Chapter 4

Creativity Support System

4.1 Introductory remarks

The author has been investigating computer-aided creativity support through construction of creativity support system called *Chie-no-Izumi*[1][39]. In *Chie-no-Izumi*, a process in which a human gets an idea is modeled using analogical reasoning. The author attempts to give a computer the ability of creative thinking by implementing the model of the process[59, 62]. A user will benefit from interaction with *Chie-no-Izumi*, because the user's own creativity will be enhanced as if she/he is interacting with a creative person.

The computer-aided creativity support can be classified into three levels[75].

1. A computer takes charge of non-creative tasks and let people to concentrate on creative tasks. The sharing indirectly enhances human creativity.

2. A computer implements (a part of) a process of a method of creative thinking, such as KJ method[34].

3. A computer gives (a prototype of) an idea.

Chie-no-Izumi is one of few examples which aim for the third level.

A study of creativity support by the author started by examining a model of creative thinking. The author defined "Paraphrasing-based Analogical reasoning", **PAR** for short, which is a framework of analogical reasoning that aims to model the creative thinking, as a result[63]. The author also builded a prototype system in order to test the effect of the creativity support by *Chie-no-Izumi*[65] with real-world data.

[1] *Chie-no-Izumi* is a Japanese phrase meaning "a fountain of wisdom". *Chie-no-Izumi*'s Japanese orthography, 知恵の泉®, is a registered trade mark of Toshiba Corporation.

In this chapter, first, the model of the creative thinking and the prototype system are reviewed, then actual creativity support sessions with *Chie-no-Izumi* are described.

4.2 Model of creative thinking in "Chie-no-Izumi"

4.2.1 Modeling creative thinking

In *Chie-no-Izumi* a process of creative thinking is understood in context of machine learning[63]. Suppose there is an *ideal* knowledge base ι and an agent[2] who has a subset γ of ι. Intuitively, creative thinking is an action that is made by the agent to make their own γ similar to ι.

The process can be characterized by an inference engine which is used to bring γ near to ι. That coincides with the fact that four patterns of creative thinking — analogical reasoning, induction, universalization and extremization — are distinguished in studies of creativity engineering[29]. Researchers usually agree that analogical reasoning is the most important among the four. Analogical reasoning is a kind of reasoning that assumes several *domains* in a knowledge base and creates new knowledge by transferring knowledge from a *base* domain, which is relatively rich in knowledge, to a *target* domain, which is a currently focused part of the knowledge base.

4.2.2 Paraphrasing-based analogical reasoning

Bottom-up analogical reasoning

Analogical reasoning consists of the following four steps[24].

1. Create internal representations of possible bases and a target.
2. Select a base as an adequate analog of the target.
3. Identify correspondences between the base and the target.
4. Augment the mapping in order to produce knowledge in the target.

There has been a long history of study of analogical reasoning[17, 19, 20]. However most studies assumed that the first step above has been finished and dealt with the second through the fourth only. The first step can be further divided into the following three substeps.

[2] A person or a computer.

a. Describe knowledge in the internal representation. (creation of a knowledge base)

b. Establish domains in the knowledge base.

c. Identify a target domain. The rest become possible base domains.

When analogical reasoning is used in problem solving, there is a *natural* way to partition the knowledge base[3], that is, the **b** above is obvious. So is **c**, because a problem to be solved is given. Furthermore **a** is a rather common problem among artificial intelligence studies, not special to analogical reasoning. Therefore the fact that most analogical reasoning studies neglected the first step is understandable.

On the other hand, creative thinking is to proceed from "the surprising fact" to theories that might explain the fact[18]. However, there are no *surprising facts* in invention nor planning but a mere vague *preferred to-be state, which could be surprising upon observation.* In problem solving, a given problem acts as a goal of reasoning. Unfortunately, the *to-be state* is often abstract and useless to guide the reasoning, because the real problem is usually unknown in the scene where the creativity support is required.

Therefore, analogical reasoning in creative thinking should not be modeled in a top-down framework like problem solving. Unlike past studies of analogical reasoning, *Chie-no-Izumi* does not assume a preset partition in a knowledge base. That is, it does not neglect the first step of analogical reasoning.

Object of analogical reasoning

Chie-no-Izumi deals with creative thinking that creates a new concept by analogical reasoning on a hierarchy of concepts of how a concept is defined using other concepts. Therefore an analogical reasoning mechanism, its objects being definitions of concepts [4], is required to model it.

PAR models analogy-based creative thinking

PAR is defined as the analogical reasoning mechanism that satisfies the above requirements. **PAR** was discussed in depth in chapter 3. Let us review **PAR** briefly.

[3]Partitioning based on problems that can be solved by knowledge in the domain.
[4]Called *paraphrasings*.

- Each of **PAR**'s object is a p's paraphrasing, i.e., knowledge in the form: $\forall x_1, \ldots, x_n(p(x_1, \ldots, x_n) \Leftrightarrow L_1 \wedge \ldots \wedge L_m)$. The righthand side is called a *body* of paraphrasing.
- Constants, functions and predicates are all called *tokens*. The tokens are classified into *internal* tokens and *external* tokens.
- A set of paraphrasings are given as a knowledge base. It is then so partitioned that a token appearing in two or more domains becomes external.
- A set of equivalence classes of tokens is called a *T-equivalence*. We costruct a sequence of T-equivalences as follows. Initially the T-equivalence makes each external token equivalent to itself. The T-equivalence is augmented to include such pairs of tokens that make two paraphrasing *analogous*, which concept is below. Two paraphrasings are analogous whenever they contain at least a pair of tokens equivalent to each other under the T-equivalence, and they will become identical if the one or more pairs of tokens not appearing in it are treated be equivalent.
- If a domain has a paraphrasing which is analogous to a paraphrasing in the other domain, then the two domains are analogous.
- Suppose a body of paraphrasing in a domain S is equivalent to a formula in the other domain O but a head (the lefthand side) of the paraphrasing is not, under T-equivalence. **PAR** creates a new paraphrasing in O by getting an atom corresponding to the head from a user and augments O and the T-equivalence.

4.3 Prototype of "Chie-no-Izumi"

4.3.1 Designing the prototype

The author implemented **PAR** system, a system performs **PAR** getting a set of paraphrasings and a set of external tokens as inputs, on a workstation. However, **PAR** system was difficult to be used as a creativity support tool for the following reasons.

1. Input of knowledge is difficult.
2. Partitioning is not fully automated.
3. Process of analogical reasoning is difficult to comprehend.

To cope with the problems, the author designs a prototype in the following policies.

1. Input of knowledge should be able to be done in natural language (Japanese). Because this is not a study of natural language processing, a simple but usable method will be preferred.

2. Partitioning should be automated. That can be implemented by creating sets of external tokens, applying heuristics to the given knowledge base.

3. Process of analogical reasoning should be displayed through a graphical user interface.

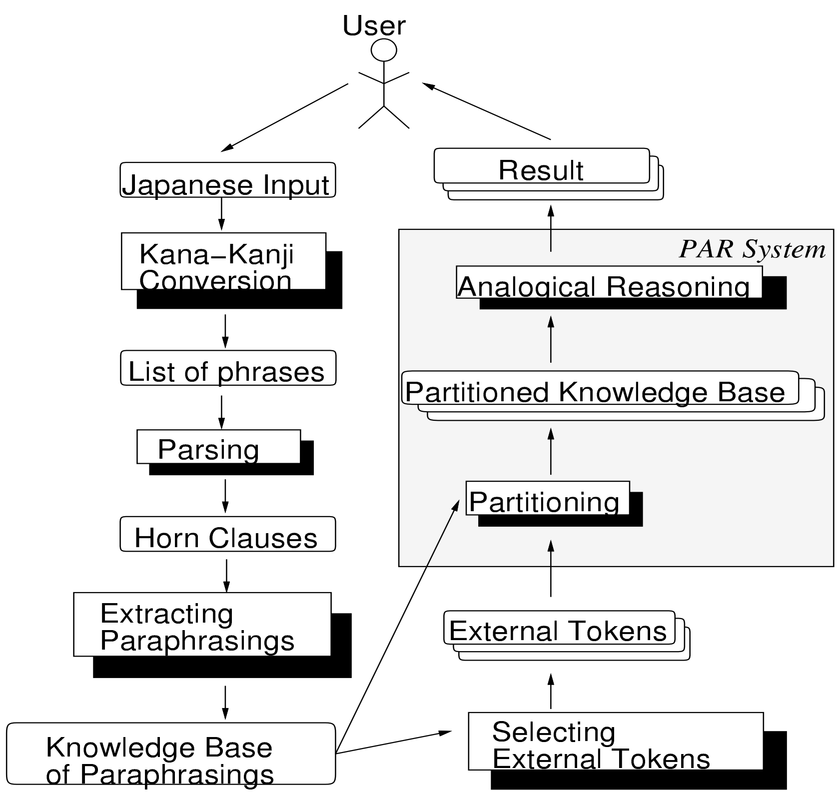

Figure 4.1: Flow of the prototype system

Figure 4.1 shows a flow of the prototype of *Chie-no-Izumi*. Implementation of each part is explained in the following subsections.

4.3.2 Input system of the prototype

An input interface for the prototype does not require semantic analysis, because **PAR** does not employ semantic information to detect analogy between domains. It is enough to parse an input sentence to produce a representation in Horn clause. Inoue's method does that.

Inoue's method

Inoue's method[2, 27] gets a Japanese sentence that is broken into a list of phrases as input and produces a corresponding Horn clause as output. Rules that produce three form of Horn clauses — facts, rules and goals — make the translation possible without dictionary, because Japanese language has the following characteristics.

- A subject usually comes first, then an object, followed by a predicate.
- A postpositional particle is suffixed to a subject and an object to indicate a case of the word.

The method will have trouble when two semantically equal atoms cannot be unified because their arranges of arguments are different. The original Inoue's method deals with the problem by dynamically adding arguments and changing places of arguments during reasoning. However it also makes a reasoning system depend on the input system. Because the author would like to use the existing **PAR** system for the prototype with as little modification as possible, the solution employed by the original method is unacceptable. As the other solution, the author normalize atoms in the following form, and extends a mechanism of unification.

```
Predicate(Argument-list, Postpositional-particle-list)
```

Furthermore, some shortcomings of the original method, such as inability to deal with a functional structure (「∼-no-…」), a parallel structure (「∼-to-…」) and a substantive modifier, are fixed. To solve the problem with the substantive modifier, a virtual postpositional particle "substantive-modifier" is introduced.

The improved method is called *jparse* system.

Lexical analysis

Unlike languages such as English, where a sentence is written in a manner that adjacent words are separated by a space, Japanese poses a difficulty for lexical analysis. However getting a word out of a phrase is relatively easy because there are not many adjunct words in Japanese.

Let us see this from a different perspective. To let a computer process Japanese language, we have to input the Japanese to the computer anyway. Today the most common tool for that purpose is a multi-phrase kana-kanji conversion system. Using the conversion system, a user is aware of phrases. Therefore lexical analysis can be easily done by saving information of the phrases during the input.

The idea can be implemented by making use of *nemacs* editor, *Wnn* kana-kanji conversion system and *egg* interface between them.

Determining paraphrasings

Jparse system recognizes a paraphrasing as a fact with a predicate such as 「～de-aru」. Therefore a system which extracts a corresponding paraphrasing from an output of *jparse* is required. That is *paraph* system.

There are the following six types of facts which is understood as paraphrasings. Here *upper-noun* denotes a paraphrased concept; namely, a predicate of head of paraphrasing. *Type* denotes an upper-noun's super concept in the *is-a* hierarchy of concepts. *Lower-noun* is a concept used to define the upper-noun with the type.

1. verb + type + de-aru, da or desu([upper-noun, .., lower-noun, ..],[ha, ga or toha, .., wo, ..])

2. type + de-aru, da or desu([upper-noun, .., lower-noun, .., verb], [ha, ga or toha, .., wo, .., substantive-modifier])

3. upper-noun + de-aru, da or desu([lower-noun, .., verb + type],[wo, .., ha, ga or toha])

4. upper-noun + de-aru, da or desu([lower-noun, .., verb, type],[wo, .., substantive-modifier, ha, ga or toha])

5. upper-noun + to-iu or to-yobu([lower-noun, .., verb + type],[wo, .., wo])

6. upper-noun + to-iu or to-yobu([lower-noun, .., verb, type],[wo, .., substantive-modifier, wo])

These facts are converted into the following paraphrasing.

$$\text{upper-noun}([X],[izIsa]) \Leftrightarrow \text{type}([X],[izIsa]),$$
$$\text{lower-noun}([izSkolem(X)],[izIsa]),\ldots,$$
$$\text{verb}([izSkolem(X),\ldots,\text{other arguments},X],$$
$$[izBefore,\ldots,\text{postpositional particles for the other arguments},izAfter]).$$

In the above formula, ... shows modifiers, *izSkolem* shows a Skolem-like function that gives a corresponding lower-noun for the upper noun, *izIsa* shows a virtual postpositional particle corresponding to the *is-a* relation and *izBefore* and *izAfter* represent virtual particles for the lower-noun and the upper-noun respectively.

Example 4.3.1 *Facts which are recognized as paraphrasings.*
nita-ryori-de-aru([stew, niku, nabe], [ha, wo, de])
stew-de-aru([niku, nabe, nita-ryori], [wo, de, ha])

When it recognizes a paraphrasing, *paraph* records a parent-child relation of the type and the upper-noun in the *is-a* concept hierarchy. There are two built-in types — *koto* and *mono*. If a super concept of the concept being recorded is unknown, it is assumed to be a child of *mono*.

If a functional form $b(a)$ appears in arguments, *paraph* expands it into 「a-no-b」 and create the following paraphrasing.

$$a\text{-no-}b([X],[izIsa]) \Leftrightarrow a([izSkolem(X)],[izIsa]),$$
$$b([izSkolem(X),X],[izBefore,izAfter]).$$

4.3.3 Automatic selection of external tokens

It is a distinctive feature of **PAR** that it is possible to draw various results of analogical reasoning from a single set of paraphrasings by giving various sets of external tokens. However the feature has not been fully utilized because of lack of guideline on what kind of external tokens should be used for a particular purpose.

An external token in **PAR** is defined as a token which may appear in two or more domains. An external token may initially exist in a single domain and later appear in the other domain as a result of analogical reasoning. Therefore a set of external tokens that gives a particular partition is not unique. Because an external token can be unconditionally mapped to a domain where it does not exist, the more external tokens, the more analogical reasoning can be carried out and therefore the more preferred. The following mechanism of automatic selection of external tokens is developed based on the argument above.

Step 1. Select a tentative set of external tokens by means of heuristics. The set is called *seeds* of external tokens. Partition the knowledge base based on the seeds.

Step 2. Select a maximal set of external tokens that gives the partition. This can be done by adding internal tokens one by one to the seeds and seeing if the same partition is made.

If a paraphrasing share at most one token with the other paraphrasings, the following theorem stands. Therefore the maximum set of external tokens that gives a particular canonical partition can be obtained through this method. The assumption is reasonable when each paraphrasing is relatively simple and a knowledge base is consist of a lot of tokens.

Theorem 4.3.2 *A family of sets of external tokens that give a particular canonical partition (the finest possible partition)[63] forms a lattice with respect to set inclusion relation.*

Proof Because a family of sets is partial order with respect to set inclusion relation, showing existence of the greatest element and the least element is enough to prove the theorem.

Let us show the existence of the greatest element. Suppose a set E of external tokens gives a canonical partition D. The canonical partition is unique according to theorem 3.4.2. For $e \notin E$, a canonical partition given by $E \cup e$ is different from D only if e is a shared token by two or more paraphrasings in a domain in D. For tokens $e_1, e_2 \notin E$, if both $E \cup e_1$ and $E \cup e_2$ give the canonical partition D, then $E \cup e_1 \cup e_2$ also gives D, because a paraphrasing does not share both e_1 and e_2 with the other paraphrasings by the assumption. Thus it is shown that there is the greatest element of the family of sets of external tokens which give the same canonical partition.

The existence of the least element can be shown by the dual argument.

■

The heuristics consist of the following three rules.

- **H1.** *Verbs, substantive modifiers and words meaning tools are external.* We assume that a predicate of the last atom of body of paraphrasing is the verb, a word corresponding to a postpositional particle *de* is the word meaning a tool and a word corresponding to a virtual postpositional particle *substantive-modifier* is the substantive modifier.

- **H2.** *The lowests of izBefore/izAfter hierarchy are external.*

- **H3.** *User-specified subtrees of the* is-a *hierarchy forms a set of internal tokens.*

The following three sets of seeds are used to partition a knowledge base.

- **Ext1.** Composite of H1 and H3
- **Ext2.** Composite of H2 and H3
- **Ext3.** Composite of H1, H2 and H3

Seeds from H3 is always used because they reflect user's knowledge.

Example 4.3.3 *Suppose that the following three sentences are input.*

1. `Steak-toha-niku-no-slice-wo-frying-pan-de-yaita-ryouri-de-aru`
2. `Bouillabaisse-toha-sakana-no-kakugiri-wo-nabe-de-yudeta-ryori-de-aru`
3. `Meunière-toha-sakana-no-slice-wo-frying-pan-de-yaita-ryouri-de-aru`

H1, H2 and H3 give the following seeds, provided that a user specified "bouillabaisse", "steak" and "meunière" as tokens which characterize domains in a hierarchy shown in figure 4.2.

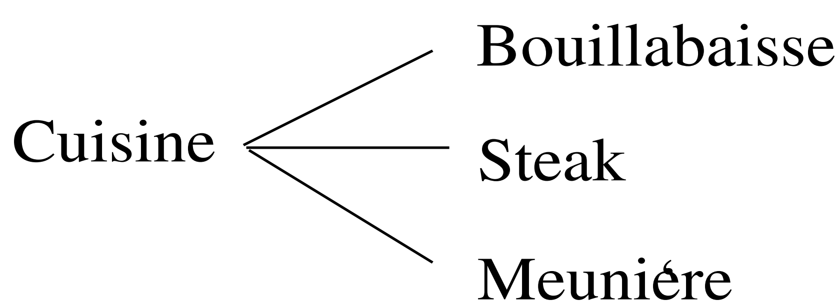

Figure 4.2: Concept hierarchy

(H1) `[kakugiri, slice, yudeta, yaku, frying-pan, nabe]`[5]

(H2) `[niku, sakana]`

[5] *Yaita* is converted to *yaku*.

(H3) `[ryori]`

Each heuristic provides the following partitions.
(Ext1): { *(1)* }, { *(2) (3)* }
(Ext2): { *(1) (2)* }, { *(3)* }
(Ext3): { *(1)* }, { *(2)* }, { *(3)* }

For Ext2, the set of external tokens are expanded in Step2 and the following set is obtained.

`[nabe, yudeta, frying-pan, yaku, sakana, niku, ryori]`

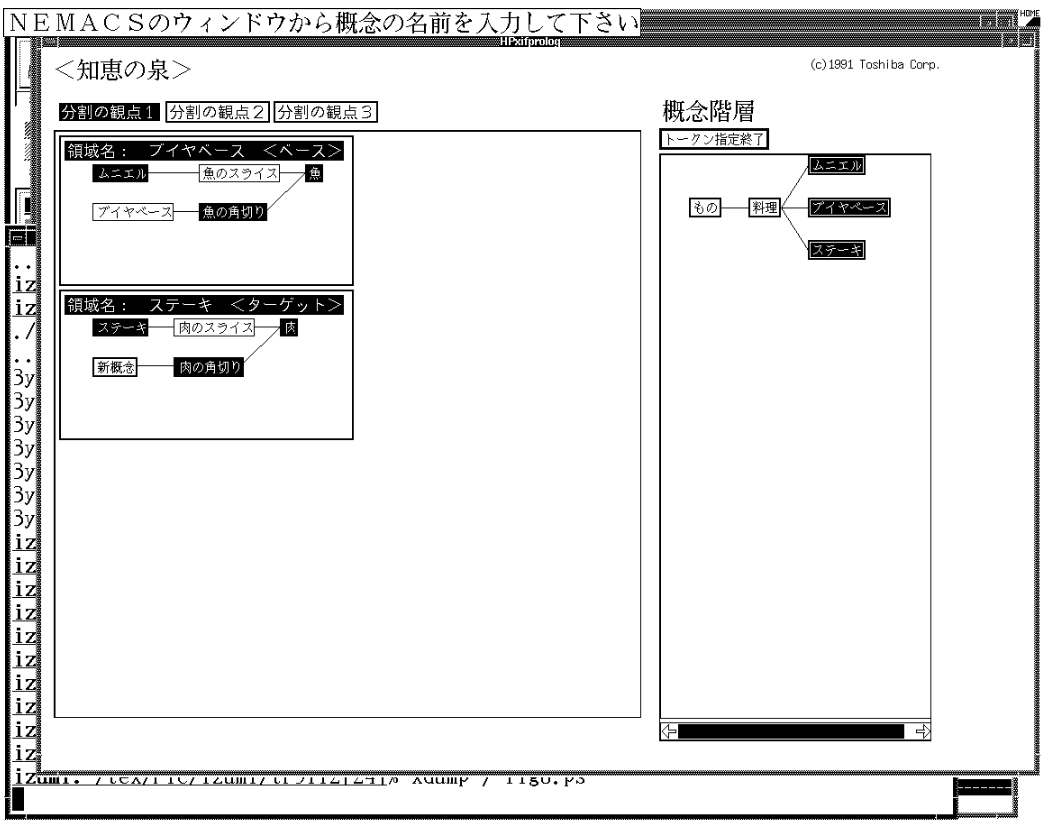

Figure 4.3: Reasoning “stew” analogically

4.3.4 Designing output system

Displaying is-a hierarchy of concepts

In order to let a user specify tokens that characterize a domain for heuristic H3, the system displays a hierarchy of concepts in a window and allow the user to click concepts to select. Righthand side windows in figure 4.3 and figure 4.4 show the hierarchy.

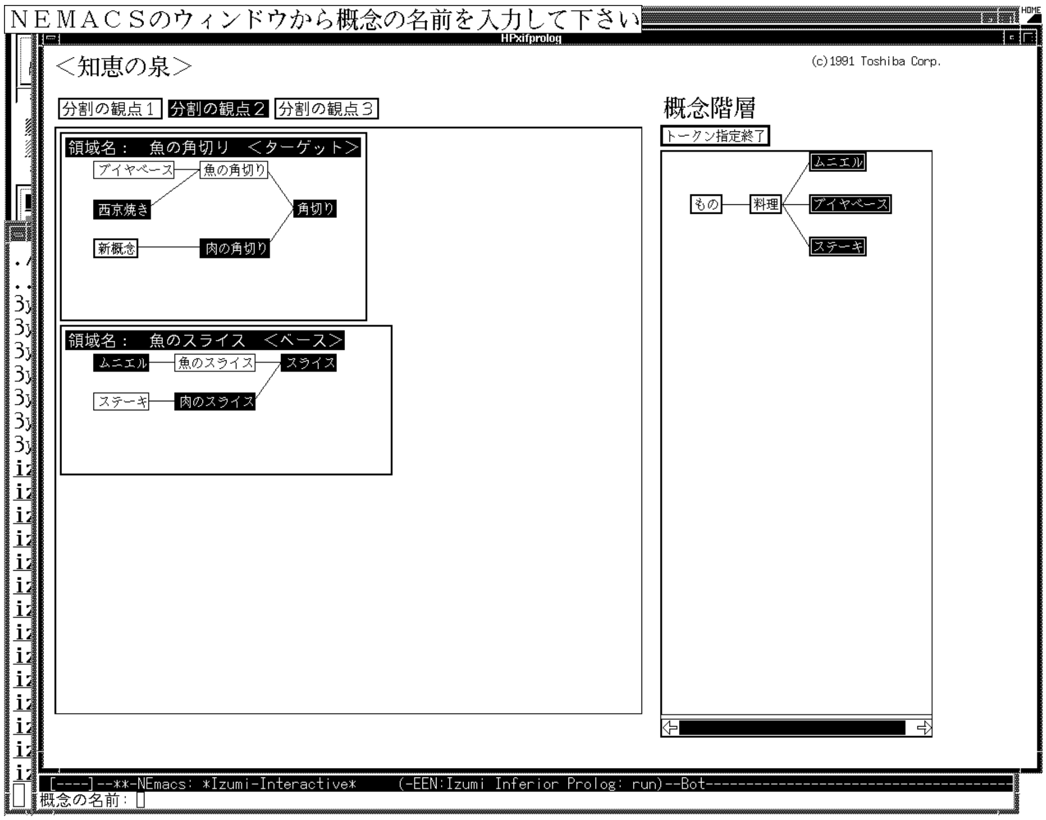

Figure 4.4: Reasoning “dice steak” analogically

The user clicks on a label of token that seems to characterize a domain to invert the appearance. In figure 4.3 and figure 4.4, “meunière”, “bouill-abaisse” and “steak” are selected. This is also a default status, defined as a status where all the leaves and nodes whose descendants have no siblings are selected.

Displaying domains

Each domain is represented by a graph of internal tokens, which means that a lefthand side token is paraphrased by righthand side tokens; namely, it is a hierarchy of how a concept is paraphrased by the other concepts.

Displaying ongoing analogical reasoning process

In **PAR** system it was difficult to understand how analogical reasoning proceeds with a large knowledge base. In the prototype, inter-domain corresponding tokens are displayed in the same color to facilitate the understanding. When a new concept is analogically reasoned, a virtual concept named "new-concept" is introduced and it and its base concepts are blinked. An audio device is also used to create sound of a gong to get user's attention.

Figure 4.3 shows that the prototype reasons a new cuisine (stewed meat block) analogically from bouillabaisse in partitioning based on Ext1 and prompts a user to name it. Although it is difficult to see it in the black and white figure, "niku-no-kakugiri" and "sakana-no-kakugiri" are displayed in the same color which indicates that they are corresponding each other. On the other hand figure 4.4 shows that the prototype has reasoned a cuisine (fried fish block, "saikyo-yaki" in the figure) from meunière in partitioning based on Ext2 then reasons another cuisine (fried meat block) from steak.

4.4 Creativity support by "Chie-no-Izumi"

So far the author has explained *Chie-no-Izumi*'s creativity model and behavior of the prototype using simple examples. In this section an example of actual creativity support session is shown.

4.4.1 "Chie-no-Izumi" for industrial use

There are two ways to use *Chie-no-Izumi* as a creativity support tool.

1. Wide-range knowledge is given beforehand. A user inputs familiar knowledge to her/him and let *Chie-no-Izumi* partition and reason. In this way, although a novel idea by analogical reasoning from a quite different domain is expected, the number of output ideas will be small. Because there is small possibility for far apart domains to share external tokens and without shared external tokens there are no **PAR** by definition. This problem can be avoided by giving a thesaurus as a theory of equivalence among external tokens.

2. Knowledge of specific area is given and a user let *Chie-no-Izumi* partition and reason. In this way, the area is partitioned into sub-areas and analogy will be made between the sub-areas. Preparation of the thesaurus is relatively easy and a lot of ideas are expected to be presented. However, truely novel ideas cannot be expected because this

is mere intra-domain analogical reasoning. In this way, either just a bottom-up analogical reasoning based on the knowledge, or analogical reasoning with more knowledge from user, are possible.

Because of ease of preparation, the author has decided to carry out experiments in a manner of 2. As the *specific area*, multimedia system is chosen, because a machine-readable surveillant database[48] of the area is available. In the experiment, *Chie-no-Izumi* will support a process of product planning for new multimedia systems.

The author would like to remark that he has chosen the method 2 not because he thought the 1 is impossible. Recently large scale machine-readable thesaurus has become available via CD-ROM. Experimenting in the manner of 1 should be included in future works.

Knowledge input

The database shown above has 871 records, which include articles that are not directly related to multimedia systems, such as standardization activity and investment by companies. Those records are omitted. Because *jparse* is not applicable to already machine-readable text, the text is divided into a list of phrase by hand. At the same time, poly-sentences and multi-sentences, which cannot be analyzed by *jparse*, are converted to equivalent sets of simple sentences.

In the database, several different words are used to describe the same meaning. Thus a thesaurus is required and prepared by hand.

Size of the experiments

- Hardware: Toshiba AS4075 (Sun SparcStation 2)
- Number of input DB records: 117
- Number of input sentences: 439
- Number of paraphrasings: 565
  441 from input sentences, 124 are automatically generated for functional structures
- Size of the thesaurus: 106 pairs of equivalent words
- Number of tokens: 1211
- Number of external tokens: 823 by Ext1, 464 by Ext2 and 1027 by Ext3
- Partitioning:
  - Token selection in *is-a* hierarchy: default
  - Time spent: about three hours

  - Number of domains:
    128 by Ext1, 74 by Ext2 and 162 by Ext3

- Number of new ideas got in 20 hours from beginning of analogical reasoning:
  22 by Ext1 and 9 by Ext2[6]

Examples of results of analogical reasoning

Three interesting new concepts, all got in partitioning by Ext1, are shown.

- *Multimedia Company Directory*: A base domain is related to *manufacture line recording software*, which facilitate analysis of a manufacture line and make improvement easy. A target domain is related to *The American Business Disk*, which includes addresses of U.S. companies, and *The-Rite-of-Spring*, which is a CD-ROM software includes Stravinsky's famous ballet and commentary. The manufacture line recording software has a function that allow a user jump from a scene to the other scene freely. *Chie-no-Izumi* creates a new concept *in a scene of The American Business Disk*, by analogical reasoning from *in a scene of the manufacture line recording software*, suggesting that give The American Business Disk images and function that jump among them. This concepts should have a name like *Multimedia Company Directory.*

- *Company Analyzer*: Continuing with the same base/target pair, *Chie-no-Izumi* proposed a system that can analyze companies using *Multimedia Company Directory*, by analogical reasoning from *manufacture line analyzer* that can analyze the manufacture line using the function of jumping among the images. This concepts should have a name like *Company Analyzer.*

- *VR hyper text*: A base domain is related to several virtual reality systems such as an aircraft design system and a battle simulator. A target domain is related to *Link Hypertext System*, which is suitable for prototyping. The VR-aircraft design system has a function that allows a user get on the plane in virtual space. *Chie-no-Izumi* recognizes correspondence between *virtual space plane* and *Link Hypertext System* by seeing similarity between *in the virtual space plane* and *in Link Hypertext System*, to propose a function to enter the inside of *Link Hypertext System.* This concepts should have a name like *VR hyper text.*

[6] The experiments is not complete for Ext3.

4.4.2 Evaluation of experiments

In this subsection the author qualitatively analyzes on *Chie-no-Izumi*'s effectiveness on creativity support. Experiments using plural subjects would be required for quantitative evaluation in order to evaluate the effectiveness independent of user's personality, and should be included in future works.

Appropriateness of the model of creativity support

Because *Chie-no-Izumi* does not employ semantic information to find similarity between objects, concepts proposed by *Chie-no-Izumi* are relatively superficial in a sense that they are new combinations of known words. However, as shown in above examples, those new combination can stimulate a user and facilitate rich images to emerge in the user's mind. This proves that *Chie-no-Izumi*'s creativity support model, where it enhances user's creativity by providing effect similar to talking with the other people, was correct.

In this model, usefulness of results of analogical reasoning is not important, but how it can stimulate user's creativity is important. The results are often used as examples of how one can bring knowledge from a base domain to a target domain. For example, in a process of proposing *Multimedia Company Directory*, *Chie-no-Izumi* simply suggested that putting images to *The American Business Disk*. However, the user has background knowledge on the base *manufacture line recording software*[7] and that can also be mapped into the target. This makes the idea *Multimedia Company Directory* possible.

The process is shown in figure 4.5. *Chie-no-Izumi* performs analogical reasoning on given, relatively small amount of knowledge (s and t in the figure). A user, who has huge background knowledge that includes the knowledge given to *Chie-no-Izumi*, performs analogical reasoning on the background knowledge to get an idea, using analogical reasoning by *Chie-no-Izumi* as a hint. Even if the user did not know the knowledge given to *Chie-no-Izumi* beforehand, like the experiment above, the user comes to know it through interaction with *Chie-no-Izumi*. Thus a framework shown in figure 4.5 can be applied.

[7] Such as "it is a software which records various aspects of a manufacture line".

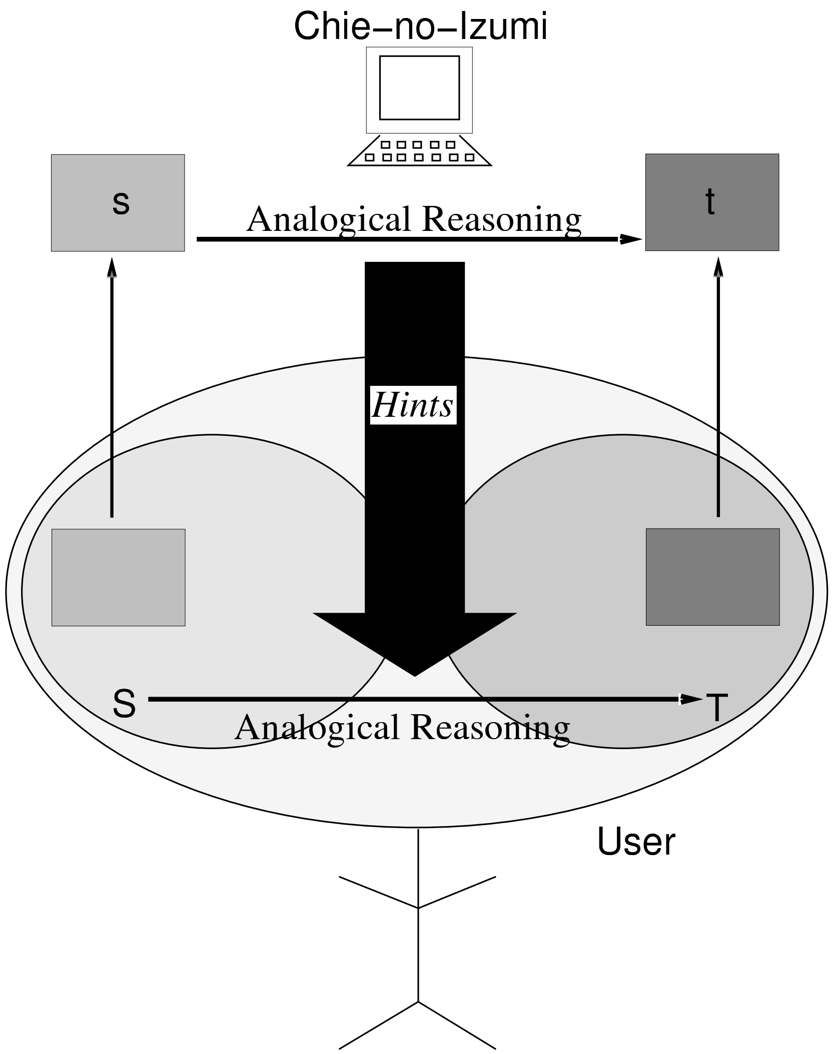

Figure 4.5: Creativity support scheme by *Chie-no-Izumi*

Therefore if a base domain and a target domain were easily understandable it would be enough to present a pair of similar domains, to support human creativity. This is intuitively confirmed when a function to display the base/target domain separately from the other domains is implemented.

On the other hand, whole framework in figure 4.5 can be included in an analogical reasoning system. Namely, instead of carrying out analogical reasoning on huge knowledge, it samples some of knowledge and recognizes similarity between domains. Then it is used as a hint to implement analogical reasoning on the original knowledge base. Although any concrete methods have not yet investigated, frameworks of theory abstraction[71, 49] can be helpful.

Time required

Although several interesting concepts have been proposed during the experiments, processing time will be a huge barrier to practical uses of *Chie-no-Izumi*. Now there are two devices are added to free a user from interaction with *Chie-no-Izumi*.

- A new concept analogically reasoned from a concept of the form 「∼-no-⋯」 is automatically named.
- When *Chie-no-Izumi* finds a concept that needs user's help with naming, it gives a temporary name to the concept and send email to the user about the concept.

A bottom-up process, where two domains are picked from the whole set of domains and compared each other, causes the slowness. For example, with Ext1 8128 compares have done in the experiments. If domains that are focused by a user can be distinguished from the rest, a mechanism like agenda in AM[41] can be used to avoid the brute-force search. However, it will be a tough task for the user to assign degrees of focus to a lot of domains, in a sense of labor and understanding of the domains. Furthermore, because *Chie-no-Izumi*'s creativity support power emerges from its ability to find surprising similarity between presumably unrelated two domains, neglecting domains that are not focused by the user may result in loss of the creativity support power.

On the other hand, it is certain that a parallel computer will improve the speed in a dramatic way because each combination of base-target pair can be compared independently each other.

Structures facilitate analogical reasoning

A lot of analogy based on functional structure of the form 「A-no-B」 are observed. For 22 concepts by Ext1, 7 of them are this type. For 9 concepts by Ext2, 4 of them are this type.

There are two cases of this kind of analogical reasoning.

- Similarity between A and A' is given. 「A'-no-B」, i.e., a new function of A' is reasoned analogically from 「A-no-B」. *Multimedia Company Directory* is an example of this case.

- 「A-no-B」 and 「A'-no-B」 are given. Similarity between A and A' is reasoned analogically for having the same function. *VR hyper text* is an example of this case.

4.5 Summary

A mechanism of analogical reasoning **PAR**, specially designed to model a process of creative thinking through analogical reasoning, is defined. A prototype system of creativity support tool *Chie-no-Izumi* are built by adding the Japanese input interface, the external token selection mechanism and the window interface to **PAR** system.

Experiments in which the prototype is used to assist product planning process of multimedia systems are carried out, resulting in some interesting ideas. Therefore effectiveness of *Chie-no-Izumi* when it is used in a limited area is experimentally shown. Although it is not shown in the thesis, *Chie-no-Izumi* has also been used to improve quality of patent documents and the result has been fine.

Recently studies on analogical reasoning become active and a number of problem solving techniques using analogical reasoning are proposed. However they are not so reliable that they can be applied to real world problems because *correct* analogical reasoning is not precisely defined. On the other hand, analogical reasoning can be used to assist human's planning work. The human may pick up or throw away ideas presented by the analogical reasoning system. In this approach even the current techniques of analogical reasoning are safe enough.

Other than items stated in the last section, future work should include development of advanced ways of partitioning. *Chie-no-Izumi* enhances human creativity because it finds surprising similarity between presumably unrelated two domains. The ability is heavily depends on partitioning. Although the selection of external tokens by heuristics, the method employed by the prototype, is efficient, a more flexible method of partitioning is required to enable advanced applications.