中國機械工程學刊第二十八卷第六期第 593~604 頁(民國九十六年)
Journal of the Chinese Society of Mechanical Engineers, Vol.28, No.6, pp.593~604 (2007)

Knowledge Representation and Reasoning Methodology based on CBR Algorithm for Modular Fixture Design

Shu Huang Sun* and Jahau Lewis Chen**

Keywords：CBR, Modular fixture, MOP, Knowledge representation.

ABSTRACT

CBR algorithm provides a better knowledge transfer and explanation than rule-based inference. It solves new problems by adapting solutions that were used to solve old problems. Based on CBR algorithm, a methodology applied in modular fixture design and focus on workpiece locating is proposed in this study. A similar solution can be retrieved from past experiences. Evaluation is applied for this retrieved case by checking degrees of freedom (DOF) to determine whether it is satisfactory for a new problem and some components would be replaced if it is not. According to this methodology, a computer-aided modular fixture design system can be established in future. In the system, three sub-bases would be included. Data base stores many function structures that are assembled by modular components to complete some functions. Knowledge base stores the qualitative knowledge that is required in considering the location of the workpieces. Case base stores previous successful design cases that can be applied to develop a new solution. MOP-based memory technique is applied to organize these complex data, knowledge and case base. A demonstrated example is finally provided in this study to illustrate how this methodology works.

This methodology principally focuses on inference process of case evaluation and modification. This is the most important and difficult issue on CBR algorithm. In the evaluation of workpiece locating, geometry recognition play a critical role. Feature recognition is beyond this study and then too detail discussion about that would not be given here. For this reason, the methodology can handle simple geometry workpiece only presently.

Paper Received April, 2006. Revised Sepfember, 2007, Accepted October, 2007, Author for Correspondence: Shu Huang Sun.

**Associated Professor, Department of Mechanical Engineering, Kun Shan University, Tainan county, Taiwan 71003, ROC.*

*** Professor, Department of Mechanical Engineering, National Cheng Kung University, Tainan, Taiwan 70101, ROC.*

INTRODUCTION

In fixture design, designers are frequently being requested to design a fixture to hold a workpiece that is similar to another one whose fixture had been designed before. Designers generally solve this problem by referring stored files to find a similar one and modify it or, for the worse, to design from initial state if a similar design is absent. Developing a methodology that has the ability to accumulate past experiences to assist designers, therefore, is absolutely essential for a task that is time-consuming for the conventional approach and based largely on previous experiences. The methodology of computer-aided design system can roughly be divided into two types, i.e., creation and variation. Creation means designing from null. Rule-based expert system is catalogued into this type. The actual question, in this way, is how knowledge is encoded into rules clearly. However, in design domain where the situation is not so clear, rules seem too brief to involve these knowledge. Consequently, rules often have many exceptions and a rule system can often suggest nothing. Variation means designing from modification of the previous design. CBR algorithm is catalogued into this type. A case-based reasoner has a case library. Experiences are saved in it and represented by storing particulars of previously design artifacts as cases. The reasoner retrieves a most similar case and modifies it to satisfy the new problem. Modifying a current design is always easier than designing from initial state. Furthermore, CBR algorithm is proposed as a more psychologically plausible model of the reasoning of an expert than the more fashionable rule-based reasoning systems.

CBR is derived from a view of understanding as an explanation process (Schank 1982). Some case-based problem-solving programs have been successful. MEDIATOR (Kolodner & Simpson 1989) solves resource disputes; JUDGE (Bain 1986), similar to a judge, sentences juvenile offenders; CHEF (Hammond 1989) created new recipes from old ones. In those earlier works, CBR systems primarily focused on text explanations and utilized sequential processing of linguistic symbolic expressions but were limited in power when dealing with numerical or graphical information. Applications of CBR in engineering have only been recently proposed. Applications in process planning are the most common within engineering domain (Tiwari et al. 2001, Humm et al. 1991, Pu & Reschberger 1991, Zarley 1991, Tsatsoulis & Kashyap 1988). Additionally, other applications can be found such as linkage design (Kramer & Barrow 1992, Bose et al. 1991), architecture or bridge construction (Fu & Wang 1999, Tah et al. 1998, Faltings & Hua 1991, Goel et al. 1991), sheet metal manufacturing (Cser et al. 1991), machine fault diagnosis (Fong & Hui 2001), structural optimization (Arciszewski & Ziarko 1991), injection moulding and process (Kwong et al. 1997) and fixture design (Sun & Chen 1993, 1995, 1996). Though different knowledge structures and algorithms are employed in these systems, some primary elements are inherent. Problem description and solution are two main ingredients of a case. Different characteristics are along with each case and are employed to assess available cases. Additionally, some systems may contain outcomes of a solution to evaluate the result or explanation and can then repair or predict failures to avoid duplicating the same mistake.

The first development of a fixture system is primarily concentrated on a dedicated fixture. Modular fixture systems have only became popular in recent years though it has the advantage of flexibility considerations, particularly in high-variety small-batch computerized manufacturing environments. Literature survey of the automatic fixture design has been presented by Cecil (2001), Bi and Zhang (2001), Trappey and Liu (1990). Gandhi and Thompson (1986) proposed a methodology to develop an expert system that synthesizes a fixtured surface and the associated modular types. Grippo and Gandhi (1987) stored empirical data into a library to aid modular fixture design. Whybrew and Nogi (1992) developed a modular fixture assembly system. Liu (1994) provides a systematic method to transfer a dedicated fixture to a modular fixture by dividing a fixture into many basic units and replacing them stepwise. Rong and Zhu (1992) proposed a modular fixture design system based on GT technology. The system developed by Hou and Trappey (2001) constructs a modular fixture by selecting proper elements and determining their location and orientations.

Even though some computer-aided fixture design methodologies had been proposed in these above papers, most of them are based on rules inference and the other experience-based systems are only give users a concept solution and have no any modification suggestions. Developments on CBR modular fixture design methodology, therefore, are still considerable.

OVERVIEW Of CBR

CBR algorithm

Case-based reasoning means adapting old solutions to meet new demands, using old cases to account for new situations, using old cases to critique new solutions, or reasoning from precedents to interpret a new situation or create an equitable solution for a new problem. The primary processes required for CBR are briefly described in the following. A detailed discussion can be found in (Kolodner 1993).

(1) Index assignment: Characterizes the input problem by assigning the appropriate features of it.
(2) Retrieval: Retrieves the relevant case from the case library.
(3) Explanation: Explains the deficiencies of the retrieved case by making a comparison of the differences between this case and the input problem. The explanation involves two aspects, i.e., which features are unsatisfactory and require modification; and how to modify these features so as to satisfy new conditions.
(4) Modification: Modifies the retrieved case to conform to new situations according to the result of explanation.
(5) Store: Saves the modified case as a new case into the case library. The case library can then be gradually expanded.

MOP memory organization

The primary function of the memory system is to store and retrieve knowledge. This knowledge is represented in the form of Memory Organization Packages (MOP) (Schank 1982) which is a hierarchical organization. Each MOP means a case or an event. It represents knowledge as a frame and contains slots. Each slot has a role or slot name to represent one attribute of the MOP and a corresponding filler, the value of the attribute, that is filled by another MOP. A MOP can have specializations or abstractions. A specialization is a more specific version of the MOP whereas an abstraction is a more generalized version of the MOP. For instance, in this fixture design system, a *Case_2* MOP, shown in Figure 1, is a specialization of the *Design_case* MOP. It contains a slot *Clamping_device* whose filler points to another MOP, *Clamping_5*, that is a specialization of a clamping structure.

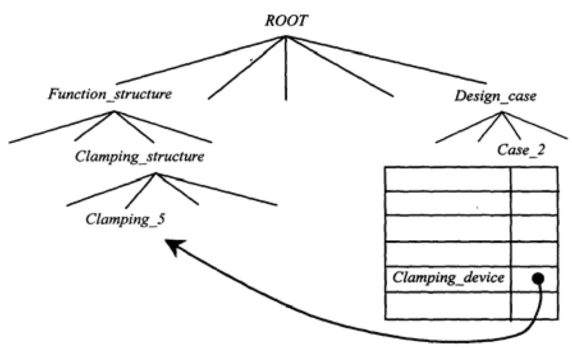


Fig. 1. MOP structure explanation

THE STRUCTURE AND LOCATION METHOD OF A MODULAR FIXTURE

A modular fixture system is directly assembled from many standard components. Many related components are assembled to be a function structure to complete some specific function and many various function structures are assembled to be a fixture. A modular fixture can therefore be divided into many structures and a structure can be subdivided into many components. A modular fixture therefore is easier than a dedicated fixture to systematize. A hierarchy database can be established to store many function structures. Depending on the required functions, some specific structures can be selected to assemble a fixture. A detailed description is provided in a later section.

The fundamental principles of the basic fixture design are classified into two types, i.e., the locating and clamping principles. The locating function is the main issue in a fixture. A locating device establishes the desired relationship between the workpiece and the fixture and this, in turn, establishes the desired relationship between the workpiece and the cutter. The selection of a locating device therefore ensures the precision of manufacturing. A free body in space has totally a maximum of six DOF, involving three linear movements and three rotational movements for x, y, and z axis. The function of location is to ensure the precision and immobility of the workpiece in some DOF those are significant in machining. The most popular principle in location is the 3-2-1 or six points principle. The interpretation of this principle is, as shown in Figure 2, three, two, one locating points are required for three preferably perpendicular planes, respectively. Although this principle is only valid for prismatic workpiece, in a wider sense, any workpiece can be located by six fixed points whose positions may not obey 3-2-1 principle. Importantly, the contact between the locating device and workpiece may not only be point-contact but also line or surface contact. For instance, the contact for a cylinder located on a V-block is two parallel lines. Four DOF are constrained in this contact. The consideration of locating a workpiece, therefore, is not to examine how many locating points are employed but to examine how many DOF are constrained.

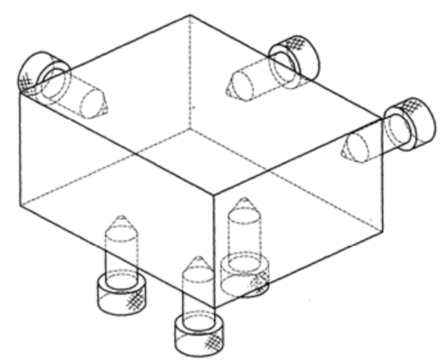


Fig. 2. Demonstration of 3-2-1 location principle

A workpiece is referred to be full-location if its six DOF are all constrained. Similarly, a workpiece is referred to be over-location or under-location if some constraints of DOF are duplicated or absent. The under-locating is not promised since it would result in the movement of the workpiece within machining. The over-locating is also not recommended since, in some situations, it would result in the instability and reduce precision. However, constraining all the six DOF in some machining operations is unnecessary. Those DOF that have effects on the precision of the workpiece are the only DOF that are required to be constrained. Restated, if the absence of the constraint of a DOF would result in the missing of the precision, then this DOF is required to be constrained. On the premise of the precision, ignoring some DOF can simplify the fixture design. For instance, to drill a hole in the center of a cylinder that is located on a V-block, the constraint in the rotational axis of the cylinder is unnecessary since it has no effect on this operation.

KNOWLEDGE REPRESENTATION AND MEMORY ORGANIZATION

The memory organization of a CBR modular fixture design system will be a hierarchical tree. The top level is the root of this tree structure and is named *ROOT*. Under *ROOT*, there are four subtrees, i.e., the function structure subtree, the machining form subtree, the contact form subtree, and the design case subtree. These four subtrees constitute the data base, knowledge base, and case base of this system. Detailed descriptions about these three bases are provided in the following.

The data base -- function structure subtree

The first subtree is named *Function_structure*. This subtree acted as a database. All modular components are pre-assembled to many different structures and stored in this subtree. The tree organization is shown in Figure 3. The essential fixture elements are grouped into five catalogued viz., locating devices, clamping devices, guiding elements, supporting elements and foundations. Under this *Function_structure* MOP, there are five specializations corresponding to these five groups. Those are *Locating_structure*, *Clamping_structure*, *Guiding_ structure*, *Support_structure*, and *Foundation_structure*. Each one groups many different structure MOP that completes the same function. For instance, these structures MOP *Clamping_l*, *Clamping_2*, etc. are all the clamping structures and under the *Clamping_ structure* MOP. A structure may be only one component or a combined unit that contains more than one component. Specializations named *Element* are also under the structure MOP. They represent the components that are assembled in this structure. Each *Element* MOP contains three slots, i.e. *Name*, *Number*, and *Quantity*. An illustrative example to clearly explain this hierarchy database is provided in Figure 4. The corresponding MOP and its specializations are also involved. The *Clamping_5* MOP contains four specializations. From the slots of these *Element* MOP, it can be understood that one Forked_clamp, two Stud_for_T_nut, three Fixture_nut and one Collar_nut are assembled to be a clamping structure to complete the clamping function.

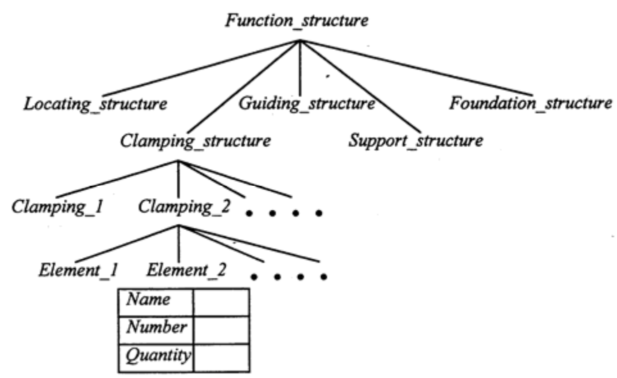


Fig. 3. The subtree of the database

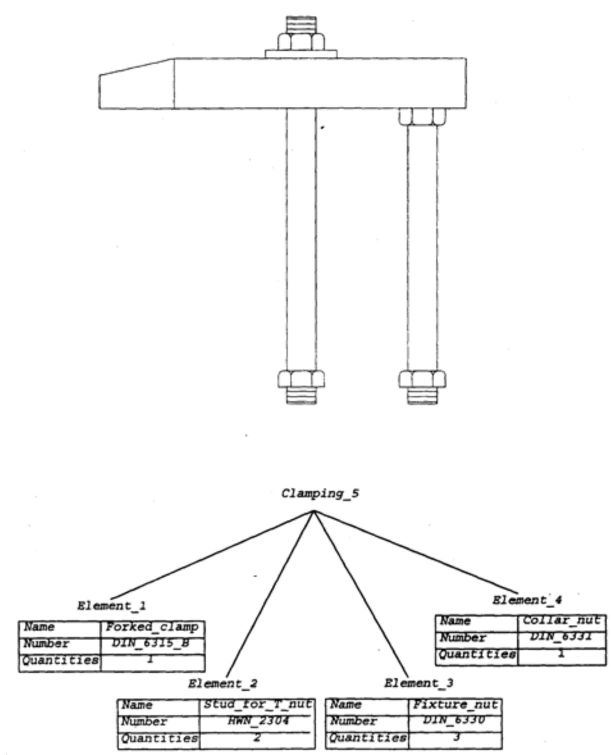


Fig. 4. An example of the Clamping MOP

The knowledge base -- machining form subtree and contact form subtree

The second and third subtree acted as a knowledge base. These two subtrees capture the qualitative knowledge that is required in considering the locating of a workpiece. Based on the assumption that only one operation is accomplished at one time, a retrieved case is evaluated whether these DOF that are required to be constrained in the coming operation are all constrained. To carry out this evaluation, the system should have proper knowledge about two issues. First, which constrained DOF are required in some machining is essential. This is stored in the second subtree. Second, which DOF had been constrained in some location method is also critical. This is stored in the third subtree. Before this consideration, however, establishing a coincident coordinate system in which the required constrained DOF and the constrained DOF are defined is necessary. It is also applied in the definition of the workpiece direction. The coordinate in this study defines the rotational axis of a rotational workpiece as the z-axis and the axis that is perpendicular to the first locating datum as the z-axis for a non-rotational workpiece.

Different operations require different constrained DOF. The second subtree defines which constrained DOF are required in all various machining. Examples for drilling operation are provided in Figure 5. The required constrained DOF, for example, for drilling a through hole in the center of a cylinder are the linear movements and the rotational movements of the x and y axis. Whereas the linear movement and the rotational movement of the z axis leave free. In another operation, to drill a blind hole in the center of a cylinder, not only these four DOF that are constrained in the above operation but the linear movement of the z axis is also required to be constrained. The organizational structure of this subtree is shown in Figure 6. This subtree is named *Machining_form*. Some specializations of it are available to represent the various machines, e.g., *Milling_machine*, *Drilling_machine*, and *Lathe*. Each specialization contains again many specializations to represent those various machining. The *Drilling_machine* MOP, for example, subdivides into many operation MOP named *Through_hole_in_center_of_cylinder*, *Blind_hole_in_center_of_cylinder*, etc. Each operation MOP contains one slot named *Required_constrained_DOF*. Its filler is the linear or the rotational movements of the x, y or z axis.

The third subtree defines which DOF are constrained in various location methods. Cataloguing location methods into some groups is deemed necessary owing to their large numbers. A classification method according to the contacted geometric feature between the workpiece and the locating device as Figure 7 shown is proposed in this study. Those location methods that constrain the same DOF are grouped together. In this manner, the constrained DOF of each location method are all explicit. This subtree is named *Contact_form*. Its specializations are named by the contacted geometric feature, such as *One_point*, *Two_parallel_line*, etc. Each of them contains a slot named *Constrained_DOF* and has many specializations named *Location_method*. Each *Location_method* MOP represents one location method and contains two slots, i.e., *Localing_feature* and *Locating_device*. All the geometric features that can be applied as locating features are pre-defined and filled in the *Locating_feature*. The filler of the *Locating_device* points to one specialization of the *Locating_structure* MOP that belongs to the *Function_structure* subtree. For instance, a MOP named *Location_method_5* that describes the location method of locating a cylinder on a V-block is a specialization of the *Two_parallel_line* MOP. The filler of the *Locating_feature* slot is *External_cylinder* and the filler of the slot *Locating_device* points to a locating structure that contains at least one V-block component. Furthermore, this location method constrains the linear and rotational movements of the x, y axis since its abstraction MOP is *Two_parallel_line* whose *Constrained_DOF* slot indicates these DOF are constrained. The tree organization is shown in Figure 8.

Description	Required constrained DOF
Through hole in center of ball	$\bar{x}\ \bar{y}$
Blind hole in center of ball	$\bar{x}\ \bar{y}\ \bar{z}$
Through hole in block	$\hat{x}\ \hat{y}\ \hat{z}$ $\bar{x}\ \bar{y}$
Blind hole in block	$\hat{x}\ \hat{y}\ \hat{z}$ $\bar{x}\ \bar{y}\ \bar{z}$
Through hole in center of end surface of cylinder	$\hat{x}\ \hat{y}$ $\bar{x}\ \bar{y}$
Blind hole in center of end surface of cylinder	$\hat{x}\ \hat{y}$ $\bar{x}\ \bar{y}\ \bar{z}$
Through hole not in center of end surface of cylinder	$\hat{x}\ \hat{y}\ \hat{z}$ $\bar{x}\ \bar{y}$
Blind hole not in center of end surface of cylinder	$\hat{x}\ \hat{y}\ \hat{z}$ $\bar{x}\ \bar{y}\ \bar{z}$
Through hole in lateral of cylinder	$\hat{x}\ \hat{z}$ $\bar{x}\ \bar{z}$
Blind hole in lateral of cylinder	$\hat{x}\ \hat{z}$ $\bar{x}\ \bar{y}\ \bar{z}$

Fig. 5. Required constrained DOF in various machining operations

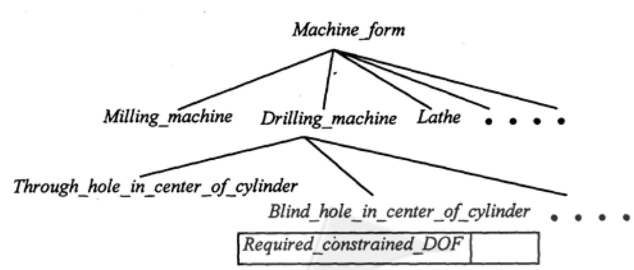


Fig. 6. The subtree that defines the required constrained DOF of each machining operation

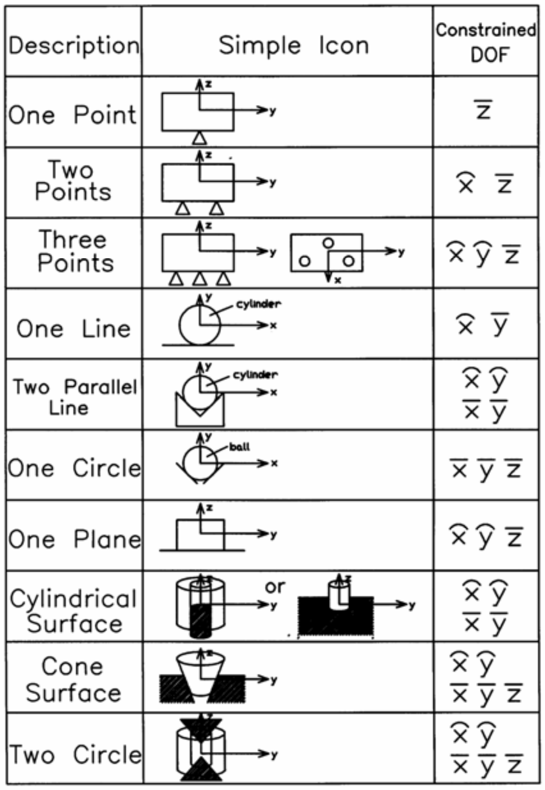


Fig. 7. Constrained DOF in various contacted geometric features

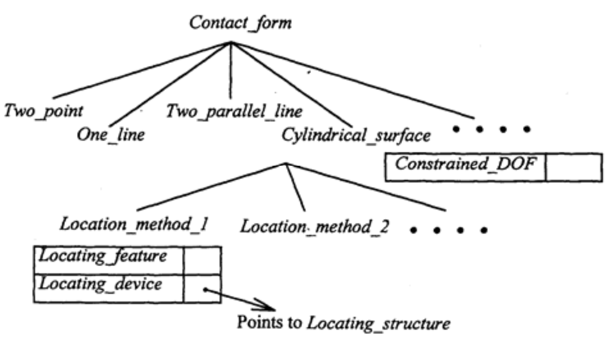


Fig. 8. The subtree that defined the constrained DOF of each location method

The case base -- design case subtree

The last subtree acted as a case base. This subtree is named *Design_case*. All previous design cases are stored in this subtree and sequentially numbered as *Case_1*, *Case_2*, etc. Each case contains many slots. These slots can be divided into two portions, problem description and solution data. Four slots, *Workpiece_class*, *External_shape*, *lnternal_shape* and *Workpiece_direction* are grouped into problem description. The *Workpiece_class* slot distinguishes the main shape of workpiece as rotational or non-rotational. The external and internal shape of the workpiece are involved in problem description since they are significant to the location method. The *External_shape* slot describes the external shape of the workpiece. For instance, a rotational workpiece may be classified as a smooth cylinder or a cone. The *Internal_shape* slot describes the internal shape of the workpiece. For instance, a workpiece may contain a hole or not. If a workpiece contains a machined hole, it can be located by a pin. The direction of the workpiece is also important in fixture design. Different directions of the same workpiece result in different fixture design. The *Workpiece_direction* slot describes the direction of the workpiece with respect to the base plate of the modular fixture. For instance, the rotational axis of a cylindrical workpiece may be perpendicular, parallel, or inclined to the base plate. If the workpiece is inclined to the base plate, an adjustable rotating component is definitely required. The direction of the workpiece is based on the coordinate that has been defined in above discussion.

The first slot of solution data is *Fixture_type*. This slot indicates which type this fixture belongs to. Fixtures can be catalogued into many different types depending on the machining operation, e.g., drilling, milling. Fixture design can then be catalogued by the *Fixture_type* slot to reduce the numbers of cases in case retrieval. This slot along with these four slots in problem description compose the indexes in case retrieval.

The other slots describe the entire fixture design. All the modular components used in this fixture are stored in these slots. The essential fixture elements, as previously stated, are grouped into locating devices, clamping devices, guiding elements, supporting elements and foundations. These slots are also grouped into these categories. Four of these are *Clamping_device*, *Supporting_device*, *Guiding_device*, and *Foundation_device*. The locating devices are subdivided into the first, second and third locating devices and represented by *lst_Location_method*, *2nd_Location_method*, and *3rd_Location_method* slots. The locating function is the most important issue in a fixture design and also is the focus of this study. It is the reason why the locating device is subdivided into the first, second and third locating devices. The fillers of these *Location_method* slots point to one of the *Location_method* MOP in the *Contact_form* subtree. Except for these three location method MOP, the filler of the *Clamping_device* points to one of the specializations of the *Clamping_structure* MOP in the *Function_structure* subtree; the filler of the *Supporting_device* slot points to one of the specializations of the *Supportmg_structure* MOP, and so on. The organization of this subtree is shown in Figure 9.

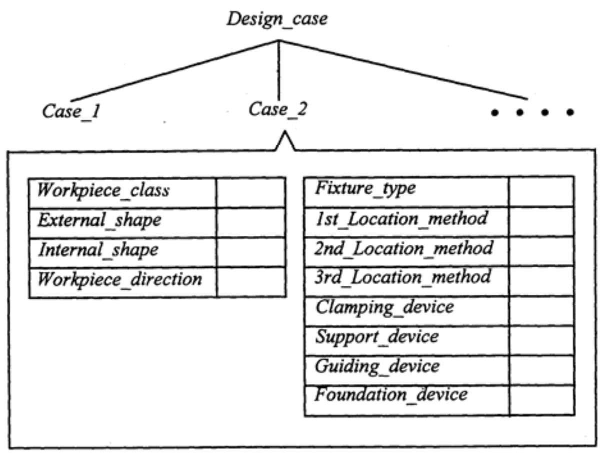


Fig. 9. The subtree of the case library

METHODOLOGY AND RUNNING PROCEDURE

Figure 10 is the flow chart of this CBR modular fixture design methodology. The procedure involves several operations: problem input, searching the case library to find the most suitable case, evaluating whether the retrieved case is satisfactory, constructing a new solution for the new problem if the retrieved case is not satisfied, and storing this new solution into the case library. Each step is described in the following.

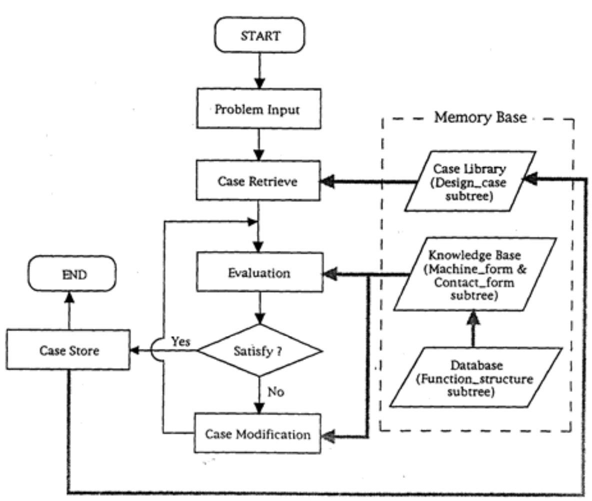


Fig. 10. The flowchart of the CBR modular fixture design system

(1) Problem input: A new problem contains many slots. Five of these slots, the same as those in an existing case, are *Workpiece_class*, *External_shape*, *Internal_shape*, *Workpiece_direction* and *Fixture _type*. Besides these five slots, there are one slot named *Machining_operation* and some *Locating_ feature* slots. Problem input entails filling the fillers of these slots. The fillers of first four slots describe the geometric features of the workpiece. The filler of the *Fixture_type* indicates the machine type of the machining of the workpiece such as milling, drilling, etc. The filler of the *Machining_ operation* involves selecting one operation from these various machining operation MOP that are stored in the *Machining_form* subtree and then points to it. From the slot *Required_constrained_ DOF* of this selected MOP, the required constrained DOF are obtained. The fillers of the *Locating_ feature* are filled by the geometric features of this workpiece that can be employed in location.

(2) Case retrieval: The *Workpiece_class*, *External_ shape*, *Internal_shape*, *Workpiece_ direction* and *Fixture_type* slots in the new problem and existing cases are the indexes and employed in case retrieval step. All cases in the case library are pre-grouped into subsets according to the *Fixture_type* slots. A most similar case can be retrieved by calculating the similarities between the fillers of these four slots in the new problem and those in all existing cases within the subset whose *Fixture_type* is the same with the new problem. Furthermore, a weighting value can be assigned to each slot. The importance of each attribute can then take effects in case retrieval. Dividing whole library into small subsets is necessary for improving the efficiency of the case retrieval when library becomes larger.

(3) Evaluation: After a case is retrieved, it need to be evaluated if it is satisfied. The fillers of first, second and third location methods in the retrieved case all point to an individual *Location_method* MOP in the *Contact_form* subtree. The abstraction of each MOP, the contacted geometric feature MOP, contains a slot named *Constrained_ DOF*. From the filler of this slot, the constrained DOF of these locations in the retrieved case can be obtained. The *Location_method* MOP contains two slots, *Locating_feature* and *Locating_device*. The filler of the *Locating_feature* records the contacted geometric feature corresponding to this location method. The filler of the *Locating_device* points to the *Locating_structure* MOP in the *Function_structure* subtree. In this manner, all the locating devices and the corresponding contacted geometric features of the retrieved case can be obtained. The evaluation involves two aspects.

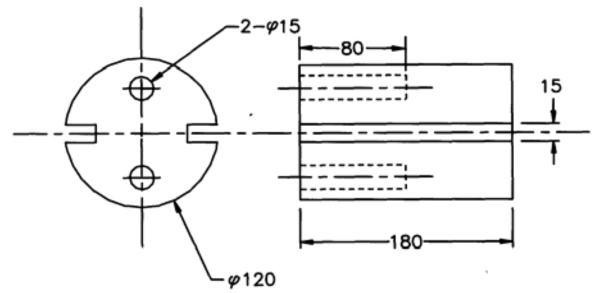
The first one is to evaluate whether all DOF that are required to be constrained in the new problem are constrained by the locating devices of the retrieved case. Another one is to evaluate whether all contacted geometric features that are corresponding to these locating devices are involved in the workpiece of the new problem. If these two evaluations are successful, the solution for the new problem is obtained; otherwise, case modification process is followed.

(4) Case Modification: Case modification progresses in two ways. The location method is implied to be unsatisfactory if the corresponding locating feature of the retrieved case can not be found in the new problem. Another location method, under the same contacted geometric feature MOP, whose corresponding locating feature can be found in the new problem is selected. Another modification way happens when the constrained DOF are unsatisfactory to the required constrained DOF. Another contacted geometric feature MOP, whose constrained DOF are satisfactory, is selected. Under the selected contacted geometric feature MOP, the locating device whose corresponding locating feature can be found in the new problem is selected. The new locating device is merged to the retrieved case to replace the locating device that is unsatisfactory. The modification of the retrieved case is then finished.

(5) Case store: Humans begin learning and obtaining new experiences when they encounter a new problem and obtain a solution by referring to former experiences. The learning ability of the CBR system resembles this learning way. This methodology obtains a new case when it encounters a new problem and modifies its old case to satisfy new requirements. The new solution is stored into the case library when the modification is finished and becomes new knowledge. Hence, a system built according to this methodology can learn from itself by expanding the case library whenever it solves a new problem. The learning capability is not only of primary concern in this study, but is also the most meaningful part in the future CAD system.

ILLUSTRATED EXAMPLE

A workpiece shown in Figure 11 is applied to demonstrate how this methodology works. Assume a fixture designer is requested to design a fixture to hold it. Abbreviated description of the design process of this methodology, following the stated in the previous section, is presented in the following.



Problem description :

(Workpiece_class Long_cylinder)
(External_shape Uniform_diameter_along_entire_length)
(Internal_shape No_internal_hole)
(Workpiece_direction Z-axis_parallel_base_plate)
(Fixture_type Drill)
(Machining_operation
Blind_hole_not_in_center_of_end_surface_of_cylinder)
(Locating_feature External_cylinder)
(Locating_feature End_surface)
(Locating_feature Through_slot)

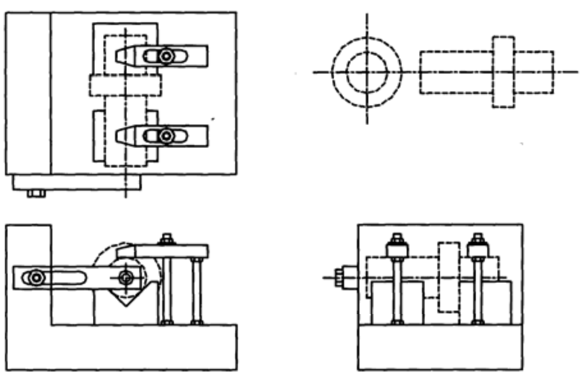
Fig. 11. A workpiece and its problem description of new input problem

(1) Fill all slots involved in the new problem. This workpiece is a rotational part with a uniform diameter along the entire length and contains no internal hole. The machining operations are to drill two ϕ15 blind holes and, in each operation, one hole is machined. These two holes lie on the end surface of this workpiece and are not coincident with the center of the end surface. Owing to operational requirements, the workpiece is placed on the base plate so that the z-axis parallels the base plate. Finally, the geometric features that can be employed in location are summed up to the *External_cylinder*, the *End_surface* and the *Through_slot*. The problem description together with all of these slots are shown in Fig 11.

(2) Figure 12 shows the retrieved previous design case that is the most similar to the new problem. It differs from the new problem only in the external shape of the workpiece and the machining operation. Workpiece and case description are also involved in this figure.

(3) Whether these three locating devices of the retrieved case are satisfied with the new problem is evaluated. The *lst_Location_method* slot points to the *Location_method_5*. From this MOP, information regarding the first location of the retrieved case is obtained. The locating feature is *External_cylinder*. It is obtained from the

Locating_feature slot. The constrained DOF are the linear movements and the rotational movements of the x and y axis. It is obtained from the *Constrained_DOF* slot that is involved in the abstraction MOP of the *Location_method_5*. The locating device and the constituent components are also obtained from the *Locating_device* slot. The second location of the retrieved case constrains the linear movement of the z-axis and the corresponding feature is *Flange*. The third location is vacant since the filler of this slot is Free. The *Machining_operation* slot of the new problem points to the *Blind_hole_not_in_center_of_end_surface_of_cylinder* MOP. This machining operation must constrain all of six DOF, as indicated from the filler of the *Required_constrained_DOF* slot. A comparison of the required constrained DOF of the new problem with the constrained DOF of these three locations of the retrieved case reveals that the locations of the retrieved case are unsatisfactory since one of the required constrained DOF, i.e., the rotational movement of the z-axis, is absent.



Case description :

(Workpiece_class Long_cylinder)
(External_shape Various_diameter_along_entire_length)
(Internal_shape No_internal_hole)
(Workpiece_direction Z-axis_parallel_base_plate)
(Fixture_type Drill)
(1st_Location_method Location_method_5)
(2nd_Location_method Location_method_13)
(3rd_Location_method Free)
(Clamping_device Clamping_3)
(Support_device None)
(Guiding_device Guiding_2)
(Foundation_device Foundation_3)

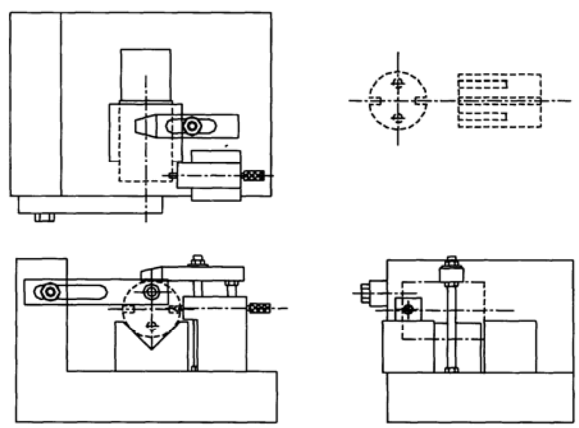
Fig. 12. The retrieved previous design case

(4) The locating feature of the first location method of the retrieved case, *External_cylinder*, can be found in the *Locating_feature* slot of the new problem. It means the first locating device of the retrieved case can be applied to the new problem. The second locating device of the retrieved case, however, can not be applied since the new problem does not contain the corresponding geometric feature. Under the same contacted geometric feature MOP, another location method whose *Locating_feature* is *End_surface* or *Through_slot* is selected. After searching, the *Location_method_l4* whose locating feature is *End_surface* is selected to replace the *Location_method_13*. *Location_method_14* locates the end surface by a block component. Five of the six DOF are constrained by the first and second locating devices and leaves the rotational movements of the z-axis free. The third location method is therefore necessary to constrain this DOF. Furthermore, the last locating feature of the new problem is the *Through_slot*. A location method that constrains the rotational movement of the z-axis and the corresponding locating feature is *Through_slot* is a satisfied one. Searches the *Contact_form* subtree again, the *Location_method_l7* that locates the through slot by a pin component is selected. This location method is then filled into the *3rd_Location_method* slot. The modified design, the workpiece, and the case description are shown in Figure 13. Summing these modification procedures, the V-block component is employed to locate the external cylinder and constrains the linear movements and rotational movements of the x, y axis; the block component is employed to locate the end surface and constrains the linear movement of the z-axis; and the pin component is employed to locate the through slot and constrains the rotational movement of the z-axis. All these six DOF are constrained and the final result for the new problem is satisfactory.

(5) The modified design is a solution of the new problem and becomes a new case. It would be stored into the case library and can be retrieved when the system faces a similar problem.

DISCUSSION AND FUTURE DEVELOPMENT

The CBR technology supports a more natural method than a rule-based expert system to simulate human thought. A modular fixture design methodology is proposed in this study based on CBR algorithm. In this algorithm, previous design case can be remembered and support a concept in a new design problem. Furthermore, valuable experiences can be accumulated in a computer-aided modular fixture design system and by saving these design cases, the system would gradually become more intelligent. The learning ability is an important property of CBR algorithm.



Case description :

(Workpiece_class Long_cylinder)
(External_shape Uniform_diameter_along_entire_length)
(Internal_shape No_internal_hole)
(Workpiece_direction Z-axis_parallel_base_plate)
(Fixture_type Drill)
(1st_Location_method Location_method_5)
(2nd_Location_method Location_method_14)
(3rd_Location_method Location_method_17)
(Clamping_device Clamping_4)
(Support_device None)
(Guiding_device Guiding_2)
(Foundation_device Foundation_3)

Fig. 13. The modified design

From an abstract perspective, the case retrieval in CBR algorithm is similar to the group technology (GT). Based on some indexes, both methodologies group similar cases together and select a case by comparing these indexes when facing a new problem. Some proposed systems that store previous experiences to assist designers can also be found. However, neither the group technology nor these proposed experience-based system possess the capability of modification.

Evaluating the retrieved case is the first necessary step before modification. From evaluation results, the system can understand why the case has failed and how to modify it. The ability of evaluation is emphasized in this methodology. Based on the evaluation, these failure locating devices in the previous design case can be modified to be a new design to solve the new problem. So far, evaluation in this methodology is only limited to workpiece locating. In the same manner, however, the evaluation of clamping, guiding, and supporting can also be developed according to some criteria. The role of this methodology is merely that of a concept supporter in the initial design state. Some details therefore are eliminated here and the consideration of the interference of the modular components is also excluded.

This study proposes a methodology based on CBR algorithm to apply in modular fixture design, especially on workpiece locating. This study is still in progress. Geometry feature recognition technology is the next issue. Including this technology, workpiece geometry can be identified more clearly. Case evaluation and modification capabilities then can be extended to handle complicated workpiece. Appling this methodology to establish a real computer-aided system is also expect in the next development.

REFERENCES

Arciszewski, T. & Ziarko, W., Structural Optimization : Case-Based Approach. Journal of Computing in Civil Engineering, 5(2), 159-174, (1991).

Bain, W., Case-Based Reasoning: A Computer Model of Subjective Assessment. Ph.D. Thesis, Department of Computer Science, Yale University, (1986).

Bi, Z. M. & Zhang, W. J., Flexible fixture design and automation: Review, issues and future directions. International Journal of Production Research, 39(13), 2867-2894, (2001).

Bose, A., Esterline, A., Riley, D. & Erdman, A., Case-Based Preliminary Linkage Design. Proceedings, International Conference on Design Productivity, Honolulu, Hawaii, February 6-9, 419-424, (1991).

Cecil, J., Computer-Aided Fixture Design – A Review and Future Trends. International Journal of Advanced Manufacturing Technology, 18, 790-793, (2001).

Cser, L., Geiger, M., Greska, W. & Hoffmann, M., Three Kinds of Case-based Learning in Sheet Metal Manufacturing. Computers in Industry, 17, 195-206, (1991).

Faltings, B. & Hua, K., Case-based Representation of Architectural Design Knowledge. Proceedings, Case-based Reasoning Workshop, Madison Hotel, Washington,D.C., May 8-10, 307-316, (1991).

Fu, C. C. & Wang, W., Sequence-Based Prediction in Conceptual Design of Bridges. Journal of Computing in Civil engineering, 13(1), 54-56, (1999).

Fong, A. C. M. & Hui, S. C., An intelligent online machine fault diagnosis system. Computing & Control Engineering Journal, 12(5), 217-223,

(2001).
Gandhi, M. V. & Thompson, B. S., Automated Design of Modular Fixture for Flexible Manufacturing Systems. Journal of Manufacturing Systems, 5(4), 243-252, (1986).
Goel, A. K., Kolodner, J. L., Pearce, M., Billington, R. & Zimring, C., Towards a Case-based Tool for Aiding Conceptual Design Problem Solving. Proceedings, Case-based Reasoning Workshop, Madison Hotel, Washington,D.C., May 8-10, l09-120, (1991).
Grippo, P. M., Gandhi, M. V. & Thompson, B. S., The Computer-Aided Design of Modular Fixturing Systems. International Journal of Advanced Manufacturing Technology, 2(2), 75-88, (1987).
Hou, J. L. & Trappey, A. J. C., Computer-Aided Fixture Design System for Comprehensive Modular Fixutures. International Journal of Production Research, 39(16), 3703-3725, (2001).
Hammond, C. J., Case-Based Planning: View Planning as a Memory Task. Academic Press, Boston, (1989).
Humm, B., Schulz, Ch., Radtke, M. & Wamecke, G., A System for Case-Based Process Planning. Computers in Industry, l7, l69-180, (1991).
Kolodner, J. L., Case-based Reasoning. Morgan Kaufinann Publishers, Inc., San Mateo, (1993).
Kolodner, J. L. & Simpson, R. L., The MEDIATOR: Analysis of an Early Case-Based Problem Solver. Cognitive Science, 13(4), 507-549, (1989).
Kramer, G. A. & Barrow, H. G., A Case-Based Approach to the Design of Mechanical Linkages. Artificial Intelligence in Engineering Design, 2, 443-466, (1992).
Kwong, C. K., Smith, C. F. & Lau, W. S., Application of Case Based Reasoning in Injection Moulding. Journal of Materials Processing Technology, 63, 463-467, (1997).
Liu, C. L., A Systematic Conceptual Design of Modular Fixtures. International Journal of Advanced Manufacturing Technology, 9, 217-224, (1994).
Pu, P. & Reschberger, M., Case-based Assembly Planning. Proceedings, Case-based Reasoning Workshop, Madison Hotel, Washington,D.C., May 8-10, 245-254, (1991).
Rong, Y. M. & Zhu, Y. X., Application of Group Technology in Computer-Aided Fixture Design. International Journal of System Automation: Research and Applications, 2, 395-405, (1992).
Schank, R., Dynamic memory: A Theory of Learning in Computers and People. Cambridge University Press, New York, (1982).
Sun, S. H. & Chen, J. L., Applying Case-Based Reasoning in Intelligent Fixture Design system. JSME-ASME Workshop on Design ‘93, Tokyo, Japan June 18-19, (1993).
Sun, S. H. & Chen, J. L., A Modular Fixture Design System Based on Case-Based Reasoning. International Journal of Advanced Manufacturing Technology, 10, 389-395, (1995).
S.H. Sun and J.L. Chen, “A Fixture Design System using Cased-based Reasoning”, Engineering Application Artificial Intelligence, Vol.9, No.5, pp.533-540, (1996).
Tah, J. H. M., Carr, V. & Howes, R., An application of case-based reasoning to the planning of highway bridge construction. Engineering, Construction and Architectural Management, 5(4), 327-338, (1998).
Tiwari, M. K., Rama Kotaiah, K. & Bhatnagar, S., A Case-Based Computer-Aided Process-Planning System for Machining Prismatic Components. International Journal of Advanced Manufacturing Technology, 17(6), 400-411, (2001).
Trappey, J. C. & Liu, C. R., A Literature Survey of Fixture Design Automation. International Journal of Advanced Manufacturing Technology, 5, 240-255, (1990).
Tsatsoulis, C. & Kashyap, R. L., A Case-based System for Process Planning. Robotics and Computer-Integrated Manufacturing, 4(3/4), 557-570, (1988).
Whybrew, K. & Ngoi, B. K. A., Computer Aided Design of Modular Fixture Assembly. International Jourani of Advanced Manufacturing Technology, 7, 267-276, (1992).
Zarley, D. K., A Case-based Process Planner for Small Assemblies. Proceedings, Case-based Reasoning Workshop, Madison Hotel, Washington,D.C., May 8-10, 363-373, (1991).

J. CSME Vol.28, No.6 (2007)

應用於模組化夾具設計之以案例為基礎推理法則內的知識表示與推理方法

孫書煌
崑山科技大學機械工程學系

陳家豪
國立成功大學機械工程學系

摘 要

比起以規則形式的知識庫系統而言，以案例為基礎的知識庫系統是一種更先進的人工智慧輔助系統。以案例為基礎的知識庫系統是透過尋找以往曾解決過的類似案例並加以修正，來對現有的問題提供解決方案。在本研究中即應用了以案例為基礎的法則提出一個著重在工件定位的模組化夾具設計方法。在進行設計時，本方法先以自由度(DOF)為依據，由案例庫中搜尋出與現有設計問題最類似的案例，然後再加以修正以求得最後結果。依據本方法，未來將可以建立一個電腦輔助模組化夾具設計系統。此系統將含有三個子系統，資料庫子系統中儲存多個模組化夾具模組，每個模組各對應其所具有之功能。知識庫子系統儲存工件的定位知識。案例庫子系統儲存以往的設計案例。本方法亦使用 MOP 知識表示法以便能將複雜的知識條理化地儲存進系統中，最後並以一個例子說明本方法的推理過程。

本方法主要著重在案例的評估及修正上，此部分是案例為基礎推理法則中最重要也困難的部分。在工件定位評估中，幾何外形辨識技術是主要的關鍵，由於這部分超出本研究的範圍，將不會在本文中作詳細討論，因此目前本方法僅能對外形較簡單的工件進行處理。