Knowledge Acquisition For Production System Design By Mapping Distinct Conceptual Knowledge Pools
Hidehiko Yamamoto
Dept. of Opto-Mechatronics
Faculty of Systems Engineering
Wakayama University
930, Sakaedani, Wakayama-shi, 640-8510, Japan
Tel and Fax: +81-734-57-8158
tiger@sys.wakayama-u.ac.jp

Abstract
This study describes a new knowledge acquisition technology for production system design using a computer system. In this study, the automatic judgement of moving on to a knowledge acquisition process is made under the control of hypothetical reasoning. As the knowledge acquisition method after the judgement that all knowledge is evaluated false by hypothetical reasoning, the new knowledge acquisition method which involves mapping part of a knowledge selected from among distinct conceptual spaces is proposed. After applying the method to an expert system that assists production system design, the new frame-type knowledge for a machine tool is acquired.

1. Introduction
Conventional computer systems that assist production system design search the design solution by using a given data base and knowledge base. These kinds of solutions are gained by directly using the information described in the data base and knowledge base [1]. This means new information is not automatically added to the data base.
On the contrary, although experienced planners proceed with the design jobs by using past data and their experience, they encounter cases that don’t directly use either their past data or experience. In such situations, they create new data by themselves and produce creative design solutions. It is necessary for a computer system assisting production system design to have an analogy for the experienced planners’ intelligent creations.

This paper describes a method to acquire new knowledge that is not directly described in the knowledge data base. It also gives the application examples for acquiring new machine tools’ knowledge.

2. Knowledge Acquisition Based On Distinct Conceptual Knowledge

When experienced production system engineers plan new production lines, the following two processes for knowledge usage are carried out.

[Process1] Planners call to mind the planning experiences that they have had and extract the knowledge perfectly conforming to the present design.

[Process2] When planners don’t have exactly conforming knowledge, which indicates a situation where the existing knowledge is insufficient, they create new knowledge by changing the existing knowledge and then apply the new knowledge to the current design.

[Process1] is the process which directly reflects expert planners’ experience and is regarded as the process to verify their memorized knowledge. They also have another situation where they don’t directly use the memorized knowledge and they are required to change their memorized knowledge in certain ways. This situation corresponds to [Process2]. The knowledge created in [Process2] means new solutions for the expert planners themselves and [Process2] is one method of new knowledge acquisition.

The first way to acquire new knowledge that expert planners take in [Process2] is to imitate relevant past knowledge and facts. However, the following problem occurs in selecting the object knowledge for imitating.

As a general expression for the searching process of solutions in the production system design process, the process to prove P (a certain design problem) by using the knowledge set K is considered, as shown in Figure 1. In this case, if the contents of knowledge set K are already known, the process to search for the knowledge element \( k_1 \), which can prove P, of K corresponds to [Process1].
Figure 1  Solution search process of production system design

Figure 2  Analogy learning example
When $k_i$ cannot be found, the knowledge acquisition process [Process2] starts. Let us consider one imitating way which was recalled in [Process2] to create another knowledge resembling a certain knowledge. The way can be made by analogy learning. The analogy learning is one of the learning methods in Artificial Intelligence to find similarity and to infer unknown facts. Many research studies on analogous knowledge acquisition by analogy learning have been done.

But, in the problem dealing with both [Process1] and [Process2], it is impossible to use analogy learning by similarity. This is because, in [Process1], proving $P$ ends in failure and, for all $k_i$, $k_i \not\Rightarrow P$ comes into existence. This means knowledge set $K$ is contradictory to the problem $P$. Even if another knowledge set $K_c$ close to $K$ is created by analogy learning, as shown in Figure 2, and another process to prove $P$ by using $K_c$ is followed, it is likely to encounter the same kinds of contradiction.

In this way, for the reasoning process including the flows from [Process1] to [Process2], it is not unsuitable to select known knowledge as an object knowledge for analogy.
For this problem, if a quite different knowledge set $K_f$ compared with knowledge set $K$ is created, as shown in Figure 3, the possibility of the above mentioned contradiction occurring will decrease.

In other words, when the deductive processes of solution searching to verify the fit of the knowledge base or data base continue one after another, (whose process corresponds to the design reasoning process problem which is needed for [Process1] and [Process2]), and when new knowledge acquisition after the failures of solution searching by each process is needed, one new knowledge acquisition method is to create a new distinct conceptual space which is conceptually distant, that is to consider the new space as a possible new solution searching space and to use the elements of the new solution searching space. In this paper, the method to use that solution searching space by using distinct conceptions whose conceptual distance is very far from is proposed.

Although the design methodology using metric space has already been studied\[5\], from the aspects that knowledge close to the old knowledge is used as well as the problem solving method of analogy learning which considers similarity\[2][3\], those research ideas are quite different from this papers idea to use different and distinct conceptual knowledge. The research for case-based reasoning to check out past case problems and use the found case for the next problem has been done\[6][7][8\]. However, case-based reasoning research is also different from this paper’s idea because case-based reasoning selects a case very similar to past known case problems as a hypothetical case.

### 3. Knowledge Acquisition Under Hypothetical Reasoning Control

#### 3.1 Production System Design And Hypothetical Reasoning

As required functions for computer systems assisting production system design include values’ decisions for many design items, it is necessary to carry out the two processes, [Process1] and [Process2] for each decision item.

Before we consider how to realize the knowledge acquisition process [Process2], it must be decided when the computer system changes from [Process1] to [Process2]. If the change point is not
decided, the system will not be a real system. Chapter 3.1 describes the concept of judging automatic change points by considering that the production system design process is under hypothetical reasoning control.

It is rare that solutions for each design item in a production system design are selected without regard for other design items. This means the solution of a later process design item is selected, depending on the solutions of pre-process design items. For example, if hydraulic control machine tools are decided as pre-process design items, as the next design solutions are based on that decision, the machining conditions of hydraulic machine tools will be searched. If NC machine tools are decided, machining conditions which are quite different from the case of hydraulic control machine tools will be searched.

In other words, in production system design process, decision processes for each design item are not always sequentially linked from top to bottom. The decision processes go up and down based on a certain dependent relation and are expressed as, so called, non-monotonic reasoning process\[9\][10]. In order to systematize such production system design process, hypothetical reasoning\[11\][12] can be used because the reasoning can express non-monotony. Therefore, by regarding solutions selected at each design process as competitive hypotheses in hypothetical reasoning, regarding one of the selected solutions as true and regarding the remaining solutions as false, all processes in production system design can be controlled by hypothetical reasoning.

Although it was mentioned before that decisions change point from [Process1] to [Process2] is important, using hypothetical reasoning makes it possible. When the contradiction occurs that nothing can be verified within the constraints despite verifying all hypotheses in hypothetical reasoning with constraints by dependency directed backtracking, [Process1] is judged as ending in failure. Such a contradiction indicates that using only solution combinations from among the known solution space does not satisfy the constraints and means arriving at an insufficient situation by using known knowledge. This situation coincides with the situation that [Process1] ends in failure. Then, [Process2] which is the new knowledge acquisition process to carry out the design process is needed. By setting the time when the above mentioned contradiction under hypothetical reasoning control occurs as the changing point from [Process1] to [Process2], it is possible for a computer system automatically to control the movement from [Process1] to [Process2].
3.2 Algorithm Of Distinct Conceptual Knowledge Mapping

When [Process2] starts, the competitive hypotheses used in [Process1] have the following character: although each hypothesis among the competitive hypotheses is a set element selected as the solution which satisfies higher positioned constraints, every hypothesis includes a certain amount of contradictory element knowledge for the constraints of the design items examined previously.

As the competitive hypotheses include such a characteristic, it is considered that the hypothesis satisfied with the last constraints can be incorporated if the element knowledge corresponding to the contradiction is revised. In order to do this, the new knowledge acquisition method by mapping distinct conceptual knowledge pools is adopted. The method first creates new element knowledge that is quite different from the element knowledge which contradicts the last constraints and, then, maps the created new element knowledge onto the contradictory element knowledge. The mapping operation is carried out with the following algorithm. Before describing the algorithm, definitions are given.

1) [Definition] “CK : Contradictory Knowledge consists of \( J(1), J(2), \ldots, J(j) \) that are considered as the causes of contradictions happening under hypothetical reasoning.”

For example, consider that a contradiction happens when the required cycle time is not satisfied. The attribute names for the knowledge which causes the contradiction are, for example, feed motor type and feed mechanism. These names correspond to Contradictory Knowledge.

2) [Definition] “\( \alpha_H \) : Mapped Knowledge is a hypothesis selected from among competitive hypotheses and consists of several knowledge elements \( \alpha_H \) \( x \) \( x = 1, 2, 3, \ldots \). Mapped Knowledge is expressed with frame knowledge expression whose slot names, facet names and values correspond to one of the knowledge elements. The knowledge elements \( \alpha_H \) \( x \) is described with variables \( x \) as a list expression, \( \alpha_H = ( \text{slot}[x] \cdot \text{facet}[x] \cdot \text{value}[x] ) \).”

3) [Definition] “A set of \( j \) kinds of elements among the knowledge elements of Mapped Knowledge whose slot names are the same as the attribute names of \( j \) kinds of elements of Contradictory Knowledge CK is defined as Mapped Attribute, \( \alpha_{\text{appliedA}} = ( \alpha_{H(1)}, \alpha_{H(2)}, \ldots, \alpha_{H(j)} ) \).”

4) [Definition] “A set of \( n \) kinds of knowledge, with frame knowledge
expression, corresponding to all kinds of knowledge in the knowledge data base, except for Mapped Knowledge applied $H$, is defined as Base Knowledge $K_n$.

**[Mapping Algorithm]**

**STEP1:** Search for each value of Mapping Attribute, value$[J(1)]$, value$[J(2)]$, $\cdots$, value$[J(j)]$ and express them as $V_{j(j)}$, ($j = 1, 2, \cdots, j$).

**STEP2:** Following from Step2-1 to Step2-4 and search for $U_{n,j(j)}$.

- **Step2-1:** Fix $n=1$ and continue to Step2-2.
- **Step2-2:** Fix $j=1$ and carry out the following rules.
  
  IF : Knowledge elements of Base Knowledge $K_n$ include the knowledge elements of the combination, slot$[J(j)]$ $\cdot$ facet$[J(j)]$.  
  
  THEN : Substitute value$[J(j)]$ of the knowledge elements of slot$[J(j)]$ $\cdot$ facet$[J(j)]$ for $U_{n,j(j)}$.

  ELSE : Substitute NIL for $U_{n,j(j)}$.

- **Step2-3:** Repeat carrying out each rule of Step2-2 by adding 1 to the value of $j$ till $j=j$. When $j=j$, move to Step2-4.

- **Step2-4:** Repeat the cycle from Step2-2 to Step2-3 by adding 1 to $n$. When $n=n$, STEP2 is finished.

**STEP3:** Calculate the distinct degree $S_n$ between $V_{j(j)}$ and $U_{n,j(j)}$.

**STEP4:** Express $U_{n,j(j)}$, whose $S_n$ is maximum value as $U_{m,j(j)}$ and express Base Knowledge including $U_{m,j(j)}$ as $K_m$. ($m \in 1, 2, \cdots, n$)

**5) [Definition]** “$U_{m,j(j)}$ gained in STEP4 is defined as Mapping Knowledge.”

**STEP5:** Delete $V_{j(j)}$ from Mapped Knowledge $H$ and map the Mapping Knowledge $U_{m,j(j)}$ corresponding to $V_{j(j)}$. The created knowledge is expressed as $H$.

**STEP6:** Add the knowledge $H$ to the elements of the competitive hypotheses and retry the design to examine the correctness of the knowledge. If the knowledge is correct, $H$ is judged as new knowledge and the algorithm is finished. If not, select another hypothesis from among the competitive hypotheses as Mapped Knowledge and repeat from STEP1.

STEP1 searches for the knowledge values which are the causes of a contradiction. For example, when $J(1)$ is regarded as chuck construction and $J(2)$ is loading method, as shown in Figure 4, those
\( J(j) \)  

| \( J(1) \) | Chuck construction |
| \( J(2) \) | Loading method |
| \( J(3) \) | Unloading method |
| \( J(4) \) | Feed mechanism |
| \( J(5) \) | Feed motor |
| \( J(6) \) | Main spindle bearing |
| \( J(7) \) | Spindle type |

- Value of \( J(j) \)  
  - \( J(1) \) : Collet chuck  
  - \( J(2) \) : Loader  
  - \( J(3) \) : Loader  
  - \( J(4) \) : Ball screw  
  - \( J(5) \) : DC servo  
  - \( J(6) \) : Radial bearing  
  - \( J(7) \) : Horizontal

**Figure 4** Example of \( V_{x(j)} \)

\[
\begin{array}{c|c}
\text{slot}[J(j)] & \text{U}_{j,1} \\
\hline
\text{slot}[J(1)] & \text{Chuck construction} \\
\text{slot}[J(2)] & \text{Loading method} \\
\text{slot}[J(3)] & \text{Unloading method} \\
\text{slot}[J(4)] & \text{Feed mechanism} \\
\text{slot}[J(5)] & \text{Feed motor} \\
\text{slot}[J(6)] & \text{Main spindle bearing} \\
\text{slot}[J(7)] & \text{Spindle type} \\
\end{array}
\]

- \( U_{j,1} \)  
  - \( U_{1,1} \) : Sleeve chuck  
  - \( U_{1,2} \) : Bar feed  
  - \( U_{1,3} \) : Bar feed  
  - \( U_{1,4} \) : Ball screw  
  - \( U_{1,5} \) : DC servo  
  - \( U_{1,6} \) : Radial bearing  
  - \( U_{1,7} \) : Horizontal

**Figure 5** Example of \( U_{n,x(j)} \)
values, value[J(1)] and value[J(2)] are value[J(1)] = \( V_{J(1)} = \) collet chuck and value[J(2)] = \( V_{J(2)} = \) loader.

STEP2 searches for the knowledge values of the remaining knowledge just like in STEP1. For example, the seven values of \( U_{1,J(j)} \), for Base Knowledge \( baseK_1 \) of Figure 5 are searched such as \( U_{1,J(1)} = \) sleeve type for slot[J(1)] = chuck construction and \( U_{1,J(2)} = \) bar feed type for slot[J(2)] = loading method.

The distinct degree \( S_n \) in STEP3 is acquired as below. As \( S_n \) indicates the scale how far it is from usable knowledge to contradictory knowledge, each distinct value \( \lambda_{nj} \) of \( j \) kinds contradictory knowledge is first acquired. Second, by summing up the distinct values, \( S_n \) is acquired. The distinct values \( \lambda_{nj} \) indicating the value difference between contradictory attribute names \( J(j) \) of Base Knowledge \( baseK_n \) and those of Mapped Knowledge \( appliedH \) takes the two values of Equation (1). That is if the both attribute names are same, \( \lambda_{nj} = 0 \), if not, \( \lambda_{nj} = 1 \).

\[
\begin{align*}
\text{IF } V_{J(j)} & = U_{n,J(j)}, \text{ THEN } \lambda_{nj} = 0 \\
\text{IF } V_{J(j)} & \neq U_{n,J(j)}, \text{ THEN } \lambda_{nj} = 1
\end{align*}
\]  
\[
\ldots (1)
\]

As the distinct values \( \lambda_{nj} \) have various \( j \) values for a single Base Knowledge, the \( j \) values are summed up by using Equation (2). The summed value is the distinct degree \( S_n \) for Base Knowledge.

\[
S_n = \sum_{j=1}^{j} \omega_j \times \lambda_{nj} \ldots (2)
\]

The \( j \) kinds of attributes of contradictory knowledge \( CK \) do not always have an equal influence on the contradiction. Something that indicates the influence degree on each of various \( j \) attributes is needed and this is expressed as the weight value \( \omega_j \) of Equation (2).

As mentioned above, the algorithm as shown in Figure 6 acquires new knowledge to map Mapping Knowledge corresponding to the most different or distant concepts from the Mapped Attribute which is the cause of the contradiction for a constraint to Mapped Knowledge in turn.

The characteristics of the algorithm are shown below. Because Mapping Knowledge \( U_{m,J(j)} \) is the farthest apart from the value \( V_{J(j)} \) which caused a contradiction, the possibility of the same
Figure 6 Mapping algorithm
contradiction happening again is remote. Because the remaining knowledge not to be mapped in Mapped Knowledge is good knowledge which has survived till the last design process that does not cause a contradiction, the new knowledge $\text{new}_H$ has the following two characteristics (1) good knowledge from the previous design process is inherited and (2) defective one is revised by mapping. Because hypothetical reasoning repeats each examination of the design process with the new knowledge $\text{new}_H$, the correctness of the new knowledge can be judged.

4. Knowledge Acquisition For Machine Tools

The new knowledge acquisition algorithm mentioned in chapter 3 is applied to the knowledge acquisition for machine tools whose knowledge is indispensable in the design process of production system. When the knowledge for machine tools is expressed with a computer system, one of the methods used is semantic networks representing the detailed specifications and features of their construction and functions. That is, the method is to describe the knowledge for each of the construction parts and functions with a frame type knowledge. By this method, the type names of machine tools are allotted as a frame name, the names of construction parts such as a spindle head are allotted as a slot and the characters of the construction parts are allotted as facet values. For example, the frame knowledge of the machine tool, STS-1 is shown as (STS-1 (Spindle-head (Value (Vertical) ) ) ). Figure 7 shows an example of frame type knowledge. The goal of the new knowledge acquisition in the application example is to obtain the machine tools knowledge expressed with a frame type knowledge as shown in Figure 7.

As the application example, the new knowledge acquisition is executed with Lisp language in the expert system assisting production line design for shaft parts which we have developed[12][13]. The expert system has drawing information and the specifications of the production system as input information and decides the necessary processing names, process sequence, machine tools and their specifications, and machining conditions as output information. In the expert system, the contradictory knowledge $\text{CK}$ which is the key entering the new knowledge acquisition process of machine tools is expressed with meta knowledge as shown in Figure 8. Contradictory names are assigned as a slot of meta knowledge and the
Figure 7 Frame knowledge example of machine tool

Figure 8 Meta knowledge

machine tool’s construction parts names and \( \omega \) values corresponding to contradictory knowledge which cause the contradictions described in the above slot are assigned as facet values. The \( \omega \) indicates \( \omega_i \) of the equation (2) and its meaning is (the value indicating) how each of the construction parts depends on the outbreak of the contradictions. The values of the \( \omega \) are chosen in order that the sum values become 1. The values are heuristic values and, in the example, they are decided by the interview of experienced planners.

After executing the developed system, the contradiction happened not to find any machine tools that satisfy a line cycle time. To pursue the contradiction, the system found the construction parts and their \( \omega \) values by searching in the meta knowledge as shown in Figure 9. Figure 10 shows the description examples for the construction parts knowledge. In the figure, the machine tool MV-5 is Mapped Knowledge \( \text{applicH} \), the machine tool CL-105 is Base Knowledge \( \text{baseK} \), whose distinct degree \( S_x \) is maximum. The distinct degree \( S_x \) of the example is \( S_x = 0 \times 0.05 + 0 \times 0.15 + 1 \times 0.15 + 0 \times 0.05 + O \times 0.25 + 1 \times 0.25 + O \times 0.05 + 1 \times 0.05 = 0.45 \) The parts of the acquired knowledge frame are shown in Figure 11. The acquired
<table>
<thead>
<tr>
<th>Construction</th>
<th>$\omega$ Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spindle control</td>
<td>0.05</td>
</tr>
<tr>
<td>Feed motor</td>
<td>0.15</td>
</tr>
<tr>
<td>Chuck construction</td>
<td>0.15</td>
</tr>
<tr>
<td>Feed control</td>
<td>0.05</td>
</tr>
<tr>
<td>Feed method</td>
<td>0.25</td>
</tr>
<tr>
<td>Loading method</td>
<td>0.25</td>
</tr>
<tr>
<td>Main spindle bearing</td>
<td>0.05</td>
</tr>
<tr>
<td>Main spindle type</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Figure 9 Contradictory knowledge and $\omega$ values

\[
(MV-5
(S-CONTROL (VALUE (MOTOR)))
(FEED-MOTOR (VALUE (SERVO-MOTOR)))
(CHUCK-CONSTRUCTION (VALUE (FINGER)))
(FEED-CONTROL (VALUE (NC)))
(FEED-MECHA (VALUE (BALL-SCREW)))
(LOADING (VALUE (LOADER)))
(SPINDLE-BRG (VALUE (RADIAL-BRG)))
(SPINDLE-TYPE (VALUE (VERTICAL))))
\]

\[
(CL-105
(S-CONTROL (VALUE (MOTOR)))
(FEED-MOTOR (VALUE (SERVO-MOTOR)))
(CHUCK-CONSTRUCTION (VALUE (SLEEVE-CHUCK)))
(FEED-CONTROL (VALUE (NC)))
(FEED-MECHA (VALUE (BALL-BRG)))
(LOADING (VALUE (BARFEED)))
(SPINDLE-BRG (VALUE (RADIAL-BRG)))
(SPINDLE-TYPE (VALUE (HORIZONTAL))))
\]

Figure 10 Description examples of construction parts knowledge
Figure 11 Description example of acquired new knowledge
machine tool has the specification representing a bar-feed NC lathe that loads a raw material by making it go through a chuck sleeve. By using the new machine tool, the design of the production line was done again. As a result, its correctness was admitted and, finally, the production line including the machine tool was acquired.

5. Conclusions

This research describes knowledge acquisition technology which is a higher intelligence function of a computer system assisting production system design.

In production system design which needs the reasoning of many kinds of decision items, the moving point to a knowledge acquisition process is judged under hypothetical reasoning. As a method to acquire new knowledge after the judgment that all available knowledge is incorrect, the new knowledge acquisition method maps parts knowledge which is selected from a distant or distinct conceptual space. In order to ascertain its usefulness, the new method was included in an expert system assisting production system design. As a result, the frame type knowledge for a new machine tool was acquired.

References