Multiagent approach to control and synchronization of teleeducation in open distributed environment

Witold Jacak*, Karin Pröll*, Jerzy Rozenblit**
*Software Engineering Department
Polytechnic University of Upper Austria, Hagenberg - Linz, Austria
**Department of Electrical and Computer Engineering
University of Arizona, Tucson, USA

1 Introduction

Many multimedia applications will be designed to run on heterogeneous computing environments or will be interconnected to offer multimedia services [1, 4, 5].
However multimedia incorporation proves ***not sufficient for training and education system implementation.*** Its a main issue that students or training participants ***not only conduct and record but also have the possibility to communicate*** their work in an interactive learning enviroment. For these reasons our research efforts focus on an agent based control system *Telesession Controller* for presentation of education units in open and distributed environments [6, 7]. The paper presents the composition of the telepresentations consisting of most different multimedia objects like text, graphics, movies, videos and also nonhypermedia objects such as executable mathematical simulation or design tasks in distributed electronic classrooms.
We assume that there exists a computer network which consists of peer to peer connected local area networks. One local network represents one virtual electronic class room. The server in one of these networks plays the specific role of the on-line sender of an education session. Other virtual class rooms are active receivers of the telepresentation session.

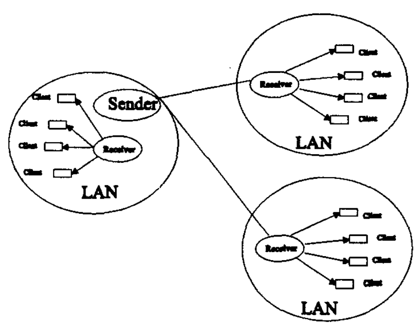

Figure 1: Communication Environment of Teleeducation Session

The control system coordinates and synchronizes the transmission of presentation units coming from different sources; i.e. from live cameras, fixed prepared multimedia presentations, sound systems and an human voice etc.. The control system is modelled using a multi agent based formalism which introduces three basic agent types
• moderator agent for coordination and synchronization of session flow,
• local agents for device controllers and transmitters e.g. camera, sound,..
• education session execution agent (receiver).

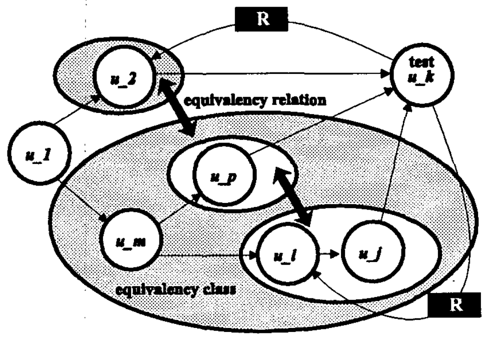

Figure 2: Education Session

2 Basic Notions

The final presentation contains different objects in one of the standardized formats (SGML, MHEG, XML)[3, 2, 8]. Each multimedia object will be modelled on two layers of specification:
• a static layer describing the standard document attributes
• a dynamic layer characterizing the object's dynamic properties with interaction dependencies.

The presentation's objects will be stored in a distributed database, which elements (multimedia objects) can be stored somewhere in the open environment. An intelligent navigation system allows selection and preview of these objects from all over the world. The multimedia objects can be used as a part of different sessions or presentations. The presentation is modelled in form of an event depending graph, representing the general flow of the education session.

2.1 Education session

When beginning the synthesis of a system for teletraining design, one specifies the family of training tasks which can be composed and performed with this system. In this section we focus our concern on the issues involved in the training process. The ***education process*** may include various linearly ordered training sessions.

$$Edu\_Process = (Ses_1, Ses_2, ..., Ses_n)$$

The basic components of each training session are ***training units***. An training unit is a complete portion of the knowledge needed to present one training item.
Presentation process is critically dependent on the training session representation. We use a general description of a training session, formally specified as follows:

$$Ses_i = (U^i, \prec^i, \equiv^i) \tag{1}$$

where:
$U^i = \{u_k^i | k = 1, ..., L^i\}$ is the set of training units connected with i-th training session,
$\prec^i \subset U^i \times U^i$ is the unit precedence relation, and
$\equiv^i \subset 2^{U^i} \times 2^{U^i}$ is the unit equivalence relation.
The partial order represents a presentation precedence, i.e., $q \prec^i u$ means that the unit q is to be completely presented before the unit u can begin the presentation. An example of training session is illustrated in Fig. 2.

2.2 Presentation Units of Education Session

The presentation unit is the multimedia based implementation of the contents of a training unit. Composition of different multimedia documents, which present the contents of a training unit

create the complex multimedia object. This object forms the ***presentation unit*** equivalent to the ***training unit*** and will be formally used as training unit representation in training process.
Let MM be a set of different multimedia objects such as text documents, figures, images, movies etc. To create the presentation unit $p\_u_k^i$ for given training unit u_k^i from training session Ses_i the composition function should be given. The composition function

$$C^i : U^i \rightarrow 2^{MM} \tag{2}$$

assign the sets of simple multimedia objects $m \in MM$ to every training unit $u \in U^i$

$$C^i(u_k^i) = \{m_j | j \in J_k^i\} = M_k^i \subset MM \tag{3}$$

The structured set of multimedia objects and documents M_k^i is called ***a presentation unit*** (p_Unit) of training unit u_k^i from Ses_i. Then, the presentation unit is composed of different simple objects belonging to different multimedia object classes. The properly presentation of the unit's components needs a time and space coordination and synchronization. It causes a multilevel control system for p_Unit execution. This control system is realized on two levels:
– execution level, which performs the presentation of each simple object - component of presentation unit,
– coordination and synchronization level, which scheduling the presentation of unit's components according to the presentation scenario.

3 CAD System for Education Session Development

The proposed CAD system allows the user to design and specify a multimedia telepresentation in a very comfortable way. The system compiles a generalized source code of all documents into JAVA language, a standard for internet applications. It envolves use of presented in previous section control models to generalize the presentation flow including interactivity. **Teletraining Session Designer** is composed from the following modules:
(A) Intelligent MultiMedia Object Manager (IMMOM)
(B) Designer for Presentation Session (DePS)

3.1 MultiMedia Objects Specification and Modelling

The desired framework should support the designer (composer) of a multimedia telepresentation in collection multimedia document objects, such as text, images, vector graphics, movies and audio–files and non multimedia executable demos, exercise or simulation programs. It provides container objects for documents in multiple file formats and conversion to document type unique formats.

3.2 Designer for Presentation Session (DePS) - *CAD–System for Interactive Telepresentation Session Architecture Synthesis*

The proposed flow system allows the user to design and specify a multimedia telepresentation in a very comfortable way. The system compiles a generalized source code, for example into JAVA, a standard for internet applications. It envolves use of systemtheoretic models like **DEVS** to generalize the presentation flow including interactivity.
The telepresentation design bases on semantic graphs. Each dragged object becomes a node of the graph, which will be meshed to other nodes by arcs defining the input–/output–relationship. The whole graph then defines the session flow of the telemedia presentation. The nodes of the graph will be represented by single elementary discrete event dynamical systems.
The DePS system is decomposed on the following modeles:

Inference–engine for session control based on node and arc specification. The inference–engine compiles a complete discrete event specified system (DEVS)[9] out of the graph, where each node, specifying a dynamic document is meant to be an atomic model and the graph itself can be interpreted as a coupled model.

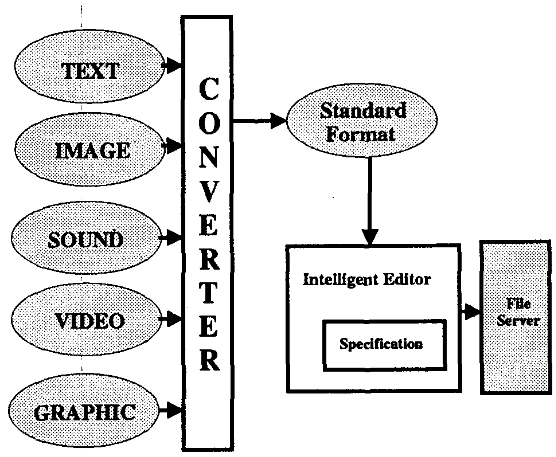

Figure 3: Intelligent Multimedia Object Manager

Generator of platform independent telepresentation session source code. The unit controller stores all DEVS and allows easy changes and additions to the session flow. It includes an DEVS–interpretation system coded in JAVA, which interprets selected objects. It's inputs are a logic query plan (interactivity) and graph for decision making and state accepting in order to run the DEVS.

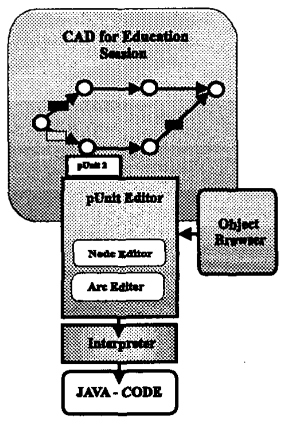

Figure 4: Designer for Presentation Session

4 The Control and Synchronization of the Teleeducation Session

4.1 Multiagent structure of Session Controller

The telepresentation coordination and control system is as a multi-agent hierarchical real time object oriented system modelled.

0-7803-5546-6/99/$10.00 © 1999 IEEE

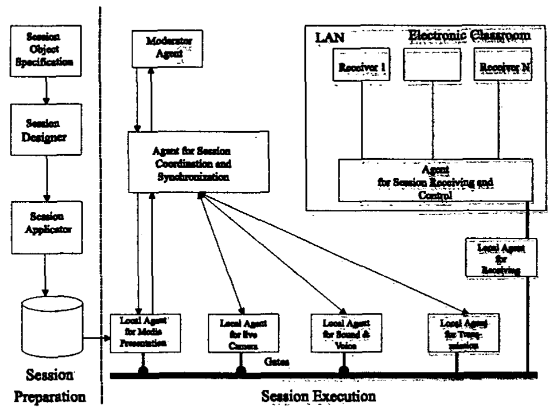

Figure 5: Multiagent Structure of Session Controller

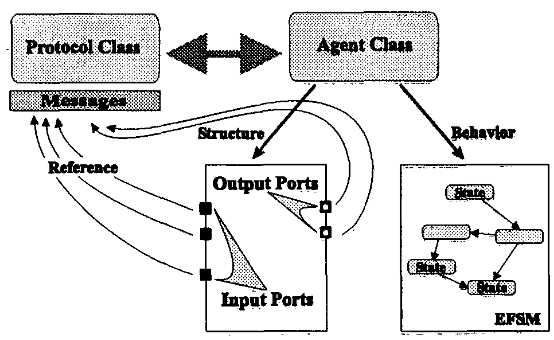

Figure 6: Structure of Agent

A Simple Agent Architecture has the following form:

$$Agent_i = (Structure_i, Behavior_i)$$

The structure component of an agent is defined the pair:

$$Structure = (Interface, Contracts)$$

where the Interface is the set of ports and the Contracts is the set of connections between agents in multi-agent system.
The *Peer_Interface* or *Interface* is a class of input/output port references representing the ports that appear on the outside of the agent

$$Peer\_Interface = \{port_1, port_2, ..., port_n\} \quad and \quad port_k = (port\_name_k, protocol\_class_k)$$

where:
• the port *name* is the name of port reference k and it must be unique with respect to other port references in the peer interface.
• the protocol class is the basic class of the protocol of messages associated with the port reference.

The contracts are defined by the set of binding b-contracts.

$$Contracts = \{b_1^{contract}, b_2^{contract}, ..., b_m^{contract}\} \quad and \quad b_m^{contract} = (end\_point_1^m, end\_point_2^m)$$

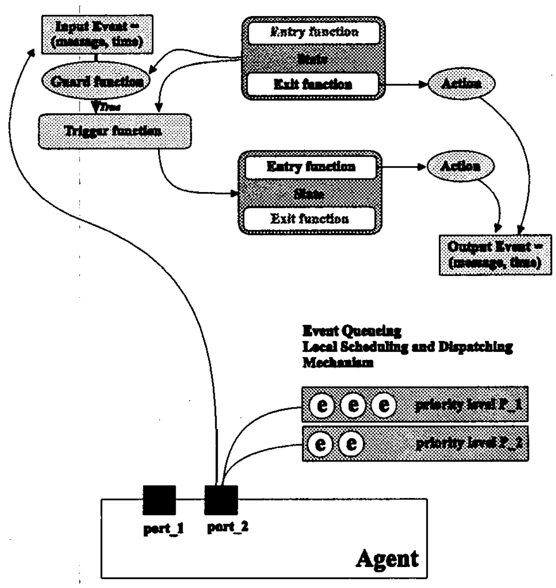

Figure 7: Behavior of Agent

Each of these end point defines the connection port on another agent

$$end\_point^{m}_{1/2} = (port\_name^{k}_{l}, agent_k)$$

where:
$port_l \in Peer\_Interface$ of an agent $Agent_k$.

The behavior will be specified as extended state machine model.

$$Behavior = State\_Machine = (M, S, A, t, g, e, x)$$

where:
• $M = IM + OM$, the set IM defines the input messages, and the set OM defines the output messages.
• S is the state set
• A is the action set
• g is the guard function.

$$g : E \times S \times Peer\_Int \rightarrow \{T, F\}$$

The guard function must evaluate to true or false. This function defines an evaluation that must be performed when a message is received, to decide whether a transition will be taken.

$$E \subset Time \times IM$$

is the set of input events

$$e_{in} = (time, i\_message)$$

• t is the trigger function (state transition function)

$$t : E \times S \times \{T, F\} \rightarrow S$$

where and $\{T, F\}$ are the values of guard function g.
$t(e, s, B) = t(e, s, g(e, s, port)) = s_{new}$ iff $g(e, s, port) = T$,
and

0-7803-5546-6/99/$10.00 © 1999 IEEE

$t(e, s, B) = t(e, s, g(e, s, port)) = s$ iff $g(e, s, port) = F$
• $e : S \to A$ is the entry function, which decides on the entry action of the agent. An entry action is performed when a state is entered by way of any transition.
The entry function can generate the output event

$$e_{out} = (time, o\_message)$$

• $x : S \to A$ is the exit function, which generates the exit action. The exit action is taken when a state is vacated by way of any transition.

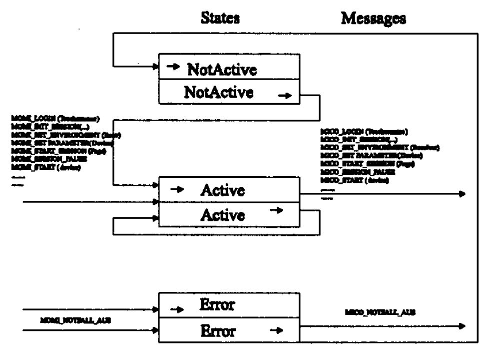

Figure 8: State Transition Diagram of Moderator Agent

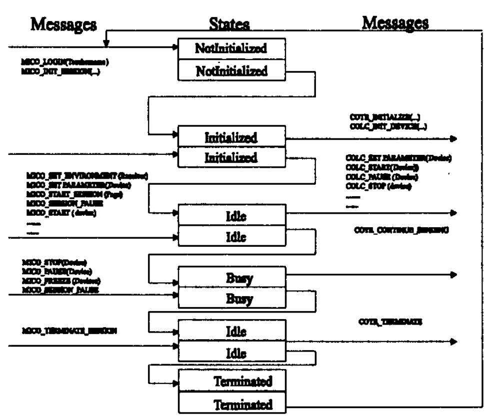

Figure 9: State Transistion of Synchronization Agent

4.2 Device control and moderator agent

Most of the control input comes from the moderator agent and from the receiver agents. The moderator (also called teacher agent) controls the local agents and directly or indirectly the receiver agents. Figure 8 shows the state transition diagram of the moderator agent.

4.3 Transmission and Synchronization Agent

The transmission and synchronization agent coordinates and synchronizes the session flow. Its state transition diagram is shown in Figure 9.

References

[1] O. Beltscheva, I. Georgiev, "The Development of a Multimedia System for Education", in *Multimedia in Open Environment*, W. Herzner (ed.), Springer Verlag, Vienna, New York, 1995

[2] T. Casey, J. Fromont, "MHEG, Scripts, and Standardization Issues", in *Multimedia in Open Environment*, W. Herzner (ed.), Springer Verlag, Vienna, New York, 1995

[3] F. Colaitis, F. Bertrand, "The MHEG Standard: Principles and Examples of Applications", in *Multimedia in Open Environment*, W. Herzner (ed.), Springer Verlag, Vienna, New York, 1995

[4] W. Jacak, "CAD System for Development of Telelearning Session in Open Environment", *Proc. of Conference TELEMEDIA'96*, Hagenberg, Austria, 1996

[5] W. Jacak, A. Wintersteiger, J. Rozenblit, " System Engineering based Design and Control of Teletraining Session in Open Environment", Proc. of ECBS'97 Conference, Monterey, USA, 1997

[6] W. Jacak, G. Jahn, A. Wintersteiger, "Design and control of teletraining session in open environment: System theoretical approach", Proc. of EUROCAST'97 Conference, Las Palmas, Spain, 1997

[7] W. Jacak, A. Wintersteiger, "Information Engineering for Teletraining Session in Open Environment", in "Information Engineering", IDIMT'96, 1996

[8] M. Marshall, W. Samson, P. Duguard, "A proposed Framework of predicting the Development Effort of Multimedia Courseware", in *Multimedia in Open Environment*, W. Herzner (ed.), Springer Verlag, Vienna, New York, 1995

[9] B.P. Zeigler.: *Multifacetted Modeling and Discrete Event Simulation,* Academic Press, 1984

0-7803-5546-6/99/$10.00 © 1999 IEEE