A Context Sensitive Model for Concept Understanding

Wen-Lian Hsu, Yi-Shiou Chen and Yuan-Kai Wang
Institute of Information Science
Academia Sinica
Taipei, Taiwan, R.O.C.
hsu@iis.sinica.edu.tw

Abstract

In artificial intelligence, knowledge representation is very crucial for simulating human understanding. Several strategies have previously been proposed for natural language understanding. But many of them have been confined to illustrations in textbooks rather than been actually implemented in large-scale natural language systems. The fact is that different representation schemes are appropriate under different situations. We present a context sensitive model for concept understanding. Our model emphasizes knowledge utilization rather than its representation. Many different representation schemes can be combined in our understanding model and the specific scheme of representation is immaterial. Our concept understanding strategy can model many human behaviors and reflect personal differences. We have applied our model to several *NLP* systems and obtained very satisfactory results. Our main goal is to reduce the problem of modeling human understanding to that of making correct selection under different contexts. The latter provides a new direction for the design and implementation of intelligent systems.

1. Introduction

In artificial intelligence, knowledge representation is very crucial for simulating human understanding. Several representations (cf. [1-4], [8-12]) have been proposed for syntactic and semantic analysis in natural language processing (*NLP*). But few of them have actually been implemented in real world *NLP* systems. We take a different route here by presenting a context sensitive model for concept understanding. Our model emphasizes knowledge utilization rather than its representation. Never the less, the model does provide a powerful knowledge representation scheme as can be seen in Section 7. It has been applied to several *NLP* systems and obtained very satisfactory results. Furthermore, certain elements of uncertainty are embedded in the model so that it can better simulate human understanding.

Our basic assumption is that the specific schemes for representing the meaning of a concept is not crucial as long as they suffice to reveal enough aspects of the concept. There are many phenomena supporting this argument. Consider two examples below:

(1) *Understand the proof of a mathematical theorem*: When a mathematical theorem is proved in a class, each student who understands the proof presumably has some kind of *representation* about the nature of the proof. Most of the time, such a representation if exists, is formed subconsciously and varies from person to person. But the exact forms of these representations are clearly not important since all of them are more or less equivalent. The teacher is also not concerned about such representations as long as the students can show that they *understand* the theorem

(2) *Describe a good tennis player*: We often hear people say that he or she has a friend who is a good tennis player. The fact is that everyone has a different notion about how a *good* tennis player (or good student, good friend, …etc.) should perform on the court. Furthermore, such a notion can be very fuzzy even for the same person since one *good* player in his mind could be significantly better than another *good* player. Thus, most people only have a rough idea about his or her own standard of a *good player* and it would be difficult to stipulate explicitly the conditions that a good player should satisfy.

To give a plausible solution to the theorem understanding phenomenon, one can adopt the following question-answer strategy. One way to evaluate whether a student indeed understands the mathematical proof is to give him a test. There are numerous ways to design such a test. For the convenience of practical implementation our test shall consist mostly of multiple choice questions (including “yes-no” questions). In our approach more attention is paid to the representation of the test questions rather than that of the concept itself. We believe the exact format and selection of the test questions has a great influence on the “effectiveness” of concept understanding. In fact, test construction is a key component in our entire design.

This paper is arranged as follows. In the next section we formally introduce our basic framework for concept understanding. This framework is applied in Section 3 to common sense description. The adoption of multiple choice questions and the detailed design of test questions and answers will be discussed in Section 4. Section 5 discusses the application of our framework to natural language understanding. In Section 5 and 6 we consider applications to the primary school mathematics and the multi-agent communication problems, respectively. Finally, Section 8 addresses our conclusions.

2. A Basic Framework for Concept Understanding

In our framework a concept no longer has a “unique” identity. It can vary from person to person, place to place. Accordingly, concept understanding, in our definition, is sensitive to contextual information. Rather than giving each concept a rigid “definition”, we describe things that are related to the concept, especially those actions and feedback that could be associated with the concept. Such descriptions can take many different forms since there are many different ways this concept can be characterized.

Consider the concept “love” for example. All of us probably have the experience of being deeply touched by some love story in a novel. We usually feel that the author of that novel really knows what true love is. Even though most people think they understand “love”, few of them can agree a hundred percent with each other on the exact description of “love”. Some said that “love is patient. Love is kind.” Others said that “love is like an ocean, full of conflicts, full of pain.” There is certainly no one definition of “love” that would satisfy everybody. It takes a whole novel for an author to elucidate what he or she thinks true love should be. A question naturally arises: “If we do not have an explicit representation of love, how do we say we understand love and how do we communicate with each other about the nature of love?” We believe the answer for most people would be: “I don’t need any clumsy representation to know what love is!”

Seriously speaking, what we try to convey here is that a rigid representation scheme of a concept is probably unnecessary (or sometimes too restricted) as far as understanding this concept goes. What is really important for a person to demonstrate that he or she understands a concept is for that person to give *correct* responses to many questions related to this concept.

We now describe our basic framework for concept understanding. It should be emphasized that, since this framework is to be implemented in a computer to build a simulation model, our definition has a strong “problem-solving” flavor. In other words, this approach is not meant to be a cognitive study of the true nature of human understanding, it merely represents a system designer’s viewpoint for a plausible simulation. Since a concept is not considered as an isolated notion, we shall refer to it as a *contextual concept*. Each contextual concept will be denoted by a pair, (concept name, query set), where the concept name C will be associated with a query set Q, in which every query has an answer (let us put aside the “correctness” issue for the time being). This CQ-pair serves as a description of the contextual concept. For example, the *frequently asked questions* (*FAQ*) about a concept could serve as a good example of the query set Q, though in practice the query set may be much larger. If the query set *Q* is obtained from a particular person's view, we say that the *CQ*-pair is a description of the contextual concept *in the mind of* that person. Thus, it is appropriate to think that the query set provides the context for this contextual concept.

For each *CQ*-pair description of a contextual concept, we say that a person *understands the contextual concept relative to its associated query set* provided this person can achieve a satisfactory score in a test sampled from the associated query set. One can measure the level of understanding by the scores obtained in the test. Since different people might have different understanding about a contextual concept, this will be reflected in our model by associating different query sets with this contextual concept. Thus, our concept understanding model is, by definition, *context sensitive*. We shall refer to our model as the *CSM*. Two people are said to *agree with each other* on a contextual concept if each of them can pass the other's test.

Another way of looking at our concept understanding model is that the above *CQ*-pair description provides a procedure for issuing a certificate (good for a certain period) that a person understands a contextual concept *C* relative to the query set *Q* much like the way a driver license is issued.

Consider the example of understanding a specific "person". For his colleagues, the queries could be related to his character, hobbies, or abilities. For his doctor, the queries could be related to his medical record. Thus, query set could contain actions, attributes and so forth. Different types of query sets are associated with different facets of the concept and could be best represented by different schemes or combinations thereof.

Another important aspect of our concept understanding model is that some sort of sampling techniques must be adopted in query selection for a test set. Since the entire query set associated with a concept can be quite huge, a smaller representative test set is often used in practice. Therefore, there is a probability factor involved in the test. A person scoring well against a specific query test yesterday might not necessarily score satisfactorily against a different query test today. This phenomenon provides a statistical uncertainty in the understanding of contextual concept, which is also very natural for human understanding. In the following we give a few more examples:

(1) *The driving test*: In this case, the government stipulates a standard to issue driver license. The *CQ*-pair description can be (eligible driver, driving test), which can be regarded as government's view on an eligible driver. However, unlike a written test whose scores can be calculated unambiguously, a driving test has to be conducted by testing officers and personal judgement is inevitable. So these officers have to be trained to have a common sense on how to conduct the test. Now, a person failing the driving test yesterday is not guaranteed to pass the test

today by only correcting the mistakes he or she made yesterday since a different test set might be adopted by a different testing officer. However, anyone who "really "knows" how to drive has a high probability of passing most tests.

(2) *A man's understanding of his girlfriend's love towards him*: First of all, this is a contextual concept that his girlfriend may not necessarily agree with. Furthermore, such an understanding could fluctuate rapidly depending on various circumstances occurring at different times. Any delicate change in her behavior or gesture could have unpredictable interpretations.

(3) *Law-abiding citizens*: A citizen who obeys the laws should at least "know" what most of the laws are. However, this is quite unlikely since government laws are so complex and change so rapidly that even the most sophisticated lawyer would have trouble keeping track of all the details, not to mention that the interpretation of each law could also be quite different. Therefore, a probability factor becomes a necessary component in such a qualitative assessment.

The depth of concept understanding in the *CSM* can be measured by the complexity of the corresponding query set. In terms of simulating human understanding, our *CSM* can faithfully reflect the following common (sometimes may be illogical) phenomena:

(1) *The individuality, subjectivity, creativity and the bias of human beings*: The degree of individuality can be reflected through the selection of different query sets or different answers to the same query. The *correctness* of an answer to a query is a very individual matter. It does not have to follow common sense.

(2) *The inconsistency of human beings*: The query set associated with an individual's understanding of a concept can change at any time. Sometimes, such a change involves changing an answer. Therefore, a person that understands a concept relative to a query set might give different answers to the same query at different times. The person might also answer query ① correctly at one time and fail at another time (for the sake of memory loss), but does exactly the opposite with query ② while still maintaining a satisfactory overall score.

(3) *The illogical behavior of humans beings*: Since answers to queries are not required to be compatible with each other, a person can make illogical choices and still pass the query test.

3. A Formal Description of Common Senses

Although all of us do not agree a hundred percent on any particular concept, we do not seem to have too much trouble communicating with each other in daily routines. Even though the *correctness* of an answer to a query can involve personal bias, we often seem to agree with each other on many issues thanks to the mysterious notion of “common sense”. Since our goal is to design a simulation model on a computer, we are less concerned about individual bias or subjectivity (even though one can never do away with these factors). In fact, we try to embed as much common sense in our computer program as possible. In the following we shall give a formal description about *common sense* based on our *CSM*. In Section 4 we shall discuss the actual construction of query sets for the understanding of common sense.

Similar to the *CQ*-pair description of concepts, we model “common sense” as a *CQG*-tuple, (common sense, query set, group). This 3-tuple consists of the name of a common sense, a query set *Q* with an answer to each query, and a group G of people. A correct answer to a query in *Q* is referred to as a *common sense answer*, which is also an answer that most people in the group would choose. A common sense about a query set among a group of people is a query set that most people in the group would be able to answer satisfactorily. We refer to such a query set as the *query set of this group for this common sense* (*QGC*). A person is said to *have this common sense of the specific group of people* provided that this person can pass a test sampled from the *QGC* satisfactorily. If a query set *Q* happens to be the associated query set of a particular contextual concept, then a person having this common sense of the specific group of people is said to *have a common sense* (*of the specific group*) *about the concept*.

As reflected in our model, we believe that common sense is subject to regional, educational, cultural and many other contextual differences. For example, English slang is difficult for a foreigner to pick up if he or she does not live in the states long enough. Likewise, it is difficult for American to get used to the “gender” issue in many European languages. The common sense about a disease among medical doctors is quite different from that among ordinary people. That is to say, a common sense for one group could be an expert knowledge for another group and vice versa. These phenomena can all be embedded in our *CSM*.

To give a more specific example of our common sense description, assume we try to let the computer understand the concept "honest". As far as the computer is concerned, without additional information, this word is just a sequence of 6 letters. To form a query set one can list many questions about what an honest person would react under different situations. Theoretically speaking, a computer that "understands" the word "honest" should be able to output a common sense answer to a query. For example, an answer to the question:

"What if he finds a parcel on the street?"
could be: "He would pick up the parcel and hand it over to the police";

an answer to another question:
"Would he cheat in an exam?"
could be simply: "No!"

Thus, what we need is a description of the effect of being honest. More examples will be given in the next section. Unlike a real human being that has a built in bias and subjectivity, we shall define personal profiles in the computer system so that each profile could respond to a query in a different manner. Profile clustering will then produce homogeneous groups in which each group is associated with certain common senses.

In the next section we shall describe the construction of query sets, a basic issue in our *CSM*. The design method can be applied to both individual concepts and common senses.

4. Implementation Issues in the *CSM*

The implementation of the above query-answer framework is quite natural for human beings since it is customary for human beings to take tests. The next questions are “How do we collect queries to test a computer?” “How does a computer system score well in a test?” and etc. Obviously, knowledge acquisition is important to constructing such a program. But a more important issue is knowledge utilization. In the *CSM,* the ability (to utilize knowledge) to answer queries satisfactorily is completely encapsulated from the construction of the query sets. Such an approach has the following two advantages:

(1) *There is no restriction on how a system designer has to do to obtain good test scores*: one may utilize heterogeneous knowledge representation schemes, various information retrieval methodologies as well as inference rules to derive answers to queries
(2) *There is no restriction on how an appropriate query set should be constructed*: query sets can be collected through various statistical sampling methods, rule-based and pattern-recognition techniques.

In other words, there is a lot of freedom in implementing our *CSM*. Nevertheless, we believe that certain restrictions on the *formats* of queries are not only helpful but also indispensable for a successful implementation of concept understanding systems on a computer. In this paper we shall emphasize the following type of queries: each query is a *multiple-choice question* (*MCQ*) that has at least one correct answer. Of course, we do not insist that all queries must be of this type. Generally speaking, in terms of entropy, answers to essay questions reveal more information, answers to *MCQ* gives less, and those to yes-no questions the least. Hence, yes-no questions are less preferable. On the other hand, it is quite difficult for the computer to figure out or narrow down an answer for an essay question. Moreover, even if an answer to an essay question is given, it is still a difficult task for a computer program to evaluate such an answer.

For most concepts, using multiple choice questions can reduce the problem of modeling human understanding to that of making decision among a finite number of alternatives. Since, for many essay questions, common sense has sufficiently limited the number of possible answers (after clustering). Through statistical sampling, we can eliminate those answers that are less probable. Such elimination can be tolerated in our *CSM* since the threshold for passing a concept understanding test is quite flexible. The remaining few candidates would then be the likely choices under different contexts and one can focus on identifying the appropriate contexts for each particular choice. Furthermore, for the problem of constructing multiple-choice questions, all kinds of information retrieval and pattern matching techniques can be used, which greatly facilitates the construction of the corresponding concept understanding system.

We now illustrate the query-design process using the following example. In understanding the concept, “wake up in the morning", the query set might contain questions asking how a person feels, what a person might choose to do right after waking up. Now suppose the designer decides to include the latter question. In constructing this *MCQ*, the designer has to

(1) figure out what the most probable answers are.
(2) identify situations under which a particular choice is preferable.

To deal with (1) and (2) above, common sense would be very useful. First of all, for (1), one might conclude that there are five most probable actions a person can take: take a shower, pick up the newspaper, drink coffee, take an exercise or eat breakfast. For (2), one might conclude that eating breakfast is the default action (probably most frequent) and we might have the following descriptions:

(a) in general, he will eat breakfast

(b) if he wants to fresh up, he would take a shower
(c) if he is eager to find out the result of an election, he would pick up the newspaper
(d) if he needs to start working right away, he would take a cup of coffee
(e) if he is very health-conscious, he would take an exercise.

So these different actions might be listed as candidate choices after a person wakes up in the morning. Alternatively, if we consider all of these actions in a script description, then the query might be to choose a priority of the above actions. The system designer might want to figure out many other conditions under which a person would take different actions. Statistical sampling would be useful here to collect "corpus" data. So we can have different queries asking what a person would do under different situations with the above five consequential events listed as possible choices. By following this procedure to construct other related queries, we would get a common sense query set for the concept "getting up in the morning".

From the above procedure for constructing *MCQ* one can see that it becomes the responsibility of the designer to explore all sorts of resources to compile sensible choices for an *MCQ*. Compared with collecting essay questions where it is difficult to judge the quality of an answer, we believe that the *MCQ* approach is more dynamic and effective for the simulation model. In the next section the reader will find that such an approach is especially useful in the construction of natural language processing systems.

5. Application to Natural Language Processing

Although it is not absolutely essential, we would like to describe our queries in natural language as much as possible. Natural language is more intuitive and has enormous descriptive power provided it is used unambiguously. To relate our concept understanding model with natural language understanding we shall classify the ambiguities in natural language into the following two types. The first type of ambiguities (referred to as the *contextual ambiguity*) are those that can be resolved by putting things in the right perspective (or context). The second type (referred to as the *intrinsic ambiguity*) are those that have different interpretations intrinsically. An example of intrinsic ambiguity is the following famous sentence linguists usually cite: "I saw a man on the mountain with a telescope". This sentence has two possible interpretations:

(1) I saw a man on the mountain and this man has a telescope;
(2) Through a telescope I saw a man on the mountain.

We are less interested in this type of ambiguities. An example of contextual ambiguity is the following sentence: “Where is Washington Memorial?” An answer could be “United States”, “Washington D.C.” or “Pennsylvania Ave.” depending on whether the question is raised outside the United States, inside the United States or within Washington D.C. As a matter of fact, we believe that it is the contextual ambiguity that provides natural language with tremendous expressive power. For our *CSM* specifically, the contextual ambiguity provides plausible candidate answers for our multiple-choice queries.

In the remainder of this section we shall illustrate our representation of the meaning of a word or a phrase. Such a representation certainly will contain typical dictionary descriptions about this word such as attributes of this word, syntactic and semantic categories, synonyms and antonyms as well as how to use this word. Besides those, our representation will contain many other items to be described below. We shall draw our examples from English-Chinese machine translation (*MT*).

We first consider the indefinite article "*a*" for example. The translation of the phrase "*a* + noun" into Chinese normally requires additional classifiers inserted between "*a*" and the "noun". There are several hundred classifiers in Chinese and each is associated with a particular type of nouns. Such a classifier-noun association can be best described in a table. Thus, we can imagine that there is an object [7] representing the article "*a*" and inside the object, there is an association table. The translation of "*a* + noun" can then be accomplished by passing a message about the noun following "*a*" to the object representing "*a*".

In the second example we illustrate a more complex word object. Consider the concept "reduce", which is synonymous to abate, lessen, etc. A possible query set for this concept is a list of questions about whether one should use “reduce” instead of “abate" or "lessen” under different situations. If one is interested in English-to-Chinese translation then a related query set could be about, among all possible Chinese translation for the word “reduce”, which Chinese word should be selected under different situations.

Phrases can be treated in a similar fashion. One can consider the translation of the phrase "*to make room for*" into Chinese. Consider the following three sentences:

(1) He moves the motor cycle away *to make room for* his car.
(2) Big trucks are forbidden on this road *to make room for* small cars.
(3) They tear down the old building *to make room for* a new road.

In each sentence "to make room for" has an implied (common sense) meaning. In the first sentence, the person wants to park his car. In the second sentence, the road would only allow small cars to pass. In the third sentence, they want to build a new road. Although in English it is unnecessary to mention the "implied" meaning in this case (since it is already well understood by everybody), a proper Chinese translation of these phrases would normally require that one spells out such implications. In different situations, the translation of the phrase, "to make room for", can vary in thousands of ways. An *MT* system that can correctly translate "to make room for" in most contexts would be regarded as “understanding” the meaning of this phrase. To achieve (or to simulate) this an *MT* system must be able to represent the background events and to make out (or compute) the expected events in different contexts.

One (natural) way to build such an *MT* system is to treat the phrase, “to make room for”, as an object and within this object, to describe all possible Chinese translations under different contexts. In order to evaluate how the *MT* system translates the phrase “to make room for” related to cars, one can form several queries for different uses related to cars and the possible choices are: drive, wash, wax, fix, race or park the car. A good *MT* system should be able to make the right choice under different situations.

Other important topics in natural language understanding include event representation of a sentence and discourse analysis. We shall touch upon some of them in Sections 6 and 7.

6. Application to the Elementary Mathematics Agent System

Given an elementary school mathematics word problem, we designed an *Elementary Mathematics Agent* (*EMMA*) system [5] that can automatically solve the problem, explain the solution procedure and the answer in natural language. We found that the most difficult part about elementary mathematics for kids is that, within the question, there are a lot of omissions or implications (more formally, ellipsis and anaphora) that maybe quite stylish for adults but confusing for kids. Thus, *EMMA* tries to discover and fill in those omissions and implications, derive the corresponding mathematical formulas and obtain an answer.

One important component of *EMMA* is that it needs to acquire enough common sense knowledge to deal with ellipsis and anaphora in the given problem. To achieve that we need to

(1) create event representation for natural language sentences. This part is related to natural language understanding.

(2) generate event association such as expected events and causal events that can be

used for inference. This would provide the candidate choices to many queries.

Another feature of the *EMMA* is that its understanding mechanism is not restricted to natural language understanding. For specific types of mathematics problems we need to create a script in which several current events and expected events are listed, the relationships among them are then inspected to make inference. Examples of such problems are:

(1) *The clock problems*: when do the minute and second of a clock overlap with each other, say, after 9 a.m.?

(2) *The cage problems*: how many chickens and rabbits are there in a cage if we know that there are 10 heads and 24 legs (note that the use of variables is disallowed in elementary school math.)?

Furthermore, instead of presenting the correct solution procedure to the student directly we can change *EMMA* to a system that asks the student to take step-by-step actions. At each step the student is asked to select an answer in a *MCQ*, where some of the candidate answers will indicate misconceptions on the part of the student. Thus, by taking a series of test questions, our modified *EMMA* can find out the potential problem with the student and provide needed help automatically. Naturally, this is a task requiring the collaboration with researchers in education methodology.

The above procedures are not much different from that used in an *MT* translation. What we did in *EMMA* is to translate a word problem into mathematical formulas that can be readily used to solve the problem. It is quite natural for our *MCQ* representation to define different levels of mathematical understanding through appropriate query set selection.

7. Application to Multi-Agent Communication

Intelligent agent is an important topic in the Internet. The role of a software agent in the Internet is similar to that of a broker who can offer certain service in the society. In order for an agent to function properly it needs to collaborate with many other agents. A natural way for agents to communicate among themselves is to use natural language [6]. The most important subject of their communication is to convey what each of them can do for the others. In other words, each agent wants to describe it so that others can *understand* what it is capable of.

In our agent society [6], each agent publishes a *handbook* describing what queries it can answer in natural language. The format of the handbook is very similar to the query-answer description of a concept. For a database agent *A*, we denote its handbook by *A*-handbook. We shall now introduce two kinds of concepts here. If we regard each database as a “concept”, then the (database, *A*-handbook)-pair would be exactly the *CQ*-pair description of the database concept in the mind of agent *A*. This can be regarded as agent *A*’s understanding of this database. Next, if we regard agent *A* as a “concept”, then the (agent *A*, query about *A*-handbook)-pair is the description of agent *A* in the mind of another agent B relative to B’s understanding about the content of *A*-handbook.

A handbook in our agent architecture serves many purposes. For example, consider an airline agent *A* who publishes a handbook containing queries regarding the availability of flights within certain periods together with their prices. Publishing such a handbook is the same as notifying other agents that, whenever necessary, they can inquire agent *A* using queries in *A*-handbook about the related flight information of this particular airline.

Furthermore, agent *A* can be tested based on a sample of queries taken from its handbook. If it passes the test satisfactorily, then a certificate can be issued (certifying that the handbook can be effectively used, namely, agent *A* really knows what it claims to know) so that other agents can send queries to agent *A* and have confidence on getting faithful answers. On the other hand, queries in the handbook provide a goal for the designer of agent *A*. In order to show that agent *A* “understands” the database it must be able to utilize knowledge in the database to answer those queries through database commands.

Besides database agents, we also create a layer of broker agents. The roles of broker agents are to support and enhance the communication between users and agents, and between agents themselves. A broker agent publishes its handbook to users or to some other broker agents. It also *collects* handbooks from other agents. An agent that *registers* its handbook with a broker agent is called its *subagent*. A broker agent is supposed to take more complicated queries from users or other agents than those given to database agents. Thus, a broker agent must have good command of natural language and the ability to utilize the handbooks of its subagents.

A broker agent answers a query in its own handbook by sending inquiries (including query words) to its subagents and integrating their answers. Thus, a broker agent needs to understand and coordinate its subagents. In other words, a broker agent needs to analyze handbooks of its subagents well enough so that it can answer a query in its own handbook by sending inquiries to its subagents and integrating their answers. For example, consider a travel agent that plans an itinerary for a customer. It should be able to communicate with the airline agents, train agents, map agents, hotel agents and etc. In addition, the travel agent needs to know how to integrate the information gathered from all these subagents to create a feasible and economical schedule.

In our model, a broker agent operates based on a collection of *events*. Using *NLP* techniques, handbook queries of its subagents are converted into events. Queries of its own handbook are also converted into events; these events are then decomposed and mapped to query events of its subagents. For each event, the broker agent can produce a set of expected events or actions, some of which simply generate queries for subagents. An event concept is considered *understood* if the broker agent takes the corresponding actions correctly.

Since the handbook of a broker agent could include those queries (or their abstraction) in its subagents' handbooks, it is "assumed" to possess the knowledge of its subagents. Therefore, handbook registration and assimilation provide a powerful way of knowledge accumulation for these agents, which is crucial for our knowledge representation scheme.

Thus, a handbook is not only a means for an agent to communicate with the outside world, it also propagates a lot of internal communication in the design process. Through the use of handbooks, a query-answer problem in an agent society can be decomposed recursively into smaller or more restricted instances so that they can be solved in a distributed fashion.

8. Conclusions

As mentioned before, an important job in constructing a concept understanding system in a computer is to first construct an appropriate query set as an evaluation criterion. Such a query set can be used as an initial goal and we then try to provide the computer enough knowledge to score well against this query set. Many different techniques such as statistical or machine learning strategies can be applied at this stage. In the process of building up such a query-answer system, the designer becomes more experienced and the query set may be augmented to be more robust and mature.

When a person is making inferences his thinking process is usually sequential. Since many decisions based on common sense can be made instantly, we suspect that decision-makings that require only common sense reflection can often be processed in parallel (biologically). This phenomenon, if verified, should explain why human beings could handle ordinary natural language so effortlessly.

Another important application of our concept understanding model is in user modeling or the construction of personal profiles in various systems. Since bias can be embedded in the query set, specific queries and answers pertaining to each individual's interest can be easily incorporated. By organizing the query sets into a hierarchical structure, feature extraction can be properly controlled at various coarse grain and fine grain levels.

In summary, we have provided a model for concept understanding with an attempt to

(1) simulate human understanding
(2) be context sensitive
(3) be useful in real world implementation

The discussion about our *CSM* is only a head start. It has been applied to a few on-going systems and the results are quite impressive. We shall demonstrate its full power through the implementation of several related application systems in the future.

References

[1] R. J. Brachman and H. J. Levesque (ed.), *Readings in Knowledge Representation*, Morgan-Kaufman Publisher, Los Altos, CA, 1985.

[2] N. Chomsky, *Syntactic Structures*, Mouton, The Hague, 1957.

[3] C. J. Fillmore, *The Case for Case*, in Universals in Linguistic Theory, edited by E. Bach and R. Harms, Holt, Rinehart, and Winston, New York, 1968.

[4] G. G. Hendrix, E. D. Sacerdoti, D. Sagalowicz, and J. Slocum, *Developing a Natural Language Interface to Complex Data*, ACM Transactions on Database Systems, Vol. 8, no. 3, 1978.

[5] W. L. Hsu, Y. S. Chen and Y. K. Wang, “Elementary school Math. tutoring agent”, *Proceedings of Agent Technology Workshop*, 1997.

[6] W. L. Hsu, Y. S. Chen and Y. K. Wang, “Natural Language Agents— An Agent Society on the Internet”, Technical Reports, Academia Sinica, 1998.

[7] W. L. Hsu, W. K. Shih and P. H. Yeh, “Object oriented concept representation", Proceedings of *ICCPOL'95*.

[8] M. P. Marcus, *A Theory of Syntactic Recognition of Natural Language,* MIT Press, Cambridge, MA, 1980.

[9] J. McCarthy, *Notes on Formalizing Context*, Proc. IJCAI, 1993.

[10] R. C. Schank, Conceptual Dependency: A Theory of Natural Language Understanding, Cognitive Psychology, Vol. 3, no. 4, 1972.

[11] T. Winograd, *Language as a Cognitive Process*, Volume 1: Syntax, Addison-Wesley, Reading, MA, 1983.