENHANCE which facilitate the use of analogies. An overview of the knowledge representation used by TEXT is first given. This is followed by a discussion of how part of this representation is automatically created by ENHANCE.

2.0 KNOWLEDGE REPRESENTATION FOR GENERATION

The TEXT system answers three types of questions about database structure: (1) requests for the definition of an entity; (2) requests for the information available about an entity; (3) requests concerning the difference between entities. It was implemented and tested using a portion of an ONR database which contained information about vehicles and destructive devices.

TEXT needs several types of information to answer the above questions. Some of this can be provided by features found in a variety of standard database models [Chen 76], [Smith & Smith 77], [Lee & Gerritsen 78].

Of these, TEXT uses a generalization hierarchy on the entities in order to define or identify them in terms of (1) their constituents (e.g. "There are two types of entities in the ONR database: destructive devices and vehicles."*) (2) their superordinates (e.g. "A destroyer is a surface ship .. A bomb is a free falling projectile." and "A whiskey is an underwater submarine ..."). Each node in the hierarchy contains additional descriptive information based on standard features which is used to identify the database information associated with each entity and to indicate the distinguishing features of the entities.

* This work was partially supported by National Science Foundation grant #MCS81-07290.

* The quoted material is excerpted from actual output from TEXT.

One type of comparison that TEXT must generate has to do with indicating why a particular individual falls into one entity sub-class as opposed to another. For example, "A ship is classified as an ocean escort if the characters 1 through 2 of its HULL_NO are DE ... A ship is classified as a cruiser if the characters 1 through 2 of its HULL_NO are CG." and "A submarine is classified as an echo II if its CLASS is ECHO II." In order to generate this kind of comparison, TEXT must have available database information indicating the reason for a split in the generalization hierarchy. This information is provided in the based DB attribute.

In comparing two entities, TEXT must be able to identify the major differences between them. Part of this difference is indicated by the descriptive distinguishing features of the entities. For example, "The missile has a target location in the air or on the earth's surface ... The torpedo has an underwater target location." and "A whiskey is an underwater submarine with a PROPULSION_TYPE of DIESEl and a FLAG of RDOR." These distinguishing features consist of a number of attribute-value* pairs associated with each entity. They are provided in an information type termed the distinguishing descriptive attributes (DDAs) of an entity.

In order for TEXT to answer questions about the information available about an entity, it must have access to the actual database information associated with each entity in the generalization hierarchy. This information is provided in what are termed the actual DB attributes (and constant values) and the relational attributes (and values). This information is also useful in comparing the attributes and relations associated with various entities. For example, "Other DB attributes of the missile include PROBABILITY_OF_KILL, SPEED, ALTITUDE ... Other DB attributes of the torpedo include FUSE_TYPE, MAXIMUM_DEPTH, ACCURACY_&_UNITS..." and "Echo IIs carry 16 torpedoes, between 16 and 99 missiles and 0 guns."

3.0 AUGMENTING THE KNOWLEDGE REPRESENTATION

The need for the various pieces of information in the knowledge representation is clear. How this representation should be created remains unanswered. The entire representation could be hand coded by the database designer. This, however, is a long and tedious process and therefore a bottleneck to the portability of TEXT.

In this work, a level in the generalization hierarchy is identified that contains entities for which physical records exist in the database (database entity classes). It is assumed that the hierarchy above this level must be hand coded. The information below this level, however, can be derived from the contents of the database itself. The database entity classes can be subclassified on the basis of attributes whose values serve to partition the entity class into a number of mutually exclusive sub-types. For example, PEOPLE can be subclassified on the basis of attribute SEX: MALE and FEMALE. As pointed out by Lee and Gerritsen [Lee & Gerritsen 78], some partitions of an entity class are more meaningful than others and hence more useful in describing the system's knowledge of the entity class. For example, a partition based on the primary key of the entity class would generate a single member sub-class for each instance in the database, thereby simply duplicating the contents of the database. The ENHANCE system relies on a set of world knowledge axioms to determine which attributes to use for partitioning and which resulting breakdowns are meaningful.

For each meaningful breakdown of an entity class, nodes are created in the generalization hierarchy. These nodes must contain the information types discussed above. ENHANCE computes this information based on the facts in the database. The attribute used to partition the entity class appears as the based DB attribute. The DDAs are a list of actual DB attributes, other than the based DB attribute, which when taken together distinguish a sub-class from all others in the breakdown. Since the sub-classes inherit all DB attributes from the entity class, the values of the attributes within the sub-class are important. ENHANCE records the values of all constant DB attributes and the range of values of any DB attributes which appear in the DDA of any sibling sub-class. These can be used by TEXT to compare the values of the DDAs of one sub-class with the values of the same attributes within a sibling sub-class. The values of relational attributes within a sub-class are also recorded by ENHANCE.

The descriptive information will be used by the generation system to indicate how the sub-classes differ. It is therefore important that the most salient differences between the sub-classes are indicated. Here again, the world knowledge axioms are used to guide the system in choosing the most salient information.

The world knowledge axioms fall into three categories which reflect the extent to which they must be changed when applying ENHANCE to a new database. They range from very specific axioms, which must always be changed, to very general axioms, which are domain independent. The axioms and their use by the system will be described after first giving an example of a question answered by TEXT based on information created by ENHANCE.

3.1 EXAMPLE USING AUGMENTED REPRESENTATION

This example illustrates how TEXT integrates the above information to form meaningful comparisons between entities. This example is taken from actual TEXT output in response to the

* these attributes are not necessarily attributes contained in the database.

question: What is the difference between an OCEAN-ESCORT and a CRUISER?*

> 1) The cruiser and the ocean escort are surface ships. 2) The ship's surface-going capabilities are provided by the DB attributes DISPLACEMENT and DRAFT. 3) The ship has DB attributes MAXIMUM_SPEED, PROPULSION, FUEL(FUEL_CAPACITY and FUEL_TYPE), DIMENSIONS, SPEED_DEPENDENT_RANGE and OFFICIAL_NAME. 4) Ocean escorts have a DISPLACEMENT between 3400 and 4100. 5) All ocean escorts in the ONR database have REMARKS of 0, FUEL_TYPE of BNKR, FLAG of BLBL and MAST_HEIGHT of 85. 6) Ocean escorts carry between 2 and 22 torpedoes, 16 missiles and between 1 and 2 guns. 7) A ship is classified as an ocean escort if the characters 1 through 2 of its HULL_NO are DE. 8) Cruisers have a PROPULSION of STMTURGRD and a LENGTH between 510 and 673. 9) All cruisers in the ONR database have REMARKS of 0. 10) Cruisers carry between 8 and 42 torpedoes, between 4 and 98 missiles and between 1 and 4 guns. 11) A ship is classified as a cruiser if the characters 1 through 2 of its HULL_NO are CL or the characters 1 through 2 of its HULL_NO are CG. 12) The ocean escort, therefore, has a smaller LENGTH and a smaller DISPLACEMENT than the cruiser.

The first sentence is derived from the fact that both ocean-escorts and cruisers are sub-types of entity class ship. TEXT then goes on to describe some characteristics of the ship (sentences 2 and 3). Information about the ship is part of the hand coded representation, it is not generated by ENHANCE. Next, the distinguishing features (indicated by the DDA) of the ocean-escort are identified followed by a listing of its constant DB attributes (sentences 4 and 5). The values of the relation attributes are then identified (sentence 6) followed by a statement drawn from the based DB attribute of the ocean-escort. Next, this same type of information is used to generate parallel information about the cruiser. The text closes with a simple inference based on the DDAs of the two types of ships.

4.0 WORLD KNOWLEDGE AXIOMS

In order for the generation system to give meaningful descriptions of the database, the knowledge representation must effectively capture both a typical user's view of the domain and how that domain has been modelled within the system. Without real world knowledge indicating what a user finds meaningful, there are several ways in which an automatically generated taxonomy may deviate from how a user views the domain: (1) the representation may fail to capture the user's preconceived notions of how a certain database entity class should be partitioned into sub-classes; (2) the system may partition an entity class on the basis of a non-salient attribute leading to an inappropriate breakdown; (3) non-salient information may be chosen to describe the sub-classes leading to inappropriate descriptions; (4) a breakdown may fail to add meaning to the representation (e.g. a partition chosen may simply duplicate information already available).

The first case will occur if the sub-types of these breakdowns are not completely reflected in the database attribute names and values. For example, even though the partition of SHIP into its various types (e.g. Aircraft-Carrier, Destroyer, etc.) is very common, there may be no attribute SHIP_TYPE in the database to form this partition. The partition can be derived, however, if a semantic mapping between the sub-type names and existing attribute-value pairs can be identified. In this case, the partition can be derived by associating the first few characters of attribute HULL_NO with the various ship-types. The *very specific* axioms are provided as a means for defining such mappings.

The taxonomy may also deviate from what a user might expect if the system partitions an entity class on the basis of non-salient attributes. It seems very natural to have a breakdown of SHIP based on attribute CLASS, but one based on attribute FUEL-CAPACITY would seem less appropriate. A partition based on CLASS would yield sub-classes of SHIP such as SKORY and KITTY-HAWK, while one on FUEL_CAPACITY could only yield ones like SHIPS-WITH-100-FUEL-CAPACITY. Since saliency is not an intrinsic property of an attribute, there must be a way of indicating attributes salient in the domain. The *specific axioms* are provided for this purpose.

The user's view of the domain will not be captured if the information chosen to describe the sub-classes is not chosen from attributes important to the domain. Saliency is crucial in choosing the descriptive information (particularly the DDAs) for the sub-classes. Even though a DESTROYER may be differentiated from other types of ships by its ECONOMIC-SPEED, it seems more informative to distinguish it in terms of the more commonly mentioned property DISPLACEMENT. Here again, this saliency information is provided by the *specific axioms*.

A final problem faced by a system which only relies on the database contents is that a partition formed may be essentially meaningless (adding no new information to the representation). This will occur if all of the instances in the database fall into the same sub-class or if each falls into a different one. Such breakdowns either exactly reflect the entity class as a whole, or reflect the individual instances. This same type of problem occurs if the only difference between two sub-classes is the attribute the breakdown is based on. Thus, no trend can be found among the other attributes within the sub-classes formed. Such a breakdown would add no

* The sentences are numbered here to simplify the discussion: there are no sentence numbers in the actual material produced by TEXT.

information that could not be trivially derived from the database itself. These types of breakdowns are "filtered out" using the general axioms.

The world knowledge axioms guide ENHANCE to ensure that the breakdowns formed are appropriate and that salient information is chosen for the sub-class descriptions. At the same time, the axioms give the designer control over the representation formed. The axioms can be changed and the system rerun. The new representation will reflect the new set of world knowledge axioms. In this way, the database designer can tune the representation to his/her needs. Each axiom category, how they are used by ENHANCE, and the problems each category solves are discussed below.

4.1 Very Specific Axioms

The very specific axioms give the user the most control over the representation formed. They let the user specify breakdowns that s/he would a priori like to appear in the knowledge representation. The axioms are formulated in such a way as to allow breakdowns on parts of the value field of a character attribute, and on ranges of values for a numeric attribute (examples of each are given below). This type of breakdown could not be formed without explicit information indicating the defining portions of the attribute value field and their associated semantic values.

A sample use of the very specific axioms can be found in classifying ships by their type (ie. Aircraft-carriers, Destroyers, Mine-warfare-ships, etc...). This is a very common breakdown of ships. Assume there is no database attribute which explicitly gives the ship type. With no additional information, there is no way of generating that breakdown for ship. A user knowledgeable of the domain would note that there is a way to derive the type of a ship based on its HULL_NO. In fact, the first one or two characters of the HULL_NO uniquely identifies the ship type. For example, all AIRCRAFT-CARRIERS have a HULL_NO whose first two characters are CV, while the first two characters of the HULL_NO of a CRUISER are CA or CG or CL. This information can be captured in a very specific axiom which maps part of a character attribute field into the sub-type names. An example of such an axiom is shown in Figure 1.

```
(SHIP "SHIP_HULL_NO"
      "OTHER-SHIP-TYPE"
      (1 2 "CV" "AIRCRAFT-CARRIER")
      (1 2 "CA" "CRUISER")
      (1 2 "CG" "CRUISER")
      (1 2 "CL" "CRUISER")
      (1 2 "DD" "DESTROYER")
      (1 2 "DL" "FRIGATE")
      (1 2 "DE" "OCEAN-ESCORT")
      (1 2 "PC" "PATROL-SHIP-AND-CRAFT")
      (1 2 "PG" "PATROL-SHIP-AND-CRAFT")
      (1 2 "PT" "PATROL-SHIP-AND-CRAFT")
      (1 1 "L" "AMPHIBIOUS-AND-LANDING-SHIP")
      (1 2 "MC" "MINE-WARFARE-SHIP")
      (1 2 "MS" "MINE-WARFARE-SHIP")
      (1 1 "A" "AUXILIARY-SHIP"))
```

Figure 1. Very Specific (Character) Axiom

Sub-typing of entities may also be specified based on the ranges of values of a numeric attribute. For example, the entity BOMB is often sub-typed by the range of the attribute BOMB_WEIGHT. A BOMB is classified as being HEAVY if its weight is above 900, MEDIUM-WEIGHT if it is between 100 and 899, and LIGHT-WEIGHT if its weight is less than 100. An axiom which specifies this is shown in FIGURE 2.

```
(BOMB "BOMB_WEIGHT"
      "OTHER-WEIGHT-BOMB"
      (900 99999 "HEAVY-BOMB")
      (100 899 "MEDIUM-WEIGHT-BOMB")
      (0 99 "LIGHT-WEIGHT-BOMB"))
```

Figure 2. Very Specific (Numeric) Axiom

Formation of the very specific axioms requires in-depth knowledge of both the domain the database reflects, and the database itself. Knowledge of the domain is required in order to make common classifications (breakdowns) of objects in the domain. Knowledge of the database structure is needed in order to convey these breakdowns in terms of the database attributes. It should be noted that this type of axiom is not required for the system to run. If the user has no preconceived breakdowns which should appear in the representation, no very specific axioms need to be specified.

4.2 Specific Axioms

The specific axioms afford the user less control than the very specific axioms, but are still a powerful device. The specific axioms point out which database attributes are more important in the domain than others. They consist of a single list of database attributes called the important attributes list. The important attributes list does not "control" the system as the very specific axioms do. Instead it suggests paths for the system to try; it has no binding effects. The important attributes list used for testing ENHANCE on the ONR database is shown in Figure 3.

```
(CLASS FLAG
       DISPLACEMENT
       LENGTH
       WEIGHT
       LETHAL_RADIUS
       MINIMUM_ALTITUDE
       ACCURACY
       HORZ_RANGE
       MAXIMUM_ALTITUDE
       FUSE_TYPE
       PROPULSION_TYPE
       PROPULSION
       MAXIMUM_OPERATING_DEPTH
       PRIMARY_ROLE)
```

Figure 3. Important Attributes List

ENHANCE has two major uses for the important attributes list: (1) It attempts to form breakdowns based on some of the attributes in the list. (2) It uses the list to decide which attributes to use as DDAs for a sub-class. ENHANCE must decide which attributes are better as the basis for a breakdown and which are better for describing the resulting sub-classes. While most attributes important to the domain are good for descriptive purposes, character attributes are better than others as the basis for a breakdown. Attributes with character values can more naturally be the basis for a breakdown since they have a small set of legal values. A breakdown based on such an attribute leads to a small well-defined set of sub-classes. Numeric attributes, on the other hand, often have an infinite number of legal values. A breakdown based on individual numeric values could lead to a potentially infinite number of sub-classes. This distinction between numeric and character (symbolic) attributes is also used in the TEAM system [Grosz et. al. 82]. ENHANCE first attempts to form breakdowns of an entity based on character attributes from the important attributes list. Only if no breakdowns result from these attempts, does the system attempt breakdowns based on numeric attributes.

The important attributes list also plays a major role in selecting the distinguishing descriptive attributes (DDAs) for a particular sub-class. Recall that the DDAs are a set of attributes whose values differentiate one sub-class from all other sub-classes in the same breakdown. It is often the case that several sets of attributes could serve this purpose. In this situation, the important attributes list is consulted in order to choose the most salient distinguishing features. The set of attributes with the highest number of attributes on the important attributes list is chosen.

The important attributes list affords the user less control over the representation formed than the very specific axioms since it only suggests paths for the system to take. The system attempts to form breakdowns based on the attributes in the list, but these breakdowns are subjected to tests encoded in the general axioms which are not used for breakdowns formed by the very specific axioms. Breakdowns formed using the very specific axioms are not subjected to as many tests since they were explicitly specified by the database designer.

4.3 General Axioms

The final type of world knowledge axioms used by ENHANCE are the general axioms. These axioms are domain independent and need not be changed by the user. They encode general principles used for deciding such things as whether sub-classes formed should be added to the knowledge representation, and how sub-classes should be named.

The ENHANCE system must be capable of naming the sub-classes. The name must uniquely identify a sub-class and should give some semantic indication of the contents of the sub-class. At the same time, they should sound reasonable to the ENHANCE user. These problems are handled by the general axioms entitled naming conventions. An example of a naming convention is:

> Rule 1 - The name of a sub-class of entity ENT formed using a character* attribute with value VAL will be: VAL-ENT.

Examples of sub-classes named using this rule include: WHISKY-SUBMARINE and FORRESTAL-SHIP.

The ENHANCE system must also ensure that each of the sub-classes in a particular breakdown are meaningful. For instance, some of the sub-classes may contain only one individual from the database. If several such sub-classes occur, they are combined to form a CLASS-OTHER sub-class. This use of CLASS-OTHER compacts the representation while indicating that a number of instances are not similar enough to any others to form a sub-class. The DDA for CLASS-OTHER indicates what attributes are common to all entity instances that fail to make the criteria for membership in any of the larger named sub-classes. Without CLASS-OTHER this information would have to be derived by the generation system; this is a potentially time consuming process. The general axioms contain several rules which will block the formation of "CLASS-OTHER" in circumstances where it will not add information to the representation. These

* This is a slight simplification of the rule actually used by ENHANCE, see [McCoy 82] for further details.

include:

Rule 2 - Do not form CLASS-OTHER if it will contain only one individual.

Rule 3 - Do not form CLASS-OTHER if it will be the only child of a superordinate.

Perhaps the most important use of the general axioms is their role in deciding if an entire breakdown adds meaning to the knowledge representation. The general axioms are used to "filter out" breakdowns whose sub-classes either reflect the entity class as a whole, or the actual instances in the database. They also contain rules for handling cases when no differences between the sub-classes can be found. Examples of these rules include:

Rule 4 - If a breakdown results in the formation of only one sub-type, then do not use that breakdown.

Rule 5 - If every sub-class in two different breakdowns contains exactly the same individuals, then use only one of the breakdowns.

5.0 SYSTEM OVERVIEW

The ENHANCE system consists of a set of independent modules; each is responsible for generating some piece of descriptive information for the sub-classes. When the system is invoked for a particular entity class, it first generates a number of breakdowns based on the values in the database. These breakdowns are passed from one module to the next and descriptive information is generated for each sub-class involved. This process is overseen by the general axioms which may throw out breakdowns for which descriptive information can not be generated.

Before generating the breakdowns from the values in the database, the constraints on the values are checked and all units are converted to a common value. Any attribute values that fail to meet the constraints are noted in the representation and not used in the calculation. From these values a number of breakdowns are generated using the very specific and specific axioms.

The breakdowns are first passed to the "fitting algorithm". When two or more breakdowns are generated for an entity-class, the sub-classes in one breakdown may be contained in the sub-classes of the other. In this case, the sub-classes in the first breakdown should appear as the children of the sub-classes of the second breakdown, adding depth to the hierarchy. The fitting algorithm is used to calculate where the sub-classes fit in the generalization hierarchy. After the fitting algorithm is run, the general axioms may intervene to throw out any breakdowns which are essentially duplicates of other breakdowns (see rule 5 above).

At this point, the DDAs of the sub-classes within each breakdown are calculated. The algorithm used in this calculation is described below to illustrate the combinatoric nature of the augmentation process. If no DDAs can be found for a breakdown formed using the important attributes list, the general axioms may again intervene to throw out that breakdown.

Flow of control then passes through a number of modules responsible for calculating the based DB attribute and for recording constant DB attributes and relation attributes. The actual nodes are then generated and added to the hierarchy.

Generating the descriptive information for the sub-classes involves combinatoric problems which depend on the number of records for each entity in the database and the number of sub-classes formed for these entities. The ENHANCE system was implemented on a VAX 11/780, and was tested using a portion of an ONR database containing 157 records. It generated sub-type information for 7 entities and ran in approximately 159157 CPU seconds. For a database with many more records, the processing time may grow exponentially. This is not a major problem since the system is not interactive; it can be run in batch mode. In addition, it is run only once for a particular database. After it is run, the resulting representation can be used by the interactive generation system on all subsequent queries. A brief outline of the processing involved in generating the DDAs of a particular sub-class will be given. This process illustrates the kind of combinatoric problems encountered in automatic generation of sub-type information making it unreasonable computation for an interactive generation system.

5.1 Generating DDAs

The Distinguishing Descriptive Attributes (DDAs) of a sub-class is a set of attributes, other than the based DB attribute, whose collective value differentiates that sub-class from all other sub-classes in the same breakdown. Finding the DDA of a sub-class is a problem which is combinatoric in nature since it may require looking at all combinations of the attributes of the entity class. This problem is accentuated since it has been found that in practice, a set of attributes which differentiates one sub-class from all other sub-classes in the same breakdown does not always exist. Unless this problem is identified ahead of time, the system would examine all combinations of all of the attributes before deciding the sub-class can not be distinguished.

There are several features of the set of DDAs which are desirable. (1) The set should be as small as possible. (2) It should be made up of salient attributes (where possible). (3) The set should add information about that sub-class not already derivable from the representation. In other words, they should be different from the

DDAs of the parent.

A method for generating the DDAs could involve simply generating all 1-combinations of attributes, followed by 2-combinations etc.. until a set of attributes is found which differentiates the sub-class. Attributes that appeared in the DDA of the immediate parent sub-class would not be included in the combinations formed. To ensure that the DDA was made up of the most salient attributes, combinations of attributes from the important attributes list could be generated first. This method, however, does not avoid any of the combinatoric problems involved in the processing.

To avoid some of these problems, a pre-processor to the combination stage of the calculation was developed. The combinations are formed of only potential-DDAs. These are a set of attributes whose value can be used to differentiate the sub-class from at least one other sub-class. The attributes included in potential-DDAs take on a value within the sub-class that is different from the value the attributes take on in at least one other sub-class. Using the potential-DDAs ensures that each attribute in a given combination is useful in distinguishing the sub-class from all others.

Calculating the potential-DDAs requires comparing the values of the attributes within the sub-class with the values within each other sub-class in turn. This calculation yields two other pieces of important information. If for a particular sub-class this comparison yields only one attribute, then this attribute is the only means for differentiating that sub-class from the sub-class the DDAs are being calculated for. In order for the DDA to differentiate the sub-class from all others, it must contain that attribute. Attributes of this type are called definite-DDAs. The second type of information identified has to do with when the sub-class can not be differentiated from all others. The comparing of attribute values of sub-classes makes immediately apparent when the DDA for a sub-class can not be found. In this case, the general axioms would rule out the breakdown containing that sub-class.*

Assuming that the sub-class is found to be distinguishable, the system uses the potential-DDAs and the definite-DDAs to find the smallest and most salient set of attributes to use as the DDA. It forms combination of attributes using the definite-DDAs and members of the potential-DDAs. The important attributes list is consulted to ensure that the most salient attributes are chosen as the DDA.

5.2 Time/Space Tradeoff

There is a time/space tradeoff in using a system like ENHANCE. Once the ENHANCE system is run, the generation system is relieved from the time consuming task of sub-type inferencing. This means, however, that a much larger knowledge representation for the generation system's use results. Since the generation system must be concerned with the amount of time it takes to answer a question, the cost of the larger knowledge representation is well worth the savings in inferencing time. If, however, at some future point, time is no longer a major factor in natural language generation, many of the ideas put forth here could be used to generate the sub-type information only as it is needed.

6.0 USE OF REPRESENTATION CREATED BY ENHANCE

The following example illustrates how the TEXT system uses the information generated by ENHANCE. The example is taken from actual output generated by the TEXT system in response to the question: What is an AIRCRAFT-CARRIER?. It utilizes the portion of the representation generated by ENHANCE. Following the text is a brief description of where each piece of information was found in the representation. (The sentences are numbered here to simplify the discussion: there are no sentence numbers in the actual material produced by TEXT).

> (1) An aircraft carrier is a surface ship with a DISPLACEMENT between 78000 and 80800 and a LENGTH between 1039 and 1063. (2) Aircraft carriers have a greater LENGTH than all other ships and a greater DISPLACEMENT than most other ships. (3) Mine warfare ships, for example, have a DISPLACEMENT of 320 and a LENGTH of 144. (4) All aircraft carriers in the ONR database have REMARKS of 0, FUEL_TYPE of BNKR, FLAG of BLBL, BEAM of 252, ENDURANCE_RANGE of 4000, ECONOMIC SPEED of 12, ENDURANCE_SPEED of 30 and PROPULSION of STMTURGRD. (5) A ship is classified as an aircraft carrier if the characters 1 through 2 of its HULL_NO are CV.

In this example, the DDAs of aircraft carrier are used to identify its features (sentence 1) and to make a comparison between aircraft carriers and all other types of ships (sentences 2 and 3). Since the ENHANCE system ensures that the values of the DDAs for one sub-class appear in the DB attribute list of every other sub-class in the same breakdown, the comparisons between the sub-classes are easily calculated by the TEXT system. Moreover, since ENHANCE has selected out several attributes as more important than others (based on the world knowledge axioms), TEXT can make a meaningful comparison instead of one less relevant. The final sentence is derived from the based DB attribute of aircraft carrier.

* There are several cases in which ENHANCE would not rule out the breakdown, see [McCoy 82] for details.

7.0 FUTURE WORK

There are several extensions of the ENHANCE system which would make the knowledge representation more closely reflect the real world. These include (1) the use of very specific axioms in the calculation of descriptive information and (2) the use of relational information as the basis for a breakdown.

At the present time, all descriptive sub-class information is calculated from the actual contents of the database, although sub-class formation may be based on the very specific axioms. The database contents may not adequately capture the real world distinctions between the sub-classes. For this reason, a set of very specific axioms specifying descriptive information could be adopted. The need for such axioms can best be seen in the DDA generated for ship sub-type AIRCRAFT-CARRIER. Since there are no attributes in the database indicating the function of a ship, there is no way of using the fact that the function of an AIRCRAFT-CARRIER is to carry aircraft to distinguish AIRCRAFT-CARRIERS from other ships. This is, however, a very important real world distinction. Very specific axioms could be developed to allow the user to specify these important distinctions not captured the the contents of the database.

The ENHANCE system could also be improved by utilizing the relational information when creating the breakdowns. For example, missiles can be divided into sub-classes on the basis of what kind of vehicles they are carried by. AIR-TO-AIR and AIR-TO-SURFACE missiles are carried on aircraft, while SURFACE-TO-SURFACE missiles are carried on ships. Thus, the relations often contain important sub-class distinctions that could be used by the system.

8.0 CONCLUSION

A system has been described which automatically creates part of a knowledge representation used for natural language generation. This enables the generation system to give a richer description of the database, since the information generated by ENHANCE can be used to make comparisons between sub-classes which would otherwise require use of extensive inferencing.

ENHANCE generates sub-classes of the entity classes in the database; it uses a set of world knowledge axioms to guide the formation of the sub-classes. The axioms ensure the sub-classes are meaningful and that salient information is chosen for the sub-class descriptions. This in turn ensures that the generation system will have salient information available to use making the generated text more meaningful to the user.

9.0 ACKNOWLEDGEMENTS

I would like to thank Aravind Joshi and Kathleen McKeown for their many helpful comments throughout the course of this work, and Bonnie Webber, Eric Mays, and Sitaram Lanka for their comments on the content and style of this paper.

10.0 REFERENCES

[Chen 76]. Chen, P.P.S., "The Entity-Relationship Model - Towards a Unified View of Data", ACM Transactions on Database Systems, Vol. 1, No. 1, 1976.

[Grosz et. al. 82]. Grosz, B., et. al., "TEAM: A Transportable Natural Language System", Tech Note 263, Artificial Intelligence Center, SRI International, Menlo Park, Ca., (to appear).

[Lee & Gerritsen 78]. Lee, R.M., and Gerritsen, R., "Extended Semantics for Generalization Hierarchies", Proceedings of the 1978 ACM-SIGMOD International Conference on Management of Data, Austin, Texas, May 31 to June 2, 1978.

[McCoy 82]. McCoy, K.F., "The ENHANCE System: Creating Meaningful Sub-Types in a Database Knowledge Representation For Natural Language Generation", forthcoming Master's Thesis, University of Pennsylvania, Philadelphia, Pa., 1982.

[McKeown 82A]. McKeown, K.R., "Generating Natural Language Text in Response to Questions About Database Structure", Ph.D. Dissertation, University of Pennsylvania, Philadelphia, Pa., 1982.

[McKeown 82B]. McKeown, K.R., "The TEXT system for Natural Language Generation: An Overview", to appear in Proceedings of the 20th Annual Conference of the Association of Computational Linguistics, Toronto, Canada, June 1982.

[Smith and Smith 77]. Smith, J.M., and Smith, D.C.P., "Database Abstractions: Aggregation and Generalization", ACM Transactions on Database Systems, Vol. 2, No. 2, June 1977.