ABSTRACT

Today’s manufacturers provide integrated product process development from product design, production and finally to marketing and sales. The whole product life cycle contains lots of data where product data management is crucial for making use of update information to facilitate collaborative product commerce. In this paper, a product data exchange standard, Product Information Markup Language, which is composed a set of well defined XML schema, is introduced to provide a flexible way to encode structured data into a format. In order to response to the rapid change of production environment or market trend, the intelligent operation is embedded in PIML such that it can provide advice for the design engineers to resolve the engineering problem and minimize the ripple effect caused by sudden changes. The intelligent operation is an artificial intelligent problem solving technique, case based reasoning, which retrieves the past most similar case by weighting various problem parameter and then adapts the past solution for the novel problem. A prototype system is developed and is proposed for applying in electroplating industry.

KEYWORD


1. INTRODUCTION

Today’s manufacturers require to face the challenge of responding to turbulent market change, meeting the escalating customer requirement and producing the qualitative product within a short product development time. The shorter time-to-market, the greater market shares the company can gain. To achieve the goal of shortening the lead time and increase the agility through product development, various technologies is evolving within the aspect of process, product and organization (Loureiro & Leaney, 2003). CAD, CAPP, CAM are used to provide better quality data and decision in manufacturing process. DFx is deployed to realize importance of design constrain in designing product. Contract manufacturing, agile manufacturing and lean manufacturing are strategy used for thriving and prospering in a competitive environment of continuous and unanticipated change. In product development cycle, there is enormous amount of scattered and loosely data dispersed in a distributed manufacturing environment. As a result, Product Data Management is essential as it helps to let the people know enough about the product development environment such that they can deal with change with update information efficiently and effectively.

There are several situations that lead to engineering changes which come from external or internal request. The fashionable market trend and changing customer requirements cause design practitioners to make design adjustment. Constrains imposed by tooling and equipment lead to technical change in manufacturing process. Financial aspect such as material cost cause change in material selection. A numerous problem have been identified and shown below

- Lack of co-ordination
- Lack of common standard for product development to exchange information between heterogeneous applications
- Lack of intelligent advise tools provide for junior engineering
- Lack of method for handling change systematically

The paper is organized to meet the following objectives to resolve the above problems:
To present a generic model of integrated product development

To propose the ‘Product Information Markup Language’

To illustrate how design agent handle design change with CBR approach.

To demonstrate the case example about applying the proposed framework in electroplating industry

To draw some conclusions and outline the needs for further research and development

2. A GENERIC INTEGRATED PRODUCT DEVELOPMENT MODEL

The generic model is composed of the three components which are data and knowledge repository, web services and intelligent operation (See Figure 1). New design is usually produced by reusing the previous innovative design with addition to the recent market needs. Therefore, data repository is importance for storing the past design information such as Computed Aid Drawing (CAD), sketch, part file, Bill Of Material (BOM), customer specification and assembly drawing. Design practitioners may solve the problem arisen during the design process based on explicit or tactic knowledge and make the decision based on the design constrain and budget. Capturing those knowledge and stored in knowledge repository can greatly reduce time consuming design iteration (Lee Branki, Aird, 2001). Among AI techniques, CBR is suitable for early product development since early product development is often based on incomplete, ill-structure and poor quality information (Haque, et al, 2000), which implies that lots of unknown factors permeate in the design process continually. The main idea of CBR is that analogies can be drawn between past situations, generally known as cases, and the current problem being solved (Lee et al, 2001). As a result, the main components of CBR which is retrieval algorithm, reasoner and adaptor are used for intelligent operation in PIML. Inference engine empowers with reasoning capability and provides advice for various problems. Since knowledge workers makes decision and adapt to environmental change with update information, interconnection between knowledge repositories to database is essential (Chung & Lau, 2000). The update data in relational database, which has been objectized and transformed as XML, in turn becomes valuable source of knowledge repository. Users can leverage the knowledge which is manipulated by the inference engine through user-interface.

Web Services provide a world-wide distributed environment that uses XML based messaging for access to distributed objects, application integration, data/information exchange, presentation aggregation, and other rich machine-to-machine interaction (W3C, 2002).

Web service describes specific functionality value delivered via Internet protocol, for the purpose of providing a mechanism for another service or application to use (Bosworth, 2001). The consumer may request Original Design Manufacturers (ODM) to provide complete concept or design rendering service. Using a Universal Description Discovery Integration (UDDI) interface, enterprises can dynamically connect to services provided by external business partners. Web service is grammar described by Web Services Description Language (WSDL) including: interface and end points (or ports), what a service can do, where it resides and how to invoke it via machine-understandable terms targeted for automated distributed communications between Web applications (Microsoft Corporation, IBM Research, 2001). To exchange information in decentralized environment and communicate with XML web service, a XML based protocol called Simple Object Access Protocol (SOAP) is deployed. SOAP is protocol binding with HTTP and HTTP-EF and it helps to provide/response model for XML web service. XML play a pivotal role in XML Web service. XML schema provides a schematic data structure for the XML document. The schema describes the vocabulary in form of rules or grammar that complying XML documents must follow in order to be considered schema-valid with that particular schema. Validation of XML documents ensures that the external data conforms to the rules (grammar) required by XML schema (MSXML, 2001).

3. PRODUCT INFORMATION MARKUP LANGUAGE

The internet appears to have latest technology and it is beneficial to globalized electronic information sharing. XML is recognized as a universal standard for the ex-
change of transaction document over the internet. PIML is a specific set of XML for product development which takes advantage of the well-structured XML schema to provide a well-understood syntax and self-defined markup language to suit particular needs in terms of product data exchange.

3.1 Product data schema

During product development life cycle, common change include changes to design method, changes to hardware, the provision of better quality data, changes to the organization of manufacturing system and changes to planning and control methods. The purposes for engineering changes are error correction and continuous improvement. Engineering changes need strong collaboration among change requestors, change brokers, change providers.

<table>
<thead>
<tr>
<th>Object Name</th>
<th>Element</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineering-Change Note</td>
<td>Name</td>
<td>string</td>
<td>Person/Organization request change</td>
</tr>
<tr>
<td></td>
<td>Date</td>
<td>date</td>
<td>The date when the DNC was generated</td>
</tr>
<tr>
<td></td>
<td>SerialNo</td>
<td>ID</td>
<td>ID for a DNC. It is recommended that it is universally unique. There is no format defined for it. Any algorithm generating such an id can be used</td>
</tr>
<tr>
<td></td>
<td>Mailing Address</td>
<td>String</td>
<td>Address of the corresponding</td>
</tr>
<tr>
<td></td>
<td>Change Type</td>
<td>complex</td>
<td>Type of change. The choice includes: adaptive, corrective, perfective and preventive</td>
</tr>
<tr>
<td></td>
<td>Problem</td>
<td>String</td>
<td>Problem leads to design change</td>
</tr>
<tr>
<td></td>
<td>Solution</td>
<td>string</td>
<td>The proposed solution for the problem</td>
</tr>
</tbody>
</table>

1. Change requestors who initiate the change, can be the clients or technicians in engineering department.
2. Change brokers are co-coordinators for the impact activities and operations for each affected object. Change requestors can interrogate change brokers in order to collaborate with required change providers.
3. Change providers effect the change to products, components, assemblies, process, document or even the activities in supply chain.

The product data schema allows user to define the structure and data type for XML documents. The change type of engineering change can be generally categorized as adaptive, corrective, perfective and preventive (OMG, 1998). Table 1 shows the major elements in engineering change note.

PIML describes the specification by XML where XML schema validates the incoming and outgoing XML document. Figure 2 shows product data schema which expresses some typical element in the Engineering Change Note based on Table 1.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.2 U (http://www.xmlspy.com) by carman (HKPU) -->
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="EngineeringChangeNote">
    <xs:complexType>
      <xs:choice>
        <xs:element name="ECN" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Requestor"/>
              <xs:element name="Date" type="xs:date"/>
              <xs:element name="SerialNo"/>
              <xs:element name="MailingAddress"/>
              <xs:element name="ChangeRequest"/>
              <xs:element name="Reason"/>
            </xs:sequence>
            <xs:attribute name="ChangeType" type="xs:string" use="required" default="Corrective"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="ECN" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Requestor"/>
              <xs:element name="Date" type="xs:date"/>
              <xs:element name="SerialNo"/>
              <xs:element name="MailingAddress"/>
              <xs:element name="ChangeRequest"/>
              <xs:element name="Reason"/>
            </xs:sequence>
            <xs:attribute name="ChangeType" type="xs:string" use="required" default="Corrective"/>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Figure 2 Product Data Schema

3.2 Mapping mechanism between relational database and knowledge repository

In PIML, close relationship between knowledge repository and database is established as knowledge workers operate and conduct activities with document and data which are stored in database. The typical tables in database are categorized as customer, order, order details, employee, supplier where the case stored in case base may include relevant piece of information from various tables in database in addition with explicit knowledge. To interconnect the heterogeneous repository, XSLT is deployed and manage axiom semantics. XSL specifies the styling of an XML document by using XSLT to describe how the document is...
transformed into another XML document that uses the formatting vocabulary (W3C, 1999). Almost all commercial DBMS such as Oracle 8i, IBM DB2, and Microsoft SQL Server have been extended to handle XML documents. XSLT processor can convert the XML document or well-format text to the specified format such as text, html or xml according to the template rules. The output data can be viewed by web browser or act as data source for knowledge base system.

XSLT possesses programming logic since XSLT supports a numerous set of flexible data types, a full set of operations and programming flow logic. The following example is to illustrate how XML combines with XSLT which capture programming logic on PIML.

Supposed the engineering change note contains the data such as requestor, date, serial number, mailing address, description and reason for change. Data stored in relational database is exported as the following XML format (See figure 3).

```xml
<EngineeringChangeNote xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="C:\My Documents\Conf\Prog\ECN.xsd">
  <ECN ChangeType="Preventive">
    <Requestor>Swiss Engineering Company</Requestor>
    <Date>2003-03-01</Date>
    <SerialNo>1485</SerialNo>
    <MailingAddress>On Wah Inudstrial Bldg, Fo Tan</MailingAddress>
    <ChangeRequest>For silver bath, Condition: Temperature is 24-32C, Current density is 0.5-1.5A/dm sq, Agitation is preferred. Cathode efficiency is 100%, Ration anode to cathode area is 1:1, Anodes is silver</ChangeRequest>
    <Reason>Maintain the tank condition.</Reason>
  </ECN>
  <ECN ChangeType="Corrective">
    <Requestor>Comfy Metal Finishing Co</Requestor>
    <Date>2002-10-26</Date>
    <SerialNo>B014</SerialNo>
    <MailingAddress>On Wah Inudstrial Bldg, Fo Tan</MailingAddress>
    <ChangeRequest>Pre-treated the parts again</ChangeRequest>
    <Reason>Peeling from the substrate</Reason>
  </ECN>
  <ECN ChangeType="Perfective">
    <Requestor>Design Dept</Requestor>
    <Date>2002-10-26</Date>
    <SerialNo>D014</SerialNo>
    <MailingAddress>Shenzhen, China</MailingAddress>
    <ChangeRequest>Consider not to include sharp edges and right angles for product design.</ChangeRequest>
    <Reason>Part receives very little plate in an acute angle. Every sharp protruding edge draws extra current and builds up with extra plate and vice versa. All sharp edges and angles should be rounded to the greatest degree design allows.</Reason>
  </ECN>
</EngineeringChangeNote>
```

Since perfective engineering change note (Highlighted by the dot line in figure 3) may contain critical information for product design in plating process, description of engineering change are stored as case solution in case base. For example, design practitioners need to recognize the function of the designed parts, the chemical and physical properties of the plating design such as strength, hardness, brightness. As they identify how plating processes improve the properties, they need to analyze the cost and benefit for the defined finish. Some geometrical design principle of the part is needed to take into consideration. Sharp edges, ridges, recesses bind hole etc introduce difficulties and increase cost into the finishing operation. In the XML code shown above, design department requests design change due to the sharp edges and right angle. Design practitioners also state the reason in engineering change note where that information in database is exported as XML format and then import to case base. The template rule (<xsl:if test="@ChangeType[="Perfective"]">) is set in XSLT and the relative data (Reason and Change Request) are retrieved and act as the solution/advice and Description for the case respectively (See figure 4).

```xml
<EngineeringChangeNote xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="C:\My Documents\Conf\Prog\ECN.xsd">
  <ECN ChangeType="Preventive">
    <Requestor>Swiss Engineering Company</Requestor>
    <Date>2003-03-01</Date>
    <SerialNo>1485</SerialNo>
    <MailingAddress>On Wah Inudstrial Bldg, Fo Tan</MailingAddress>
    <ChangeRequest>For silver bath, Condition: Temperature is 24-32C, Current density is 0.5-1.5A/dm sq, Agitation is preferred. Cathode efficiency is 100%, Ration anode to cathode area is 1:1, Anodes is silver</ChangeRequest>
    <Reason>Maintain the tank condition.</Reason>
  </ECN>
  <ECN ChangeType="Corrective">
    <Requestor>Comfy Metal Finishing Co</Requestor>
    <Date>2002-10-26</Date>
    <SerialNo>B014</SerialNo>
    <MailingAddress>On Wah Inudstrial Bldg, Fo Tan</MailingAddress>
    <ChangeRequest>Pre-treated the parts again</ChangeRequest>
    <Reason>Peeling from the substrate</Reason>
  </ECN>
  <ECN ChangeType="Perfective">
    <Requestor>Design Dept</Requestor>
    <Date>2002-10-26</Date>
    <SerialNo>D014</SerialNo>
    <MailingAddress>Shenzhen, China</MailingAddress>
    <ChangeRequest>Consider not to include sharp edges and right angles for product design.</ChangeRequest>
    <Reason>Part receives very little plate in an acute angle. Every sharp protruding edge draws extra current and builds up with extra plate and vice versa. All sharp edges and angles should be rounded to the greatest degree design allows.</Reason>
  </ECN>
</EngineeringChangeNote>
```

Figure 3 XML of engineering change note

XSLT transforms the data in text format with delimiters and those delimiters act as anchors of the text file. The text followed by delimiters will be imported into the corresponding case of the case base where the defaults delimiters include Case Name:, Description:, Solution: and Questions:. The output format transformed by XSLT processor is shown in figure 5.

```xml
<ECN ChangeType="Preventive">
  <Requestor>Swiss Engineering Company</Requestor>
  <Date>2003-03-01</Date>
  <SerialNo>1485</SerialNo>
  <MailingAddress>On Wah Inudstrial Bldg, Fo Tan</MailingAddress>
  <ChangeRequest>For silver bath, Condition: Temperature is 24-32C, Current density is 0.5-1.5A/dm sq, Agitation is preferred. Cathode efficiency is 100%, Ration anode to cathode area is 1:1, Anodes is silver</ChangeRequest>
  <Reason>Maintain the tank condition.</Reason>
</ECN>
```

Figure 4 XML Style sheet Language Transformation
and angles should be rounded to the greatest degree design allows.  
Solution: Consider not to include sharp edges and right angles for product design.

Figure 5 The neutral data file for case base

The text file shown in figure 5 is imported into the case base and acts as knowledge source of the CBR system since most CBR system allows text to case conversion. The data in corporate can be migrated to the knowledge repository of CBR. The knowledge representation relevant to CBR can assist junior process or product engineer to formulate the possible solution for particular domain.

4. INTELLIGENT OPERATION BY MULTIPLE AGENTS

Case based reasoning is proposed to be the intelligent operation in generic integrated product development model. CBR, which is one of artificial intelligence techniques, solves the novel problem by reusing past similar cases. In CBR terminology, case usually denotes as an episode with problem and solution. Case based reasoning is cycle loop which includes retrieving the past similar case, reusing the most similar case, revising the case for the novel problem and finally retaining the seed case for future use (See figure 6).

Figure 6 Collaboration among multi-agents

Search agent---it identifies the features of the problem that is the description of the problem. A smart search agent not only identifies the words inputted, but also understands the meanings of the description. Otherwise, search agent can only retrieve the case with some surface feature. Some important keywords should set up the definition, synonyms and thesaurus in order to let the system understand the semantics of the problem description. As search agent executes the initial match with the most a set of plausible candidates, it selects the most similar cases for reuse among the numerous cases in the case base. The method to identify the cases similarity includes nearest neighbor algorithm, induction, and knowledge guided induction.

Advisor agent---All retrieved cases are presented to advisor agent for further examination. The advisor agent tries to solve the new problem by modifying the old solution in order to conform the new situation. Advisor agent will revise the case by evaluating the case solution generated by adaptation agent. If the case can solve the problem successful, it will retain, and otherwise it may seek manual advice or predefined rules in the system to repair the cases.

Adaptation agent --- it takes account of the difference between the retrieved case and the novel problem such that adaptation agent can adapt the past solution and provide a reasonable solution for the current problem. Adaptation approach can be generally classified into substitution adaptation, transformational adaptation, derivational adaptation and compositional adaptation.

Maintenance agent---it stores the case in a manageable structure such that it allows retrieval agent access the case efficiently. It is probably true that more cases in case bases, the more successfully the problem is solved. However, increasing number of cases may decrease the efficiency of retrieval. Maintenance agent finds a balance between storing method that preserves the competency of cases and the efficiency of case retrieval. The case should be competent and assigned with indexes such that the retrieval agent can access and retrieve the relevant cases proficiently.

These generic procedures of CBR can be deployed to support the enhancement of adaptability in product design by virtue of its knowledge capturing and features reusing. The reasoning mechanism is the essential part to support knowledge discovery, which is instrumental in product design process where reuse of previous experience and creation of new ideas are the two key catalysts to support an innovative and rapid product design environment.

5. SYSTEM IMPLEMENTATION AND DATA EXCHANGE

PIML has been experimentally implemented in a Hong Kong manufacturing firms. The following case study illustrates how the present system works.

Step 1: Define XML Vocabulary. PIML is a set of XML vocabulary which acts as XML schema for product data model that allows data exchange between customer and enterprise. Define the major elements, attributes, data types and constraints on a complex content definition as section 3.1.

Step 2: Build up the case for case base. During the product and process design, experienced design practitioners usually recall the past cases and modify the previous design previous case. Update product data about customer requirement and design change as well as some explicit knowledge can be stored in form of cases such that some junior engineering can base on those cases to solve the novel problem.
Step 3: Replicate the data from relational database to case base. Mapping mechanism between relational database and knowledge repository has been mentioned in section 3.2. The ultimate aim of such an approach is to achieve the automatic updating of information on cases when there is any change of data on the relational database.

Step 4: Case based reasoning is proposed as intelligent operation for capturing knowledge in engineering design. The engineers input the problem encountered in electroplating process. The case retrieval group box (upper left) is the list area of messages from search agent which finds the relevant cases in case base. The text box (lower left) is used to display the suggestion from Advisor agent and the information is come from solution of the case which replicated from relational database (See Section 3.2). The text box in lower right part collect the user recommendation of the retrieved case such that it can be used to evaluate the competent of the retrieved case.

6. CONCLUSION

In this paper, a dynamic product information schema is proposed for supporting responsive product development. The empirical feature of PIML, which is the importance of polymorphism and inheritance in the design of a dynamic mechanism to deal with engineering change, is introduced to facilitate rapid product development. The reported methodology is a step toward the development of a generic model for effective communication with information exchange in order to achieve ease authoring, browsing at any-time anywhere and efficient content analysis and therefore attain the goal of short time-to-market and development of innovative product with latest customer requirements. A case study is conducted at Hong Kong electroplating manufacturer which set up two plants in the mainland and one office in Hong Kong. The result shows that PIML capitalizes data exchange among business partners with the latest advances XML technology related to product development, thereby enhancing design agility. The proposed Product Information Markup Language enables efficient data exchange among various data objects which resides in different platforms by providing a meaningful semantic framework for managing the various activities during product development. Agents possess social ability, proactiveness, reactivity and autonomy (Woodridge, Ciancarini, 2000) thereby enhancing the creation of a conducive product development environment.

ACKNOWLEDGMENTS

The work described in this paper was substantially/partially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region (Project No. G-W065).

REFERENCES

OMG(1998) PDM Enablers Joint Proposal to the OMG in Response to OMG Manufacturing Domain Task Force
Author Biographies

CARMAN LEE is currently a research student of the Department of Industrial and Systems Engineering at The Hong Kong Polytechnic University. She received her Bachelor degree at The Hong Kong Polytechnic University in 2000. Her current research areas include manufacturing information systems and product development. She can be contacted by e-mail at <00900653r@polyu.edu.hk>

HENRY LAU is currently the Assistant Professor of the Department of Industrial and Systems Engineering at The Hong Kong Polytechnic University, involving in research and teaching activities. He received his Master degree at Aston University in Birmingham in 1981 and his Doctorate at the University of Adelaide in 1995. His current research areas cover manufacturing information systems and artificial intelligence applications, and have authored and co-authored over 90 academic papers covering multi-agent modeling, object technology, global manufacturing and computational intelligence applications. He can be contacted by e-mail at <mfhenry@inet.polyu.edu.hk>

KAI-MING YU received his BSc (Eng) in Mechanical Engineering from the University of Hong Kong in 1985. He obtained his PhD from the University of Hong Kong, Department of Mechanical Engineering in 1991. He worked in the Research Centre and Mechanical Engineering Department of the Hong Kong University of Science & Technology until 1993. He is currently an Assistant Professor in the Hong Kong Polytechnic University Industrial and Systems Engineering Department. His research interests include CAD/CAM, CAE, and PDM, reverse engineering and rapid prototyping technologies. He is also a Senior Member of the Society of Manufacturing Engineers. He can be contacted by e-mail at <mfkmyu@inet.polyu.edu.hk>