Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

EXPLANATION-BASED MANIPULATOR LEARNING: ACQUISITION OF ASSEMBLING TECHNIQUE THROUGH OBSERVATION

Lei WANG*. Yajie Tian**
Tetsuo Sawaragi***

** Graduate School of Engineering, Kyoto University*
Kyoto, Japan, (Tel: 81-75-753-5044; e-mail: wlei81@gmail.com).
*** Graduate School of Engineering, Kyoto University*
Kyoto, Japan, (Tel: 81-75-753-5044; e-mail: tianyj@po1.dti2.ne.jp).
**** Graduate School of Engineering, Kyoto University*
Kyoto, Japan, (Tel: 81-75-753-5044; e-mail: sawaragi@me.kyoto-u.ac.jp).}

Abstract: This paper describes a robot manipulator system currently under development which learns assembling technique from observation of skilled worker's demonstrations. The system is an application of EBL (Explanation-Based Learning) to the robot manufacturing domain. The system improves the operating capability of the robot manipulator through observing, analyzing the skilled worker's assembling operation, and learning the assembling technique, that is, the tacit knowledge of skilled workers which makes them complete the assembling task more efficiently. The learning process is a knowledge-based deduction approach, requiring sufficient background knowledge, to understand the observed sequence.

1. INTRODUCTION

Today, robotics affects a broad sector of economic activities from automotive and electronics industries to food, recycling, logistics, etc. To remain competitive in the global arena, future manufacturing scenarios throughout all industrial branches will have to combine productivity and flexibility with minimal life-cycle-cost of manufacturing equipment. Thus, it can be expected that manufacturing competence is further concentrated on robot systems which are expected to become a key component in the digital factory of the future. Robot technology development challenges related to the development of robot assistants concern the required intelligent system behavior (Martin Haegele, et al., 2005). Learning is an important one of the underlying relevant functionalities.

This paper focuses on learning the tacit knowledge, that is, the assembling technique of skilled workers who perform the assembling task with the robot manipulator. We are constructing a system which acquires assembly concepts from observation. The system monitors the manipulator operation by a skilled worker, and reads in the commands one by one from the skilled worker's program. Then it searches through the knowledge base to analyze the actions or commands. When it recognizes completion of the goal, the system generalizes the observed manipulator sequence to create a new concept that is to be used for solving similar assembling problem efficiently. The learning process in our system emphasizes human collaboration with automation system. The aim of our system is to explain why the assembling operation of skilled workers is better, and learn the better assembling technique, thus it is able to improve the assembly efficiency of the robot manipulator. Our system does contribute to skill succession even after the skilled workers were retired.

In this paper, the robot manipulator system consists of four parts: the robot arm, the robot controller, a teaching box, and the PC. Figure 1 is a picture of the robot manipulator system but without the PC. The controller sends commands for operation to the robot arm. Only one device is allowed to operate the controller at the same time, even though several devices, such as the teaching box or the PC, are connected to the controller. The robot arm can be operated automatically with reading in the program, or taught by human workers with the teaching box.

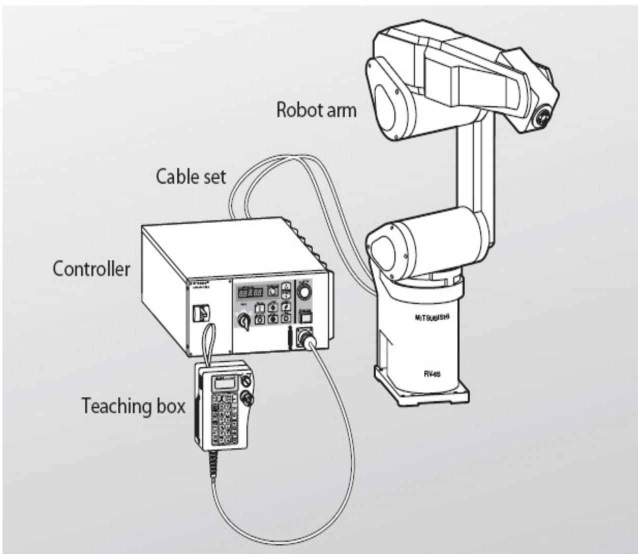

Fig. 1. The Robot Manipulator System

2. LEARNING METHOD & SYSTEM ARCHITECTURE

2.1 Learning Method

Learning from observation is particularly promising for acquiring knowledge from human experts for knowledge-based systems (Gerald DeJong, Raymond Mooney, 1986). Explanation-based systems differ from more traditional correlational machine learning paradigms in that it is possible to acquire useful knowledge on the basis of a single problem-solving episode. For this to be possible, explanation-based systems rely on extensive domain-specific knowledge both while observing the sample problem-solving episode, and in guiding the generalization process (Alberto Maria Segre, Gerald DeJong, 1985).

The Acquiring Robotic Manufacturing Schemata (ARMS) system is an application of EBL to the manipulator domain created by Alberto Segre about 20 years ago. The ARMS domain consists of a prototypical industrial robot arm maneuvering rigid pieces in a finite three-dimensional workspace (Alberto Segre, 1991). Given a set of parts, their location on a tabletop and a description of a mechanical assembly's desired behavior, the AMRS system learns a physical realization of this behavior as well as new operator schemata from observing an expert's solution.

Partially inspired by the ARMS system, the learning method in our system is also based on EBL. However, other than the ARMS system that learns the physical realization and assembly plan to fulfill a mechanical behavior, our system is engaged in analyzing the assembly knowledge hidden in the skilled worker's operation to the robot manipulator, and acquiring the assembling technique by generalizing the reason that makes the skilled worker's assembling operation better. Thus, our system and the ARMS system apply EBL for different targets in the same domain. The goal in our system is to learn how the skilled worker masterly applies the assembling rules, which makes the domain theory in our system is different from the ARMS's. Compared to the ARMS system, our system pays more efforts on studying the operating details in the skilled worker's assembling process.

2.2 System Architecture

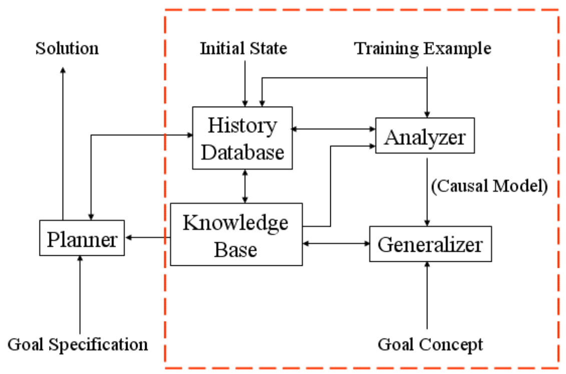

Fig. 2. System Architecture

Figure 2 shows the architecture of our explanation-based manipulator learning system.

There are four inputs to our system: the initial state, the training example, the goal concept, the goal specification.

The initial state is input into the history database, which is a recorder of the world state of the work cell. The history database will update when the world state of work cell is changed after a certain operation. And, the past data of the world state are stored in the database.

The training example, which is an observed skilled worker's assembling operation and is input to the analyzer, is in the forms of a sequence of primitive robot manipulator commands. After reading in the commands one by one, the analyzer explains each command and analyzes the operation sequence while referring to the history database and the knowledge base. When certain command changes the world state, the analyzer will inform the history database to update. The product of the analyzer is a causal model explaining how the tacit assembling technique is implemented within the provided primitive command sequence.

The goal concept is the assembling technique to be learned. The generalizer reads in the goal concept, combines it with the causal model, and constructs an explanation tree by conferring with the knowledge base. Then, the generalizer generalizes the explanation tree, abstracts new knowledge about the goal concept, and stores it into the knowledge.

When the system is used to solve an assembly problem, the assembly task is described as the goal specification and is input into the planner. The planner will search through the knowledge base while consulting the reference world state in the history database, and give a solution to the goal specification if possible.

As showed in the dashed frame in figure 2, this paper concentrates on the learning element of the system, mainly introducing the assembling technique acquisition mechanism with explanation-based learning.

3. REPRESENTATION OF THE LEARNING PROBLEM

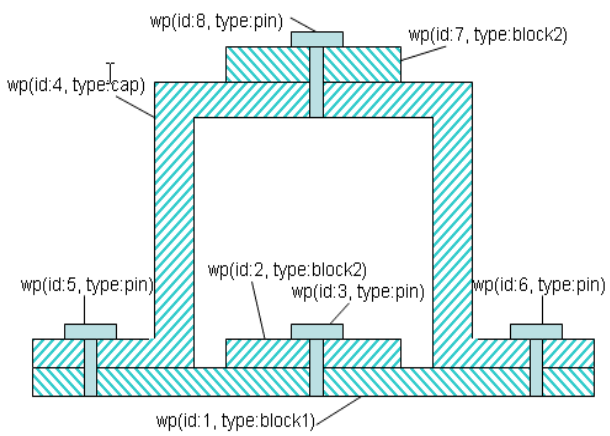

Fig. 3. The Test Assembly

The learning problem in our system is an explanation-based generalization problem, which is defined by Gerald DeJong (Gerald DeJong, Raymond Mooney, 1986) as: given the domain theory, the goal, the initial world state, and the observed operator/state sequence, then the new knowledge that achieves the goal is determined in a general way.

This paper illustrates the learning problem through a test assembly instance. Figure 3 is a sketch of the test assembly. All the workpieces are cylindrical in shape. The assembling sequence is designated as the 'id' number shows. There are four types of workpieces in the test assembly: block1, block2, cap, and pin.

3.1 *Initial State*

Initial state is a specification of the initial world state of the work cell. In our system, the initial state describes the initial position of the workpiece based on the robot coordinate system. The illustration of it is showed as figure 4.

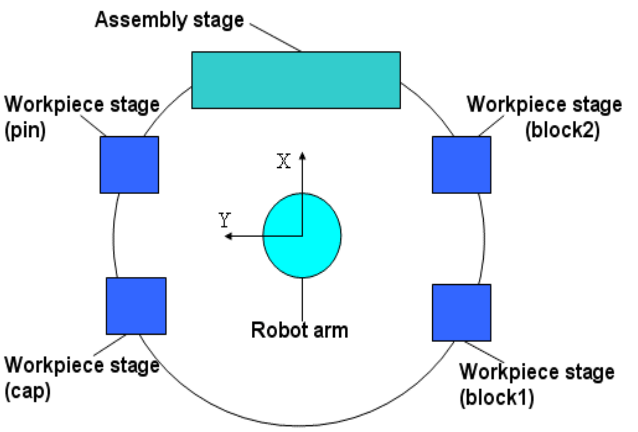

Fig. 4. Initial State of the Work Cell

The robot manipulator is in the center of the work cell. The robot coordinate system is designated as figure 4 shows, and the direction of the Z axis is upward. The workpiece stages and the assembly stage are distributed in a circle around the robot manipulator. The workpieces are placed vertically on the workpiece stages. The assembly stage is the workbench for the assembling task.

The work piece's initial position is defined as:

pos(id:1, type:block1, p(-337,-337, 200)).

It specifies the position of the first assembled workpiece ('id' here refers to the assembling order) and the type of it. The position of a workpiece denotes the center of the underneath surface of the workpiece. It is also the origin of the workpiece coordinate system. All the initial states of the workpieces are defined in the same way as above.

3.2 *Domain Theory*

The domain theory in our system is stored in the knowledge base and can be updated as new knowledge is acquired. It is composed of three parts: the workpiece base, the basic base, and the rules base.

The Workpiece Base

The workpiece base provides the workpiece data, such as its type, catch point, figure, and so on. The workpiece data are indexed and specified according to their types. Catch point is the point at which the robot hand grips the workpiece, and it is designated by the skilled worker according to the type of the workpiece. An instance is showed as the following:

```
wp(
 type:block1,
 catchpoint(0, 0, 2),
[ figure(radius:26, height:4),
 hole1(p(0, 20.5, 4), 1.5),
 hole2(p(0, 0, 4), 1.5),
 hole3(p(0, -20.5, 4), 1.5) ] ).
```

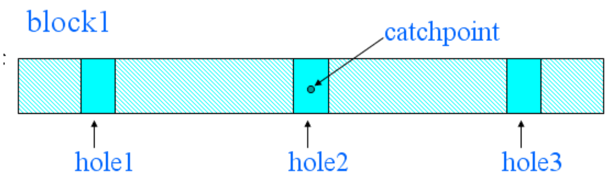

Fig. 5. The Illustration of 'block1'

The workpiece data are based on the workpiece coordinate system, and the origin of the system is the center of the underside of the workpiece. The axes of the workpiece coordinate system are parallel to those of the robot manipulator coordinate system while the directions are the same. As figure 5 shows, in the above instance, the origin of the coordinate system of block1 is the center of the underneath round surface of block1. The position coordinate of the hole is the center of the upper round surface of the hole, and the radius of the hole is given. Other workpieces' data are given in the similar way.

The Basic Base

The training example is composed of the robot commands and parameters only. To analyze the training example, the system must understand the meanings of every command and its parameters. Thus, the basic base supplies the commands of the robot arm with the explanation of their function. The command here has a name with parameters and a body. The body is almost the same as the primitive robot command. The parameters of the name are pointers that relate the command with the workpiece and world state of the work cell. The name shows the purpose using the command.

```
approachp(P, D):- mov(P,-D).
```

The above is an example of the item stored in the basic base. The primitive command is 'MOV P,-D'. 'mov(P,-D)' is the result of the case changer in the analyzer. This item explains that the purpose of the command is to approach point 'P' with the distance 'D' above the point, which is moving to a position retracted 'D' from 'P' in the robot hand direction. In different situations, the same command may have different explanations, in which the proper one is decided depending on the temporal world state and the command's order in the program. For instance, 'MOV P' means that moves to point 'P' from current position 'P_CURR'. When P.X=P_CURR.X and P.Y = P_CURR.Y, if P.Z>P_CURR.X, then it will be explained as 'raiseto(P): mov(P).'; else if P.Z<P_CURR.Z, then the command will be explained as 'lowerto(P): mov(P)'.

The Rules Base

When a worker carries out the assembling task, he/she must obey some rules even though he/she did not notice the rules he/she is using. Different workers use different rules in different manners, which is why the skilled worker performs the assembling better than the green hands (i.e., novice workers). The rules base contains the rules in the assembling task, including both previously encoded rules and new rules learned from the training example.

There are two types of the rules: inference rule and control rule. Inference rule deals with the world state information in the history database and the workpiece base to calculate the positions of important points. For example, the seizing point position of workpieces in the robot coordinate system. Control rule is the operating rule that should be obeyed in the assembling process. The control rule can be classified in to general control rule and soft control rule. General rule is common sense or general knowledge that is to be paid attention to in assembling. For example, the approach point must be higher than the upper surface of the target workpiece to avoid collision. Another example is defining the essential or indispensable actions that must be included in a complicated operation concept to complete a certain assembling task. As the general rule assures the completion of the assembling task, the soft rule is the rule to improve or optimize assembling operation. For instance, it specifies when it is better to change the speed.

3.3 Goal Concept

As mentioned above, there is a great difference between our system and the ARMS system in that the two systems have different learning targets. Our system is interested in the combination of the assembly rules that the skilled worker employs while the ARMS system is limited in learning a physical realization and two new operator schemata for fulfilling a mechanical function. In an actual manufacturing process, the product is designed by other experts, and the robot manipulator is only in charge of assembling the product with given workpieces. Therefore, the goal concept in our system is the assembling technique to be learned. Our system focuses on learning skilled workers' assembling technique. The technique is the way that how the expert applies the rules to accomplish the assembling task in less time and with less or no error.

In the assembly process, there are three typical phases: picking up the workpiece, moving the workpiece along a certain path, placing the workpiece at the target position. Therefore, the goal concepts in our system are efficient 'pickup' and 'place' movements and optimized moving 'path' to be learned. There two criteria for evaluating 'efficient': speed and stability. The former requests a less assembly time, and the latter requires a less error in the assembling operation. There is a contradiction between the speed and stability criteria, and it is difficult to mediate the contraction. For example, increasing the speed may cause instability, and often the stability will have the priority compared with the speed. To different workpieces and world states, two different 'efficient' criteria are to be emphasized. The skilled worker has the experience to solve the speed and stability contradiction properly.

3.4 Training Example

The training example in our system is a robot manipulator program made by the skilled workers. It can be a ready-made program or a program being input by the skilled workers using the teaching box. So, our system can learn both in offline from the skilled worker's program and in online from the skilled worker's operation. To explain the knowledge that our system learns, two training examples are given in this paper as below.

The first one is an instance how the skilled worker 'pickup' the first assembled workpiece block1 in the test assembly above:

```
10 OVRD 20       'Set the speed as 20% of the Maximum
20 MOV P1,-15    'Move to a position retracted 15mm
                  from P1 in the robot hand direction
30 MVS P1        'Move to P1 in line
40 HCLOSE 1      'Close hand
50 MOV P1,-15
```

The second one is an instance how the skilled worker 'pickup' the third assembled workpiece pin in the test assembly above:

```
10 OVRD 20
20 MOV P5,-10
30 OVRD 5
40 MVS P5
50 DLY 0.5       'Delay 0.5 second
60 HCLOSE 1
70 DLY 0.5
80 OVRD 20
```

90 MOV P5,-10

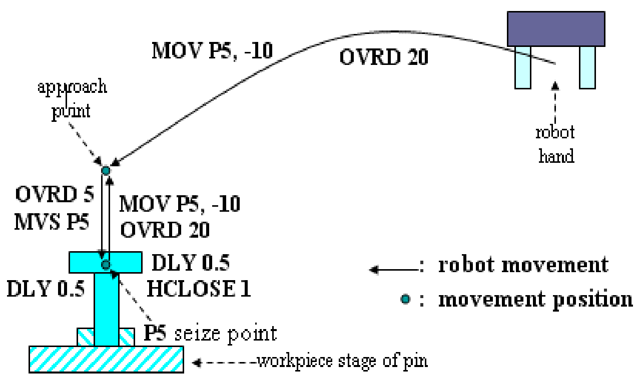

Fig. 6. Pickup 'pin'

Figure 6 is the illustration of the second training example. Since the robot manipulator is not allowed to run in maximum speed, the speed is set to 20% of the maximum. The speed slows to 5% of the maximum when the robot hand moves in line from the approach point to the seize point. After delaying 0.5 second, the robot hand closes, and delays 0.5 second. Then accelerating to 20% of the maximum speed, the robot hand retreats to the approach point. The pickup of block1 is different from the pickup of pin in the following: in the process of block1 pickup, the speed is faster, no slow down before getting to the seize point and no time delay after closing the hand. The reason for these differences is given in the next section.

4. KNOWLEDGE ACQUISITION

The knowledge acquisition mechanism in our system is based on EBL. As Charles Elkan and Alberto Segre pointed out (Charles Elkan, Alberto Segre, 1989): In order to support the application of EBL techniques, a knowledge-representation formalism must meet certain conditions. Most obviously, the formalism must be declarative so that it could allow the construction of explanations. Less obviously, knowledge must be represented in a way that supports the construction of explanations that are capable of being generalized. Knowledge should be expressed in terms of individuals and relationship between them, so that specific individuals in explanations can be replaced by generic individuals. If knowledge is represented as logical facts and rules, then explanations are simply equivalent to proofs. Moreover, all proofs have the same tree structures, because a fact can only be proven by matching it with the consequent of a rule, and then recursively proving the antecedents of the rule.

In this paper, the knowledge to be acquired is the assembling technique of the skilled worker. In other words, the knowledge is the rules followed by the skilled worker in his assembling operation. To obtain this kind of knowledge, our system takes certain efficient assembling operation as the goal concept, and attempts the explanation construction for the training example, which is a proof tree that explains why the training example is a successful instance of the goal concept, and then extracts the assembling technique as a new rule from the constructed explanation. Therefore, the explanation construction is the basis for knowledge acquisition.

According to the training examples above, the explanation construction mechanism in our system is illustrated with the 'pickup' examples.

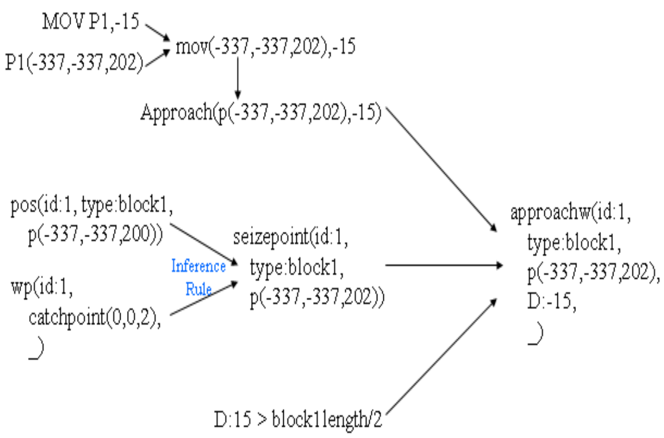

Fig. 7. Explanation of Robot Command

As shown in figure 7, our system explains the robot commands read in first. From the 'pickup block1' training example, the system reads in the command 'MOV P1,-15' and the parameter P1(-337,-337,202). The case changer changes the case of the command and the analyzer understands the purpose using this command as approaching point p(-337,-337,202) with a distance 15mm retracted from the point in the robot hand direction. With the initial state data 'pos(id:1, type:block1, p(-337,-337, 200)).' in the history database and workpiece data 'wp(id:1, catchpoint(0,0,2),_)' from the workpiece base in the knowledge base, the system gets the seize point of the first assembling workpiece block1 according to the inference rule. As the distance 'D:15' is bigger than half length of block1, which means the approach position is higher than the upside of the approaching target and is a general state control rule, and the calculated seize point is the same as the target point read from the program, the system recognizes according to the general operating control rule that approaching the first assembling workpiece block1 is realized.

While explaining for every read in command, the analyzer stacks them in a list one after one. As the blue lines in figure 8 show, there is a general control rule that 'MOV P,-D', 'MOV P', 'HCLOSE 1' and 'MOV P,-D' are indispensable actions in the complicated movement 'pickup'. When all the four indispensable commands are detected in the list, the analyzer realizes that the list of commands is 'pickup' and gets a causal model for the commands.

The generalizer reads in the goal concept and constructs the explanation tree for the goal concept with the causal model. As shown in figure 9, the explanation tree explains how the skilled worker uses the other commands besides the indispensable commands of 'pickup' to make the 'pickup pin' operation more stable. Then, the generalizer extracts the assembling knowledge as a new rule from the top-level subgoals of the explanation tree.

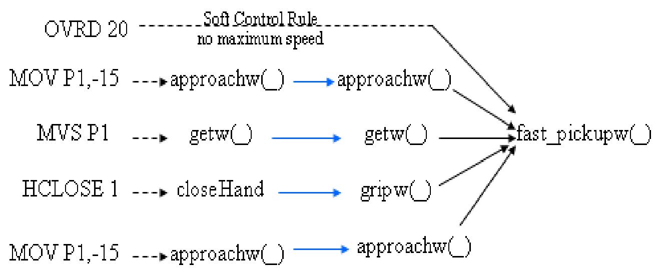

Fig. 8. Explanation Tree for 'pickup block1'

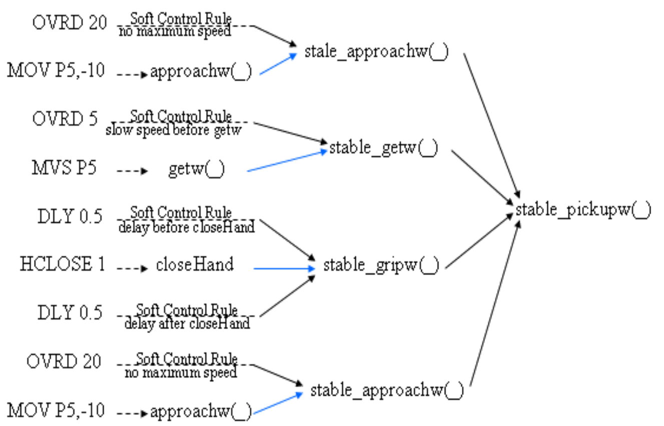

Fig. 9. Explanation Tree for 'pickup pin'

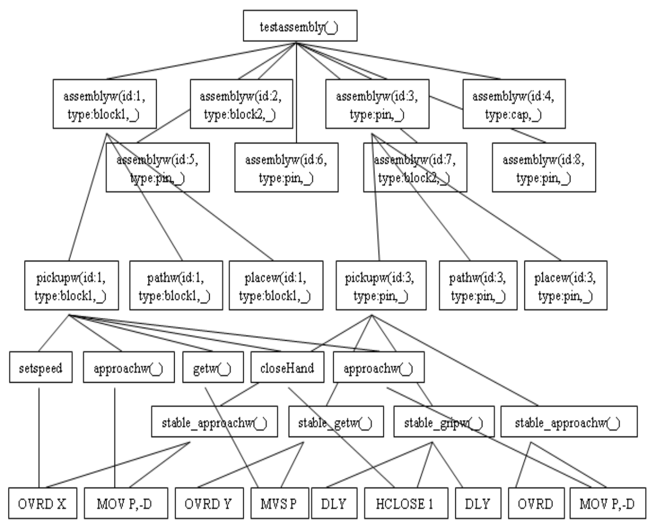

Fig. 10. Partial Explanation Tree for the Test

Figure 8 and figure 9 show the explanation trees of goal concepts 'fast pickup' and 'stable pickup' for the 'pickup block1' training example and 'pickup pin' one, respectively. Dashed arrows and lines mean the detailed explanations for the robot commands as shown in figure 6 are overleapt. Corresponding to different types of workpieces, different assembling technique is employed. As different precision degrees are required to different workpieces and picking up block1 is easier than picking up pin, the assembling technique used in the two training examples has different preferences. Pickup block1 does not have to require high precision, higher speed is its preference. So in pickup block1, only the soft control rule that the operation should not run in maximum speed is appeared. On the other hand, to pickup pin, precision is emphasized; therefore, the measures such as slow-down before grip and delay before and after grip making the pickup more stable are used. Figure 10 is a part of the explanation tree for the test assembly. From this explanation tree, the new rule extracted is a rule classification that to a certain type of workpiece some assembling technique preference is emphasized.

5. CONCLUSIONS

This paper presents a human-robot collaboration system that makes the robot have the ability to learn human expert knowledge by observation. Our system shows the operationality for applying EBL on the manufacture domain and provides a new method. Our future research direction will be on making our system have the ability to quantificationally analyze and evaluate the efficiency of the assembling operating as marking the application of the rules and applying our system on assembly error recovery problem.

REFERENCES

Alberto Segre (1991). Learning how to plan. *Robotics and Autonomous Systems,* 8, 93-111.

Alberto Maria Segre, Gerald DeJong (1985). Explanation-Based Manipulator Learning: Acquisition of Planning Ability Through Observation. *Robotics and Automation. Proceedings. 1985 IEEE International Conference,* 2, 555-560.

Charles Elkan, Alberto Segre (1989). Not the Last Word on EBL Algorithms. In: *Technical Report 89-1010.* 1-2. Department of Computer Science, Cornell University, Ithaca, NY.

Gerald DeJong, Raymond Mooney (1986). Explanation-Based Learning: An Alternative View. *Machine Learning,* 1, 145-176.

Martin Haegele, Tomas Skordas, Stefan Sagert, Rainer Bischoff, Torgny Brogardh, Manfred Dresselhaus (2005). White paper – Industrial Robot Automation. *European Robotics Network,* 1, 1-3.