DOCUMENT RESUME

ED 165 793 IR 006 872

AUTHOR Van Der Mast, Charles
TITLE A Modular CAI System.
PUB DATE 78
NOTE 24p.; Paper presented at the International APLET Conference (Pontypridd, Great Britain)

EDRS PRICE MF-$0.83 HC-$1.67 Plus Postage.
DESCRIPTORS *Computer Assisted Instruction; Computer Programs; Concept Formation; Data Processing; Educational Development; Experimental Programs; *Models; *Programing Languages; Teaching Machines; *Technological Advancement; Technology Transfer

ABSTRACT

The experimental CAI system which is being tested at Delft University of Technology is structured in a modular manner to account for high changeability. The concept formulated for this project was the outcome of research into technological, organizational, and educational developments in CAI, and the enumeration of the common aspects of the courseware, different author languages, systems, and methods in current use. This paper discusses this concept and the implementation of the system, including the presentation of lessons, recording activities, designing a lesson, evaluation, distribution of data, and management. While experience with large numbers of students and course authors is not yet available, the system is functioning technically in accordance with expectations at this time. It is expected that it will be used more extensively by lecturers and students at the university in the future, and that extensions in the form of a compiler for the TUTOR subset and the CAISCO interpreter for a microcomputer system will make it possible to test the changeability and flexibility of this modular system. (JEG)

A Modular CAI system

by

Charles van der Mast

Informatics Group
Delft University of Technology
Julianalaan 132
Delft
Netherlands

presented at the
International APLET
Conference ETIC78 in
Pontypridd, Great Britain

SUMMARY.

In the field of Computer Assisted Instruction (CAI) many developments take place. In order to have the opportunity to respond to forthcoming new possibilities and requirements in a flexible way one needs a changeable CAI system.

After having made an analysis of the characteristic functions of an operational CAI system a concept will be introduced to implement the main functions independently. According to this concept it is possible to construct a CAI system consisting of rather isolated modules. A CAI system structured in such a modular way is presented.

INTRODUCTION

Computer Assisted Instruction (CAI) has become popular in the last 10 years, all over the world. Courseware has been presented to tens of thousands of students, mostly in an experimental stage. CAI does not prove to be widely accepted, practised and disseminated, however, according to the expectations in the initial period. This may be due to the problems encountered in creating high quality lessons, which might be able to demonstrate the advantages of the new medium in such a way that people other than the creators can be convinced. A second reason may be the fact that there have been no obvious financial advantages of CAI in comparison to conventional educational methods (Fielden, 1977). Another reason is probably the unnecessary creation of many different authorlanguages and CAI systems. This hampered the dissemination of CAI to all levels in education. Within 20 years there have been some tens of authorlanguages developed for convenient and problem oriented programming. Many CAI systems have been specially designed, some are small systems, whereas others are large. Some have specially designed terminals to present the teaching materials with those systems.

At the Delft University of Technology, Department of Informatics, a CAI project has been initiated several years ago. The objective of the project was research concerning the implementation of CAI, using the most recent technological, organizational and educational developments. A few comparative studies (Bode, 1972; Leiblum, 1974; Zinn, 1969) show that the state of the art of CAI has been sufficiently developed in order to integrate several fundamental aspects of different streams. For such an enterprise a first stage would be to enumerate the common

aspects of both courseware and of the different authorlanguages, including the systems and methods used to practise CAI.

This has led to the formulation of a new concept in the practical realization and organization of CAI. This will be termed 'Modular CAI'. An experimental CAI system was designed following this concept to evaluate advantages against disadvantages. This paper will discuss the concept, its implementation and the present modest experience in using this experimental CAI system.

FUNCTIONAL ANALYSIS OF CAI

Our starting point is an analysis on the most important functions of a CAI system in-operation.

In CAI the lesson is stored in the memory of the computer. The computer's main task is to entertain a dialogue with the terminal. This should lead to a certain transport of information from and to the student. This two-way system of transportation is quite essential in CAI, both in order to maintain a good functioning of the dialogue during all sessions and to record the performance of the students for the measurement by the teacher of the impact of the lessons on the student. This means that except for presenting lessons, also a record of the activities of the students must be kept for analysis later.

The teacher who designed a lesson, whether on his own or in a team, should be able to evaluate continuously the quality of the lesson on the basis of the registered activities of the student.

The fact that many teachers and students are involved in the same CAI process means that one of the most important functions of CAI is the distribution of data.

Management of all mentioned activities is also one of the main functions in CAI. This takes place partially in the computer. The just mentioned functions are depicted schematically in figure 1.

insert figure 1 about here

A closer examination of the functions shown in figure 1 shows that each is a whole class of activities.

For example we mean with the design the completion of the process of making a lesson i.e.:

- formulating the objectives;
- structuring the coursematerial;
- choice of the educational strategy;
- programming;
- compilation or translation into a final shape;
- testing.

With the presentation of a lesson we mean:

- presenting the instructional material;
- analysis of the responses from the students;
- deciding on the proceeding of the lesson.

Recording consists of the storage of:

- all responses of students;
- the response times;
- the path followed through the lesson.

These three elements form the 'historical data'.

Evaluation concerns the analysis of the recorded historical data of students and lessons and of interviews. The objective of this evaluation process is the improvement and adaptation of the lessons.

Distribution takes place in two directions i.e. the lesson moves from the teacher to the student, whereas historical information is given the other way around.

Management of the whole process includes a multiplicity of activities, such as the administration of all data, which should be distributed, and storage and distribution of all identification codes of students and teachers, each with their own authorizations within the CAI system. In the management function also such storage of data is included which is required to restart a lesson after a system failure or other disruptions caused by the student.

Note that the above enumeration is far from complete, e.g. the contact between teacher and student on a conventional basis has not been taken into consideration.

In the technical realization of CAI we can observe two developments in two opposite directions, against the background of figure 1.
On the one hand there is the existence of large operational CAI systems with up to 500 terminals per computer, such as PLATO and IIS of IBM.
One advantage of this approach is that the lesson, or at least the experience in writing a lesson, is easily exchangeable. A disadvantage is the technical and organizational dependence on one central machine, in which all data of students, teachers and lessons are stored.
An entirely different trend is the usage of small computers which can locally present lessons via terminals. The lessons are usually developed somewhere else and distributed mechanically.
An advantage of this approach is that its small scale allows to involve local computers, already in use for other purposes before CAI is being introduced. This is why the costs are perceived to be low. One other advantage is the larger flexibility when new technological developments take place. A disadvantage is that one does not have the power of a large computer at one's disposal.
It is still an open question, however, to what advice the balance of the advantages and disadvantages of small versus large CAI systems will lead. For example, the necessity of centralization is questionable for course materials which have taken more or less firm shape after some years of experimentation, and which are virtually self-supporting. The power of a large system is convenient in the early stage of the design of a lesson, when many changes still are to be made.

A CONCEPT FOR A MODULAR CAI SYSTEM

Searching for a modular CAI system it was deemed important that a system should have:

- well defined interfaces;
- simple possibilities for adding or modifying modules.

The latter possibilities are required in view of technological advancement and changes in user demands. For a system as described above two issues are important.

Firstly the large diversity in terms of languages and systems for which many and mostly good teaching materials are available.

Secondly the rapid pace of development of performances versus prices, which differ for all sorts of machines. For example, compare the price and performance of very large computers, of microprocessors with floppy or video disks, of intelligent graphical terminals, of communication lines, etc.

In the near future technological innovations could have great influence on the possibilities of the realization of several functions shown in figure 1.

These considerations were the reason why this concept was formulated. The six main functions mentioned before could possibly be executed independently by using different computers. The only condition is that the data to be transferred should have a well-defined structure. Therefore the distribution, in particular the characteristics and the form of what has to be distributed is central in the concept, as shown in figure 1.

The functions which should be carried out are in principle assumed to be independent one of another. In other words: the operation of a CAI system is in principle independent on the manner of implementation of

the different functions. It is even possible that different ways of implementation of one and the same function can exist simultaneously in one and the same CAI system. We refer to the possibility that some lessons can simultaneously be followed by using a large computer in a time-sharing system and by using stand-alone intelligent terminals. This is considered as important both with respect to the technological developments to be expected in the near future as well as to the different demands made by the users concerning hardware and software in the CAI environment. The courses should be distributed in such a form that their execution can be carried out without too many readjustments, even at different local computers. This form should have the nature of a code in which all necessary CAI actions are represented as a set of instructions. This means that this representation has to be some type of assembler code, executable and interpretable by a hypothetical machine (e.g. an interpreter in a particular computer). This implies that teaching-material which is written in an authorlanguage should be susceptible for analysis and transfer by a compiler into this CAI assembler code.

Hence, this code should not be more detailed than is required for the complete description of the features of the different authorlanguages. The design of the code should not be limited by the restricted possibilities of certain computers, but should meet the conditions imposed by fundamental CAI actions as well as possible.

insert figure 2 about here

We defined such a code and termed it as CAIASCO (CAI ASsembler COde) as described in figure 2. The compilers and interpreters can be situated in various kinds of computers. One compiler is necessary for each authorlanguage and one CAIASCO interpreter is necessary for

each type of computer. The CAIASCO code is always identical. In this way the courses which are written in one authorlanguage can be executed by various kinds of computers. We shall now give a brief description of this CAIASCO code.

CAIASCO

Analyzing the different kinds of authorlanguages we found that their facilities all exist of a chain of actions which all are part of a limited set, such as:

1 send a text to the terminal;
2 move the cursor on the display panel;
3 send a picture to the terminal;
4 execute a calculation;
5 set a clock;
6 wait for a response from the student;
7 switch the response processor;
8 analyze a response;
9 test a condition;
10 jump to another part of the lesson.

Some actions have a quite broad and general significance, like action 1, while others are rather technical or apply only to the features of one authorlanguage, like action 7. The number of such elementary CAI activities necessary to describe most of the authorlanguages turns out to be no larger than about one hundred. The actions involved in these activities are so elementary that only from their appearance or from characteristic sequences of such actions we can recognize what authorlanguage is in use.

Due to the elementary nature of these actions it is not simple to write a lesson in such a code. In the structure of the CAIASCO code we took more account of the possibilities to make an interpreter than of the readability for users. However, the code remains readable like usual assembler codes.

An other important starting point was that the code had to be such that the description of a course in a new authorlanguage should not require changes in the code: the code will at most undergo some extensions. (cf. Wiechers, 1975 and Van der Mast and Van der Valk, 1976 for a complete description of CAIASCO.)

THE MODULAR CAI SYSTEM IN DELFT

A CAI system has been designed and implemented for our research in order to try out the new concept (Van der Mast, 1977). The system is functionally split up over several computers, see figure 3.

insert figure 3 about here

Firstly we use an IBM 370/158 to transpose the courses which are fed by batch or interactively in an authorlanguage into CAIASCO code. This CAIASCO code is stored on tape or floppy disk for further distribution to the presentation computers. The presentation of the courses can take place in two ways:

- By using a dedicated PDP11/45 (28 kw) with eight terminals. The courses are interpreted in this computer. The interpreter is part of an experimental system (CAISYS), which enables users to follow courses in a multi-access way. The CAIASCO code is stored on disk before interpretation takes place. Historical information is recorded on tapes. The content of these tapes can be sorted and analyzed by the IBM 370/158, see figure 3. An interactive graphics package for the design of static pictures and texts is available at the PDP11/45. The pictures are stored in a device-independent picture code during the stage of design, such that the use of pictures does not depend strongly on the use of certain computers or types of terminals. These pictures may be called by name from the lesson.
- We started during the past year with the implementation of an interpreter for a microcomputer. This microcomputer is a stand-alone small computer system with a dual floppy disk unit (LSI11). We decided for microcomputers because of the steep

decrease of their prices and their growing possibilities in usage. The distribution to and from this system has to take place by means of floppy disks.

The development of teaching-materials

The design of a course is carried out in the same way as elsewhere. During the stages of programming and input in the computer differences can arise compared to other systems. Since the authorlanguage PLANIT was popular in The Netherlands around 1975 it is natural that we first built a compiler for this language. This compiler was written in ALGOL and has been operational since two years. The compiler informs us not only of syntactical but also of a certain number of semantical errors. Various courses which were designed somewhere else using a PLANIT system could be transformed into CAIASCO without any difficulty. At the moment a compiler for a subset of TUTOR language of the PLATO system is in development. For this purpose the CAIASCO code has to be extended with some new actions. This is especially interesting because these two languages which do not resemble one another can be derived into one and the same CAIASCO code.

The Presentation and Registration

CAISYS

The experimental CAI system CAISYS contains a CAIASCO interpreter and also software for managing and monitoring lessons in a multi-access way. Special measures have been taken in order to keep response times of the system below two seconds. After technical failures or after interruptions of courses, one can proceed with the courses without difficulties. Four of the terminals have a wide variety of graphical

possibilities.

During sessions students can make use of simple calculating facilities. They may always evaluate but one arithmetic expression. Besides they can define variables and functions necessary for their calculations.

The path followed through a course is monitored by CAISYS for use during the lesson. This information is not the same as the historical data on tape, which is for evaluation afterwards.

In the CAIASCO code reference can be made to pictures. These pictures can be inserted during the loading of the CAIASCO code. The pictures can be designed by using an interactive graphics package and can be stored in libraries, see figure 3.

CAIMIC

As a consequence of the latest technical developments in microcomputers we are now amidst the development of a CAIASCO interpreter for a microcomputer (CAIMIC). The possibilities of presenting courses will be similar to that in CAISYS. Students receive a floppy disk with a course recorded on it. Historical data will be stored on another floppy disk which is not available for the student. These stand-alone microcomputers can be easily transported.

The use of the system

We distinguish 3 categories of users of the system:

- teachers, authors and programmers;
- the manager;
- students.

The set up of the system makes an interactive design of courses impossible, so that the author cannot follow the course in a student-

mode in order to test it.

Experience shows that writing a course is not as simple as we thought and it is not advisable to present the courseware on the display in a too early stage of the design: the actual design and structuring process should not take place at a terminal. Experience also shows that programming should be carried out by programmers in the design team. In our case, the task of the programmer is not to make a good run of a lesson but to write parts of a lesson in the authorlanguage and to convert it into CAIASCO, using the appropriate compiler. The manager of the CAI system can then take care of the CAIASCO code such that the course can be followed at terminals. A course in such a design phase should be evaluated on its correctness, should be judged didactically and tried out by other persons of the design team. The necessary changes and corrections can then be communicated to the programmer.

The task of the manager of the system is to insert the CAIASCO code into the CAISYS subsystem. It is also his task to do the accounting of students and lessons in CAISYS and to distribute passwords to students. He should take care of all tapes containing historical data and it is his duty that all evaluation programs are available at the large computer. The students are the most important users of the system. The objective they have to take place at the terminal is simple; they want to learn something from the courses. The CAI system itself can contribute to the learning process by being as inconspicuous as possible to the student. The characters should be well readable and the response time should not be too long. The terminal must be simple in use. Also the in- and outlogging procedures should be simple. In this respect the CAI system will not differ too much from other CAI systems.

CONCLUSIVE REMARKS

The modular CAI system here described was built in order to check the fundamentals of the underlying concept in practice. Research in this project is being done particularly in the area of computer technology. In line with this approach the organization and planning were not meant to produce a large number of courses.
Experience with large numbers of students and authors is also not available until now. Technically the system is functioning in accordance with our expectations, however.
In the near future the system will be used more intensively by lecturers and students at the Delft University of Technology.
The extension the system will undergo with the compiler for the TUTOR subset and the CAIASCO interpreter for a stand-alone microcomputer system will enable us to test the changeability and the flexibility of this modular CAI system.

REFERENCES

Bode, A. and Dütting, M. (1972), Computer-Unterstutzter Unterricht: Probleme, Autorensprachen, Systemvergleiche, Dokumentation. Diplomarbeit, Karlsruhe.

Fielden, J. (1977), The Financial Evaluation of NDPCAL. British Journal of Educational Technology, No. 3, Vol. 8, pp. 190-200.

Leiblum, M.D. (1974), An Analytical and Comparative Study of Computer-Assisted-Instruction Programming Languages, their Characteristics and Usage. Katholieke Universiteit, Nijmegen.

Mast, Charles van der and Valk, Martin van der (1976), Voortgangsrapport 1976, CAI-project. Delft University of Technology, Delft.

Mast, Charles van der (1977), Voortgangsrapport 1977, CAI-project. Delft University of Technology, Delft

Wiechers, G. (1975), Design and Implementation of a System for Computer Assisted Instruction. University of South Africa, Pretoria.

Zinn, K.L. (1969), Comparative Study of Languages for Programming Interactive Use of Computers in Instructions. University of Michigan, Center for Research on Learning and Teaching, Ann Arbor.

Biographical notes

Charles van der Mast is a member of the CAI project of the Department of Informatics at the Delft University of Technology.

Address for correspondence:
Delft University of Technology, Julianalaan 132, Delft, Netherlands

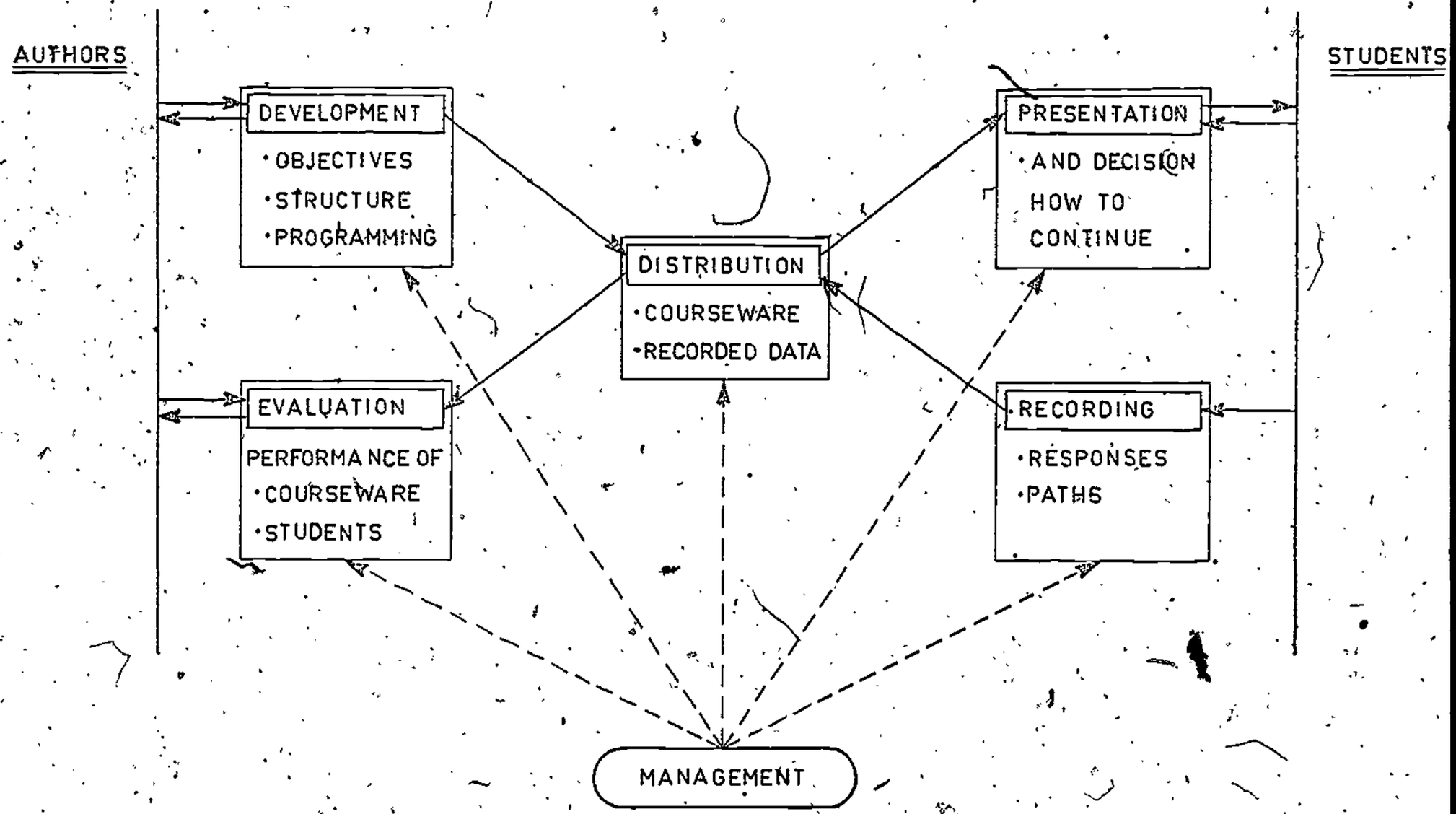

Fig.1.
Functional characteristics of the implementation of Computer Assisted Instruction.

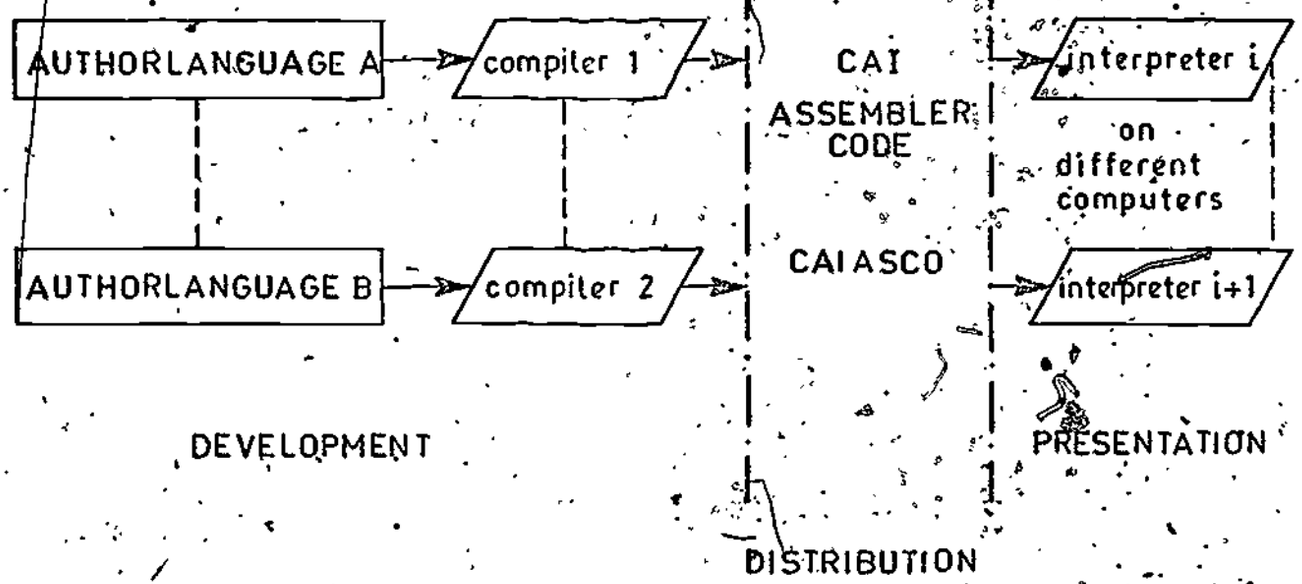

Fig. 2:
The main principle of the modular CAI system: isolation of the development and the presentation of courseware.

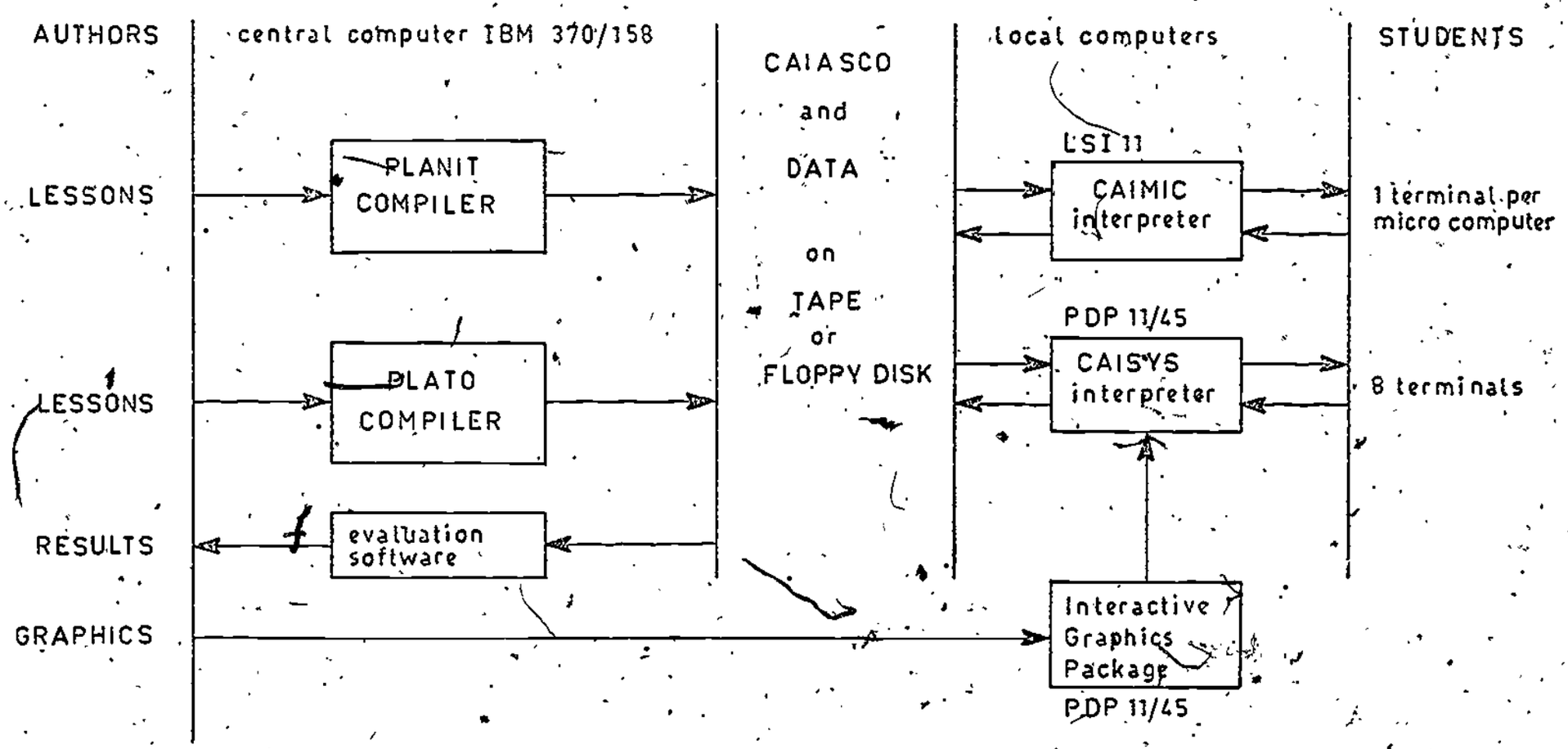

Fig.3.
Overview of the experimental modular CAI-system of Delft University of Technology.