# Research on Agile Infrastructure for Collaborative Manufacturing and Agile Supply Chain

J. L. Su [a, b, c] Zhonghui Ouyang [b] Yimin Chen [a]

[a] *School of Computer Engineering and Science, Shanghai University,200027,shanghai, P. R. China*
[b] *TSL School of Business and Information Technology, Quanzhou Normal University, 362000, Quanzhou, P. R. China*
[c] *Yang En University, 362014, Quanzhou, P. R. China;*

*su_jinlong@yahoo.com.cn*

***Abstract*—The Information Economics and its platform can input the vigor into manufacturing industry, i.e. garment industry, and help to increase its ability for good return, can promote the level of technology and management of Garments from a labor-intensive industry, and can move it toward the accurate management. The platform is designed as Comprehensive Information Platform by our research team. This paper is one of the series of papers about the design of Agile Infrastructure for Collaborative Manufacturing and Agile Supply Chain by us. This one lays emphasis upon the design of data access model. The design of data access model, MVC model2, is touched upon. The three parts of Model-View-Controller model are elaborated in this paper in turn. The paper tries to introduce the research of Agile Infrastructure and its key technology, showing how well it works in some traditional manual industry.**

***Keywords—Agile Infrastructure, manufacturing industry, MVC model2, Platform***

## I. INTRODUCTION

With the growing trend of economic and information-based globalization, manufacturing industry has to face the increasingly fierce competition as well as the frequent and unpredictable market fluctuations. An Agile theory thereupon emerges as the requirement of global market fluctuations. The life cycles of products are shortened and the pace of the product renewal is quickened, and the demands of customers are becoming more and more specific and diversified, thereupon, the producing and organizing models transfer from product-oriented to customer-oriented, requirement-oriented and service-oriented, and the aims of enterprises transfer from enterprise profit-driven to market and social profit-driven. The key for enterprise to gain its markets and customers is to improve some factors, such as time, quality, cost, service and environment. Agile Enterprise and Manufacturing Enterprise Alliance, as the running models of future enterprises, will fully make use of new technologies and coordinated operation which is more agile, compartmentalized, order-driven and dynamic to adapt to the markets.

The relationship among Agile Manufacturing Enterprise, Agile Supply Chain and Agile Infrastructure is like the relation between the sharpness of knife and the knife itself, which cannot be divided apart. Agile Infrastructure for Manufacturing System is the platform where the Agile Enterprise, Agile Supply Chain, Agile Manufacturer, Virtual Enterprise are put to good use. The essential condition of Agile Enterprise and Virtual Enterprise is the Agile Infrastructure which is reliable, cross-enterprise, cross-industry and trans-regional. The Agile Infrastructure is established to normalize the managerial practices of enterprise, such as production, sale, policy-making, financial affairs and personnel affairs. The member enterprises can be inserted flexibly, just like the circuit module with standard output and input jacks. In the Agile Infrastructure, member enterprises run business with common rules and establish dynamic Agile Enterprise Alliance, i.e. Virtual Enterprise. The members of Agile Enterprise Alliance negotiate abiding by common rules and accomplish the task of production and sale, which is a game of cooperation. According to the outside market environment and the group intention of inside members, the Virtual Enterprises make identical judgment and macrocosmic layout.

In this paper, the authors research into the construction of Agile Infrastructure for Collaborative Manufacturing and Agile Supply Chain and its key enabled technologies. It is supported by the achievements of some projects, such as the "Demonstration Projects of the Information-based Technology of Manufacturing Industry in Fujian" which is part of the major national supporting project of "the Eleventh Five-year Plan" (i.e. "Information-based Project in Manufacturing Industry"), the "Creative and Information-based Demonstration Platform of Modern Port with Large Logistics" (Project Number: RJZ20063500037) which is a major national project of software and integrated circuit, the "Research on Data Acquisition and Large Information Platform Construction of Distributed Information System" (JA06014) which is a project at provincial and ministerial level, the "Enterprise Informational Public Service Platform (EIPSP)" (2006H0106) which is a subject in textile and garment industry, "Research on the City Distributional System of GIS Platform" (2005J056).

This paper，which lays emphasis upon the design of data access model, is one of the series of papers about the design of Agile Infrastructure for Collaborative Manufacturing and Agile Supply Chain by our research team. The overall organization of the paper is as follows. After the introduction, in Section II we present an example of Agile Infrastructure in application, the garment information platform, which is a practical platform for garment collaborative manufacturing. The relationships of the main GIP functions and the structure of GIP are also clarified

in this section. In section III, the design of data access model, MVC model2, are touched upon. Then, the program design of model is presented on Section IV. In Section V, the authors elaborate the program design of View, and program design of controller introduced in Section VI. Finally in Section VII the authors conclude the paper.

## II. AGILE INFRASTRUCTURE AND IT'S APPLICATION

Agile Infrastructure for Collaborative Manufacturing and Agile Supply Chain can standardize the output and input information of its member enterprises. Our team had done a lot on Agile Infrastructure research, and named it as CPI (Comprehensive Information Platform). CIP stores the manufacturing and supply chain and human resource information in CIP's database and helps the enterprises to run their business in an Agile way designed by CIP, i.e. Agile Collaborative Manufacturing Execution Systems (ACMES), Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), Product Data Management (PDM), Supply Chain Management (SCM). The Comprehensive Information Platform tries to affect every aspect of its members in their running mechanisms, and each registered enterprise becomes a standardized module of the Agile Infrastructure. All its members can buildup Agile Manufacturing Enterprise, or construct Agile Supply Chain temporarily or permanently. Thus the Enterprises can concentrate on their core competences and they are able to recombine rapidly their interior and exterior capabilities and resources, thereby to respond rapidly market opportunity.

CPI theory has multiplications; one of them is GIP (Garments Information Platform) which is designed for CM (Collaborative Manufacturing) in garment industry in China. The garment industry was considered as the labor-intensive industry during the past several decades in China. The rough developing type was considered as its basic developing path. The expansion of information economics in garment industry changes these considerations.

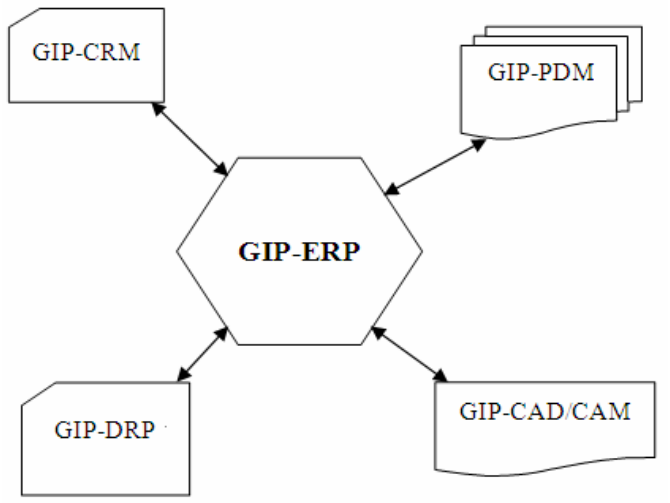


Figure 1. Relationship of the Main GIP Functions

In China, there are thousands of small garment factories which work for a same merchant and cooperate to produce a same kind of overcoat or football shoes. One of them only manufactures very small part of finished production or small number of them. Those factories may not belong to a same company and most of them possibly are located in different places, thus, the information sharing becomes one of the bottle-neck of the garment industry. The information which needs to be communicated includes material management, cost control, manpower arrangement, quality control and manufacture technology sharing. All of them are becoming crucial in the chain of garment industry.

The garment industry can optimize the garments management by utilizing Garments Information Platform Enterprise Resource Planning (GIP-ERP), can provide the advantages of manufacturing management by integrating GIP-ERP and GIP-PDM (Product Data Management), can combine the garments' Computer Aided Design/Manage (GIP-CAD/CAM), the integrated GIP Distribution Resource Planning (GIP-DRP), can change the traditional sales channel's process, the garments Customer Relationship Management (GIP-CRM), can satisfy the relationship between company and customers.

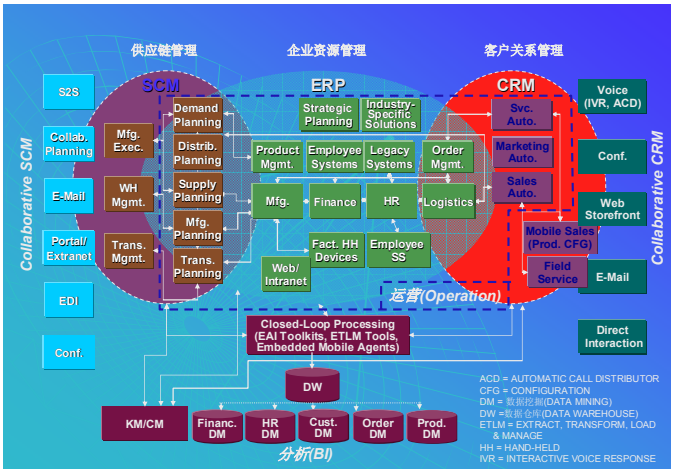


Figure 2. The Design of Agile Infrastructure for Garment Industry

The garments Enterprise Resource Planning (ERP) is the centre of GIP, where CRM, CAD/CAM, DRP, PDM system combined together, useful enterprise information perfectly shared. GIP-ERP is a system which can efficiently process information and effectively communicate with other parts in the GIP system. Thus the enterprise resource can be well managed and adjusted, and the efficiency of production and management of enterprise can be much improved.

A structure map of GIP with more detail is shown in Fig.2, where we can see the structure of GIP is complicate, including office automation system (OA), management information system (MIS), knowledge base, electronic commerce (EC), CAD/CAM/CAE (3C) and product data management (PDM).

## III. DESIGN OF DATA ACCESS MODEL

Concerning the data access strategy, we promote an optimized programming and algorithm based on the XML and MVC model. As we know, data output and input is the most normal activity of Agile Infrastructure. The data is always

transferred to viewer and changed by customs at terminals, then is stored back to the database. The data stream between database and terminals is the main stream of the platform. Model-View-Controller model is design to separate the View module and logical module, which makes the program more maintainable, extendable, transplantable and reusable. Trygve Reenskaug design MVC model for Smalltalk platform in the late 1970's, which turns out to be a very useful and mature theory after thirty years evolution.

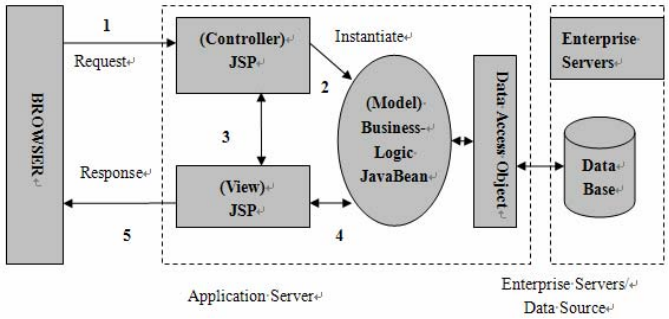


Figure 3. MVC Model 2

The MVC model is always designed by J2EE, and there still not perform very well in the View-Controller separating, and there still hard to realize the MVC idea by J2EE program. The research team redefines the MVC Model2 based on XML and uses XML and Java to build an MVC model in data access strategy of Agile Infrastructure.

In the research on data access of Agile Infrastructure, the author analyzes the main ways of implementation in the three parts of Model-View-Controller, and redesigns the program of MVC Model2 based on XML. In the programmer of MVC model, the whole process of data access of MVC model2 based on XML is described as follows.

Firstly, the terminal browser sends signal of request about URL address, and the front controller transmits the request through URL after the request of HTTP arrives at the controller.

Secondly, Path, the parameter of URL, accesses the page layout of system function and orientated the definition function. Together with the XML document of page logic, it is input to the XML interpreter.

Thirdly, Funname, the parameter of URL, makes use of the application logic of system and accesses the system function Class through Reflection. The Java Class of systemic function logic acquires data by database accessing and takes back the logic running result.

Fourthly, the XML interpreter forms HTML document which is identifiable to client, and returns to the client browser by HTTP agreement.

The advantage of XML-based design of MVC model 2 is that the system categorizes the functional page-layout and adopts different configuration rules for different kinds. Developers need only to make functional logic of page-layout in XML format, while the XML Parser in bottom layer of system forms the HTML document which is identifiable to client by analyzing the XML document and transmits it to terminal. The aim of completely separating the layout technology from system logic is realized by the way of setting the View module with XML configuration technology, which reduces the difficulties of system maintenance and secondary development. Other advantage is that the interface of system in infrastructure is changed by altering XML Parser rather than XML document.

## IV. PROGRAM DESIGN OF MODEL

The platform applied JDBC to connect the Oracle database, which is needed to open the database before use and close the database. The system encapsulates the open and close operation in SQLConn.java Class. Algorithm1 is the database opening program.

***Algorithm 1 Java.SQLConn Code***

```
Connection conn = null; //Open new data based
String jndi = SQLGlobal.getSQLGlobal().getJNDIByFacid (facid);
if(ctx == null){
try{
   ctx = new InitialContext();       //initialization
}catch(Exception e){
   throw new Exception("error in  initialization for: + e.getMessage()");}
try{
  DataSource ds = (DataSource)ctx.lookup(jndi);
  conn = ds.getConnection();
  return conn;
 }catch(Exception e)
{
   throw new Exception("error in database connection for:" + e.getMessage()
+ "!<BR>The Enterprise number is not exist ");
 }
```

Algorithm2 is for database close.

***Algorithm2 Java.SQLConn Code***

```
public void closeConnection(Connection conn) throws Exception{
try{
    if(conn != null){
        conn.close();//close the database
    }
 }catch(Exception e){
   throw newException("error in database closing for:" + e.getMessage());
   }}
```

Algorithm3 is to perform the sql execution.

***Algorithm3 Java.SQLConn Code***

```
public boolean exeCommand(String sql, Connection conn, HttpServletRequest
request)throws IstException{
      boolean openflag = false;
      boolean flag = false;
      if(conn == null){
         openflag = true;
         conn = this.getConnection(request);  }//Set up the connection
      PreparedStatement ps = null;
      try{
         ps = conn.prepareStatement(sql);
         ps.execute();
      }catch(Exception e){
throw new Exception("error in execute" + sql + "in database DDL, for:" +
e.getMessage(), e);
      }finally{
         if(ps != null){
            try {
               ps.close();
            } catch (SQLException e1) {……
            }}
         if(openflag){
```

```
            if(conn != null){
               this.closeConnection(conn);
            }}}
      return flag;}
public java.sql.ResultSet exeQuery(String sql, Connection conn)throws IstException{
      if(conn == null){
    throw new IstException("SQLConn.exeQuery parameter conn can not be empty"); }
      try{ return conn.prepareStatement(sql).executeQuery();
  }catch(Exception e){ throw new IstException("error in execute " + sql + "for:" + e.getMessage(), e);      }}
```

Database affair includes beginning the database affair; committing database affair and rollback database affair.

***Algorithm 4 Java.SQLConn Code***

```
   public void beginTransaction(Connection conn)throws IstException{
      try{
         if(conn != null){
            conn.setAutoCommit(false);   }// start the database affair
      }catch(Exception e){
         throw new Exception("error in execute start the database affair：" + e.getMessage(), e);      }   }
   public void commitTransaction(Connection conn)throws IstException{
      try{
        if(conn != null){
            conn.commit();   }
      }catch(Exception e){
      throw new Exception("error in commit database affair for:" + e.getMessage(), e);
      }   }
   public void rollbackTransaction(Connection conn)throws IstException{
      try{
         if(conn != null){
            conn.rollback();   }
      }catch(Exception e){
    throw new Exception("error in rollback database affair for:" + e.getMessage(), e);
      } }
```

## V. PROGRAM DESIGN OF VIEW

The page layout with logical function is encapsulation in XML file by some rules. When users access that functional layout, the front controller will interpret URL and transmit the Request and Session Object to XML Parser. XML Parser takes charge of XML file interpreting as well as checking request & session parameter or data accessing operation.

XML Parser locates physical address of in functional XML file in application server and loads it into the memory, after receiving the request from front controller.

***Algorithm 5 Java.SQLMgr Code***

```
String path = SQLGlobal.getSQLGlobal().getDefXmlPath() + "\\" + tablename;
java.io.File file = new java.io.File(path);//locate the XML file
SAXReader sa = new SAXReader();
doc = sa.read(file);// read file
this.rootem = doc.getRootElement();
java.util.Iterator it = rootem.elementIterator();
tagmap.clear();
while(it.hasNext()){
   Element em = (Element)it.next();// check the element
   if(em.attribute("id") == null){
      continue;
   }
   tagmap.put(em.attributeValue("id"), em);}
```

In program 5, the system translates the XML request address to the physical file address. And then it uses SAXReader of DOM4J to load this file into the memory. XML file interpreting process is to analyze XML and translate it into HTML file.

Program 6 is to generate the title of HTML file, which is a title to be shown in the web page.

***Algorithm 2-6 Java.EIPSPGlobal Code***

```
public String getTitleStr()
{
  Element titlenode = this.tagnode.element("page-title");
  String title = "";
  if(titlenode != null){
    title = titlenode.attributeValue("title");//analyzing XML node
    if(title.indexOf(">") > 0){
      title = title.substring(title.indexOf(">") + 1, title.length());
      if(title.indexOf("<") > 0){
         title = title.substring(0, title.indexOf("<"));// get the value of title string
      }
    }
    java.util.Iterator it = titlenode.elementIterator("page-title-param");
    while(it.hasNext()){
       Element em = (Element)it.next();
       String id = em.attributeValue("id");// check the id value
       if(id == null){
      throw new Exception("one of the page-title-param of page-title without an id value!");
       }
       id = "$" + id;
    String val = SQLGlobal.getSQLGlobal() .getObjString (ParamMgr.getParamMgr(em, request, response, this.parammap, this.conn, this.sc);
          title = SQLGlobal.getSQLGlobal().replaceStr(title, id, val);
       }
     }
    return title;}
```

In this function, this.tagnode Object saves the XML node, which is analyzed and generated by the DOM4J. By analyzing the title of sub node, the DOM4J returns title string. The hole HTML file is compose by analyzing result of each part of the XML file, and reloads to terminate by statement response.getWriter().write(ps.getHTML(0)). The response Object is the browser-reload-interface of servlet Class opened by servlet Class. After XML Parser analyzes and reloads it to browser, thus, the terminate can browses the web page through functional page accessing by URL in the browser.

## VI. PROGRAM DESIGN OF CONTROLLER

The system front controller offers two services, one is to access the functional page, which provides searching, adding, deleting, updating function, and the other is to access the system application logic, which can afford several kinds of service. These two services are encapsulated in several Java Classes. The front Viewer controls the different functional call by URL parameter accessing of terminates.

Rule 1: The format of functional page accessing of URL is '/EIPServlet?path=<name of module>/<service page>. <functional node>'. The front controller locates the XML file of system by name of module and service page in URL page; and the functional node is id node defined by XML request service.

Rule 2: The format of logical page accessing of URL is '/EIPServlet? funname=<name of Java Class>', front Controller execute the logical function through Java Reflection, Java Class use by URL path.

The Controller module calls for different service by input URL. It transmits request to relative module, to perform the function of Controller, which is realized by EIPServlet Class. EIPServlet comes in for javax.servlet.http.HttpServlet of J2EE API, which are Servlet Class. In web.xml, the web application program, configuring Servlet Class, EIPServlet, is to process the request of URL rule, which is shown by Algorithm 7.

***Algorithm 7 file Web.xml Code***

```
<web-app id="WebApp">
.....
<servlet>
   <servlet-name>EIPServlet</servlet-name>
   <display-name>EIPServlet</display-name>
   <servlet-class>ist.servlet.programe.EIPServlet</servlet-class>
</servlet>
<servlet-mapping>
   <servlet-name>EIPServlet</servlet-name>
   <url-pattern>/EIPServlet</url-pattern>
</servlet-mapping>
```

We define a Servlet, EIPServlet, whose relative Java Class, ist.servlet.programe.EIPServlet, is defined at the same time, for terminate accessing those URL begin with '/EIPSerlvet'.

EIPServlet are Controller Class, which comes in for its farther Class: Java2 Enterprise Edition API' and 'javax.servlet.http.HttpServlet'. There are two functions to get data sent by terminate, doGet and doPost for GET method or POST method respectively. Sometimes we change GET method to POST method, which can be seen in Algorithm 8.

***Algorithm 8 Java.EIPServlet Code***

```
public void doGet(HttpServletRequest req, HttpServletResponse resp)
   throws ServletException, IOException {
   doPost(req, resp);  // doGet redirect to doPost
}
```

Thus, the getparms of URL is performed by function doPost, which has two ideographs: 'public void doPost (HttpServletRequest req' and 'HttpServletResponse resp)'. There are two parameters for rewriting the data to terminate in the function, one is reg, the 'javax.http.HttpServletRequest'; the other is resp, the 'javax.http.HttpServletResponse'.

Controller transmits different services by parsing the parameters of URL, as can be seen in Algorithm 9.

***Algorithm 9 Java.EIPServlet Code***

```
//to find whether there are any funname in the URL parameter,
String funpath = req.getParameter("funname");
if(funpath != null){
   // turn to functional logical service after confirmation
   ........
}
//to find whether there are any path in URL prameter
String funpath = req.getParameter("path");
if(funpath != null){
   //turn to functional page service after confirmation
   ........
}
```

EIPServlet Class performs the function of front Controller, and decides which logical processing needs being turned to, by parsing the parameters of URL.

In comparison with the traditional program of J2EE, this method is more logic, stable and convenient for maintenance and data access. This method theoretically provides new ways of thinking in the structure, function and operation of MVC model, the new design plan of the model realization, as well as a practical way of the structural design of data access.

## VII. CONCLUSION

In this paper, Comprehensive Information Platform designed for collaborative manufacturing was presented with an application case in garment industry. Specifically, the relation of GIP functions as well as the structure of CIP in the garment industry application was introduced. In addition, technical details about data access model, especially, MVC model2, were discussed in the following sections. The three parts of Model-View-Controller model and their program designs are elaborated in turn. More elaborate performance report will come up after long period of observational and practical use. All technical details touched upon run in application, and the main improvement in its performance can only come with application. Some parts of the Agile Infrastructure are almost completed, i.e. GIP, which seems to work well and more details can be seen on the internet on http://www.istqz.com.

## REFERENCES


[1] J.M. Bull, R.W.Ford, A.Dickinson, A Feedback Based Load Balance Algorithm for Physics Routines in NWP, *Proceedings of Seventh ECMWF Workshop of the Use of Parallel Processor in Meteorology*, World Scienti_c, 1998.

[2] S. Lucco, A Dynamic Scheduling Method for Irregular Parallel Programs, *Proceedings of ACM SIGPLAN'92 Conference on Programming Languages Design and Implementation*, San Francisco, CA, June 1992.

[3] C.D. Polychronopoulos and D.J.Duck, Guided Self-Scheduling: A Practical Scheduling Scheme for Parallel supercomputers, *IEEE Transaction on Computers*, C-36 (12),1987, pp. 1425-1439.

[4] T. Tabirca and S.Tabirca, An Application of the Smarandache Inferior Part Function to Loop Scheduling, *Smarandache Notions Journal*, vol.12, nr.1-2, 2000, pp.35-46

[5] Y. Yang, C.Jin and X.Zhang, Adaptively scheduling Parallel Loops in distributed Shared-Memory Systems, *IEEE Trans. on Parallel and Distributed Systems*, vol.8, nr.1, 1997, pp.70-81.

[6] Nathan Einwechter. "An Introduction To Distributed Intrusion Detection Systems"[M]. Security Focus. Jan 2001.

[7] Eugene H. Spafford and Diego Zamboni. Intrusion detection using autonomous agents. Computer Networks, 34(4):547-570, October 2000.

[8] Mark Slagell. The design and implementation of MAIDS (mobile agent intrusion detection system) [R]. Technical Report TR01-07, Iowa State University Department of Computer Science, Ames, IA, USA, 2001.

[9] Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy and Salvatore Stolfo. A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data. Applications of Data Mining in Computer Security. Kluwer 2002.

[10] J. Han and M. Kamber. Data Mining, Concepts and Technique. Morgan Kaufmann, San Francisco, 2001.

[11] Jinlong SU. Research on Some Key Technologies of Integrated Infrastructure for Agile Enterprise [Doctoral Dissertation]. Shanghai University. March, 2008