World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:5, No:11, 2011

A Collaborative Framework for Visual Modeling on Web 2.0

Song Meng, Dianfu Ma, Yongwang Zhao, Jianxin Li

***Abstract*—**Cooperative visual modeling is more and more necessary in our complicated world. A collaborative environment which supports interactive operation and communication is required to increase work efficiency. We present a collaborative visual modeling framework which collaborative platform could be built on. On this platform, cooperation and communication is available for designers from different regions. This framework, which is different from other collaborative frameworks, contains a uniform message format, a message handling mechanism and other functions such as message pretreatment and Role-Communication-Token Access Control (RCTAC). We also show our implementation of this framework called Orchestra Designer, which support BPLE workflow modeling cooperatively online.

***Keywords*—**colllaborative framework; visual modeling; message handling mechanism

I. Introduction

JOBS are not as simple as we considered. OK and affirmative is our visual modeling work, which needs more rules, steps and participants, has become much more complicated than ever before. Nowadays, almost every large visual modeling work has a requirement that participants must be able to complete work in teams and collaborate with coworkers effectively[1]. Since participants are often separated geographically, a work environment allows visual modeling cooperatively among participants is required.

Collaborative visual modeling is a new kind of product development way[4], take aircraft designing and modeling for instance, which allows experts from at least two different fields to increase corrective rate of decision, decrease rework times, and speed up decision-making by exchanging aircraft-designing information, using cooperation mechanism, and sharing information and knowledge of aircraft-designing, and thus improve modeling efficiency. Compared to local visual modeling, cooperative visual modeling faces new challenges such as user management, access control of resource, operation-conflict handling, message pretreatment, and so on. One effective way of resolving these challenges is provision of collaborative framework for establishing collaborative environment.

Traditional cooperative environments such as PC Anywhere, VNC (Virtual Network Computing) and NetMeeting has explored much and implemented many functions in collaborative fields, such as remote access, remote control, whiteboard, and point to point audio and video[5, 6, 7]. But under Web2.0 environment, they has exposed many weakness, for example, limit the number of participants in cooperation, region limitation of system running, poor performance of cross platform, and so on.

To overcome these obstacles, we need to build a new kind of cooperative visual modeling platform, which can be used anytime, anywhere, by anyone, and all it need is a browser. A collaborative framework contains message transmission mechanism, error handling, role allocating and pattern matching in cooperation is what this platform could be built on.

This paper provides such a visual modeling framework for collaborative platform which has following features. First, this paper set a message format for collaborative modeling by using XML Schema. Second, this paper establishes a mechanism of message notification which supports variety ways of notification and variety kinds of devices. Third, this paper approach supports the ability of message authentication and resending message while missing to ensure the integrity of cooperation. Forth, this paper shows a way of access control base on Role-Communication-Token for controlling cooperation.

The remainder of paper is organized as follows. Section 2 surveys related work. Section 3 presents a collaborative visual modeling message format and message-transmitting mechanism for cooperation. Section 4 describes how to implement the framework and shows the application based on our framework. Finally, Section 5 concludes this paper.

II. Related Work

Many IT companies have developed visual modeling tools for designing and modeling workflow. Oracle BPEL Process Manager (OBPM) is a visual modeling tool and it supported equipping a series of synchronous and asynchronous services to a BPEL workflow[20]. OMPM provides a more reliable solution for developers on designing, deploying and managing BPEL workflow. WebSphere Integration Developer is an Eclipse-based business integration module development tool providing visual drag or drop programming. It can include a Web Services compliant service and provide a set of tools and process for supporting top-down or bottom-up development model[21]. ActiveBPEL Designer is a commercial modeling tool for designing and developing WSBPEL process based on Eclipse which provides a open-source WSBPEL execution engine called ActiveBPEL[2]. These tools have a very authoritative status in their professional fields, but they do not focus on cooperation in modeling.

The popularity of internet has lead to several researches about web-based cooperation, especially focus on access

Song Meng is with Department of Computer Science, Beihang University, Beijing, China (e-mail: mengsong@act.buaa.edu.cn).
Dianfu Ma is with Department of Computer Science, Beihang University, Beijing, China (e-mail: dfma@nlsde.buaa.edu.cn).
Yongwang Zhao is with Department of Computer Science, Beihang University, Beijing ,China (e-mail: zhaoyw@act.buaa.edu.cn).
JIanxin Li is with Department of Computer Science, Beihang University, Beijing ,China (e-mail: lijx@act.buaa.edu.cn).

control, some of which are summarized below. RBAC[8] (Role-Based Access Controls) can be used as the basis for a common definition in platform based on user roles, which is a type of non-discretionary access control that is more central to cooperation processing. Peyman[9] present an approach for access control mechanism within collaborative working environments based on trust, context, and social networks. This type of access control is more suitable than RBAC for internet environment which enables users to express and apply flexible access control rules based on their relationships with other participants and context information. [22] propose an access control model and mechanism for grid-based collaborative applications. It provides an access control model built around a service or resource provisioning agreement. These methods are very classical in the field of Access Control, but they can't fit certain environments such as collaborative visual modeling environment.

Also, framework and Architecture of cooperation are discussed in scientific research. A framework[11] for collaborative is presented which enables collaborative participants anywhere in the world can collaborate efficiently on the internet without the need of a powerful client machine. The framework includes a hierarchical collaborative schematic editor based on Java with interfaces to distributed Web tools and cell libraries, infrastructure, protocols for tool communication, message passing and collaboration. An agent-based service-oriented integration Architecture[12] is presented for manufacturing enterprise collaboration in general. It provides a unified framework to support sophisticated dynamic and automatic services collaboration, and integrates software agents and Web Services for manufacturing enterprises. A multi-agent-based service-oriented architecture (SOA)[3, 13, 14] for inter-enterprise cooperation system has been proposed which establishes the basis for transforming inter-enterprise cooperation business models into multi-agent-based SOA components, utilize existing web services standards UDDI specifications and design UDDI service agent tool for extension. These cooperation frameworks are famous in cooperation field, but they pay less attention on cooperation modeling.

Although numerous researches on collaborative Architecture are available, there has limited attention on designing a collaborative framework, which focuses on internet environment and could be compatible of various kinds of message notification, ensure integrity while transmitting, enable to establish collaborative platform and so on. To fill these gaps, we propose a web-based visual modeling cooperation framework. Our primary contribution in this work is the presentation of a collaborative mechanism which contains RCTAC access control, message integrity, several cooperation models, multitasking and various kinds of message notification support.

III. APPROACH

A collaborative environment is essential for participants in visual modeling fields who coordinate works from different regions. A framework for building such environment is presented in this section. It includes how to format message while cooperating, what kind of message transmission mechanism should be provided in, how to check and deal with incomplete message in cooperation, and which kind of access control method should be used.

A. Overview

To ensure web applications running in cooperative environments, a cooperation component set is need for communicating by sending, receiving, and handling messages from users who use several kinds of device. Cooperation component set is built on the cooperation framework showed in Fig.1 and it interactive with business logic of Application.

Data Processing Component is a direct connection part with Business Logic of Application in Cooperative Framework. The Component contains two parts, Data Classification and Info Assembler. The former is for classifying data for Business Logic from Cooperation producer. The latter is for collecting information from Business Logic for sending message to other Cooperation consumers. To control the access of cooperation resources, Access Control Component with following functions is need: Token Distribution and Role Adjudication which reduce conflicts by resources access. Message pretreatment part, which contains packet-loss checking and message integrity checking, is necessary for both sending message and receiving message. It could avoid loss caused by Network transmission fault. Notification Component is designed for receiving/sending messages, adapting message from several sources such as PDA, PC, PPC, even Mobile Phone and converting to united code while communicating with others. To assist communication in cooperation, Cooperative Tools present some tools such as IM, E-mail, and E-Whiteboard which can be used in the process of cooperation.

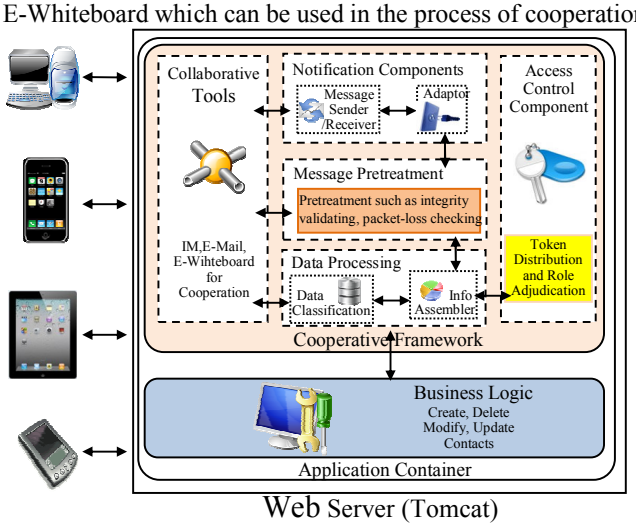

Fig. 1 Framework for collaborative visual modeling

B. Message Format

When cooperative message is received from server, certain operations will be executed by information which is carried on the message. To decrease network data transmission, improve cooperation efficiency, and avoid inconsistency caused by different origins of message, a uniform XML Schema format of message is required for collaborative framework based on Web. Fig.2 shows the XML Schema for message we provided.

The structure of message is divided into three parts: header, which contains description data; body, which contains application data; and others which contains affiliated data.

Description data is a field for describing message. It contains Sender Indentify Information (SII), Interaction Method (IM), Channel (CHN), Topic (TPC), and Protocol (PRT). SII is identification information which can uniquely determine sender of the message such as ID; IM is a method of interaction while message transmitting as HTTP, Web Service or RPC (Remote Procedure Call); CHN is a transport tunnel for message transmitting; TPC is to ensure receiving certain message from related topic when multi topics exists in the channel; and PRT is to identify the protocol of the message notification.

Application data is the main field of cooperation message. It contains Project ID (PID), Operation (OP), Priority (PRR), Object ID (OID), and Object Value (OV). PID identifies project which the cooperation message belongs to; OP shows the operation type of message; PRR is the priority of operation which can help sorting the sequence of operation when multi messages arrived in one time; OID is ID of Object(s) which is(are) operated; OV is new value(s) of Object(s).

Affiliated data is the error-checking filed for cooperation which contains Integrity Information (ITI), Message Number (No.), and Others (OTH). ITI stores integrity-checking data of message such as MD5 code; No. is the sequence number of message which can help receiving message from server again when message in error; OTH stands for other field for integrity-checking such as code type of ITI.

C. Message Handling Mechanism

In our cooperative Framework, message handling mechanism is a key part to ensure framework working stably and efficiently. It should resolve such problems: How to notice other participants? How to combine messages from various devices and transport protocols? How to ensure integrity of message? What to do with packet-loss? How to handle messages after message-pretreatment? How does Access Control Module work? How to interactive with the business logic?

Major mechanism of message handling is described as Figure 3.

Pre-Processing Component is the outer layer where client and Server interact. The component contains notification components which for sending and receiving messages. Adaptor component is for adopting message to various devices. And message pretreatment is for message integrity and sequence checking.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema">
  <element name="Description_Data" id="Header">
    <complexType>
      <sequence>
        <element name="Application_Data" type="string" id="SII"/>
        <element name="Interaction_Method" type="string" id="IM"/>
        <element name="Channel" type="string" id="CHN"/>
        <element name="Protocol" type="string" id="PRT"/>
        <element name="Topic" type="string" id="TPC"/>
      </sequence>
    </complexType>
</element>
  <element name="Application_Data" id="Body">
                                    <complexType>
      <sequence>
        <element name="Project_ID" type="string" id="PID"/>
        <element name="Operation" type="string" id="OP"/>
        <element name="Priority" type="string" id="PRR"/>
        <element name="Object_Value" type="string" id="OV"/>
        <element name="Object_ID" type="string" id="OID"/>
      </sequence>
    </complexType>
  </element>
  <element name="Affiliated_Data" id="Others">
    <complexType>
      <sequence>
        <element name="Integrity_Information" type="string" id="ITI"/>
        <element name="Message_Num" type="string" id="No"/>
        <element name="Others" type="string" id="OTH"/>
        <element name="Body" type="string"/>
      </sequence>
    </complexType>
  </element>
</schema>
```

Fig. 2 Message format for collaborative framework

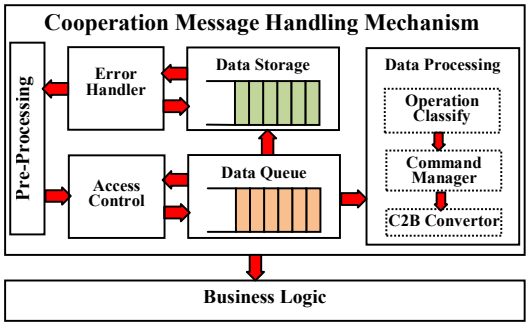

Fig. 3 Message handling mechanism for cooperation

The message will be divided into two ways after being treated by message pretreatment component. If message passed pretreatment, it will be send to Data Queue, waiting for data-catching. And in the meanwhile, a backup of message will be added into Data Storage to support Error Handler. If message was detected as uncompleted or packet-loss, it will activate Error Handler, which will ask server to resend message or catch data from Data storage for rolling back.

Once caught from Data Queue, data will be sent to Data Processing Component, which contains Operation Classification, Command Manager and Command to Business (C2B) Convertor Module. After classified by Operation Classification Module and command-switched by Command Manage, data will enter C2B Convertor Module, which interactive directly with business logic of the application, for completing cooperation.

Access Control Component is necessary before catching message from Data Queue and pushing message into Data Queue. Access Control Component is for ensuring security of data access. In our framework we use token, which is managed

by Token Distribution Module, to lock data from Data Queue. Role Adjudication Module will judge users into different user groups, which have different access-authority, by token they get and each group related one certain mode, user will enter modes by Mode Matching Module.

D. Message Pretreatment

Fig.4 illustrates a message pretreatment mechanism we presented. As soon as message arrives at destination, Message Pretreatment is activated as flow below. One process is to check packet-loss by validating the sequence of message. If the sequence is not consistent, Packet-loss Handler will request Error Handler to roll back message. The other process is to check whether the message is integrated by contrasting verification data unpacked by affiliated data from message and verification data produced by receiver. If those data is not consistent, Verification Error Handler will request Error Handler to resend message. After pretreatment, no-error message will be carried to Resource Handling Model for storing or sending to Data Processing Model.

E. RCTAC

Access Control is a component which enables an authority to control access to resources in system[10]. So how to control nodes' access rights is an important content in collaborative designing, because nodes will compete for resources when their self-governance is strong. So in Fig.5 we present RCTAC, a method of access control based on Role-Communication-Token.

Users will be grouped into Role Groups by usernames they use when log in, and each Role Group has a set of authorities. Then users can apply for token which stand for the authorities in set. When system has token undistributed, users will get token and be project-master in cooperation. If not, users have to wait for token until it is released. In addition, users can use tools to communicate with others, getting the token he wants or discussing anything about token. In RCTAC, operation will only be activated by users who have related token in cooperation. Thus RCTAC can make cooperation process more orderly.

IV. IMPLEMENT AND EXAMPLE

In this section, we describe how to implement the collaborative visual modeling framework to server and client, and show the architecture of client and server. Then we introduce an example based on our framework.

A. Server Architecture

The Architecture of server is illustrated in Fig.6. In this architecture, message transmitting between users is based on XML, a standard of data exchanging among applications over the Web[15]. This message follows the format described above, which can ensure performance and transmission-speed on PC, Mobile Phone, even PPC. Major components of server architecture are described in Fig.6.

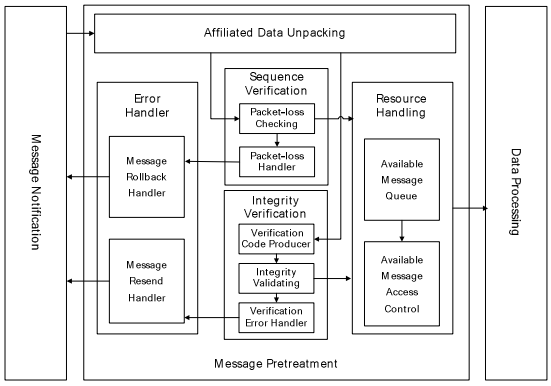

Fig. 4 A message pretreatment mechanism

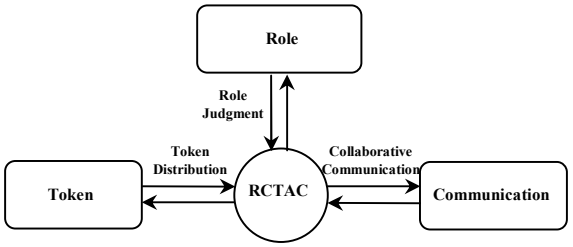

Fig. 5 Mechanism of RCTAC

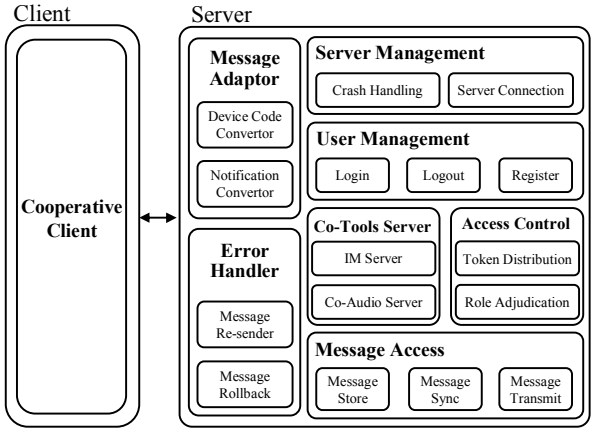

Fig. 6 Server Architecture

The architecture of server is designed for seven parts: Server Management, Message Adaptor, Error Handler, User Management, Message Handler, Access Control, and Co-Tools (Collaborative Tools) Server. Server Management contains Crash Handling which detecting and handling the server while server in crash, and Server Connection which for setting connection or disconnection between client and server. To handle user interactive with server, User Management provides login, logout, and register on sever for simple management. Co-Tools Server supports some servers of tools while in cooperation such as E-mail, IM, E-Whiteboard, and so on. Access Control provides some management module on access limitation such as token management, user group management and model management. Message Access allows clients set or get message from server, it contains message storage module, message synchronization module and message transmit

module. Message Adaptor is a coordinator between server and several kinds of clients by transcoding message to various kinds of devices and translating message to certain notification type. Error Handler deal with the messages which are detected error in clients. It provides two ways of error handling: resending message when message is incomplete and rolling back message when packet-loss in transmitting.

B. Client Architecture

The Architecture of client provides the function of interactive with cooperation server and local business logic, which is illustrated in Fig.7.

The architecture of client is designed by six parts: User Interface, Pattern Matching part, Notification Component, Data Processing Part, Message Transmitting Part, and Channel Set Part.

User Interface is the interface of cooperative client. Pattern Matching is a set of cooperation toggle, by which users can select single cooperation modes or combine multi modes from modes such as Silence Mode, Sign Model and Compare Mode. Notification Component plays a role of connection which controls message flowing between clients and servers. Data Processing has two components: One is unpacking message from receiving message; the other is packing information on message before sending message. Message Transmitting suggests translation between message and business logic data. Channel Set collects transmitting channels and relates invoking functions which build the way between server and client.

C. Example

Nowadays, we have many workflow modeling tools to resolve the united-rules-problem between participants while transforming documents, information and missions in the fields of E-Commerce, Office Automatic, Finance, and Communication. But with complexity growing with the work, cooperation is necessary. So how to make departments in different areas take part in one modeling process and how to realize the collaboration between business staff and technical staff has become an important feature needs to be improved.

An example of application based on collaborative framework we presented resolving the problem above is given in this section, whose name is Orchestra Designer (OrDesigner)[18]. OrDesigner is an online BPEL modeling tool which provides a more business-oriented functional view of generating BPEL (Business Process Execution Language), an OASIS standard executable language for specifying actions within business processes with web services[19], than pure BPEL view and supports modeling collaboratively. OrDesigner is based on Flex programming language, which is a software development kit released by Adobe Systems for the development and deployment of cross-platform rich Internet applications based on the Adobe Flash platform[16,17].

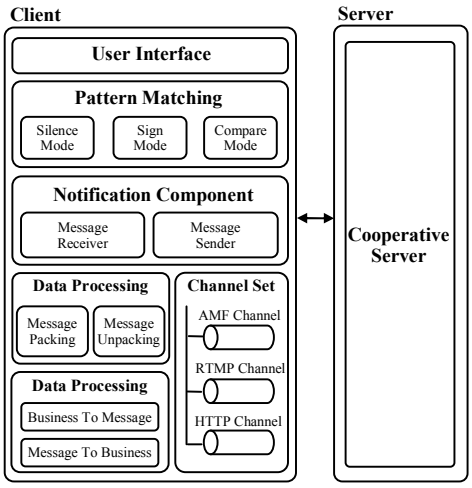

Fig. 7 Client Achritecture

Based on our framework, OrDesigner provides three cooperative modes: Silence Mode, Sign Mode, and Compare mode. In Silence Mode, cooperation works on background, multi-thread-modeling is supported in this mode so that observer of cooperation can do another project modeling while one modeling project in cooperation. In Sign Mode, figures which were modified in cooperation will be marked by red box(s). In Compare Mode, figures belong to last state are showed semi-transparently on figure canvas after they are modified just now.

In Fig.8, user B has built a simple loan approval Web Service that provides a port where customers can send their requests for loans, but it has problem of one case process handling. User B invited his workmate User A who works in different place to join the cooperation to check the modeling, and User B enter Sign Model and User A get the token to modify the model. After User A fixing the problem, one figure was encircled by a red box on Use B's figure canvas.

V. Conclusion and Future Work

This paper has presented a collaborative framework for visual modeling based on web2.0 environment. A uniform format of message based on XML Schema is proposed to reduce data transmission costs and ensure message handled more efficiently. Based on the uniform-message we present a message handling mechanism on web environment. Message pretreatment module and RCTAC module, as parts of collaborative framework, can ensure message no-error before handling and control the access to resources for reducing resources access conflict. The framework has been applied in some collaboration system and we showed a demonstration of Orchestra Designer, which is an online workflow designer that generates BPEL process and supports cooperation.

In the future, we plan to extend our framework in supporting user management and handling message-conflict to emphasize expansibility and stability.

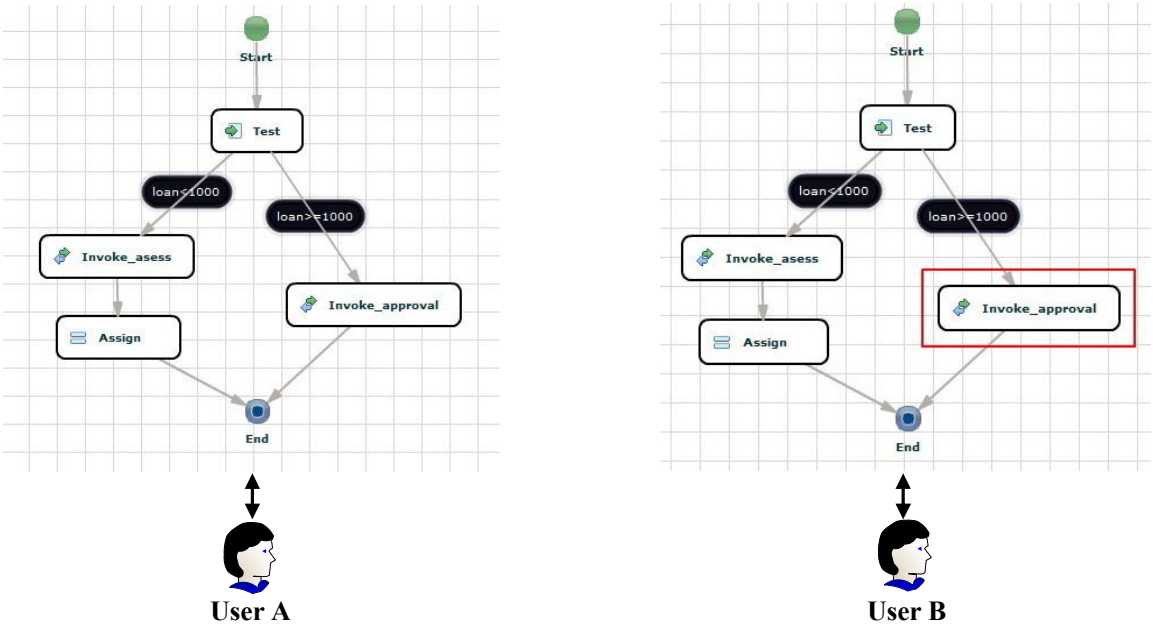

Fig. 8 Snapshot of Or Designer

REFERENCES

[1] K. Figl, M. Derntl and S. Kabicher, "Visual Modeling of Competence Development in Cooperative Learning Settings," Lecture Notes in Computer Science, 2009, pp.176–185.
[2] A. Marconi, M. Pistore, P. Poccianti, and P. Traverso, "Automated web service composition at work: the amazon/mps case study, " ICWS, pp. 767–774.
[3] Y. Zhao, D. Ma, C. Hu, M. Liu, Y. Huang, "SOCOM: A Service Oriented Collaboration Middleware for Multi-User Interaction with Web Services based Scientific Resources" , in Proceedings of the Sixth international Symposium on Parallel and Distributed Computing (ISPDC'07). IEEE Computer Society, Washington, DC, 2007.
[4] F. Detienne, "Collaborative design: Managing task interdependencies and multiple perspectives, " Interacting With Computers, 18, 2006. 1–20.
[5] M. SATYANARAYANAN, B. GILBERT, M. TOUPS, N. TOLIA, A. SURIE, D. R. O'HALLARON, and et al., "Pervasive Personal Computing in an Internet Suspend/Resume System, " IEEE Internet Computing 11, 2 (2007).
[6] Shen WM, Wang LH, Hao Q, "Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey, " IEEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and Reviews , 2006 ,36(4):563–77.
[7] Microsoft. Microsoft Windows NetMeeting 3 SDK[EB/OL]. http://msdn2.microsoft.com/en-us/library/ms708704.aspx,2007
[8] Q. Wei, J. Crampton, K. Beznosov, and M. Ripeanu, "Authorization recycling in RBAC systems, " in Proceedings of the thirteenth ACM Symposium on Access Control Models and Technologies (SACMAT). Estes Park, Colorado, USA:ACM, June 11–13 2008, pp. 63–72.
[9] Peyman Nasirifard, "Context-Aware Access Control for Collaborative Working Environments Based on Semantic Social Networks, " Sixth International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'07), Roskilde, Dinamarca.
[10] Access Control, http://en.wikipedia.org/wiki/Access_control
[11] X.F. Zha, H. Du, "Knowledge-intensive collaborative design modeling and support," Part I. Review, distributed models and framework, Computers in Industry, 57 (1) (2006), pp. 39–55.
[12] Weiming Shen, Qi Hao, Shuying Wang, Yinsheng Li, and Hamada Ghenniwa, "An agent-based service-oriented integration architecture for collaborative intelligent manufacturing. " Robotics and Computer-Integrated Manufacturing, " 2007, 23(3),pp. 315–325.
[13] X. Liu, "A Multi-Agent-Based Service-Oriented Architecture for Inter-Enterprise Cooperation System, " In: Proceedings of the Second international Conference on Digital Telecommunications (ICDT'07). IEEE Computer Society, Washington DC (2007)
[14] H. Demirkan, R.J. Kauffman, J.A. Vayghan, H.G. Fill, D. Karagiannis, P.P. Maglio, "Service-oriented technology and management: perspectives on research and practice for the coming decade," Electronic Commerce Research and Applications 7(4) (2008), pp. 356–376.
[15] Serge Abiteboul, Omar Benjelloun, and Tova Milo. "The Active XML project: an overview, " VLDB J., 2008, 17(5), pp. 1019–1040,.
[16] Adobe Flex, [OB/EL], http://www.adobe.com/products/flex/.
[17] Dou Sun, Zhuqing Li, Yongwang Zhao, Dianfu Ma. "Orchestra Designer: an open-source tool for scientific workflow modeling", Open-source Software for Scientific Computation (OSSC), 2009 IEEE International Workshop on, 2009, pp. 39-43
[18] Orchestra Designer, http://www.trustie.com/projects/project/show/OrchestraDesigner
[19] BPEL, http://en.wikipedia.org/wiki/Business_Process_Execution_Language
[20] F. van Breugel and M. Koshkina, "Models and Verification of BPEL," Technical Report, York University, Toronto, M3J1P3, Canada (2006)
[21] J. Noel. "BPM and SOA: Better Together," IBMWebsite,WhitePaper, 2005.
[22] Demchenko, Y., Gommans, L., Tokmakoff, A., van Buuren, R.: Policy Based Access Control in Dynamic Grid-based Collaborative Environment, in International Symposium on Collaborative Technologies and Systems. 2006: IEEE Computer Society.