DOI: 10.2298/CSIS100915027F

Problem Solving by soaking the concept network

Xixu Fu and Hui Wei

College of Computer Science, Fudan University
ShangHai, China
081024009@fudan.edu.cn

Abstract. Because of the complexity and fuzziness of the real world, it's hard to build a dense knowledge system and reason in it with traditional methods. But man can deal with such tasks freely. Inspired by cognition and linguistics, a solution is advanced for reasoning dense knowledge in this paper. Objects and concepts are organized in the form of concept graph. Soaking the nodes in the graph until the result is represented in the graph the final graph can be the explanation of the scenario. With the naïve algorithm, monotonic scenario reasoning problem can be solved in dense knowledge environment.

Keywords: cognition, scenario reasoning, knowledge representation, object-oriented, concept graph, soak.

1. Introduction

Natural world is a complex world. P. J. Hayses defined a good knowledge system to be a dense knowledge system with detailed description for objects [27]. However, it is hard to explain how a scenario taken place for machines.

There are two main problems in the construction of expert systems [30]: the bottleneck of knowledge acquisition and the narrow scope of knowledge system. The former involve that how to translate the knowledge in the real world into the knowledge that the expert system can use, while the latter involve that expert system only adapt to domain-restricted problem solving, once related to the issue of domain-opened, its capacity of problem solving will become very vulnerable. The main reason for these problems is that the knowledge of expert system is almost mechanical and lack of the support of the underlying semantics. They are not from the perspective of the development of concept to construct a concept system.

Association is a main reasoning method of human. It's a powerful method in reasoning. For dense knowledge system, association can be an efficient method. Let's see the example below.

Once, a little horse went across a little river with a bag of salt on its back. It bumped into water by mistake. But after the mistake, it found the bag become light and feel happy. The next day the little horse went cross the river with a bag of cotton on its back. It felt into the water on purpose. But this time it felt heavier after stood up.

Men use association to solve these problems while computers search state space for such tasks. We can naturally understand the story. But for computer algorithms, it’s hard to define and get the result. There are even many muddy tasks without precise definition. For these problems, traditional state-space searching can do nothing.

Relations are complex in actual problems. Connectionism [7] is suitable for complex problems, but connectionism doesn’t maintain accurate relations. This may cause errors in the reasoning process.

In developmental psychology, objects are always the focus of children in knowledge acquisition. The thinking process also starts from object. When all the details loaded into the mind, the process is built.

Inspired by psychology and linguistics, a method is advanced to simulate man’s association and solve many actual problems. Objects and concepts are regarded as basic elements. Begin with known objects, the method soak the knowledge network and activate more and more concepts and objects. When all the nodes are related, the problem is solved.

In this paper, related works are first analyzed first. Then, basic elements and concept network is defined. After that, dynamic soaking process is analyzed with an example. The algorithm is provided next based on the analysis. At last, some discussions are carried about efficiency and future works of the method.

2. Related Work & Motivation

2.1. Symbolism and connectionism

The advocates of symbolism try to represent knowledge with symbols. Symbols can form reasoning precisely, but it is hard to build a complex system with symbols. Connectionism [7] is suitable for complex problems, but connectionism doesn’t maintain accurate relations. This may cause errors in the reasoning process. The problem is to find basic elements that can both represent complex semantics and relation.

2.2. Linguistic Semantics & Pragmatics

Language reflects the method man’s comprehension of the world. Words are potential elements describing the world. They can be important reference in building knowledge systems. Words and grammar build the semantic of language [12].

Real nouns [11] are basic elements of language and cognition. Children first recognize them. Verbs and rhetoric words such as adjectives and adverbs are acquired consequently. Reference words such as prepositions and pronouns are aware last [13]. These are basic element of human knowledge. Because of homonymy, polysemy, semantic motivation and the changing of background linguistic Semantics and Pragmatics varies a lot. Language cannot be a unified representation and reasoning tool for machines.

2.3. Linguistic Based Knowledge Systems

There are many knowledge systems based on linguistic semantics.

CYC [19] is a well known linguistic based knowledge system. It advanced the idea to build a large-scale knowledge system that can solve various artificial intelligent problems. The system maintains a large knowledgebase with common sense. With the reference of common sense, the system can solve many common sense concerned problems such as NLP.

WordNet [18] is a well-known semantic dictionary. It organizes words by their semantic and lists semantically related words and syntax. Many tasks have been carried out with this system.

These linguistic based system solved many problems about semantics and syntactic. But process concerned reasoning cannot be implemented based on these systems. Semantics of these systems are not abundant to solve complicated problems.

2.4. Semantic Network

Semantic network [14, 17] is a popular knowledge representation. Many knowledge-based applications [15, 22] are built based on knowledge in this format. Objects are related with slots named isa and ako. The semantic of semantic network is not rich enough to support process reasoning.

2.5. Searching based Methods

Searching based methods have been widely used in simple problem solving tasks. These methods include state-space searching and encoded searching methods such as evolutionary computing and genetic algorithm.

State-space searching [3] algorithms can find a way to solve some model problems in which all states are known. But state-space searching is time consuming and limited by knowledge structure [1]. For complex problems, it's almost impossible to find what happened.

State-space searching has its shortcomings. According to Alexander's paper [1], five problems are concerned:

- State Space Search has excessive space requirements.
- State Space Search is time-consuming.
- State Space Search is difficult to implement.
- State Space Search can only be used in uniform trees.

– State Space Search dominates directional searches.

Further more, states are not easy to list and potential states are of great number. Although many heuristic algorithms such as A* [3,4] are invented to reduce the complexity. They can't be applied to complex process reasoning tasks.

Evolutionary computing and genetic algorithm [29] belong to another type of searching algorithms. State-space is encoded into bits or characters. Searching is carried out by intersection and mutation starting from random individuals.

2.6. Reasoning & Association

Fortunately, man's knowledge acquisition and thought can be important reference of building knowledge systems. Association is an important ability of human. Some works related to human association have been advanced in field of neural network and pattern recognition [28].

Children recognized objects first, and then acquire the features of recognized objects. They relate objects together by their reactions and similarity. When thinking, we focus on objects first too. For example, when we see a fly and a magazine, we first focus on fly and magazine, then know we hate the fly and we can kill it with tool such as a magazine, after that we kill the fly with the magazine.

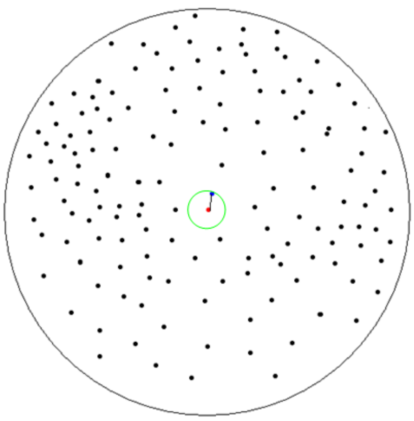

Fig. 1. shows a task that to find nearest point to the red point in a large space. There are large amount of points in the whole space, but most of them needn't be concerned in humans' opinion although computers trends to search all or many of them.

As shown in figure 1, traditional computer algorithms focus on the whole state space while men focus on related concepts only. Although many heuristic algorithms have been advanced to limit the searching space, traditional methods can't tell how a natural scenario takes place. Men's association process can provide a way for reasoning in complicated knowledge environment.

2.7. Previous Work and Basic Theory

Some work has been done before this work. Objects, concepts and scenarios and related operations are defined in Xixu Fu and Hui Wei's previous work about knowledge architecture [20]. Michael Freund's work [16] and John F. Sowa's book [17] provide basic theory about knowledge representation and identification.

3. Concept Network

The concept network is the core of the problem solving system. Problems are resolved into objects which relate each other with concepts and operations. Objects activate each other according to the relations, and the activating process constructs the whole process of the problem.

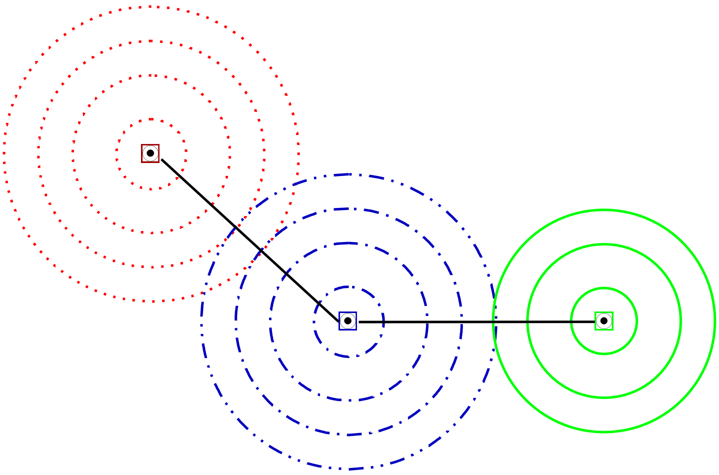

Fig. 2. This figure shows the process of association. Soaking begin with known nodes which are represented by points in the figure. Related nodes are activated to connect the initial nodes together.

3.1. Definition of Elements in Concept Network

As an object-oriented system, objects and concepts are the basic elements. Anything are regarded as objects and abstracted into concepts. All the objects consists the knowledge network. Objects and concepts are regarded as nodes of the concept network. Nodes and relations consists the whole network.

3.1.1. Various types of Objects

There are many types of words including nouns, verbs, and adjectives and so on. Nouns, verbs and adjectives and adverbs are regarded as lexical words. Other are mainly regarded as grammatical words [25]. Lexical words are semantically important. Various kinds of objects exist in the world too. Similar to natural languages, we can categories these objects into four classes:

– Nominal objects are independent objects that can be represented with nouns.
– Actions are objects to describe the behavior of nominal objects.
– Rhetoric objects are objects used to describe features of nominal objects and actions.
– References are objects that indicate the temporal or spatial situation of objects and actions.

Nouns can represent nominal objects whether concrete or abstract. For example, desk and thought are both nominal objects. Nominal objects can have other nominal objects as parts. A proposition or scenario can be regarded as a nominal object.

An action often has an actor and a target. Some acts may have two targets, one is direct and the other is indirect. An action can be represented as a pentad ACTION<Actor, DirectTarget, [IndirectTarget], Process, Result>. Actor is the object that carries out the action. DirectTarget is the object receives the act. IndirectTarget is the media of the act. Result is the result of the action. Process is the process of acting.

Rhetoric objects are rhetoric to nominal objects or actions. Adjectives and adverbs can be regarded as attributes.

A reference is a dyad REFERENCE<Precedent, Hind>. For example, in the proposition A is on B. A is the precedent of on while B is the hind.

Every kind of objects has its distinctive attributes. For the sake of reasoning, every object has an attribute to describe whether it is activated. An activate operation is used to determine whether related objects can be activated and set them activate.

From the analysis of objects, we can define object as below:

Definition 1: Object is a dyad O<D,E>. D is the description of the object which includes the name of objects and components described by private or protected elements. E is the set of components can be described by public elements.

Variables and functions are regarded as objects. For functions parameters are regarded as public elements.

Figure 3 shows the representation of four kinds of objects.

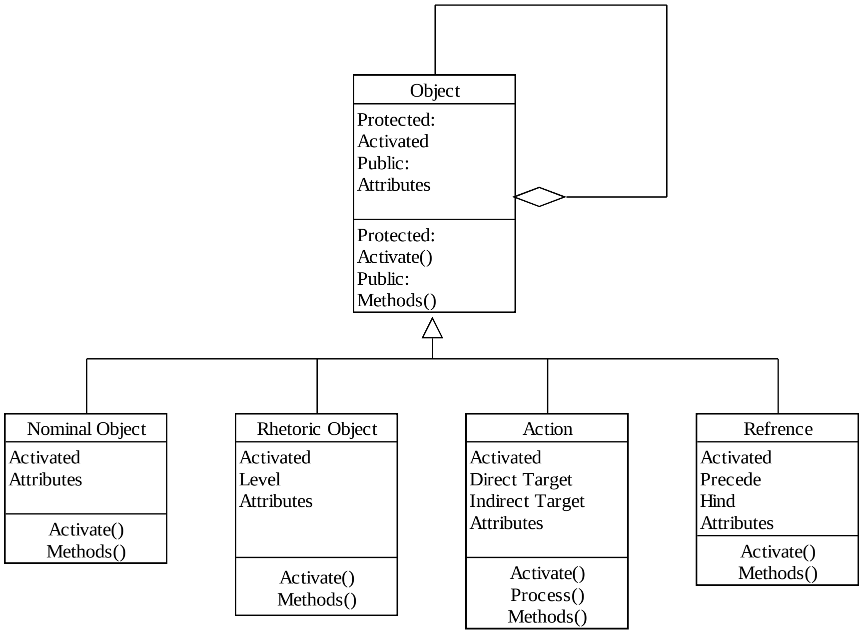

Fig. 3. The figure organized representations of objects and four kinds of objects. Attributes of an object can be regarded as object too.

As shown in Fig. 3, all objects need an activation method to construct the activating process. This method can be included in the constructor of object.

3.1.2. Concepts

Concepts are abstracted from objects. The set of concepts can be represented by a dyad C<D,E> too. It's defined as below:

Definition 2: Concept is a dyad C<D,E>. D is the description of the object which includes the name of objects and components described by private or protected elements. E is the set of components can be described by public elements.

3.1.3. Nodes

Objects and concepts are nodes of concept network. They can have a unified definition as below:

Definition 3: Node is a dyad N<D,E>. D is the description of the object which includes the name of objects and components described by private or protected elements. E is the set of components can be described by public elements.

3.1.4. Relations and Scenarios

Objects and concepts are related with three types of relations: inheriting, convergence and forth putting. When an object or a concept is activated by another object or concept, they become related and a relation is build between them. A scenario is build up with objects, concepts and relations. A scenario can represent a status or a process.

Scenarios are defined to describe things. A scenario is defined as the collection of concepts, objects and their relations. It's defined as below:

Definition 4: Scenario is a triad S<C, O, R>. C is the set of concepts in the scenario. O is the set of objects in the scenario. R is the set of relations in the scenario.

3.1.5. Concept Network

A scenario can be described by concept network which described by nodes:

Definition 5: Concept network is a dyad NW<N, R>. N is the set of nodes in the concept network. R is the set of relations in the scenario.

Concept network can be regarded as a structured node.

3.2. Knowledge Representation with Concept Network

A problem can be represented by a graph consists of concepts and objects as nodes. Nodes are connected with relations. When nodes are activated, they can represent definite fact.

For example, the statement Salt can be dissolved in water can be represented by figure 4.

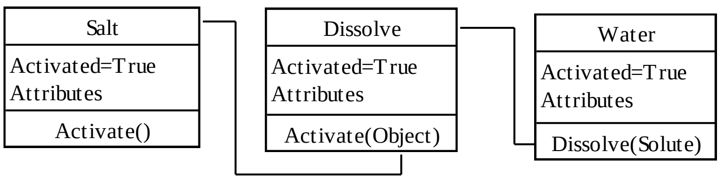

Fig. 4. The Representation of the Statement Salt can be dissolved into water

When all three nodes are activated, the graph means that water can dissolve salt.

3.3. Handle Negations

A graph plus a negative node has the graph as a sub graph. This may course the error that A is not B implies A is B. Negative words such as *no* and *not* are not regarded as objects or concepts. All objects and are positive objects. Negations should be merged into the node it charges. Conflict should be avoided. For example *not important* may be represented by the object unimportant or not_important. Anything can't be both important and unimportant.

Definition 6: A concept network NC_1 is a sub network of network NC_2 when and only when all the relations and nodes in NC_1 are also in NC_2.

Theorem 1: A concept network implies any of its sub networks.

4. Graph Reasoning

4.1. Overview of the Concept Network Reasoning

4.1.1. Activation of Nodes

A node is activated when the definition has been fulfilled or when it's related to activated objects. There are two major relations between objects. One is inheriting the other is convergence. The activation of nodes can be divided into static activate and dynamic activate.

Definition 7: Static activate is defined as the activation by node structure. Such as a node activated by its components or parent nodes.

Definition 8: Dynamic activate is defined as the activation that a node activate other nodes by its operations.

The activation of nodes can be divided into possible activation and definitely activation.

Definition 9: A node is definite activated if the node can be activated without decision by its activate operation.

Definition 10: A node is possible activated if the node can't be activated without decision by its activate operation.

The activation of nodes can be described below:

- Nodes mentioned as input are regarded as activated objects or concepts.
- If a node is activated then its parent concepts can be activated definitely.
- If a node is activated then objects or concepts as parts of it can be activated definitely.
- A node can be activated definitely by fulfilling the definition in the activate operation.
- An object or concept can be possibly activated when it inherits from an activated concept.
- An object or concept can be possibly activated when some activated objects or concept composes it.

Definite activation can be used in normal reasoning. Possible activation is not used in definite reasoning and only being useful in some possibility reasoning such as imagination. When an object is activated, its attribute activated is set true.

4.1.2. Static View of Problem Solution

Any problem has a start scenario and a result scenario. All nodes in both scenarios and relations in the start scenario are extracted to build a concept network. Then the soaking process begins. When the result scenario can be represented in the concept network, the problem is solved.

4.1.3. Activation Process

An object can activate and relate objects or concepts as its parameters. When a node is activated, an edge between it and the node activate it is added into the reasoning graph. Let's take the problem represented by figure 4 as an example. Let salt in water be input scenario. Figure 5 presents the process of reasoning. The expected result is salt dissolve in water.

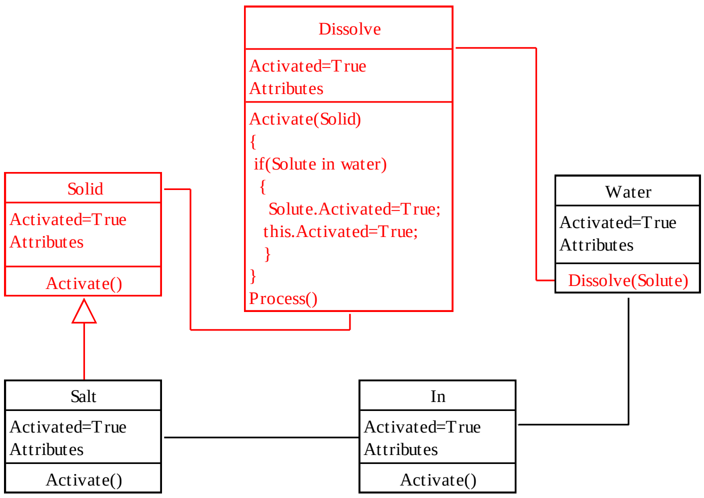

Fig. 5. The Solution of the Problem Salt in Water

The objects *water*, *in* and *salt* are connected in a known scenario. Object dissolve is activated as an operation of water. The concept solid is activated as a parent concept of salt. Then dissolve activates solid and the problem is solved.

4.2. Explain Complex Scenarios by Soaking Concept Network

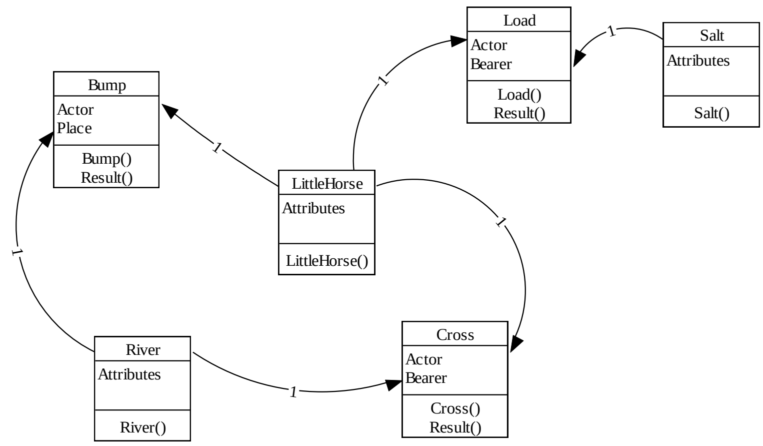

Fig. 6. Initial status of the story can be shown in this figure. The little horse is crossing a river, loading some salt. Arrows mean activation.

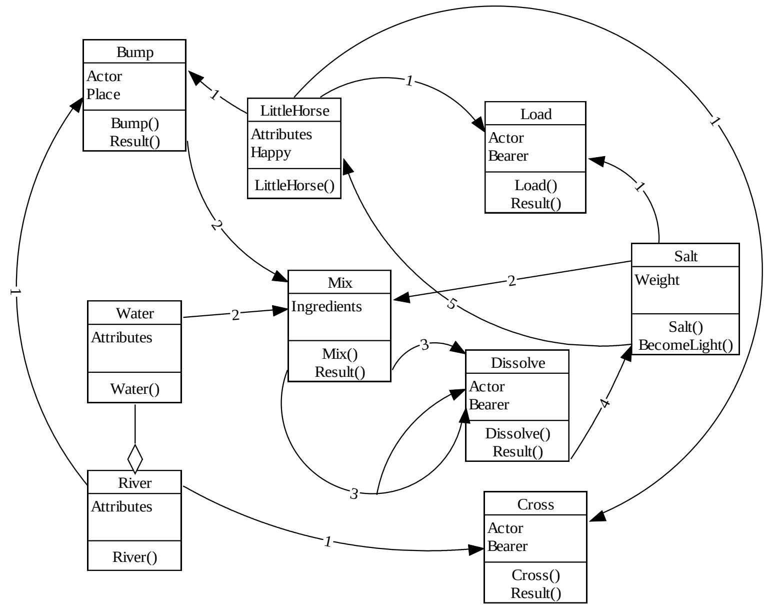

Fig. 7. This figure shows the overall activation process of related concepts. The graph can explain the process of the first scenario. The little horse bumped into river when crossing it. Salt mixed with water and dissolved in the water. So salt become light and the little horse felt happy.

Let's take the story mentioned in the introduction session as an example for scenario comprehension. The story can be represented into two scenarios. The first describes the first time the little horse crossing the river. The second show the next bumping in the river. These scenarios need complex knowledge to explain. This section shows how the story can take place. To represent the process neatly in the paper, the figures are simplified. Attributes are not represented as a node in the figures. Unrelated nodes are ignored too.

The beginning of the first scenario can be described as Fig. 6. Nodes such as bump and river are regarded as initial nodes. Initial relations are also given out. The reasoning process can be shown in the next figure.

In the second scenario, some activated nodes such as happy should be brought in. The process can be represented as figure 8.

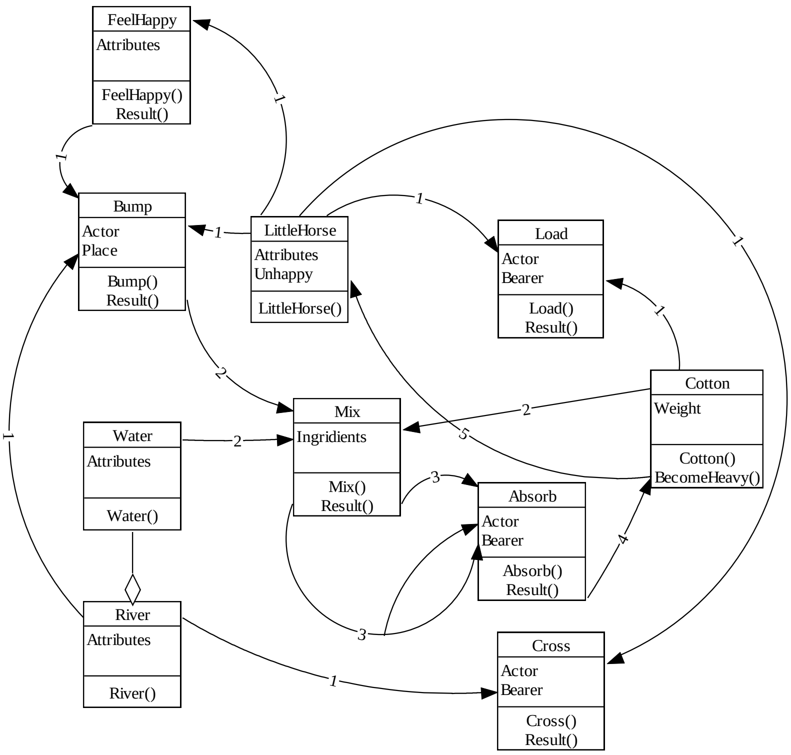

Fig. 8. This figure represents the bump happened the next day. Because the horse fell happy, it repeated the same bump in the same river. What's different is cotton can absorb water and become heavy. That cause the little unhappy.

The story can be explained by the soaking the existing knowledge begins with the given nodes. The soaking results should include the explaining of the target scenario.

5. Problem Solving Algorithm

5.1. Reasoning Process

Before problem solving, known scenarios are analyzed. A directed graph of reasoning is set up according the concept networks represent them. Nodes of concept network are regarded as nodes of the graph. Known relations are added to the graph as edges. Then new nodes are found by searching the knowledge network and activated. When activated, they become known nodes and take part in the next round of soaking. When all the known nodes are activated, a reasoning graph is built which is called G. If the reasoning graph of result scenario is the sub graph of G, G can map into a concept network which represent the answer of the problem.

5.2. Naïve Algorithm

The simplest method is to soak in a hierarchical way until all needed nodes in the result graph are activated. The algorithm may cost a lot of time and space, but the output can include the full process of the scenario. That's to say the scenario can be explained by the algorithm. Let CNBegin and CNresult be the scenarios for the concept networks represent the beginning status and the expected result. Let Ni be nodes of the joint of SBegin and Sresult. Let Ei be the edges of CNBegin. Let G be the result graph. Let Gresult be the reasoning graph of G. Fig. 9 shows the algorithm.

After application of Algorithm 1, a graph G is returned as the answer. The answer is a solution of the problem, but not a simple one.

5.3. Deduction

To find a simple answer of the problem, redundant nodes and edges should be removed from G. For the deduction process, key nodes are defined as the initial nodes of the problem graph. Nodes other than key nodes are called normal nodes. They are activated in the soaking process. If a node can't activate any key nodes, it's unnecessary for the problem solving. Figure 10 shows the deduction algorithm.

With the algorithm described in Fig. 10, unnecessary nodes can be deleted to make the representation simple and clear.

```
Algorithm 1:
ProblemSolve(CNBegin,CNResult)
{
  //Initialization
 for all Ni in CNBegin or CNResult
     if(Ni∉G.Nodes) G.AddNode(Ni);
  for all Ei in SBegin
     if(Ei∉G.Edges) G.AddEdge(Ei);
  //Soak
 while(GResult is not a subgraph of G)
     {
      for all Nj in G
         {
          for all Nk as parent concept of Nj
              {
               Nk.Activated=True;
               G.AddNode(Nk);
               G.AddEdge(Nk,Nj);
              }
            for all Nk as part of Nj
              {
               Nk.Activated=True;
               G.AddNode(Nk);
               G.AddEdge(Nk,Nj);
              }
             //For activate operation with more than one parameters, Try all combinations
             for all Ns in G
                if(Ns can be parameter of Nj)Nj.Activate(Ns);
            }
         }
 return(G);
}
```

Fig. 9. The Soaking Algorithm

6. Experiments and Results

Because the evaluation method can't be reasonably defined, heuristic algorithms and genetic algorithms can't be applied effectively. Only traditional state-space searching algorithm can be compared with soaking method. Hops and soaked nodes can reflect the complexity of soaking algorithm while potential states can be regarded as the complexity of traditional state-space searching algorithm.

We tried some dense monotonic scenarios with the soaking method and state space searching method. Any value change of any object can cause a new state. Because continuous value can cause infinite states, they were ignored in the experiments. The result is presented as below:

Algorithm 2:

```
Deduct(G)
{
Node Neighbor;
int KeyPaths;
 For all normal node Ni in G
     {KeyPaths=0
      for all neighbor nodes Nj of Ni
          {Neighbor=Nj;
            if(FindKey(Nj))KeyPaths=KeyPaths+1;
           }
       if(KeyPaths<1)G.DeleteNode(Ni);
      }
}

Boolean FindKey(Node)
{
  if(Node is a key node)
     return ture;
  else
     FindKey(Node.NotVisitedNeighbor);
}
```

Fig. 10. The Deduction Algorithm

Table 1. Result of Soaking Method and Traditional State-space Searching Method in Scenario Comprehension Tasks.

Problem	Hops/Soaked Nodes of Soaking Method	Number of Potential States for State space Searching Method
Salt solute in water.	2Hops/36Nodes	64
A bag of salt drop into water.	4Hops/257Nodes	9216
A little horse with a bag of salt pass a little river and felt into it, then the horse felt happy.	8Hops/1026Nodes	Not available

From the result of three simple problems, we can see state space searching method can cause sharp increase of complexity in dense knowledge systems. What's more, if all elements of the problem are not explicitly stated, the complexity may be related to all the details of knowledge base. It means the problem can't be solved by with such methods. Object-oriented soaking method can solve such problems in dense knowledge environment.

7. Discussion and Future Work

7.1. On Knowledge Network and Complexity

Object oriented method provide rich semantics. For state searching algorithms, every value of any concept may course a change of state. Because the total number of object is unknown, states of the problem are unknown. Let totally N concepts and objects concerned in the problem and every concepts and objects have n values. The number of states is n^N. Be aware that N should be an unknown great number. Even heuristic algorithms can't handle this in limited time.

The basic element of this method is object, so the number nodes become N*n. Because only a few objects and concepts can be activated, N can be a relatively small number. Provide the answer is fixed, when the knowledge network become more complex, the method become slower. But more simple answer may be found.

7.2. Enhance the Method with Association Rules

As experience can help people find answer quickly, mining results such as association rules can reduce object activation or even find answer directly. This is a promising way to enhance the method.

7.3. More Flexible Methods

In this paper, only definite activation is used to activate nodes. Other mechanisms such as association need to activate nodes with similar features. These mechanisms may be concerned in future works.

8. Conclusion

Inspired by human's association, a naive object-oriented method is advanced to comprehending how things happened without knowing all the states. By soaking knowledge network, the process is discovered and represented as a reasoning graph. The method is much less complex than state space searching methods in dense knowledge systems.

Acknowledgements. This work was supported by 973 Program(Project No. 2010CB327900) and Shanghai Science and Technology Development Funds(Project No. 08511501703).

References

1. Alexander Reinefeld, and Peter Ridinger: Time-Efficient State Space Search. Artificial Intelligence Vol. 71, No. 2, 397-408. (1994)
2. Meuleau Nicolas et al.: A heuristic search approach to planning with continuous resources in stochastic domains. Journal of Artificial Intelligence Research, Vol. 34, 27-59. (2009)
3. Rüdiger Ebendt and Rolf Drechslerb: Weighted A* search – unifying view and application. Artificial Intelligence, Vol. 173, No. 14, 1310-1342. (2009)
4. Yngvi Bj¨ornsson, Vadim Bulitko and Nathan Sturtevant: TBA*: Time-Bounded A*. In proceeding of IJCAI2009, 431-436 (2009)
5. K. Araki, S. Gnesi, and D. Mandrioli: Using Abstractions for Heuristic State Space Exploration of Reactive Object-Oriented Systems. In Proceeding FME 2003, 462-481 (2003)
6. Jean Piaget: Structuralism. New York: Basic Books, USA. (1970)
7. Adrian Cussins: The Connectionist Construction of Concepts. The Philosophy of Artificial Intelligence. Oxford University Press, UK. (1990)
8. Jonathan Lawry: A framework for linguistic modeling. Artificial Intelligence, Vol. 155, 1–39. (2004)
9. LinXiang Jia: Connectionist Cognitive Psychology. Shanghai Education Press, China. (2006)
10. Fodor J. A. and Pylyshyn Z. W.: Connectionism and Cognitive Architecture: A Critical Analysis. Cognition, Vol. 28, 3-71. (1988)
11. David Barner, Shunji Inagaki and Peggy Li: Language, thought, and real nouns. Cognition, Vol. 111, 329–344. (2009)
12. M. A. K. Halliday: An Introduction to Functional Grammar. HODDER ARNOLD. (2008)
13. Michael Eysenck and Mark T. Keane: Cognitive Psychology: A Student's Handbook. Psychology Press, a member of the Taylor & Francis Group. (2000)
14. Lokendra Shastri: Default reasoning in semantic networks: A formalization of recognition and inheritance. Artificial Intelligence, Vol. 39, No. 3, 283-355. (1989)
15. Hai Zhuge: Communities and Emerging Semantics in Semantic Link Network: Discovery and Learning. IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 6, 785-799. (2009)
16. Michael Freund: On the notion of concept I. Artificial Intelligence, Vol. 172, 570–590. (2008)
17. John F. Sowa: Knowledge Representation: Logical, Philosophical, and Computational Foundation. Machine Press, China. (2003)
18. Chenghua Dang et al.: Wordnet-based summarization of unstructured document. WSEAS Transactions on Computers, Vol. 7, No. 9, 1467-1472. (2008)
19. Douglas B. Lenat: CYC: A Large-Scale Investment in Knowledge Infrastructure. COMMUNICATIONS OF THE ACM, Vol. 38, No. 11, 33-38. (1995)
20. Xixu Fu and Hui Wei: Cognition Inspired Object Oriented Knowledge Warehouse Architecture. Journal of Software, Vol. 5, No. 9, 926-933. (2010)
21. Adelman, JS and Brown, GDA.: Modeling lexical decision: The form of frequency and diversity effects. Psychological Review, Vol. 115, 214-227. (2008)

22. Zdravkovic M, Trajanovic M.: Integrated Product Ontologies for Inter-Organizational Networks. COMPUTER SCIENCE AND INFORMATION SYSTEMS, Vol. 6, No. 2, 29-46. (2009)
23. Garcez Artur S. D'Avila, Gabbay, Dov M. and Lamb Luis C.: Value-based argumentation frameworks as neural-symbolic learning systems. Journal of Logic and Computation, Vol. 15, No. 6, 1041-1058. (2005)
24. Cruse D. A.: Language, Meaning and Sense: Semantics, An Encyclopedia of Language. Routledge, London. (1990)
25. Barber C.: The English Language: A Historical Introduction. Cambridge University Press, UK. (1993)
26. Otto Jespersen: The philosophy of grammar. Beijing World publishing company, China. (2008)
27. P.J Hayses: The naive physics manifesto. The Philosophy of Artificial Intelligence, Oxford University Press, UK. (1990)
28. Anthony Morse and Malin Aktiusa: Dynamic liquid association: Complex learning without implausible guidance. Neural Networks, Vol. 22, No. 7, 875-889. (2009)
29. Angelo Cangelosi: Evolution of Communication and Language Using Signals, Symbols, and Words. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, Vol. 5, No. 2, 93-101. (2001)
30. Forsythe D.E., Buchanan B.G.: Knowledge acquisition for expert systems: some pitfalls and suggestions. IEEE Transaction on Systems, Man and Cybernetics, Vol. 19, No. 3, 432-435. (1989)

Xixu Fu was born in 1981. He received his master degree in 2007. Now he is a Ph.D Candidate in Fudan university. His research interests include artificial intelligence, cognitive science and data mining.

Hui Wei was born in 1971. He received his PhD in department of computer science at Beijing University of Aeronautics and Astronautics in July 1998. From September 1998 to November 2000 he was a postdoctoral in the department of computer science and institute of artificial intelligence at Zhejiang university. In November 2000 he jointed the department of computer science and engineering at Fudan University. He is currently a professor in Fudan university. His research interests include artificial intelligence and cognitive science.

Received: September 15, 2010; Accepted: February 17, 2011.