TRIZ Oriented Artificial Intelligence

Mehdi Akbari
Islamic Azad University of Khomeini Shahr
Akbari@creatology-triz.com

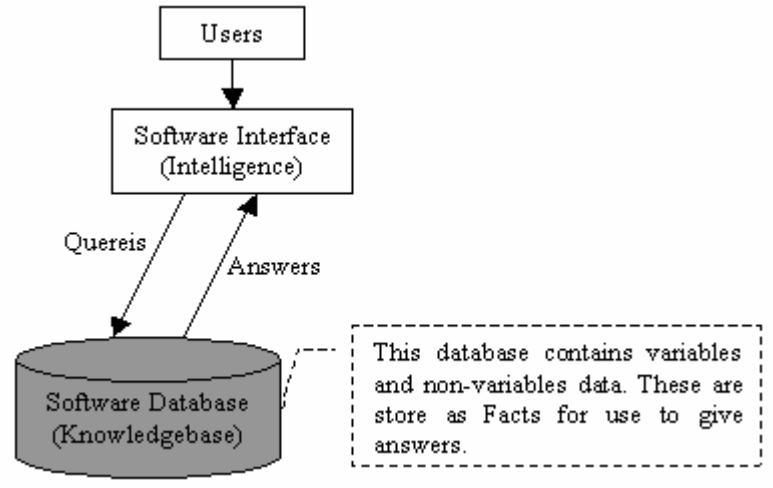
Abstract
One of the central challenges of computer science is to get a computer to do what needs to be done, without telling it how to do it. Artificial Intelligence (AI) addresses this challenge by providing a method for automatically creating a working computer program from a high-level problem statement of the problem it is one of the best ways to improve software's performance for solving complex problems. We have a various methods for simulate intelligence such as Semantic Network, Artificial Neural Network (ANN) and so on, but all of them, does not have to confine itself to methods that are biologically observable. In the TRIZ oriented Artificial Intelligence we present new approach for use in software to make them more useful and intelligence. 40 inventive principles of TRIZ is a powerful methodology to form models for design new software to have a more similar characteristic from human intelligence.

Introduce of Artificial Intelligence
Artificial Intelligence is a branch of Science, which deals with helping machines, finds solutions to complex problems in a more human-like fashion. Artificial Intelligence aims to improve machine behavior in tackling such complex tasks. This generally involves borrowing characteristics from human intelligence, and applying them as algorithms in a computer friendly way. Computers are fundamentally well suited to performing mechanical computations, using fixed programmed rules. This allows artificial machines to perform simple monotonous tasks efficiently and reliably, which humans are ill suited to. For more complex problem, things get more difficult. Unlike humans, computers have trouble understanding specific situations, and adapting to new situations. Together with this, much of AI research is allowing us to understand our intelligent behavior. Humans have an interesting approach to problem solving, based on abstract thought, high-level deliberative reasoning and pattern recognition. Artificial Intelligence can help us understand this process by recreating it, then potentially enabling us to enhance it beyond our current capabilities.

Background
AI is a way to simulate human brain in software. Computer software helps us to solve our problems with using algorithms. Whatever we used exact algorithms we have powerful software to solve problems. In AI programming, we try to design new algorithm for increase creativity in software. There are many different approaches to Artificial Intelligence. The main idea in AI is using knowledgebase for inference as illustrated in Figure 1. In knowledgebase, we have to keep a lot of records named "Facts" to form inference as results. AI uses deferent way to create knowledgebase, store Facts and use them for give result such as Semantic Network and Artificial

Neural Network. In this research we present TRIZ oriented AI to create knowledgebase.

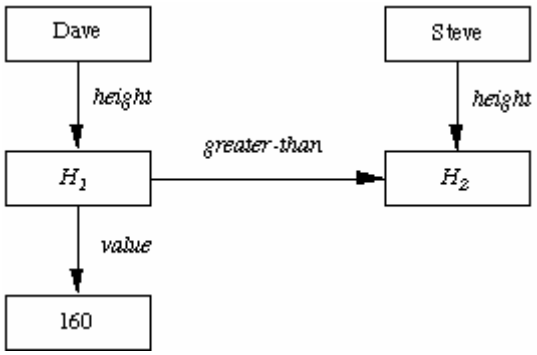
Figure 1 How Intelligence software work.



Knowledgebase Representation

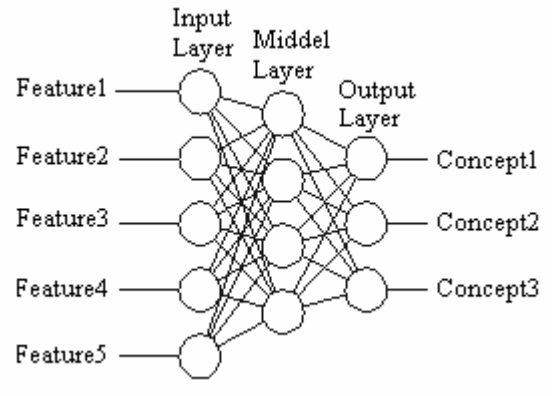
Knowledgebase represented in deferent way. One of the first is Semantic Network. A Semantic Network is a graphic notation for representing knowledge in patterns of interconnected nodes and arcs as presented in figure 2. Computer implementations of Semantic Networks were first development for Artificial Intelligence and machine translation, but earlier versions have long been used in philosophy, psychology, and linguistics.

Figure 2 a Semantic Network example



Artificial Neural Network is a second way to represent knowledgebase. Neural Nets are a widely used technique for learning by changing the weights assigned to the nodes or arcs of a network. Their name, however, is a misnomer, since they bear little resemblance to actual neural mechanisms. Figure 3 shows a typical Neural Net, whose input is a sequence of numbers that indicate the relative proportion of some selected features and whose output is another sequence of numbers that indicate the most likely concept characterized by that combination of features. In an application such as optical character recognition (OCR), the features might represent lines, curves, and angles, and the concepts might represent the letters that have those features.

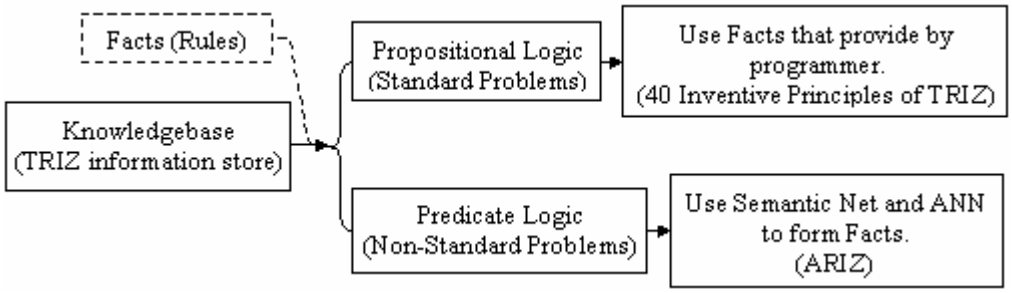
Figure 3 Neural Network



Using TRIZ methodology to represent knowledgebase

In TRIZ methodology we have two types of problems: standard and none standard. Standard problems can be solved with TRIZ principles (40 inventive principles), but none standard problems solved with creativity and intelligence algorithms (In TRIZ methodology it's named "ARIZ"). These algorithms needed when we have new problems. In figure 4 we compare TRIZ methodology with Artificial Intelligence knowledgebase. Facts that used in Artificial Intelligence, always are in standard group and we used them for conclude results. These results can help us to solve our problems. Intelligence software uses Facts for give result. Whatever we have exactly Facts; we can give most perfect result from software. In this section we present some ways to create and store Facts with TRIZ that help software to get more intelligence.

Figure 4 Compare TRIZ methodology with AI knowledgebase



Create and store Facts with using inventive principles of TRIZ

TRIZ is a powerful methodology for represented knowledgebase. In this approach, we can use inventive principles of TRIZ to dynamically create Facts and save them in knowledgebase. In this article, we have two discussions: First, we discuss about how to create Facts, afterward we explain the structure for maintain Facts that define with TRIZ principles.

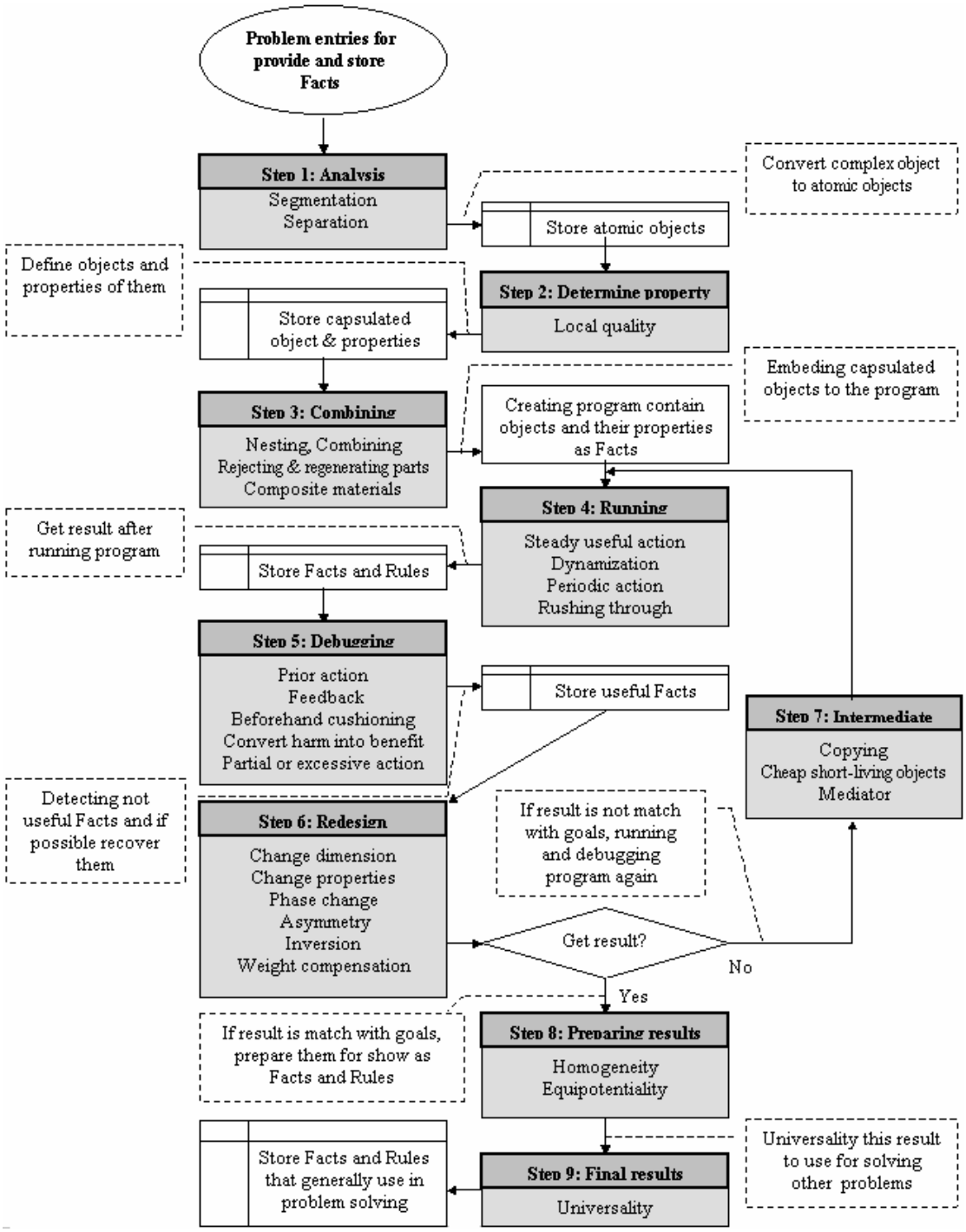
How to create Facts?

The nine major preparatory steps for the basic version of TRIZ Oriented Artificial Intelligence are:

1. Analysis
2. Determine property
3. Combining
4. Running

5. Debugging
6. Redesign
7. Intermediate code
8. Preparing results
9. Final results

Figure 5 Use TRIZ for represented knowledgebase



1. Analysis

In the software engineering, first step for develop is design and analysis and the main approach in design is separated problem to the entities. (Atomic objects that we want

to get information about them.) TRIZ methodology presents two principles to form it shown in figure 5:

- Segmentation
- Separation

In this step we separate problems to the atomic objects, store each separated objects and prepare them for the next step.

2. Determine property

This step defines the properties for each object. Every object has special properties that define by Local quality in TRIZ principles. We can say "Local quality in software means define private properties of each object that utilized with their functions". We say local because objects or entities have functions that work with their parameters and the other objects cannot access to them.

In this step we capsulated every objects with their properties and store them in the storage media.

3. Combining

Every provided object in the previous steps must be joining together to work as a collective program because we want to understand and detect relationship between objects. It is applied with four TRIZ principles that contain of:

- Nesting
- Combining
- Rejecting & regenerating parts
- Composite materials

At the final, we have a program include objects and their properties that we want to use them as Facts in knowledgebase.

4. Running

After prepare program and determine relationship between capsulated objects, it is ready to run. In this section we want to now is the program working properly or not? This process that applied with four TRIZ principles that contain of:

- Steady useful action
- Dynamization
- Periodic action
- Rushing through

5. Debugging

When the program running, it maybe causes an error or gives not useful results. This step manages it and filters the result and finally gives benefit result as useful Facts to store. We use five TRIZ principles in this step that contain of:

- Prior action
- Feedback (for get results and compare them with final goals)
- Beforehand cushioning
- Convert harm into benefit
- Partial or excessive action

6. Redesign

If results taken in step 5 are not match with goals, we have to redesign program, objects and relations. TRIZ methodology explains below principles to change them:

- Change dimension
- Change properties
- Phase change
- Asymmetry
- Inversion
- Weight compensation

7. Intermediate code

Before this step, we check results for matching with our goals and if not, we create a template copy of the program for do step 4 again. It is do with these TRIZ principles:

- Copying
- Cheap short-living objects
- Mediator

8. Preparing results

Now we can prepare Facts that provided with above steps. Two TRIZ principles help us to do it:

- Homogeneity
- Equipotentiality

9. Final results

In the final step, we can universality taken results for use in other or related problems. We use Universality in TRIZ methodology to do it. Results finally store as general Facts and Rules to use in Artificial Intelligence. (Shown in figure 5)

How to maintain and store Facts?

We can use Semantic Network or ANN to form this diagram (figure 5). Every step can be store as a Neural in ANN or a Node in Semantic Network. Every Neural or Node has a storage media (buffer) to keep produced objects in each step. Saving the Facts as a separated objects with their related processes (Inventive principles of TRIZ) is a one of the basically difference between TRIZ Oriented and the other ways to manipulate knowledgebase.

Conclusion

Over the past six decades, AI research has mostly been focusing on solving specific problems. In TRIZ oriented AI, we use some of 40 inventive principles to solve problems. For example one of the most important problems in Intelligence programming is a presenting human emotion. We have exiting ways to represent it, but all of them are very theoretical and not yet practical use in the software. TRIZ principles can be simulated it easily because it is very closely with human behavior.

We can use TRIZ for present knowledgebase and create Facts to store in knowledgebase and give a more intelligence software.

Endnotes

1. The Innovation Algorithm. Dr. Genrich Altshuller, Technical Innovation Center, Inc. July 2000
2. Artificial Neural Networks for creativity complex problem solving. M. Akbari, Iranian journal of TRIZ. Summer 2004 ISSN 1735-1464.
3. Artificial Neural Networks and creative softwares. M. Akbari, Iranian journal of TRIZ. Winter 2004 ISSN 1735-1464.