Multi-User Interfaces and Applications
S. Gibbs and A.A. Verrijn-Stuart (Editors)
Elsevier Science Publishers B.V. (North-Holland)

CNLS - COMPUTER NETWORK LECTURING SYSTEM

Peter STANCHEV (*) and Valentin SABEV (**)

(*)Institute of Mathematics
Bulgarian Academy of Sciences
G. Bonchev Str. 8, 1113 Sofia, Bulgaria, Fax. +359-2-752078

(**) Center on Informatics and Computer Technology
Bulgarian Academy of Sciences
G. Bonchev Str. 25A, 1113 Sofia, Bulgaria, Fax. +359-2-707273

CNLS - a computer network lecturing system is presented in this paper. It supports the lecturing process in a network environment and provides tools for text and image query processing.

1. INTRODUCTION

Recently many computer lecturing systems, which support the display of images, have been developed in various application areas [1]. These are usually simple systems for demonstration, with or without animation. Early examples include well-known systems such as: ShowPartner [2], Story Teller (including PC StoryBoard) [3], etc. As a rule, these systems operate on a single computer. Another kind of such a system is one which possesses certain properties for tele-conferencing. An example is the Bulgarian PC-VIDEO System [4]. This system operates on an Ethernet LAN. Another, recently developed system, which utilizes a new approach in image presentation, is MODSIM [5]. This system includes an object-oriented language and supports animation. These types of systems are characterized by maintaining only a "one-way" channel, through which information is transferred directly to the user. This is, in a sence, a restriction - there is no computer-supported feed-back channel, i.e., no possibility of asking a question or conducting a dialogue.

This paper presents a natural and useful development of computer-supported lecturing systems - the Computer Network Lecturing System CNLS: it operates on a computer network; possesses demonstrational abilities for conducting lectures, tools for lecture preparation and storage, query processing services, text and image information services. The query processing for image and text information is realized through a query language available in both menu-driven and command forms. Two ways of communication, dynamically changeable during lecturing, are avaiable: real-time transfer or store-and-forward; They are realized through a synchronizing procedure.

The organization of the paper is the following: after discussing the architecture of CNLS in section 2, section 3 describes CNLS performance phases. The example contained in section 4 enlightens the functioning of the system.

2. CNLS ARCHITECTURE

In CNLS, a lecture is considered consisting of: (1) pages, and (2) a scenario. Each page is divided into one or more frames. Each frame is characterized by its location on the page and consists of image or text boxes. Images can be either simple or compound: simple images are graphical data (vector data as pointed out in [6]); compound images are obtained from others by composition operations. Other types of images are raster ones [6]; They are treated as simple images and obtained by a scanner device. The scenario is a presentation plan determined by rules. The rules introduce the page order, and the duration of displaying.

There are two types of participants in CNLS: "lecturers" and "listeners". The "lecturer" developes the lecture, and makes it available to users. Each "listener" observes the received pages. He may put up questions in the end of each lecture, or if absent, he may get trough it not concurrently - in a more convinient time for him.

A formal description of the CNLS objects is presented in Appendix 1.

The functional model of CNLS is similar to the client-server model [7], [8]. Its structure (Figure 1) consists of functional elements which are aimed at both the handling of information processing and exchanging of information among participants. CNLS is comprised of two types of functional elements - Server and End_User processes.

There are two types of CNLS servers: Training_Server and DBMS_Server.

The Training_Server supports communication between the "lecturer" and the "listener". The relation supported is one-to-many. A dinamically changed distribution list [9] is used for this purpose. It contains participant data, participant status, participant location, and communication modes. The communication takes place in two modes: (1) real-time transfer and (2) store-and-forward transfer. The real-time transfer ensures real-time receipt - "listeners" are allowed to observe the pages simultaneously with the "lecturer". The store-and- forward transfer permits the "listeners" to receive the pages at a convenient time.

The DBMS_Server maintains the lectures in CNLS. It does not differ from a DBMS which comprises of graphical and text data in a manner which accomplishes information processing. This means that DBMS_Server may operate as a server for remote access to the data base [10].

The End_User processs consists of two parts: (1) a subscriber and (2) a client.

The Subscriber performs local information processing. It sends requests to both the DBMS_Server and Training_Server. Each request results in a certain service ensured by the respective server. The Subscriber can be either Coordinator, or Constructor (Lecturer or Query one). The Constructors prepare the lectures, pages, and the questions to the "lecturer". It runs independently of the Coordinator. The Coordinator supports: (1) the "lecturer's" work by sending the pages entered in the Training_Server, and (2) the page display for the "listeners".

The Training_Client and DBMS_Client are intermediate elements between the Subscriber, Training_Server and the DBMS_Server which ensure their connection. Usually the End_User process and the servers are located at various nodes of a computer net-

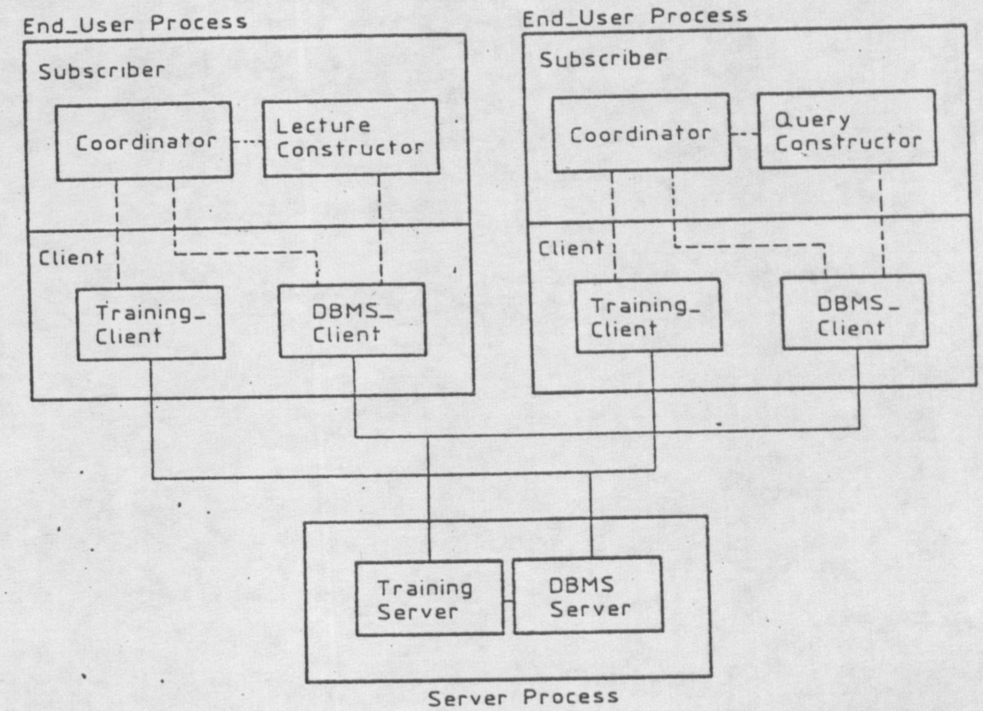

Figure 1. *Functional Model of CNLS*

work (e.g. various workstations and dedicated stations of local area networks). For this reason, communication between the Training and DBMS_Clients and their servers should be realized through a protocol.

The exchange of bit-map information is a crucial part of the CNLS communication. An optimization approach for the exchange of this kind of information is used. Real transfer of page data is obtained via a protocol between the DBMS_Server and DBMS_Client in the End_User process. This form of transfer, called "third-party-transfer" is described in [8].

3. CNLS PERFORMANCE PHASES

There are two regimes in CNLS: (1) Lecture Creation and (2) Lecture Organization. Lecture courses are elaborated during the first regime. Lecture presentation is performed during the second regime.

3.1. Lecture Creation Regime

The lecture creation regime includes the following phases: image input and editing; text input and editing; page construction; and scenario construction.

3.1.1. Image Input and Editing Phase

Images can be entered with the use of a scanner device or created via a graphical image editor. The latter allows images to be constructed out of elements (primitives): points, segments, arcs, rectangles, curves, text or other images. A graphical editor, with two different levels [11], is created: the first level is for object creation; the second is for image construction. A set of editing operations perform certain image transformations, providing a different perspective of the image to be considered. Some of these operations are:

- rotation operation which gives a different view of the image or objects within it;
- zooming operations enabling the user to view an image at different levels of detail (vertical zooming) or view parts of an image (horizontal zooming);
- superimposing of one image over another.

Graphical image representation is stored in a file, containing the graphical image, as a sequence of graphical primitives. Information on the diveloped by the editor image, is stored as an image description. It contains the image name and a list of the objects contained within the image. For raster images, the corresponding bit-map matrix, in compressed form, is stored together with the image name and a list of objects contained within the image.

3.1.2. Text Input and Editing Phase

During this phase, a text editor is used to enter the lecture text into text boxes. It is possible to enter text using a variaty of fonts, point sizes and line spacing.

3.1.3. Page Construction Phase

During this phase, an editor is used which provides tools for dividing the page into frames, as well as integrating image placement and text boxes into specific frames.

3.1.4. Scenario Construction Phase

During this phase, an editor that provides tools for obtaining the scenarios is used. Its operations are similar to those of PC StoryBoard [3].

3.2. Lecture Organization Regime

The lecture organization regime includes two phases: lecture presentation and query processing. The second one is operational after the end of the lecture presentation.

3.2.1. Lecture Presentation Phase

During the lecture presentation there are two processes: showing and observing. The first process depends on the "lecturer" - he determines the lecture and the pages being displayed on the "listeners" screen. The observing process is determined by the "listeners" who can choose the communication mode: real-time transfer or store-and-forward transfer.

During the presentation phase, each "listener" is in one of the following states (Figure 2):

- store-and-forward transfer state (SF);
- real-time transfer state (RT);
- fast-display state (FD).

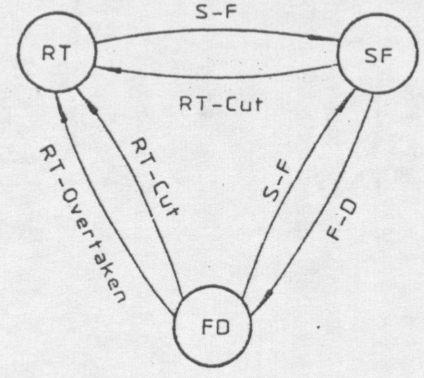

Figure 2. *State Machine for "listener"*

The SF and RT states correspond to real-time and store-and-forward communication modes. In the SF state, pages shown by the "lecturer" are stored and forwarded to the "listeners" on request. In the RT state, showing and observing proceed simultaneously. The FD state is a secondary one. It is introduced in order to avoid the "lecturer" overtaking the "listener". During this state, the "listener" may quickly observe the pages that are being shown by the "lecturer" and not yet observed by the "listener", and those that will be shown by the "lecturer" until the observing process "overtakes" the showing process.

A change to either SF or FD state depends on events determined by the "listener" via a menu command. In Figure 2 these events are denoted as S-F (to store-and-forward transfer state) and F-D (to fast-display state) correspondingly.

A change to the RT state is more complicated than switching to other states. This is in relation to the problem of overcoming the delay of pages that are shown by the "lecturer" and are not yet observed by the "listener". Switching to the RT state depends on the events RT-Cut (to real-time transfer state with "cutting") or RT-Overtaken (to real-time transfer state with "overtaken") (Figure 2). The RT-Cut event means that pages, not observed by the "listener", will be skipped. Such an event is triggered by the "listener" by a menu command. The occurrence of the RT-Overtaken event indicates that all non-observed pages have been seen by the "listener", i.e. showing and observing processes have been running "simultaneously". Such an event depends on the synchronization procedure. For its automatic determination the following approach is used:

Let $[t_1, t_2]$ is the time interval in which the "lecturer" shows pages which have not yet been observed by the "listener". Let k ($k > 1$) is a coefficent showing how many times the observation speed can be increased. After time $(t_2 - t_1)/(k - 1)$, during which the "listener" is observing with a high speed, normal speed will be used.

3.2.2. Query Processing Phase

Query processing includes issuing a query and communication between the "listeners" and the "lecturer".

Query language for page searching is a recently developed language for page retrieval by content. According to it, the user specifies the retrieval condition in the form: **RETRIEVE <query_clause>**.

The **<query_clause>** contains **COURSE <course_clause>** or **LECTURE <lecture_clause>** or **PAGE <page_clause>** or boolean combinations of them.

The **<course_clause>** gives the course name or course number. The **<lecture_clause>** gives the lecture name or lecture number. The **<page_clause>** gives the page number, or **<condition_clause>**. The **<condition_clause>** is a boolean combination of condition groups, image_boxes, and text_boxes. The conditions group is an expression including boxes and **<position_relation>**. The **<position_relation>**s are "E", "W", "N", "S", "NW", "SW", "NE", "SE", where "E", "W", "N", "S" denote the four cardinal points. The condition group is interpreted from left to right. An **<image_box>** gives the image name or a boolean list of objects contained in the image. A **<text_box>** gives the boolean list of words contained in the box.

All the stored pages, fulfilling the query, constitute the query answer set. Since page retrieval is not an exact process, and the user may forget essential characteristics of the sought pages, not one, but several pages are usually retrieved as a result of a query. Thus the existence of a relevant feed-back and query reformulation [12], become essential for at any moment the user can: (1) go back to the query formulation phase, if dissatisfied by the result he has obtained, and (2) change some aspects of the query specification.

A formal description of CNLS query language is presented in Appendix 2.

4. AN EXAMPLE

As an illustration of CNLS, we briefly discuss the use of the system.

4.1. Lecture Creation Regime

Suppose a mathematical course lecture has to be created.

4.1.1. Image Input and Editing Phase

During this phase, images have to be entered or created. In Figure 3 a. an image, generated with the help of graphical editor, is presented.

4.1.2. Text Input and Editing Phase

During this phase, text is entered. A box with text is shown in Figure 3 b.

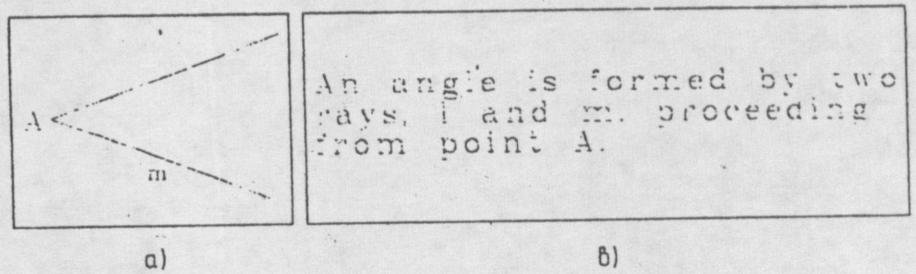

Figure 3. *An image and text boxes of the example*

4.1.3. Page Construction Phase

During this phase, pages are constructed from the text and image boxes. In Figure 4, a page created by the Page Constructor is shown.

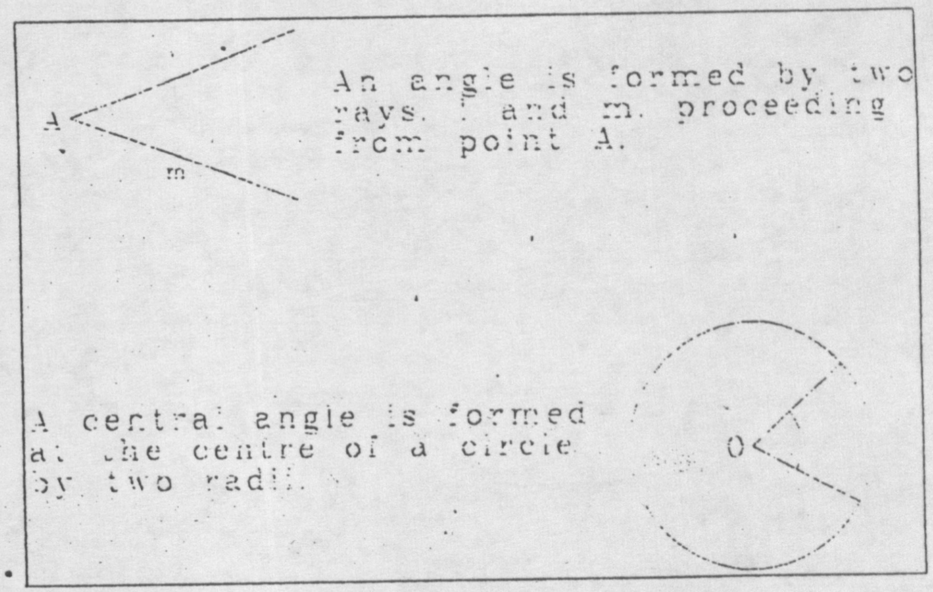

Figure 4. *An example page*

4.1.4. Scenario Construction Phase

During this phase, the time for each lecture page must be given. It is presumed that for the particular page the time is 25 sec.

4.2. Lecture Organization Regime

4.2.1. Lecture Presentation Phase

During this phase, the "listeners" obtain the lecture on their computers.

4.2.2. Query Processing Phase

After the lecture, the "listeners" can ask some questions. Suppose pages with the following features have to be retrieved from the lecture database: an image containing an angle, or rectangle or circuit which is to the west of a text box containing the word "angle". Another requirement is the presence of either a text box containing the words "circle" or "pie" or an image with the name "central_angle".

An expression of this query is shown in Figure 5.

```
RETRIEVE PAGE
 (((IMAGE_CONTAINS angle OR rectangle OR circuit)
  W
 (TEXT_CONTAINS angle))
   AND
 ((TEXT_CONTAINS circle OR pie)
  OR
 (IMAGE_NAME = central_angle )))
```

Figure 5. *The example query*

As a result of the query, a set of pages are retrieved from the lecture database. One of the pages retrived is the page from Figure 4.

5. CONCLUSION AND FUTURE WORK

CNLS is under development. It is realized on IBM PC computers and on Ethernet LAN. LAN software includes a NETBIOS emulator. It involves ordinary workstations and special servers such as File and Print servers.

DBMS_Clients communicate with the DBMS_Server by means of a protocol being a functional subset of the RDA protocol [10]. The communication with the Training_Server is realized through a protocol based on primitives of P1 and P7 protocols of X.400 [13], [14].

The Subscribers are written in the C programming language.

Computerized lecturing is still technically limited and cannot effectively support all desirable functions at the present moment. The future development of this research depends on including voice handles.

FOOTNOTES

This is a joint research project: the lecture construction and query language are being developed at the Institute of Mathematics, the computer supported groupwork part of the research is being realized at the Center on Informatics and Computer Technology.

REFERENCES

[1] DeFanti T. and Brown M., Visualization - Expanding Scientific and Research Opportunities, IEEE Computer, Vol. 22 (Aug. 1989)

[2] ShowPartner, User's Guide, Microsoft.

[3] StoryTeller, User's Guide, IBM.

[4] Boyanov K., Sabev V., Avramov N. and Illiev M., Computer systems for teleconferencing, in: Heimer V., (edt.), Informatik reporte AW DDR, No. 12 (1987) pp. 31-35.

[5] A Quick Look at MODSIM, CACI (1990)

[6] Bertino E., Gibbs S., Rabitti F., Thanos C. and Tsichrizis D., A Multimedia Document Server, in: Proc. of the 6-th Advance Database Symposium (1986)

[7] Framework for Distributed Office Application, ECMA-TR/42 (1987)

[8] Distributed Office Application Model, ISO/TC97/SC18/WG4 N 564 (1987)

[9] OSI Directory access service and protocol, ECMA-TR/32 (1985)

[10] Remote data base access service and protocol, ECMA-TR/30 (1985)

[11] Conti P., Corbo M., Pancaccini, Rabitti F. and Stanchev P., Second Prototype Specifications. Image Classification and Retrieval by Content in MULTOS, Tech. Report OLI/IEI-87-03 (1987)

[12] Salton G., McGill M., Introduction to Modern Information Retrieval, McGraw-Hill (1983)

[13] MIDA-Distributed Application for Message Interchange, ECMA-93 (1985)

[14] MIDA-Mailbox Service Description and Mailbox Access Protocol Specification, ECMA-122 (1987)

APPENDIX.1.

Objects in CNLS are characterized by the following description.

a. Course. A course is characterized by its name and number, and includes the sequence of lectures. Formally a course can be described as a sequence (1):

(1) $$C(C\_name, C\_number, L_1, L_2, \ldots, L_{ln}),$$

where *C_name* is the course name, *C_number* is the course number, L_i ($i = 1, 2, \ldots, ln$) is the contents of the i–th lecture.

b. Lecture. A lecture is characterized by its name and number, and includes the sequence of pages. Formally a lecture can be described as a sequence (2):

(2) $$L(L\_name, L\_number, P_1, P_2, \ldots, P_{pn}),$$

where *L_name* is the lecture name, *L_number* is the lecture number, P_i ($i = 1, 2, \ldots, pn$) is the contents of the i–th page.

c. Page. A page is characterized by its number, the show time, and is divided into frames. Formally a page can be described as a sequence (3):

$$P(P\_number, P\_time, F_1, F_2, \ldots, F_{fn}), \tag{3}$$

where P_number is the page number, P_time is the time that the page stays on the screen, F_i $(i = 1, 2, \ldots, fn)$ is the contents of the i-th frame.

d. Frame. A frame is characterized by its location in the page and includes an image box or a text box. Formally a frame can be described as a sequence (4):

$$F(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, Box\_contents), \tag{4}$$

where $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$ are the frame coordinates, $Box\_contents$ is the contents of the image box or text box.

e. Image box. An image box is characterized by its name and a list of objects in the image. Formally an image box can be described as a sequence (5):

$$I(I\_name, I\_object_1, I\_object_2, \ldots, I\_object_{in}), \tag{5}$$

where I_name is the image name, I_object_i $(i = 1, 2, \ldots, in)$ is the object name contained in the image box.

f. Text box. A text box is the text put in a frame. Formally a text box can be described as a sequence (6):

$$T(text_1, text_2, \ldots, text_{tn}), \tag{6}$$

where $text_i$ $(i = 1, 2, \ldots, tn)$ is the i-th word in the text. The text also includes control words describing the fonts, line spacing, etc.

g. Participant. A patrticipant is characterized by his/her name, status during the lectures and location. Formally, a participant can be described as a triple (7):

$$P(P\_name, P\_status, P\_location), \tag{7}$$

where P_name is the participant name, P_status is the participant status, $P\_location$ is the participant location.

APPENDIX 2.

Formal description of the proposed query language, using UNIX YACC syntax is:

```
query            : 'RETRIEVE' query_clause
                 ;
query_clause     : 'COURSE' course_clause
                 : 'LECTURE' lecture_clause
                 : 'PAGE' page_clause
                 : query_clause 'OR' query_clause
                 : query_clause 'AND' query_clause
                 ;
```



```
course_clause    : 'COURSE_NAME' '=' identifier
                 : 'COURSE_NUMBER' '=' number
                 ;
lecture_clause   : 'LECTURE_NAME' '=' identifier
                 : 'LECTURE_NUMBER' '=' number
                 ;
page_clause      : 'PAGE_NUMBER' '=' number
                 : condition_clause
                 ;
condition_clause : '(' box ')'
                 : '(' condition_clause 'AND' condition_clause ')'
                 : '(' condition_clause 'OR' condition_clause ')'
                 : '(' box position_relation box ')'
                 ;
box              : image_box / text_box
                 ;
image_box        : 'IMAGE_NAME' '=' identifier
                 : 'IMAGE_CONTAINS' list_of_objects
                 ;
list_of_objects  : object_name
                 : list_of_objects 'OR' list_of_objects
                 : list_of_objects 'AND' list_of_objects
                 ;
text_box         : 'TEXT_CONTAINS' list_of_text
                 ;
list_of_text     : text
                 : text 'OR' text
                 : text 'AND' text
                 ;
position_relation : 'E' / 'W' / 'N' / 'S' / 'NW' / 'SW' / 'NE' / 'SE'
                 ;
number           : 'CURRENT' / 'PREVIOUS' / INTEGER
                 ;
identifier       : STRING
                 ;
text             : STRING
                 ;
object_name      : STRING
                 ;
```

(66.) Stanchev P., Sabev V., "CNLS - Computer Network Lecturing System", " ***Multi-User Interfaces and Applications***", S.Gibbs and A.Verrijn-Stuart (eds.), North-Holland 1990, 323-333.