XML-based Representation in a CBR System for Fixture Design

Fan Liqing[1] and A. Senthil Kumar[2]

[1]National University of Singapore, mpeflq@nus.edu.sg
[2]National University of Singapore, mpeask@nus.edu.sg

ABSTRACT

Fixture design is a complex process and largely relies on a designer's experience to solve current problem. Case-Based Reasoning (CBR) provides a promising methodology for solving design problems in this complex domain based on the idea that past problem-solving experiences can be reused and learned from in solving new problems. In order to realize a CBR design system to assist designers to solve fixture design problems in the Internet-enabled environment, case representation is described using XML (eXtensible Markup Language) format in this paper. The XML-based representation structure using UML (Unified Modeling Language) notation is illustrated and discussed. A design process with an example is presented in the developed system applied with XML-based representation. The Internet-enabled CBR system for fixture design is developed and implemented in client-server mode architecture.

Keywords: Case-based reasoning, fixture design, case representation, XML, UML.

1. INTRODUCTION

Fixtures are devices which are designed to repeatedly and consistently maintain the orientation of a workpiece during machining, assembling, welding, inspection, *etc.* [1]. They are an essential part of manufacturing production. As part of manufacturing tooling, fixture design makes significant contributions to the production time and cost in daily production. Flexible fixtures play important roles in modern flexible manufacturing systems (FMS) as well as computer-integrated manufacturing system (CIMS).

Fixture design is a highly complex process because it must consider the workpiece, the cutting tools and the machining environment. In addition to these aspects mentioned above, Senthil Kumar *et. al.* [2] illustrated all factors considered in fixture design that are categorized into three basic constraints, including technical, economical and resource availability.

Fixture design is also experience-based. Designers prefer to use previous designs because they save time and effort and use the concepts have proven effective in previous situations. In the design of fixture, based on all the information pertinent to the product as given by the engineering specification and the process sheet, a tool designer configures a fixture setup appropriate to the workpiece depending on his/her experience of fixturing a similar product [3]. Meanwhile, the selection of surfaces on the workpiece and fixture elements for locating and clamping during machining is flexible and largely relies on the prior experience of the designer.

In today's product development context, the design of products is sub-contracted out to other firms. This creates a scenario where the designers and manufacturing engineers may be globally dispersed. Implementing distributed manufacturing systems would offer rapid manufacturing capacity. An Internet-enabled manufacturing system which has the ability to cooperatively design not only saves costs and time, but also creates a seamless collaborative manufacturing environment to resolve problems in real-time. Therefore, to realize a collaborative functional fixture design system, care must be taken such that the design activity can be performed on the internet.

For these reasons, a case-based reasoning (CBR) approach which organizes previous experiences as cases to solve new problem is attempted in this work. Except CBR approach, a new paradigm in Computer Integrated Manufacturing (CIM), namely Internet-based Manufacturing, is also adopted in this work. The objective of this research is to develope a system where design of fixture is done in case-based reasoning environment over the Internet.

Case-based reasoning (CBR) is an artificial intelligence technique which is a general problem solving method using past experiences to solve novel problem [4]. The major consideration in a case-based reasoning approach to design is as following:

- Representation of design cases;
- Indexing and retrieving similar cases from case-base;
- Adapting the retrieved cases for current design.

Case representation is generally regarded as one of the most important issues and is crucial to success of case-based reasoning system. In this distributed fixture design system, case representation is concerned with two areas of research. First, as a fundamental aspect of the CBR paradigm, case is not only described as problem, solution and outcome in fixture design domain, but also represented in a manner that allows efficient retrieval, easy maintenance and transmission over network. Second, a case representation can be devised as an open standard in fixture design domain and exchange information with other computer-aided manufacturing systems.

Although Case-Based Reasoning is a knowledge-rich methodology, there is no standard means of representing the case in a case base. CASUEL, a Common Case Representation language, is developed in the INRECA project [5]. CASUEL is a flexible, object-oriented frame-like language for storing and exchanging descriptive models and case libraries in ASCII files. M. Marefat and J. Britanik [6] present an object-oriented model representation in case-based process planning for 3D prismatic parts. The model consists of the feature-based part information model, the process plan information model and tools knowledge concerned with process plan. These two are only developed on local computer but not over network. Hayes *et. al.* [7] have developed an XML-based Case Mark-Up Language (CBML) for e-commerce through Internet. This language allows user to make the formal definition of the structure of our cases completely independent of the application code. CBML also makes system for industry extensibility, easy of reuse and interoperability. Although CBML supports complex case representation, such as hierarchical cases and object-oriented cases, as well as the commonly used flat vector format, it does not suit for Internet-based CAD applications in which sometimes a large amount data are required to transfer over the Internet.

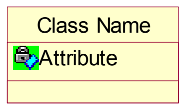
In fixture design domain, vast amount of research works is reported on developing computer aided system. Various approaches have been attempted in fixture design, i.e. Rule-based expert system, Genetic Algorithm, Multi-agent Approach, Machine Learning, Geometric Analysis. Some have been integrated and hybrid systems have been developed. However, there is no information exchange between these systems and hence the fixture design output information can not be shared with other manufacturing domain. Jerard and Ryon [8] developed Numerical Control Markup Language (NCML) based on XML as data exchange format to permit buyers and sellers of custom machined parts to conduct e-commerce over the Internet. In this language, information regarding workpiece, setups, tools, and tolerance is provided to allow users to judge the manufacturability of a part. However, NCML does not provide any information about fixture.

This paper is primarily focused on representing XML-based fixture design cases using Unified Modeling Language (UML) [13] for the developing case-based reasoning system. XML is a description language that supports meta-data description for particular domains and these meta-descriptions allow applications to interpret data marked up according to this format. Besides being a recommendation of W3C, XML [12] is used in our work because it is neutral, platform-independent, and flexible and allows structured data in Web applications. This means we can collect data from various place and exchange structure rich data in various applications over existing network protocol. Today, there is also ongoing development that will include a query language in the XML standard. UML was chosen because it is a popular method for designing software and has proven to be valuable for data modeling [11].

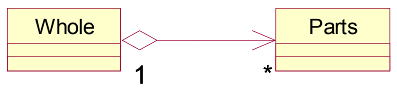
2. XML-BASED REPRESENTATION USING UML

This section illustrated XML-based fixture design cases and modeling XML file represented in the UML. The graphical UML notations that are used to represent XML model are summarized in Fig. 1. The UML depicts a class as a rectangle with its name at the top and its attributes in the middle (Fig. 1a). Relationships between classes are depicted as lines that link them. The inheritance relationship is identified by a hollow triangle at the end of line that indicates the superclass (Fig. 1c), while the aggregation relationship is identified by a diamond at one end of line that indicates the owning class and by an arrow at the other end (Fig. 1b). Multiplicity indicates how many instances of one class are related to a single instance of another class at a given point in time. For example, Fig. 1c means one class whole has many parts. The multiplicity notations are located at the end of a link. The meaning of them is shown in Fig.1d.

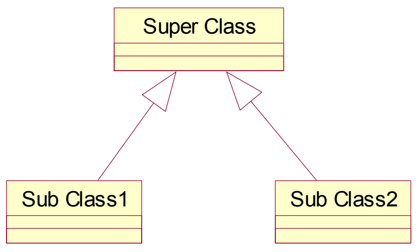
(a) Class:



(b) Aggregation:



(c) Generalization (Inheritance):



(d) Multiplicity of Association:

*	many
0	zero
1	one
0..*	zero or more (optional)
1..*	one or more
0..1	zero or one

Fig. 1 Unified modeling language notation

2.1 Case Structure

To identify the content of a design case, it is important to find out how design and design requirements are represented in practice. Requirements for a fixture are the workpiece to be fixtured, manufacturing resources and fixture elements provided. For modular fixture design, design outcomes are fixture planning that deals with overall design concepts and fixture layout that produces a spatial layout of the fixture. Therefore, the representation of a case in fixture design is divided into three parts: part representation, setup representation and fixture representation. A workpiece is described using feature-based geometry and material properties. The machining features are grouped according to their orientations and machining constrains into setups. In one setup, only one fixture is associated with it. By this way, the three parts are linked (Fig. 2). The setup information including manufacturing resources and fixture plan is usually provided by process planners who can access the system and co-operate with fixture designers, while fixture design which includes fixture layout and fixture configuration is final solution for the requirement.

2.2 Part Representation

Part design is the input to the fixture design system. In case representation, its role is similar to the problem description. It is represented not only as a part stored in case base but also as a new case to be solved. Part representation contains not only geometric shape but also material property.

The geometric information is composed of a set of features, their interactions, faces, points for clamping, locating and supporting, as well as engineering information (tolerance, dimensions, etc.) pertaining to the features. Feature class represents the complete machining area in a workpiece by showing the size and type of features present. The features include the following classes, i.e. boss, pocket, hole, slot, step, and each class can be classified further. Fig. 3 shows the part representation using UML notation.

Inheritance is exploited in representing the knowledge of the features. In Fig. 4, Feature class is an abstract class only acting as interface for basic shape feature classes, i.e. Hole, Slot, etc. These subclasses inherit the common attributes from Feature class and other features are created from basic shape features. For simplicity, subclasses of Tolerance class and basic feature shape classes are not displayed in Fig. 3. However, in order to clarify the inheritance, the Hole class and its subclasses are shown in Fig. 4. In Fig. 4, the classes in the third level refer to the implementation class; the classes in the first two levels are Meta-class. In the class "Hole", the class "Couterbore Through Hole" is inherited from its super class "ThroughHole" which is inherited from metaclass "Hole". The class "Counterbore Through Hole" not only includes its own attributes "CounterDiameter" and "CounterDepth", but also inherited the attribute "Diameter" which represents the diameter of an inner hole from its super class "ThroughHole".

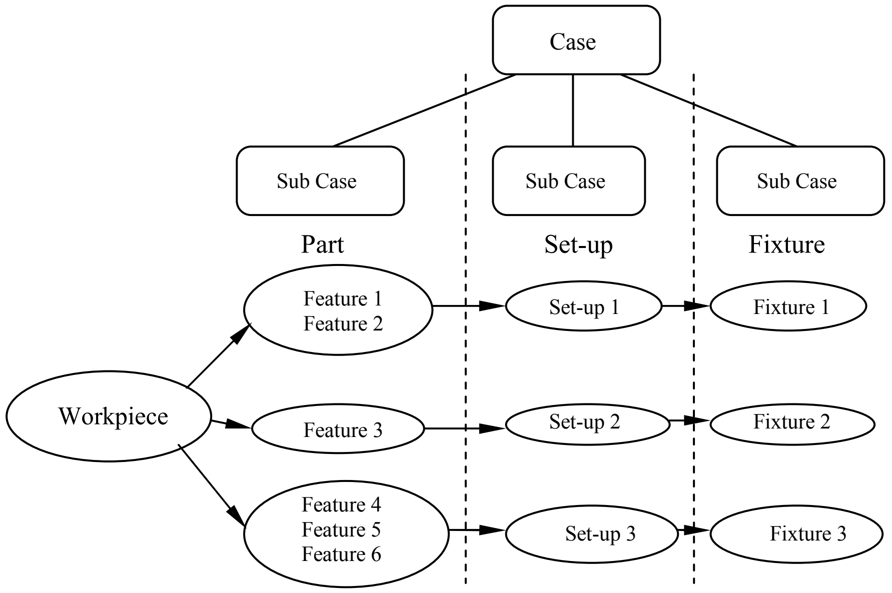


Fig. 2. Case structure

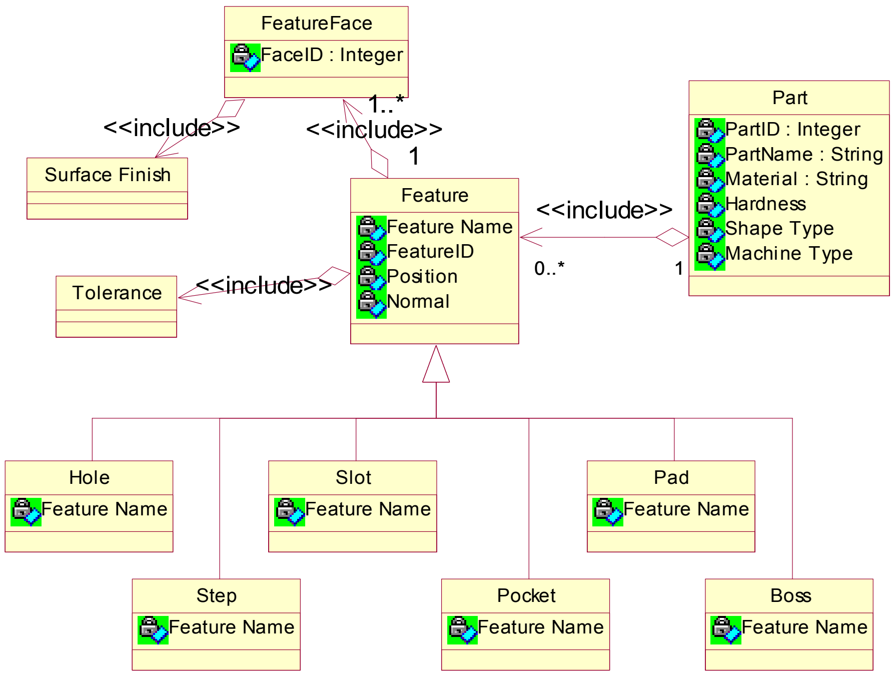


Fig. 3. Feature representation in UML notation

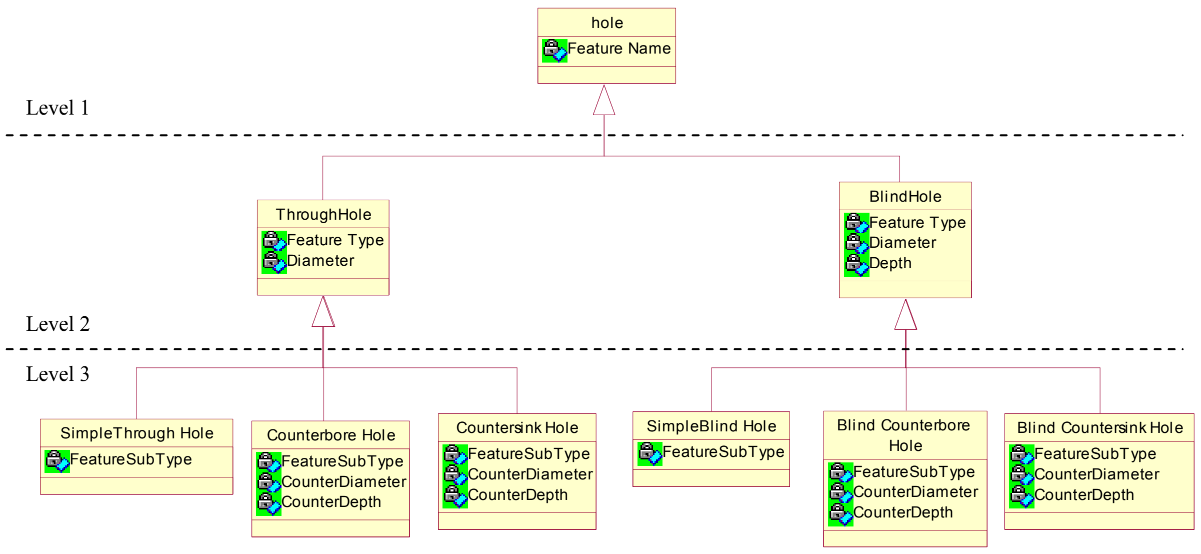


Fig. 4. Inheritance in the Hole class

2.3 Fixture Design Representation

A case obviously includes the solution to the problem. In fixture design domain, the solution is usually a complete fixture layout configuration, which includes details on the work-piece, base plate, supporting surfaces, points and elements, locating surfaces, points and elements and clamping surfaces, points and elements. This information can be provided to fixture analysis and process planning. In the developing system, the focus is on modular fixture that inherits from fixture. The modular fixture is composed of fixture elements that can be mainly categorized into four functional units: Base plate, Locator, Support and Clamp class. Locating elements, supporting elements and clamping elements are located and assembled on a base plate through “LocatorPointID”, “SupportPointID” and “ClampPointID” respectively in a fixture. Accessory elements associated with the four functional units are assembled in a fixture if necessary. These fixture elements contact with workpiece through assembling Faces. The detailed fixture design representation is shown in Fig. 5.

2.4 Setup Representations

Set-up planning is one of the important issues in process planning and requires extensive experience. A set-up is defined as a task of determining a sequence of features and the number of set-ups required in fixturing configurations. Set-up planning information enables the consideration of the fixture design configuration, positioning locators, clamps and supports. Setup links the workpiece and its fixture designs together, and contains information that includes machining features in workpiece, workpiece orientation, fixture planning, support faces, locating faces, and clamping faces. One workpiece may require many set-ups, while each setup only requires one fixture (Fig. 6)

3. SYSTEM IMPLEMENTATION AND CASE STUDY

The developed system together with the representation described above is implemented in three-tier client-server architecture and uses Java as the programming language, Java3D as the graphics API and XML as the information exchange file format so as to provide the flexibility of interoperability on a variety of operating platforms. The system consists of three main parts: the client, the server and the case library. The detailed implementation refers to [9].

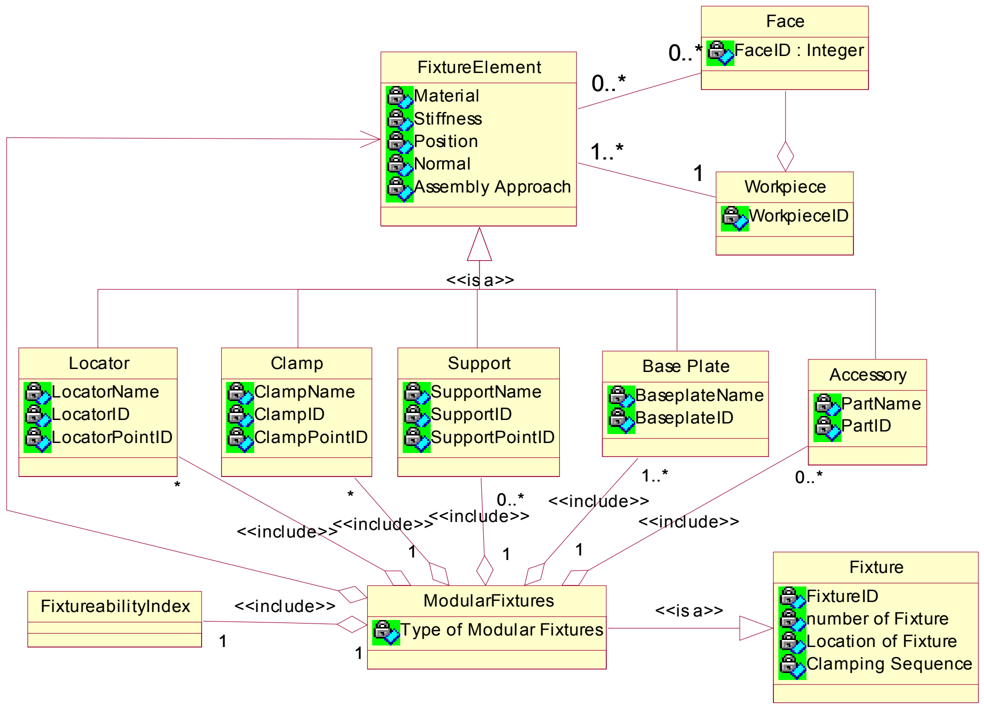


Fig. 5. Fixture Design Representation Model in UML Notation

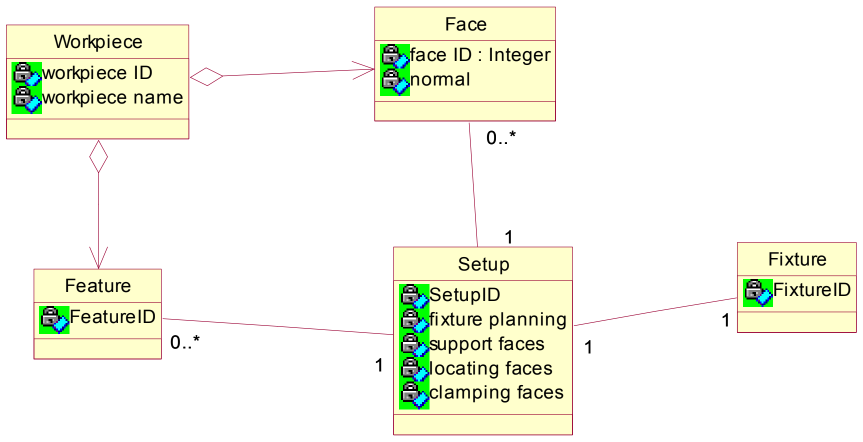


Fig. 6. Setups in UML notation

The server side consists of back-end CBR engine, Parasolid model kernel, server classes, data compression, and the Remote Method Invocation (RMI) interface. The back-end CBR engine includes case indexing and case retrieval module. When server side receives and processes the request from the client end, CBR engine searches and matches new case with cases in Case Library, retrieves desired candidates, and sends them back to original client. Current proposal is to perform CBR retrieval on the server-side, because case retrieval module including indexing, searching and matching cases in case library can be computationally intense process, especially for a huge case library. We do so in order to make high-end server to handle heavy computing.

The client application handles visualization, CBR front engine, RMI interface to server and XML parser. The front-end CBR engine is composed of case selection, case storage, and case adaptation & repair module. It is integrated with 3D solid modeling so that a user can visualize information from case-based reasoning via the models. Since a distributed CBR system could potentially have many clients connecting simultaneously, case retrieval is computationally intense process, and current PC performance at client-end is increasingly well, distributing as much work as possible to the client side could possibly speed up the reply time for all users on the network. Under these circumstances, the trend seems that the thin-clients are growing fatter or have the ability to put on the weight. So that, the front-end CBR engine is put in client-side.

Case library developed using Apache Xindice XML server [10] includes fixture design, workpiece feature information and manufacturing information which are in XML file format. The benefit of a native solution is that you do not have to worry about mapping your XML to some other data structure. You just insert the data as XML and retrieve it as XML. You also gain a lot of flexibility through the semi-structured nature of XML and the schema independent model used by Xindice.

The proposed system using CBR is also following certain sequential steps; hence the user can follow the design stage and interact with the system. The detailed work flow is shown in Fig. 7. The first three steps, new workpiece import, feature extraction and setup information setting, are considered as design-problem specification. The setup information may be provided by a process planner.

Fig. 8 shows a workpiece whose shape type is prismatic is loaded in the system and the machining features are extracted. The feature information is shown in XML file. We can know the part name is “model3” from <PARTNAME> tag and part id is 32 from <PARTID> tag. The feature information is provided in the <FEATURE> tag. Fig. 8 also shows Slot class, its subclass and its attributes in the <FEATURE> tag.

Fig. 9 shows XML schema for the imported workpiece setup information. The XML file may be retrieved and displayed on the Internet Browser. For this workpiece, only one setup is applicable; we can know that the setup ID is “101” from <SETUPID> tag, the fixture planning is “3-2-1” from <FIXTUREPLANNING> tag, and the machine type for manufacturing operation “Mill, Drill” is “Vertical Machine” from <MACHINETYPE> tag.

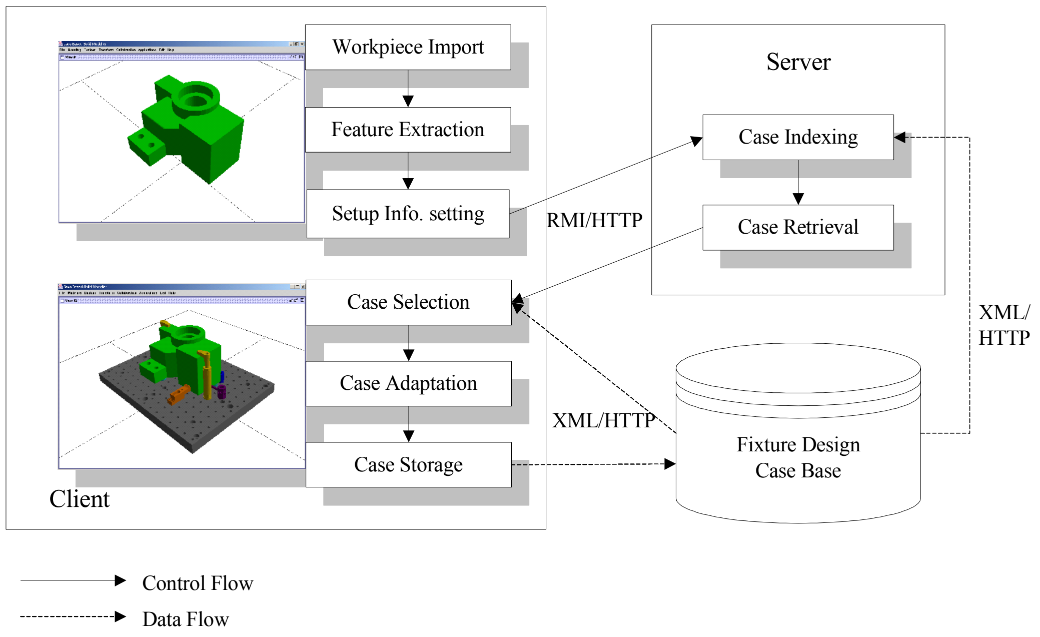


Fig. 7. Work flow of CBR on fixture design

The process of case indexing and case retrieval including case similarity matching and ranking reside in server. The input part compares with the models in the case-base. The compared models are ranked based on the similarity values. The ranked cases are returned back to client and the fixture designs are displayed on the client for selection (Fig. 10). A desired design can be chosen from ranked list for adaptation. After the selected design case is adapted, the new design case shown in Fig.11 is directly stored to the case library in case storage process.

Fig. 11 shows the complete fixture design and its corresponding XML file describing the detailed fixture design. From <SELECTEDWORKPIECENAME> tag, we can know the workpiece name is model4. The locator information is in the <LOCATOR> tag. The name of one of the locators in the fixture design is BJ400-12050 and its associated face ID with workpiece is 267.

4. CONCLUSION

Case representation is one of the most important research issues in developing case-based reasoning system. This paper mainly discusses the case representation in XML format in the Internet environment. XML-based representation developed in fixture design domain is illustrated in the aspect of part representation, setup representation, and fixture design representation using UML notation respectively. Through the case representation, the new problem and the solution in the area of fixture design are described in XML schema. One reason XML schema is adopted to represent case is that it can facilitate network transmission, case retrieval and case storage over the Internet. Another reason is that it can be regarded as open standard to exchange manufacturing information with other CAD/CAM systems. Based on this XML-based representation, the distributed CBR system for fixture design is implemented and developed.

5. REFERENCES

[1] Nee, A.Y.C., Whybrew, K. and Senthil Kumar A. (1995). *Advanced fixture design for FMS,* Springer-Verlag, c1995.

[2] Senthil Kumar, A. and Nee, A.Y.C. (1995), “A framework for a variant fixture design system using case-based reasoning technique”, *Manufacturing Science and Engineering; ASME, MED,* Vol. 2-1, pp. 763-775.

[3] Nnaji, B.O. ; Lyu, P. (1990), Rules for an expert fixturing system on a CAD screen using flexible fixtures, *Journal of Intelligent Manufacturing*, Volume 1, Issue 1, 1990, Pages 31-48

[4] Kolodner, J. L. (1993). *Case-Based Reasoning*. Morgan Kaufmann.

[5] INRECA consortium (1994). “CASUEL: A Common Case Representation Language”, at http://wwwagr.informatik.uni-kl.de/~bergmann/casuel/

[6] Marefat, M.; and J. Britanik. (1997), Case-based process planning using an object-oriented model representation, *Robotics & Computer-Integrated Manufacturing* 13(3):229-251.

[7] Hayes, C. & Cunningham, P. (1999) Shaping a CBR View with XML. *Proceedings of the Third International Conference on Case-Based Reasoning, ICCBR’99,* Seeon Monastery, Germany. LNCS Vol. 1650. Althoff, K.-D., Bergmann, R.,Branting, L.K. (Eds.) Springer-Verlag Berlin/Heidelberg 1999, pp.468-481

[8] Robert B. Jerard and Okhyn Ryon, "NCML: A Data Exchange Format for Internet Based Machining," *2002 Japan USA Symposium on Flexible Automation (JUSFA),* July 15-17, 2002 Hiroshima, Japan.

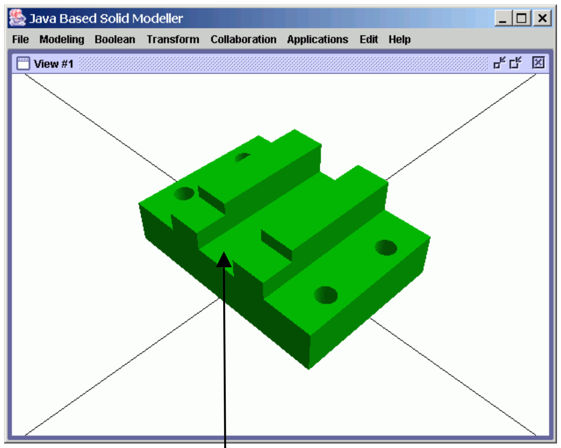
[9] Fan, L.Q.; A. Senthil Kumar and F. Mervyn; “The Development of a Distributed Case-Based Fixture Design System”, *2004 Japan USA Symposium on Flexible Automation (JUSFA),* July 19-21, 2004, Denver, Colorado, USA.

[10] Xindice, (2002), The Apache Software Foundation, at http://xml.apache.org/xindice/

[11] David Carlson, (2001), *Modeling XML applications with UML: practical e-business applications*. Addison-Wesley, c2001.

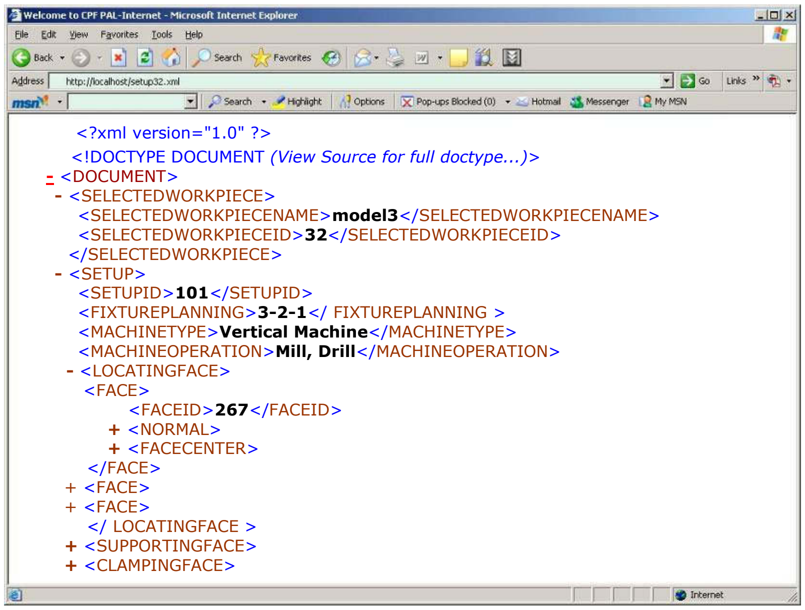
[12] http://www.w3.org/XML/

[13] http://www.uml.org/


Java Based Solid Modeller
File Modeling Boolean Transform Collaboration Applications Edit Help
View #1
Slot

```
    <?xml version="1.0" ?>
-<PARTDOCUMENT>
-<PART>
    <PARTNAME>model3</PARTNAME>
    <PARTID>32</PARTID>
    <ShapeType>Prismatic</ShapeType>
    <Material>Cast Iron</Material>
    <HeatTreatment>Normalizing</HeatTreatment>
    <MachineType>Vertical Machine</MachineType>
   +<ProfileSize>
   +<ProfileFace>
 </PART>
.
.
.
- <FEATURE>
   - <Feature>
       <FeatureName>Slot</FeatureName>
       <FeatureType>ThroughSlot</FeatureType>
       <FeatureSubType>Rectangular</FeatureSubType>
     </Feature>
    -<Dimension>
        <Width>0.03</Width>
        <Depth>0.015</Depth>
     </Dimension>
   + <Position>
   + <Normal>
   + <FeatureFace>
  </FEATURE>
+ <FEATURE>
```

Fig. 8. Loaded workpiece and its feature XML schema


Welcome to CPF PAL-Internet - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/setup32.xml
Go Links
Search Highlight Options Pop-ups Blocked (0) Hotmail Messenger My MSN
Internet

```
    <?xml version="1.0" ?>
    <!DOCTYPE DOCUMENT (View Source for full doctype...)>
 - <DOCUMENT>
 - <SELECTEDWORKPIECE>
     <SELECTEDWORKPIECENAME>model3</SELECTEDWORKPIECENAME>
     <SELECTEDWORKPIECEID>32</SELECTEDWORKPIECEID>
   </SELECTEDWORKPIECE>
 - <SETUP>
     <SETUPID>101</SETUPID>
     <FIXTUREPLANNING>3-2-1</ FIXTUREPLANNING >
     <MACHINETYPE>Vertical Machine</MACHINETYPE>
     <MACHINEOPERATION>Mill, Drill</MACHINEOPERATION>
   - <LOCATINGFACE>
       <FACE>
           <FACEID>267</FACEID>
         + <NORMAL>
         + <FACECENTER>
       </FACE>
   + <FACE>
   + <FACE>
       </ LOCATINGFACE >
   + <SUPPORTINGFACE>
   + <CLAMPINGFACE>
```

Fig. 9. XML schema for setup

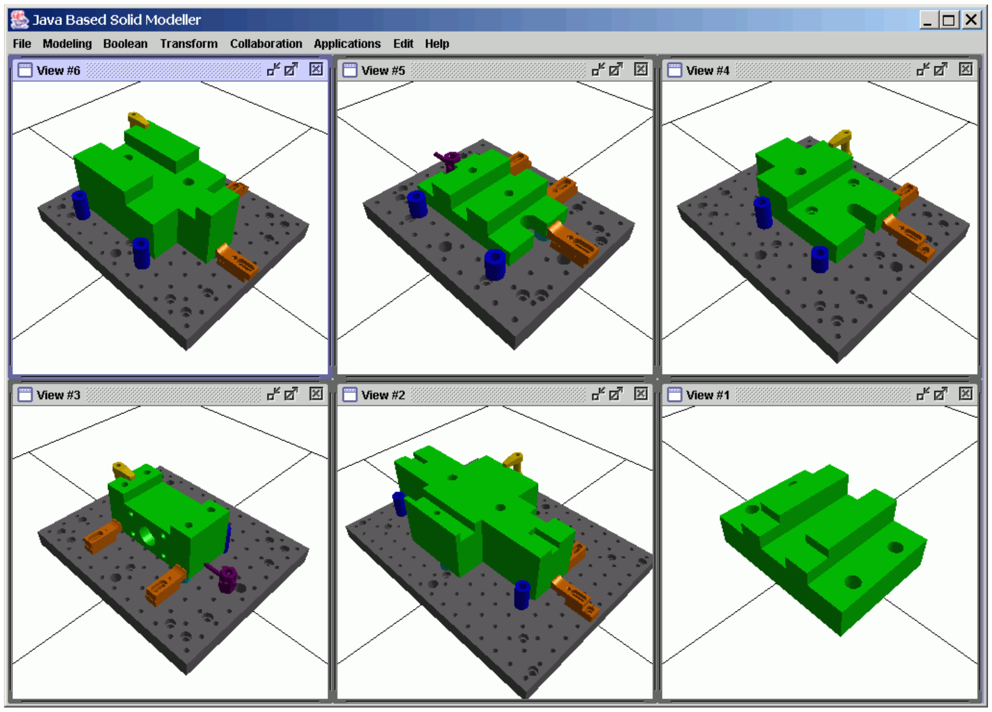
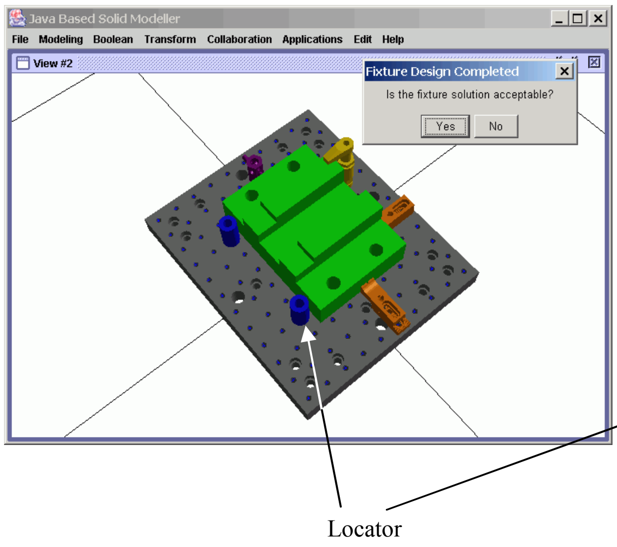

Java Based Solid Modeller
File Modeling Boolean Transform Collaboration Applications Edit Help
View #6
View #5
View #4
View #3
View #2
View #1

Fig. 10. The new workpiece and fixture designs of its similar parts


Java Based Solid Modeller
File Modeling Boolean Transform Collaboration Applications Edit Help
View #2
Fixture Design Completed
Is the fixture solution acceptable?
Yes
No
Locator

```
- <SELECTEDWORKPIECE>
    <SELECTEDWORKPIECENAME>model4</SELECTE
        DWORKPIECENAME>
    <SELECTEDWORKPIECEID>32</SELECTEDWORKP
        IECEID>
  </SELECTEDWORKPIECE>
- <BASEPLATE>
    <BASEPLATENAME>BJ010-4050-
        12</BASEPLATENAME>
    <BASEPLATEID>126</BASEPLATEID>
    <BASEPLATETHICKNESS>0.05</BASEPLATETHICK
        NESS>
        .
        .
        .
- <LOCATOR>
    <LOCATORNAME>BJ400-12050</LOCATORNAME>
    <LOCATORID>471</LOCATORID>
    <LOCATORPOINTID>13</LOCATORPOINTID>
+ <LOCATORPOSITION>
- <WORKPIECE>
    <WORKPIECEID>32</WORKPIECEID>
-   <FACE>
    <FACEID>267</FACEID>
   + <NORMAL>
   + <FACECENTER>
      </FACE>
    </WORKPIECE>
  </LOCATOR>
```

Fig. 11. The final fixture design of the new workpiece and its XML schema