Unified Modeling Language (UML)-based CASE Tool for Computer Supported Collaborative Work:

Riri Fitri Sari, Bagio Budiardjo, Kalamullah Ramli, Astha Ekadiyanto, Stefan Werner, Axel Hunger
riri@eng.ui.ac.id, bbudi@eng.ui.ac.id, k.ramli@eng.ui.ac.id, astha@eng.ui.ac.id, swerner@uni-duisburg.de, axel@uni-duisburg.de

1Department of Electrical Engineering, Faculty of Engineering, University of Indonesia, Indonesia
2Faculty of Engineering, University of Duisburg-Essen, Duisburg, Germany

Summary
This paper presents our work on providing tools and platform for Pervasive Multimedia Learning Environment. We review the state of the art technology in the field of tele-education, particularly on CSCW tools and pervasive computing. We develop a Passenger-UML which is a Computer Supported Collaborative Work (CSCW) tool to support Computer Aided Software Engineering (CASE) based on the latest Unified Modeling Language (UML) approach. Performance evaluation and additional functionality to Passenger-UML such as change graph has been conducted and reported in this paper.

Key words: CASE tool, Multimedia e-Learning, Passenger, CSCW

INTRODUCTION
The University of Duisburg Essen (UDE) Germany, the Universitas Indonesia (UI), and the Universiti Kebangsaan Malaysia (UKM) have established a joint research group (JRG) in the field “Technologies and Applications for The Future Internet”. From the year 2002, within this field the JRG concentrates on the topics: (1) knowledge based Internet System, and (2) the Internet communication technologies.

The objectives of the collaboration are to facilitate live tele-teaching and to conduct software engineering laboratory practice with distributed student teams. An important character of this cooperation is the fact that the institutions are connected each other through devices and environment with heterogeneous bandwidth capacities. Multimedia applications connection to remote places needs mechanisms that can guarantee fairness in terms of bandwidth capacity and the quality of service. Some work on this issue has been conducted.

In our research we address the prominent technologies to provide facilities for tele-education with streaming of multimedia applications containing lecture videos between these partners who have different bandwidth accesses and capacities. Those technologies are pervasive computing, active networks and computer supported collaborative works.

This paper is organized as follows. First, we introduce the notion of pervasive multimedia learning environment. Subsequently we explain our work on content development, software called Passenger-UML which is a Computer Supported Collaborative Work (CSCW) tool to support Computer Aided Software Engineering (CASE) based on the latest Unified Modelling Language (UML) approach.

PERVASIVE MULTIMEDIA LEARNING ENVIRONMENT

We define multimedia pervasive learning as an interactive teaching model that gives mobile students access to a wide range of educational resources—teachers, peers, contents such as lecture notes, readings and exercises, and various mode of media delivery such as voice and video, independent of place and time. It leverages computing, communication, and multimedia technologies to create richer learning environments and support more flexible modes of interactions, more scalable, and more cost-effective compared with the standard classroom or lecture hall.

We believe that, in the near future, the popularity of pervasive learning environments in both professional and academic settings will steadily increase due to the rising demand from traditional students as well as working adults for postsecondary
and professional education, advances in information technologies, the Internet’s ubiquity, the emergence of high-capacity wire and wireless networks, as well as the pervasiveness of networked personal computers in both businesses and homes.

Our pervasive learning multimedia environment consists of four main components. Those components are audio video tools for teleconference purpose, shared working space to support collaborative works, mobile IP component to support mobility of users, and pervasive devices to allow access anywhere, anytime and on-demand. The component diagram is shown in Figure 1.

![Component Diagram](image)

**Fig. 1** Components of the Environment

Pervasive computing, or known as location-aware computing in which information and communication technology will be an integrated part of our environments. In some cases, a pervasive computing environment has some integrated processors, sensors, and actuators connected via high-speed networks and combined with new visualization devices ranging from projections directly into the eye to large panorama displays.

To support the delivery of multimedia content to clients with heterogeneous bandwidth we use our own active network technique called Friendly Active Networks (FANS). FANS covers the functionalities required for these applications: bandwidth adaptability, mobility support, tracking mechanism, and communication and media transfer capabilities. Moreover FANS supports applications to include mobile devices provided that their URL addresses are announced. With its tracking and transcoding capabilities FANS applications and possible research extension can include pervasive devices or users.

**CASE TOOLS FOR COMPUTER SUPPORTED COLLABORATIVE WORK**

In order to provide an advanced e-learning environment, we develop a Computer Aided Software Engineering tools which support on line and collaborative works. Our CASE tools utilize the state of the art technology which provides a fast, efficient, and reliable connection to bridge differences in facilities. This multiparty collaborative software product is based on Passenger CASE tool [2, 3, 4, 5] from the University of Duisburg Essen, Germany. The system provided a platform that supports geographically distributed students and lecturers.

The Passenger software is purposely built for aiding educational process in Software Engineering area. It provides facility for synchronous communication such as the publicly available package such as Microsoft Netmeeting or CU-See-Me, but facilitating effective collaborative work for educational purposes. It has been tested that Passenger, the software engineering CASE tool for synchronous work have more functions than publicly available tools. Passenger provides a better support for discussions (chatting mechanism), the setup of partners’ images and relationship management. It also provides mechanisms for preventing communication breakdowns. Overall this software which originally support multiparty to draw Ward and Mellor structured programming diagrams, support inter-students and trainer cooperative actions.

In our work on contributing to Passenger, we provided extra facilities and improvement to cater the newly develop de facto standard of software engineering approach such as UML (Unified Modelling Language) to Passenger [6].

UML supports object-oriented technology. Previously some Object Oriented methods have been used world-wide, such as Object Modelling Technique (OMT) of Rumbaugh, Object Oriented Analysis/Design (OOAD) from Shlaer-Mellor, Booch methods. Responsibility-Driven Design/Class Responsibility Collaboration (RDD/CRC) of Wirfs-Brock, and Object
Oriented Software Engineering (OOSE) [6, 7, 8, 9, 10, 11]. Each method has its own notation to model a system. The notation should be expressive and provide consistent interpretation to the software developer which read them.

A single standard language was considered necessary to be provided since mid 1990s to prevent miscommunication of software design. The Object Management Group (OMG) eventually made a consortium from some organization to create a single language called Unified Modelling Language (UML) which becomes a de facto standard for a well define modelling and can be used for variety of purposes [6].

PASSENGER UML

Object oriented programming normally used to create software as a group of unique object in which the object consists of attributes and operation which complement each other. The main construction of object oriented programming consists of attribute (something own by the object) and operation (what the object can do, determined as procedures and functions in software).

The software engineering concept brought forward by the implementation of UML into passenger brings the advantage such as [10]:

- The code being written is easily understandable, due to the fact that the code can be classified into classes based on the problem being solved.
- The modules of the code can be developed and changed,
- Software will be more stable due to the fact the programming focus is the small part (classes) of the software.
- Classes can be reusable for other application.

There are 9 modelling diagrams provided by UML, which are Use Case, Class, Object, Sequence, Collaboration, State, Activity, Component and Deployment. The nine diagrams can be divided into four point of views:

- Requirement view: Use Case diagram.
- Logical view: Class and Object diagrams.
- Dynamic view: Collaboration, Sequence, State, and Activity diagrams.
- Physical view: Component and Deployment diagrams.

In our on-going work we have developed new facilities which can be acquired by means of clicking the new icons. We developed the Passenger-UML using Borland Delphi 7 and DRED (Delphi Resource Editor) [12, 13, 14].

The graphical user interface of our newly developed UML-Passenger teleconference and collaborative working tools is shown in Figure 2.

![Passenger-UML User Interface – 9 views of the diagram](image)
Passenger has been equipped with teleconference and online-session which includes audio, video, and whiteboard, and chatting facilities. Figure 3 shows the discussion and chatting facilities for passenger.

![Figure 3](image)

**Fig. 3** Discussion and Chatting facilities for Passenger

Figure 4 shows the result of collaborative work on drawing sequence diagram using Passenger-UML. Several components such as Actor (equipped with lifeline), Object (equipped with lifeline), Object Message, Message to Self, return Message, has been provided.

![Figure 4](image)

**Fig. 4** Sequence Diagram Facility of Passenger - UML

Use case diagrams can also be drawn, as can be found in Figure 5. Figure 6 depicted the activity diagram produced by Passenger-UML. The facility in our software includes some activity diagram components such as: state, initial state, end state, activity transition, note, anchor note to item, activity state, branch of decision, synchronization or fork.

![Figure 5](image)

**Fig. 5** Use Case Diagram Facility of Passenger – UML
Figure 7 shows the State diagrams, in which some components such as State, Initial State, End State, State Transition, Note, Anchor note to item can be drawn.

For the purpose of providing facilities to draw collaboration diagrams, some modules to enable the drawing of collaboration diagram modules such as: Actor, Class Instance, Object, Object Message, Message to Self, Anchor Note To Item, Note, and Text Box, have been provided. This is depicted in Figure 8.
Figure 9 shows the client-server activation mechanism for Passenger UML, which cater the need for 4 users to collaborate through the Internet.

We have thoroughly gone through the software life cycle to provide the Passenger-UML. The module and unit testing has been conducted. We are also aware of the existence of the commercially available CASE tools which has been widely used such as Argo UML, Poseidon, Visual Paradigm and Rational Rose [8, 13-19]. We compared the characteristics of those packaged with our newly extended Passenger UML. The comparison results can be found in Table 1. The main advantage of Passenger over the available packages is the online and collaborative work supports which are not available in Visual Paradigm and Rational Rose.

We have tested and used the Passenger-UML case tool. However, sometimes we faced the bandwidth availability problems which caused some facilities such as video session do not work well, particularly when more that one diagram concurrently being drawn by several users which take over the token.

We have performed some efforts to enhance the Passenger-UML case tools for the purpose of providing synchronization and verification. In addition to that the problem of consistency checking is still being discussed among the e-learning environment investigators. This is due to the fact that initially consistency checking is not provided in the previous version of Passenger, because the purpose of the CASE tool is to enable the trainer to remind the students of the paper conduct in drawing the right software engineering diagrams.
Table 1. The Characteristics Comparison of Three CASE Tools: Visual Paradigm, Rational Rose, and Passenger.

<table>
<thead>
<tr>
<th>Application</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Canvas/drawing media</strong></td>
<td>Excellent professional canvas</td>
</tr>
<tr>
<td><strong>Icon notation</strong></td>
<td>Good, provide 9 UML diagrams</td>
</tr>
<tr>
<td><strong>Pulldown menu</strong></td>
<td>Used for main menu</td>
</tr>
<tr>
<td><strong>Pop up menu</strong></td>
<td>Used for editing menu</td>
</tr>
<tr>
<td><strong>Standard function</strong></td>
<td>Cut, paste, copy, new, delete, etc</td>
</tr>
<tr>
<td><strong>Help menu</strong></td>
<td>Available</td>
</tr>
<tr>
<td><strong>File management</strong></td>
<td>Good</td>
</tr>
<tr>
<td><strong>Facilities</strong></td>
<td>The most complete</td>
</tr>
<tr>
<td><strong>User friendliness</strong></td>
<td>Easy to use</td>
</tr>
<tr>
<td><strong>Online system</strong></td>
<td>Not available</td>
</tr>
<tr>
<td><strong>Code generator</strong></td>
<td>Supported</td>
</tr>
<tr>
<td><strong>Collaborative work facilities</strong></td>
<td>Not available</td>
</tr>
<tr>
<td><strong>Data compatibility</strong></td>
<td>Good</td>
</tr>
</tbody>
</table>

NEW FUNCTION IN PASSENGER UML

The function of change graph module on PASSENGER application is to indicate the changes of UML notations such as objects and lines of notation in user local history. When one user had designed an UML diagram and sent it to Public Window, it will automatically be saved in local history. Every action in UML notation summary change will influence the change graph. A new component is used to support the change of the graph module called BWFileFinder. The function of this component is to search and classify the extension file.

Each UML diagram will be saved in different file’s extension name and the name of the file always relates itself to the timestamp. For example, the file history 19_23_32_19062006.um9 represents the hour, minute, the date, month, and year when the file is saved respectively. The file’s extension of um9 indicates that it is a Use Case diagram.

BWFileFinder is used for matching history files in directory of PASSENGER application and inspect the attribute’s value from it. These values are numbers of the existing UML notation and Hint values related to the user who made them. The module reads each file in sequence based on the index of the 9 UML diagrams (represented in extension uml until um9). Then it reads each line to find the key word “Hint” as the attribute value for user identification and notation UML he/she has made. When a new file is saved in local history, then BWFileFinder will compare the new file with the previous one and compare them with rank 1 and rank 2. The changes of attribute values are then transferred dynamically into the unit file uHistogram for display. The overall process can be seen in Figure 10.

The change graph module will also automatically create a new file in the local history whenever a user transfers the design from private window to the public window. When the file is uploaded to server, it will automatically stored in the global history file with the extension “cst”. Other users can then view the newly uploaded design from the server. However, when they want to modify the design, they should save the file into their own local history and updates the UML notation summary for each diagram as well as the username that creates and deletes notations in the file. After the summary of the new version of UML notations has been generated, it will then be compared to the old one and then be represented in the change graph.
The change graph component is implemented in a file named `Histogram.pas`. Every time a user wants to put a notation in the working area of PASSENGER, the new object dialog will come up. The user has to key in the object’s name and hint field. The hint field is automatically filled with the user name only when the user logged in and connected to the server. All users who notice an update in the global history located in the working area of PASSENGER, the new object dialog will come up. The user has to key in the object’s name and hint value becomes the value for user name from each UML object created.

There are some rules in order to use the component. The Hint box needs to be filled by the name of user who created UML notation although they aren’t connected to the server. All users who notice an update in the global history located in the server must allow the changes to be saved into their own local history file. Only then, the change graph will be able to indicate changes.

The information presented in the change graph will visualize how changes are performed on each diagram by all users. Every part of the change graph has different function and component type, which can see in Figure 11. The component is also equipped with a function for saving the graph result into a bitmap file. An example of such bitmap file is shown in Figure 12, which also presents some significant activities in a period of time.

**Change Graph Component**

![Change Graph Component Diagram](image)

The created object will be categorized based on the Hint value. The number of Object Count & Line Count, becomes the value of Object ID. Every time a user wants to put a notation in the working area of PASSENGER, the new object dialog will come up. The user has to key in the object’s name and hint field. The hint field is automatically filled with the user name only when the user logged in and connected to the server.

![Change Graph Window](image)
Interface in the new PASSENGER-Client

The new PASSENGER-Client is modified by adding some splash animation during the time the program load and add the new change graph component by allowing the PASSENGER client to display it using a button in the main toolbar as shown in Figure 13.

CONCLUSION

We define multimedia pervasive learning as an interactive teaching model that gives mobile students access to a wide range of educational resources. Content-wise, in our work on contributing to Passenger, software for aiding educational process in Software Engineering area, we provide extra facilities and improvement to cater the newly develop de facto standard of software engineering approach such as UML. The main advantage of Passenger-UML over the available packages like Rational Rose and visual Paradigm is the online and collaborative work supports which are not available in Visual Paradigm and Rational Rose. Our collaboration helps in achieving the implementation of collaborative tool.

Our on going and future work covers the latest development which covers the area of ad hoc network, mobile IP, sensor network, context-aware system, and implementation of WiMAX technology. This research is conducted in our laboratory to provide the state of the art pervasive learning environment.

ACKNOWLEDGMENT

This research is funded by RUTI 3 program of Ministry of Research and Technology Republic of Indonesia and conducted at the Mercator Laboratory at Faculty of Engineering University of Indonesia. We thank the contribution of Ibnu Maulana, Edi Rachmadi, Arrino, Tri Fajar, and Prima Dewi to this work.

REFERENCES


