COOPERATIVE AND INTEGRATION MECHANISMS IN COLLABORATIVE PRODUCT DEVELOPMENT

Li, W. D. and Roy, R.
Department of Manufacturing, School of Applied Science
Cranfield University
MK43 0AL, U. K.

ABSTRACT

In product development companies, an important progress is the extensive development and deployment of collaborative design and manufacturing methodologies and systems, further to form a Collaborative Product Development (CPD) environment. It is in line with the growing trend of global cooperative design and outsourcing in product development. In a CPD environment, people are able to participate in intra-/inter-enterprise or even global product development chains, and to collaborate with each other and overseas partners to pursue competitive advantages. Collaboration and integration are two key research issues, and the development of effective mechanisms is crucial for a successful CPD environment. In this paper, developed collaborative and integration mechanisms are discussed. The collaborative mechanisms are summarized from the perspectives of collaborator awareness, negotiation and conflict management, collaborative decision-making, development process management (e.g., workflow management, design change management and review management), team management, etc. The integration mechanisms are investigated from a data-centric integration level, which combine modern markup and Internet-compatible language and product exchange standards, and a service-centric integration level, in which services are developed to build up a dynamic CPD environment. In the end, some future trends are discussed.

Keywords: Collaborative Mechanism, Integration Mechanism, Collaborative Product Development

1. INTRODUCTION

To respond to modern markets that are in faster change and keener competition, product development companies strive for shortening the lead time of new products, improving product quality, and increasing the efficiency and flexibility of resource utilization. Recently, some philosophies have come into existence to facilitate product development processes. In the late 1980s, the philosophy of Concurrent Engineering (CE) appeared to integrate the design of products with related manufacturing processes systematically. To establish a CE environment, two fundamental mechanisms are necessary. The first mechanism is effective and quick bi-communications between designers, process planners and manufacturers to facilitate their interactions. Another mechanism is the close integration of software systems in product design, analysis, simulation and manufacturing optimization to realize obstacle-free product information exchange and sharing. Based on CE, manufacturing expertise can be brought into the early process of design decision-making to optimize the overall performance of products. The implementation of the communication mechanism usually depends upon some popular Internet communication facilities, such as emails, e-discussion forums, Microsoft Net-Meeting, etc. To realize the integration mechanism, neutral data exchange schemes, such as STEP (STandard for the Exchange of Product model data) and IGES, was initialized and popularized to represent product information in a common computer-interpretable form to remain the major features and data consistence of products when exchanged between the various software systems. From 1990s, the philosophy of Collaborative Product Development (CPD) began to be actively investigated to further increase the innovation of new products and gain the time-to-market. CPD has extended CE to support much wider inter- and intra- enterprise integration of product development systems, which are more heterogeneous in terms of information representation, data structure and programming language. Another significant advancement of CPD is to setup a more
intensive network to enable a designer to cooperate with other people, not only from the designer’s own company, such as marketing and sales staff, other designers and manufacturers, but also outside of the company, such as customers, suppliers, sub-contractors and distributors, for exchanging ideas and comments. Therefore, people, who work across multiple functions, organizational boundaries, disparate geographies, time zones and languages, can participate in global competitiveness and respond to market responsiveness rapidly.

Nowadays, key mechanisms are being in active development to pave ways to establish advanced CPD environments. Based on the state-of-the-art communication tools and augmented with the modern artificial intelligence techniques, collaborative mechanisms are being developed to allow people and software systems to work at various levels of cooperation with rapid access to distributed knowledge and information repositories. New integration mechanisms, such as service architectures, are also in quick evolution to wrap and incorporate various product development systems into a CPD environment as required, so as not to interrupt organizational links previously established. In this paper, some developed collaborative and integration mechanisms of CPD are summarized. The collaborative mechanisms include collaborator awareness, negotiation and conflict management, collaborative decision-making, development process management (e.g., workflow management, design change management and review management), team management, etc. The integration mechanisms are not only in a data-centric integration level based on the combined strengths of modern markup and Internet-compatible language and product exchange standards, but also in a service-centric integration level to setup CPD environments with dynamic service supporting.

2. COLLABORATION MECHANISMS OF CPD FOR DISTRIBUTED TEAMS

2.1 Collaborator Awareness

Due to different design habits of collaborators, it is usually a challenging task to organize a CPD activity. Awareness of collaborators and their characteristics/activities is important for establishing a ground for mutual understanding in a team, and further achieving the success of the whole collaborative activity. Awareness information, which depends upon the cues of a shared environment, can include who is around and what they are doing, the state of artifacts in the environment, people’s face-to-face communications, body language, physical gestures, etc. Awareness can support team collaboration in many ways. Based on the understanding of others’ situations and activities, a person can determine the context of his/her own activity. This context is used to ensure that individual contributions are relevant to a group’s activity, and to evaluate and adjust individual actions with respect to the whole goal and progress. Awareness information thereby allows a group to manage the progress of collaborative working. On the other hand, when people work together, they shift back and forth between individual and shared work. When people realize that a collaborative opportunity arises, when they need to discuss or decide something, or when their current design task requires the involvement of another person, they usually move from their own foci to participate in a joint activity. With awareness of others’ activities, people can decide who they can work with and when to make the transitions from their personal work to a team work, without missing opportunities to collaborate or interrupt other people inappropriately. Awareness can also facilitate the coordination of actions to realize cognitive synchronization. For example, in a collaborative task, a concurrency control mechanism, which constrains that only one person can manipulate an artifact or modify its attributes at a single moment, has been popularly used to avoid potential conflicts. When people are aware of the artifacts that others are currently using or manipulating, the concurrency control mechanism is less important or even unnecessary [20][23].

Awareness information can be naturally acquired in a co-located work place for developing intra-team relationships. When a team works in a distributed environment, only a fraction of the information available in a physical environment is provided. Many of physical cues for establishing awareness are absent or become narrowed, so that collaboration between team members becomes difficult. To collect and utilize the awareness information effectively, and to keep people apprised of important changes in a team without distracting them from their focal tasks, mechanisms for establishing distributed awareness are in investigation, primarily from two aspects: (1) to choose suitable communication tools and means to collect perceptual information, and (2) to optimize the organization of perceptual information for requested awareness and to decide the suitable presentation of the information.

Communication tools for perceptual information collection

When people work in separate places, many awareness resources are disrupted. The possibility to use gesture is limited, facial expressions are eliminated or constrained, auditory cues are diminished, tools and artifacts cannot be easily shared, and exchanged information is delayed by seconds or
even minutes [6]. Some advanced communication tools have been developed to enhance awareness to overcome the spatial or temporal separation of people. For example, shared repositories are data stores that allow group members to view and update shared works, such as project data and documents [18][30]. Monitoring systems provide means to track where group members are currently located or what they are currently doing [11][13]. Image and sound information can be conveyed using Audio/video conference systems [16][17][48]. Shared application facilities allow a remote group to work together simultaneously on an artifact or platform, such as e-document, e-whiteboard, and Computer-Aided Design system [21]. Annotation functions embedded in some systems can enable collaborators to be informed of each other’s comments or marks on shared documents [32][40][15]. Notification systems, such as email alerts and chat messaging, help people keep aware of events peripheral to their current tasks.

Organization and representation of perceptual information

To maintain effective awareness, perceptual information needs to be organized and represented strategically. The awareness information is generally represented as three ways: (i) embodiments, (ii) expressive artifacts, and (iii) visibility techniques [23].

- **Embodiments**
  Watching how other people work is a principal way for developing awareness, and thereby an embodiment technique has been developed as visible representations for people’s bodies and their statuses. There are three primary types of embodiments used in distributed groupware: telepointers, avatars and video images. Telepointers, which indicate the location of each team member’s mouse cursor, provide implicit information about presence, identity and activity of the member. Avatars represent team members with stylized pictorial descriptions, such as more humanlike bodies and some kinds of gesture. Video images of people and their actions are more realistic and expressive while full-fidelity images are usually weighed against the high performance in the Internet environment with the limited bandwidth.

- **Expressive artifacts**
  The interactions of an artifact by people in an environment can be used as one of the primary ways for awareness. To illustrate interactions more obviously and interpretably in a distributed workspace, expressive artifacts have been devised. For example, a cursor and a button (i.e., artifacts) manipulated by a person in a computer system can indicate the operation of the person. When the person’s cursor moves over the button, it is shown being pressed on all screens of other collaborators so that all of them know the operation of the person. However, the operation is very brief and easily ignored. To augment the effect, an alternative way is to make the action more visible, such as making a clicking sound or adding a graphic that lingers longer than the actual press [22].

- **Visibility techniques**
  Visibility techniques provide a broader overview of people in a shared environment. Radar view, over-the-shoulder view and cursor’s-eye-view are three popular visibility techniques. A radar view is an overview representation that shows the entire workspace in miniature, and it can facilitate people to maintain high-level awareness of presence, location and general activities of each other in the environment. An over-the-shoulder view shows a reduced version of another person’s main view. It allows a person to observe another person more closely. A cursor’s-eye view shows a small area around a person’s mouse cursor in full detail. This view is particularly useful when the precise details of a person’s work are required.

### 2.2 Collaborative Organizations

An organization mechanism is an essential building block for establishing a CPD environment for a team to conduct a groupware task and avoid process deadlock. The developed mechanisms can be classified based on task and temporal organizations.

The task-based organization consists of two primary ways: (1) members of a team focus on the same task simultaneously, and the process is carried on through interactive discussions, negotiation and manipulations among members; and (2) a complex task is decomposed as some sub-tasks, and each member is responsible for one or more sub-tasks. In the second way, interfaces between the sub-tasks are defined as “communication protocols” between team members to coordinate and communicate. The goal of the whole project is achieved through the combined contributions of all sub-tasks. The progress of the project is scheduled and controlled based on milestones, critical review, project meetings and discussions in necessary moments, etc.

In terms of temporal organization, there are synchronous and asynchronous collaborations. Synchronous collaboration requires efficient communication among team members to work together in a real-time manner. Asynchronous collaboration supports individuals to work in the different time so that it provides the convenience for the people to work based on their own schedules and time. In Table 1, some mechanisms and tools to support teams to carry on various collaborations are shown. Significant mechanisms and tools are discussed as below.
1. Concurrency control mechanisms

Concurrency control is the foremost mechanism to organize synchronous activities to avoid conflicts. Locking is a primary means in managing concurrency control to prevent people from colliding. Generally, locking has three types: non-optimistic, optimistic and visual.

- **Non-optimistic locking mechanism**
  The non-optimistic locking is the most straightforward mechanism. It enables a person to gain privileged access to a shared object for a length of time, and he/she can edit the attributes of a shared artifact actively during the time. Other people in the same community are observers to receive the updated information. When the lock-holder no longer requires the lock, it is released and made available to the collaboration community [2][8][34].

- **Optimistic locking mechanism**
  An optimistic mechanism has been proposed to improve the performance of a CPD activity. The optimistic mechanism can allow a person to acquire a tentative lock immediately once he/she requests the lock. With this tentative lock, the person can start manipulating a share artifact before he/she knows whether it is really unlocked or not. If the unlock is then approved, the process will continue as normal. Otherwise, the artifact must be returned to its original state. The advantage of the optimistic mechanisms is that waiting time can be minimized and the occasional repair is worthy.

- **Visual locking mechanism**
  A locking mechanism can be enhanced by a visual technique to indicate which artifacts are in use clearly so as to provide more flexible management. The mechanism relies on social protocols and people’s awareness of each other’s action to prevent conflicts. For example, Bidarra, et al. [4][5] proposed a traffic light mechanism to lift some strict restrictions or limitations of some locking mechanisms. Traffic lights can be used to indicate the busy states of team members. The light displayed at a member switches from green to yellow when another member starts specifying an edition operation on a shared object, and to red when a central server starts executing an operation on the object.

---

| Table 1: Functionalities for collaborative activities centrally and remotely located [48] |
|---|---|---|---|
| **Same Place** | **Different Place** | **Synchronous Collaboration** |
| Meetings | Video conferencing | Video conferencing |
| Whiteboards | Teleconferencing | Teleconferencing |
| Face-to-face communications | Screen saving | Screen saving |
| Electronic whiteboards | WWW forums | WWW forums |
| Live CAD sharing and interoperation | | |
| **Same Time** | **Different Time** | **Asynchronous** |
| Mailboxes | E-mails | E-mails |
| Shared folders | Workflow | Workflow |
| Bulletin boards | Routing | Routing |
| Documents | notification | notification |
| Meeting minutes | | |
Some research projects [43][19][25][28][26][54][31] have been carried out to enhance PDM systems to support pre- and post-design stages. Huang, et al. [28] developed a Web-based system to manage Engineering Changes (ECs) in a CPD activity. ECs are frequently happen during a design process, and managing the ECs in a Web-based system can facilitate better information sharing, simultaneous data access and more prompt communications among team members. In this system, four ways are used to aid the management of ECs during the whole life-cycle of CPD, including: (1) a log of ECs to present an overview of the whole ECs generated in the system; (2) a request of ECs to record the originator to create the ECs; (3) evaluation of ECs, through which an analysis to evaluate the ECs on the product development can been made; and (4) notice of ECs to transmit and share the ECs to the corresponding people. The system can play as a complimentary tool to a PDM system to enhance its capability in the management of ECs. Meanwhile, Huang extended the Web-based system to support product design review and bid to extend the system to support a design chain [27][26]. The design review system can simulate an on-line central review meeting room through equipped with a VRML whiteboard for visualizing an on-line design model, a review coordinator to provide a set of facilities for a project manager to plan the activities and resources involved in the review process, and a BOM explorer to store and share review comments and some relevant documents. The bid system utilized the potentials of Web for better information sharing and access to establish a bid space to enable customers to define their requirements and potential suppliers to specify the supply capabilities.

3. Negotiation and conflict management

As an important way, negotiation can formalize and implement the mediation and facilitation functions among people to handle conflicts. Earlier coordination mechanisms used the direct supervision to allow an individual designer to issue instructions and monitor the execution of the tasks. Working process, output and skills are required to be standardized to reduce the mutual communication and information exchange. Negotiation improves the rigidity of these mechanisms by introducing some more flexible and rational conflict arbitration mechanisms. The negotiation model proposed by Case and Lu [7] includes identification of conflicts, dissemination of information about conflicts, exchange of rationale and recording of conflict resolution. The whole process is initiated by identifying a conflict event. A set of users who are involved in the conflict event is notified and a deadline is set for negotiations. After receiving notice of an event, each user can express an opinion about the desired outcome of the conflict, which will then be sent to the arbiter. Each user is kept informed about the status of the whole negotiations and he has the option to change his opinion at any time until the deadline is reached or the conflict is resolved. If the conflict cannot be resolved through the negotiations and the time reaches the deadline of resolution, the arbiter will select one of the opinions, label it as the resolution, and then broadcast it to all of the interested users. Wong [51] proposed four methods of conflict resolutions, including enquiry, arbitration, persuasion and accommodation. In the method of enquiry, designers find and collect the underlying data and beliefs about a conflict. They then resolve the conflict by appealing to the retrieved data and beliefs and some shared principles for interpretation. The arbitration methods are based on some fair social-choice theory to select an outcome out of many competing alternatives. These methods usually include an agenda to contain a series of criteria for judge. Individual preferences by qualitative means (such as preference relationships) or quantitative means (such as utilizations or probabilities) are first formed from the competing alternative plans according to the criteria. Some procedures are then applied to select an outcome out of the individual preferences. The persuasion method aims to achieve agreement among designers, but not in the way enquiry does. The goal of the enquiry is to settle the conflict by smoothing out the differences, while the persuasion starts off from the premise that an agreement is not possible so it attempts to re-shape the agenda to figure out new acceptable solutions. The accommodation method takes account of the history of decision-making process for conflicts, and some rejected criteria might be reconsidered to be included into an agenda as compromise or compensation. This method is used to address a situation that a former decision differs in what the designers now want so as to rectify the previous action. Wang and Chien [50] proposed a Web-based group decision support system, which includes idea generation and sharing, discussion and decision making engine. The decision making engine consists of two mechanisms – deterministic decision inference and probabilistic decision inference. The deterministic decision inference facilitates decision-making by a set of rules and what-if analysis, and the probabilistic decision inference handles situations with uncertainty based on a Bayesian network.

4. Workflow management

Recently, workflow management systems have been popularly used to create, manage and execute collaborative activities asynchronously based on their business logic [1][24][12][33]. Workflow is a preferred way to streamline collaborative product
development process. However, the application is compromised due to its inherent rigidity, that is, business processes have to be mapped into a workflow before their execution, and the support for dynamic change or exception of workflow remains weak. One of the imperative research topics for a workflow system is to manage dynamic changes to the workflow process in an effective way and avoid the problem that the affected workflow has to be terminated before the changes are applied and the new workflow gets instantiated again.

To solve this problem, some MASs (Multi-Agent Systems) have been developed to utilize the characteristics of the MASs to provide more flexibility, scalability and portability in managing and scheduling workflows. The applications of the MASs in CPD are usually in two categories: (1) providing an open and heterogeneous platform to link product design and manufacturing effectively using facilitators and wrappers for legacy system integration [14][49][44][46][55][29] and (2) providing a mechanism to map a multi-disciplinary design process to a dynamically organized network [25][47][34][35][38]. Huang and Mak [25] made an attempt to use an agent to represent a design task, and therefore a group of such agents and their relationships can define a workflow. Observing that modularization/parameterization of design tasks and well-defined dependencies among the tasks are the keys to achieve a good CPD process, Li, et al. [34] developed a MAS, which is more applicable to practical situations, to specify modular tasks, design parameters and their dependencies. In the system, agents can take some modular design tasks such as computing, reasoning or enquiring, and an agent controller is responsible to coordinate information flow and task allocation. Xiang, et al. [53], Tang [47] and Madhusudan [38] applied the developed MAS-based workflows to some application design areas respectively, including fluid power system design, die-maker for stamping part design and aircraft brake design. With the help of some service agents (such as cost analysis, FEA (Finite Element Analysis), manufacturability evaluation) and coordination agents, systematic design environments can be set-up to develop and optimize complex products.

3. INTEGRATED MECHANISMS OF CPD FOR DISTRIBUTED SYSTEMS

An integration mechanism is imperative to seamlessly connect dispersed software systems from upstream design to downstream manufacturing and services. The mechanisms could also provide convenient and secured accessibility to databases of companies and suppliers with critical information for product development. Logically, the appeared mechanisms can be generally classified as two levels: (1) data-centric integration, and (2) service-centric integration. In the following part, previous works and systems are summarized from these two aspects.

3.1 Data-centric integration

To realize the integration of software systems in a CPD environment, a conventional way to exchange and interoperate product models of the systems is through neutral information exchange standards, such as IGES or STEP. Recently, XML, which is regarded as the new-generation Internet mark-up language and has inherent relationships with some emerging Internet technologies (e.g., Web services), has gotten more attentions. XML offers several potential advantages and allows for tag definition reflecting the tailored structure of information for various applications, so that it has been chosen a base for developing exchange formats of design and manufacturing data in a CPD environment. Extensions have been made for XML to incorporate some existing product exchange standards to combine their strengths to represent product models and the relevant information, including geometric and topological information (e.g., points, lines, surfaces, relationships between geometric entities and attributes, etc.), validity constraints and attributes.

(1) EXPRESS/XML

EXPRESS (ISO 10303-11) is the modeling language of STEP. Various data types, including simple types (i.e., Integer, Real, Boolean, Logical, String and Binary), aggregate types (i.e., Bag, List, Set, and Array), enumeration types and user-defined types, can be presented in schemas of EXPRESS. EXPRESS also has a powerful inheritance mechanism and can represent complex inheritance relationships and functions between product data. To leverage the semantic power of STEP, there is a desire to develop a combined strategy of STEP and XML, equivalently as the combination of EXPRESS and XML or EXPRESS/XML. The ISO 10303-28 (XML representation of EXPRESS schemas and data) launched in the late 2000 is a representative example towards this direction. Another main advantage of ISO 10303-28 is that it combines the rich semantics of EXPRESS/XML and the Web. Thus, it provides a simple way to describe product data for the Web, and therefore it is very useful for collaborative partners to use the Web to carry out product development. Currently, the published resources of ISO 10303-28 include: XML DTD (STEP PDM Schemas), specifications (product identification and classification), and technical overview of the STEP Object Serialization Early Binding (OSEB), which is
based on the EXPRESS data specification language which supports structurally object oriented inheritance.

(2) Web3D/XML

The large size of EXPRESS/XML files brings the problem of low-efficient data transferring across networks. On the other hand, product models are the most important knowledge and properties of product development companies. In some cases, companies are reluctant to share the models directly to avoid leaking their commercial secrets to their competitors. Due to it, the full potentials and benefits of collaboration cannot be realized. Active research has been recently conducted to develop a series of light-weight Web3D schemes to represent concise product models and execute them in the Web, such as VRML (Virtual Reality Modeling Language), X3D (eXtensible 3D), W3D (Web 3D), MPEG-4, U3D (Universal 3D), etc. These schemes retain the essential visualization information of proprietary product models to support display-based manipulations, such as rotation and zooming, annotation, mark-up, etc. With the new representation schemes, a lot of collaborative activities can be carried on, such as taking on design discussion, reviewing and remarking product, and conducting a customer survey to get design feedback as early as possible. Since only the visualization information is included in the schemes, crucial design information is secured.

An important trend is to combine the representation with XML. A typical example is X3D developed under the Web3D Consortium’s (www.web3d.org) standardization process. As a major upgrade from VRML, X3D retains a backwards compatibility with a huge base of available 3D content. A major feature of X3D is that it is XMLised. Nodes in X3D are represented in XML tags so as to take full advantage and potentials of XML on the Internet. Meanwhile, X3D utilizes an open profile/components-based architecture enabling custom-crafted scalable implementations. It has a layered and componentized architecture that enables extremely compact 3D clients. These clients can be extended with plug-in components to create standardized profiles with the functionality to meet the demands of sophisticated applications. X3D incorporates numerous advanced 3D techniques including advanced rendering and multi-texturing, NURBS surfaces, GeoSpatial referencing, Humanoid Animation (H-Anim), and IEEE Distributed Interactive Simulation (DIS) networking.

3.2 Service-centric integration

Several significant service-centric mechanisms for CPD integration have been reported, including interface-wrapping services, MAS services and Web services.

To address the inherent complexity of structures and interactions among software systems in a CPD environment, a common “component interfaces” mechanism has been developed, through which various systems can be redeveloped as services. Towards this direction, several competing technologies, such as Microsoft COM, CORBA and J2EE, have been launched. A work proposed by Liu [37] is a Microsoft COM interface-based framework to wrap and expose API functions of CAD/CAPP/CAM systems as services for remote invocation. The concept of developing standard interface specifications, namely the core interfaces, was proposed to encapsulate the functions of the systems as a generic service layer according to some international standards. Chao, et al. [9] proposed a CORBA-based framework as a wrapper to support communication between design and manufacturing systems. Within the CORBA-based framework, an Interface Definition Language (IDL) is used to define the interfaces of functional systems for services to realize their interoperation. The objective of the FIPER (Federated Intelligent Product EnviRonment) system (FIPER Project, www.fiperproject.com/fiperindex.htm) [3] funded by NIST is to develop a Web-based distributed framework for design analysis and product lifecycle support based on component mechanisms and configurable workflow mechanisms. It can provide open and flexible capabilities to incorporate existing analysis and design tools/methods through Java-based wrapping mechanisms including Java Native Interface (JNI) and the FIPER SDK toolkit. A workflow for a design process can be conveniently organized and configured by users through assembling components in the distributed environment. Xiao, et al. [53] developed the Web-DPR system as an infrastructure to support collaborative design and manufacturing. Based on the Java Remote Method Invocation (RMI) mechanism, agents and an event-based mechanism, the functional modules of the systems can be linked and coordinated effectively. In the Web-based fixture design and manufacturing system presented by Mervyn, et al. [39], Java-RMI (one of the J2EE technologies) was used to wrap the remote fixture design and manufacturing methods as services for remote invocation and manipulation. In order to alleviate the complexity of the whole system caused by the add-on wrapping structures and the huge programming effort for implementation, Li, et al. [34] proposed a multi-layer wrapping mechanism based on Java Servlets (one of the J2EE technologies) to extract some common wrapping parts from different functional modules to form an independent middle layer. This middle layer plays as a communication channel with the wrapped functional modules and
other services in the Internet environment, and the weights for each functional service can be reduced.

The aforementioned interface-wrapping mechanism setup a foundation to build up other advanced mechanisms such as MASs and Web services. A MAS, which is a self-adaptive distributed artificial intelligent technology to organize simple systems into complex one, provides an advanced service integration solution for collaborative systems. Existing and legacy software systems can be encapsulated as agents to integrate design, planning, scheduling, simulation, execution, and product distribution, with those of their suppliers, customers and partners to collaborate in an open and distributed intelligent environment via networks. A Web service, which is based on XML schemas and a communication protocol - SOAP (Simple Object Access Protocol) to provide a neutral data exchange format, can be published, located, and invoked across the Web. Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service. The developments of MASs and Web services are towards more standard to define services and their communication. For example, MASs specify a KQML (Knowledge Query and Manipulation) language for communication, and Web services use XML for realizing more generic communication. On the other hand, the interface-wrapping mechanism is just for developing and deploying services in the network, while in MASs and Web services, coordination mechanisms are one of the essential components to support collaborative activities. From this aspect, they are facing the same or similar research challenges with those collaborative design paradigms mentioned earlier, including system architectures, communication and cooperation, dynamic system reconfiguration, knowledge representation, conflict resolution and management, distributed dynamic scheduling, etc. Some developed systems based on MASs and their characteristics are summarized in Table 2.

### Table 2: Some agent-based systems for CPD

<table>
<thead>
<tr>
<th>Works</th>
<th>Functional Characteristics</th>
<th>System Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shen, et al.</td>
<td>A MetaMorph agent architecture ensuring the coordination among design parts and resource agents to support distributed design and manufacturing activities.</td>
<td>The architecture is a mediator-centric hybrid agent organization. AutoCAD with AME 2.0 is used to support product design. TCP/IP protocol is used to support high-level KQML communication among agents.</td>
</tr>
<tr>
<td>Gerhard, et al.</td>
<td>An event-based and agential framework to communicate design and manufacturing information through agent channels, and manufacturing analysis functions are enveloped as agents to support the establishment of an open and plug-in environment.</td>
<td>Java RMI technology is used to establish the agent infrastructure. Exchanged information is wrapped as events for communications in the environment.</td>
</tr>
<tr>
<td>Sun, et al.</td>
<td>An agent architecture integrating design, manufacturability analysis, process planning and scheduling.</td>
<td>The multi-agent organization is based on JATLite multi-agent system. TCP/IP protocol is used to support high-level KQML communication among agents.</td>
</tr>
<tr>
<td>Kotak, et al.</td>
<td>A framework for holonic design and operations based on three functional modules – holonic control, virtual simulation and human/system integration.</td>
<td>The multi-agent organization is based on JADE platform, which is an open-source Java agent development framework.</td>
</tr>
</tbody>
</table>

### 4. CONCLUSIONS AND FUTURE TRENDS

Research and development have been actively carried out to develop technologies and methodologies to support CPD environments. As two key mechanisms of CPD, collaboration and integration can ensure people to work across multiple functions, organizational boundaries, disparate geographies, time zones and languages, and to deliver the right data of product development systems that correct information to the right person or system in the right format. In this paper, a summary is given for some developed collaborative and integration mechanisms of CPD. The collaborative mechanisms are discussed in collaborator awareness, negotiation and conflict management, collaborative decision-making, development process management, team management, etc. The integration mechanisms are investigated from two levels, i.e., a data-centric integration level, which combine modern markup and Internet-compatible language and product exchange standards, and a service-centric integration level, in which services are developed to build up a dynamic CPD environment.

In summary, research of CPD is on the following aspects:

- Enterprise integration for the distributed organizations and systems. Manufacturing companies and enterprises can be integrated with their distributed systems and partners, such as customers and suppliers, via networks to establish an CPD for product development.
For instance, the distributed design and manufacturing analysis systems can send the demands and requirements from the customers directly to the design department of a company to support global competitiveness and rapid market responsiveness.

- Heterogeneous environments and interoperability of software tools. A CPD will allow the integration and interoperability of heterogeneous software and hardware. Information environments and legacy systems in companies are usually based on different programming languages, representation languages and models for product information, and computing platforms. To achieve an effective and efficient interoperation and interaction of sub-systems and software components in such heterogeneous environments, automatic information conversion and interpretation capabilities are necessary to realize obstacle-free information communication and workflow control.

- Open and scalable computing structure and services. There is a need to provide possibility to dynamically integrate new sub-systems into or remove existing sub-systems from an ICE-enabled product development with high convenience, security, reliability and without stopping and re-initializing the entire environment. New kinds of service architectures to wrap software tools have to be developed to incorporate them into the environment as required so as not to interrupt organizational links previously established.

- Cooperation between humans, and between systems and humans. People and software systems need to work at various levels of collaboration, and with rapid access to knowledge and information repositories. Bi-directional communication infrastructures are necessary to allow effective and quick communication between systems or between humans and systems to facilitate their interactions.

The future trends for CPD the relevant mechanisms include, but not limited to:

1. Optimization of the selection and integration of various collaborative mechanisms. Many collaborative mechanisms are complementary in functions. It is important to establish a CPD environment equipped with convenient, effective and economic collaborative mechanisms to meet the requirements of the interpersonal linkage among people not only in design but also in the whole lifecycle of products. More research is required to develop intelligent strategies and leverage existing artificial intelligent technologies to make the organization of people and systems in a more autonomous, robust and mobile way. Meanwhile, the mechanisms should be smarter to learn from the past experience and surrounding environment.

2. Security and Interoperability of data and information. As customers, suppliers and designers from different places move to Internet-based collaboration, security must be considered carefully. While much of the security solutions offered by the current collaborative systems will be handled at the transmission level, they can also benefit by incorporating additional security features into their data models. Enhanced interoperability between systems should be achieved. The enhanced interoperability is not only for the heterogeneous collaborative systems, but also for the collaborative systems and other enterprise systems, such as ERP (Enterprise Resource Planning), portfolio management, PLM (Product Lifecycle Management), need to be achieved. Another significant goal for the interoperability must include the possibility to access and manipulates legacy enterprise systems and various product development data in their native file formats for various applications.

REFERENCES


37. Lan, H. B., Ding, Y. C., Hong, J., Huang, H. L.
ABOUT THE AUTHORS

Dr Weidong Li is now working at the Coventry University as a senior lecturer. His research areas are collaborative product development, CAD/CAM/PLM. In the areas, he published 2 books, and about 50 research papers in international journals and conferences. He has worked with various companies for industrial applications.

Dr Rajkumar Roy is now a professor working at the Coventry University. His research areas are cost engineering and product service systems.

(Received May 2006, revised November 2006, accepted April 2007)
Collaborative product development (collaborative product design) (CPD) is a business strategy, work process and collection of software applications that facilitates different organizations to work together on the development of a product. It is also known as collaborative product definition management (cPDM). Collaborative product development (collaborative product design) (CPD) is a business strategy, work process and collection of software applications (More). Wikipedia. Quired products, an order is signed and submitted to the retailers, and the retailers contact with the enterprise according to the order products; if the enterprise stock fails to meet the needs, get ready to produce the products listed in the order, and meanwhile contact with first and second level. suppliers to get spare parts; and the suppliers contact with raw material suppliers to purchase materials for production; in the entire business flow, logistics enterprises should provide logistics channels [5]. A typical supply chain network is shown in Fig.1. Figure 1. Supply chain network. Development and reuse [15]. Customize the agent of different granularity according to the specific roles of these entities in the supply chain. Figure 5. A multi-agent SCM system model based on the physical entity.