

Computer-Human Interaction and Collaboration

Subjects: [Computer Science](#), [Artificial Intelligence](#)

Contributor: Manuel Ortega

Through a series of projects carried out by the Computer–Human Interaction and Collaboration (CHICO) group of the University of Castilla-La Mancha, some proposals are presented to improve the current e-Learning systems by making use of different paradigms of human-computer interaction. Synchronous and asynchronous collaborative systems, ubiquitous computing, and augmented reality can improve the current learning environments.

human-computer interaction

CSCL

CSCW

ubiquitous computing

e-Learning

1. Introduction

Computers have been used for decades to teach and learn in many different branches of knowledge. Since the year 2020 and as a consequence of the COVID-19 pandemic, their use has become widespread, as they are, in many cases, the only way to attend classes, giving rise to the generalized use of asynchronous and synchronous collaborative tools that have been significantly improved since the 1990s. In an article in 2000 ^[1], Manuel Ortega predicted the predominance in the immediate future of collaborative systems and ubiquitous computing in eLearning from the perspective of three factors that would participate in the process: teachers, software, and hardware. In the article, the author comments on the necessary changes in the software, taking into account the following ^[1] (pp. 14): "On the one hand, in order to foster knowledge acquisition in a constructivist way, software should be developed aiming at real and complex projects with problem "scaffolding" and the use of the so-called "Artificial Intelligence" techniques in order to guide the students without overwhelming them. On the other hand, we must focus the solution of those problems in a collaborative way." Of the three factors mentioned above, the teacher component highlighted by Ortega has proved to be of vital importance in this crisis, and his dedication and effort in the transition from face-to-face to online classes have allowed teaching work to continue during the ongoing pandemic ^[2].

2. Computer Supported Collaborative Learning

Collaborative learning systems are divided, in one of their most commonly used divisions, into asynchronous and synchronous. Asynchronous tools are used at different times by the participants in the interaction. A typical example of this kind of system is email. In this type of application, users utilize the tool at different times to send messages to each other, and it is not necessary to reply immediately to the received email. When the email account is opened, they can answer the received emails immediately. By not having to reply in real time, the user

can think about how to respond to the asynchronous proposed action. Asynchronous tools, therefore, are used in many cases to plan solutions to complex problems in e-Learning systems, in many cases requiring artificial intelligence mechanisms. As an example, we can propose the PlanEdit tool ^[3] (see [Figure 1](#)), which was developed by students to solve domotics problems within the DomoSim-TPC (Domotics Simulation—Telematic Planned Collaborative) system ^[4]. In this case, the student solves a planning problem where they can be tutored by artificial intelligence mechanisms. In this asynchronous system, the students select actions individually in the "Individual Workspace", which are then shared in a "Collaborative Workspace" where they discuss possible solutions. When they reach a consensus, and if the solution testing mechanisms used by artificial intelligence are in accordance with what was planned, the proposal is moved to the so-called "Results Space". If the discussion in the collaborative space is too long, the system can suggest changes according to the level of scaffolding that the problem allows. The scaffolding helps the student more in the initial problems, even if they are easier, and less in the later problems presented, even if they are more difficult. In this way, help is offered according to the level of knowledge acquired by the student.

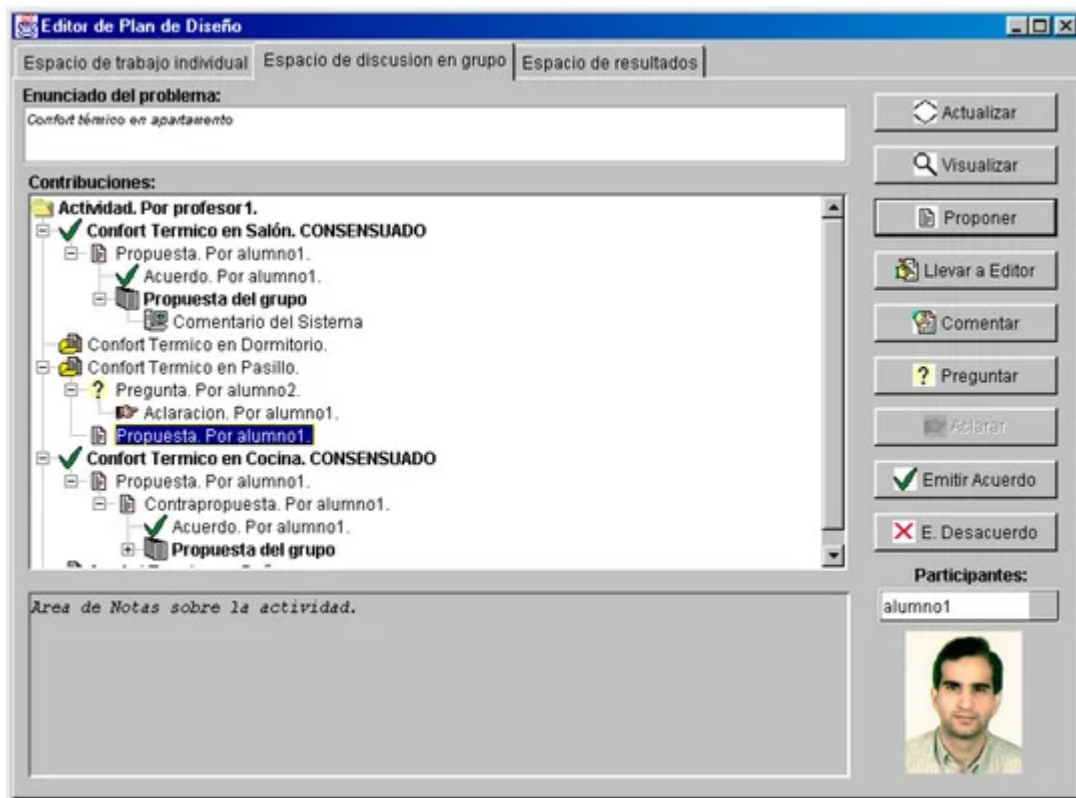


Figure 1. Asynchronous Computer Supported Collaborative Learning (CSCL) environment of DomoSim-TPC.

Synchronous environments ^{[5][6]} (see [Figure 2](#)) are those in which students collaborate to solve a problem at the same time, and usually from different locations. These systems usually use different forms of awareness ^[7] to maintain the feeling of working collaboratively, so each participant knows at all times what the other participants are doing. [Figure 2](#) shows two students collaborating to solve a home automation problem. One of the awareness methods is the use of telepointers—red in the case of the student alumno2 and green for alumno1 (see [Figure 2](#)). In synchronous systems, the problems are solved in a limited time and therefore with decisions that are made

immediately, thus requiring the solution to be planned in many cases. In this way, synchronous environments should be used after a stage of asynchronous collaboration, which allows for greater reflection on the solution of the proposed problem. These systems consider the individual contributions of the students and the collaboration between them, as well as whether the collaboration improves the results for the proposed problems, using a large number of metrics that reveal the students' levels of development.

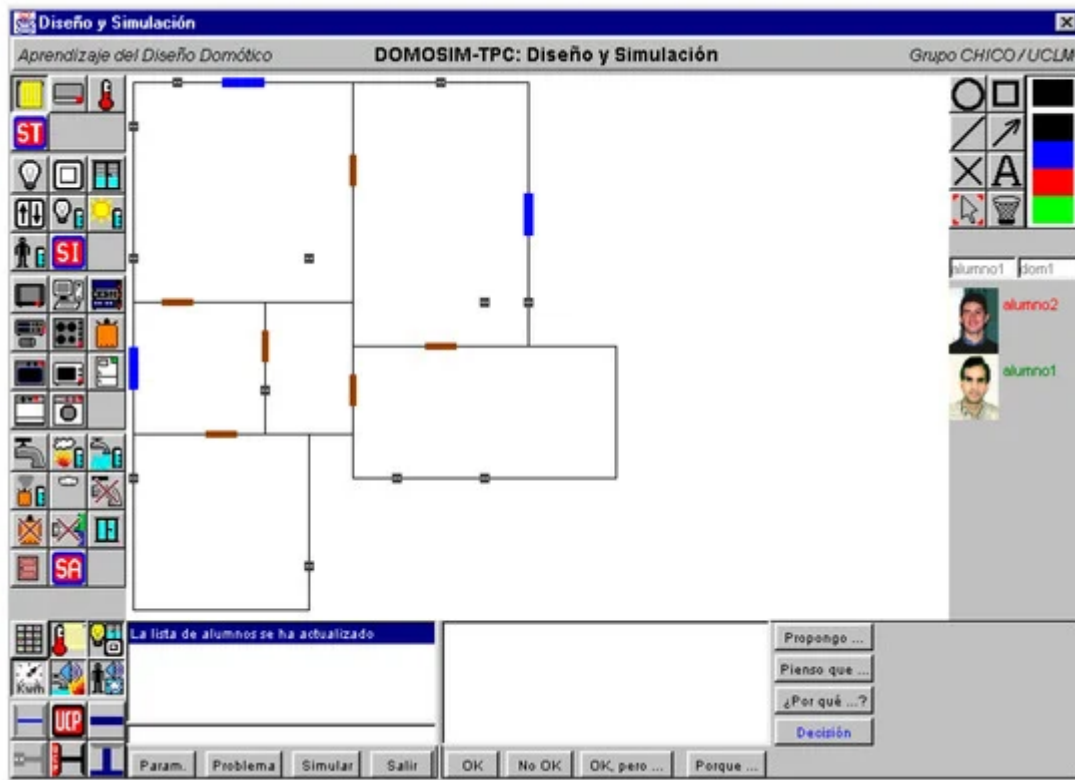


Figure 2. DomoSim-TPC synchronous CSCL environment.

Students must solve home automation problems of increasing complexity, and the system uses artificial intelligence-based mechanisms to check the quality of the proposed result with an aid implemented through scaffolding. As mentioned above, the collaboration process and the quality of the solution proposed by the students are monitored on the basis of a set of qualitative and quantitative variables expressed by means of linguistic variables based on fuzzy logic. To obtain these conclusions, we use variables modeled with fuzzy sets. These sets are compound attributes or linguistic labels which define the domain of each variable. We considered variables characterized by five specific attributes corresponding to the following labels: very low, low, normal, high, and very high. In DomoSim-TPC, we incorporated the necessary functionality to draw conclusions about initiative, creativity, elaboration, conformity and disconformity.

The process-product analysis in CSCL environments carried out in the DomoSim-TPC environment is very important in any collaborative system. The quality of the proposed solution—i.e., the product—and the quality of the collaboration—i.e., whether a fruitful collaboration between the participants has taken place—must be known. Finally, it is also important to know whether the quality of the collaboration has influenced the quality of the product.

To this end, a line of research in collaboration analysis was developed. It allows detailed reports of the collaborative activity to be obtained using a framework called FAPPEC (Framework for Process-Product Analysis in Collaborative Environments) [8]. This framework can be used in different tools to monitor the collaborative activity of the participants in a collaborative environment and can be defined by means of a tool with direct manipulation of the different parameters to be visualized. In [Figure 3](#), we can see this authoring tool designed to model the intervention of a CSCL system.

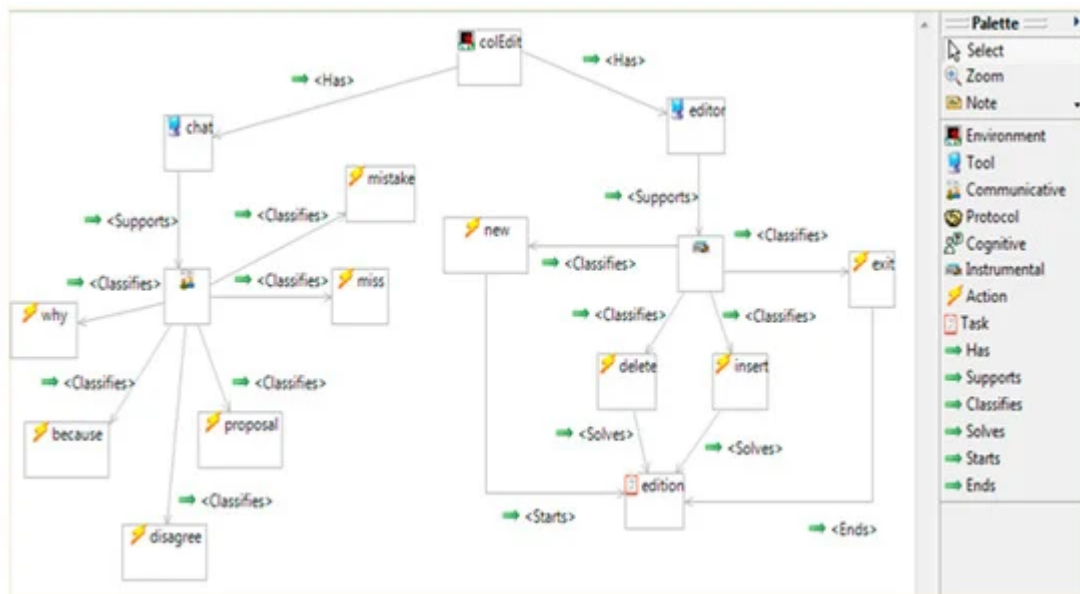


Figure 3. Authoring tool for intervention modelling.

Through these frameworks, precise metrics can measure the collaboration of students to solve complex problems. The metrics used come from studies of the influence of different parameters upon the collaborative process [7][8][9]. Some of these metrics for the collaborative process are:

- Number of accesses to the system that each student carried out to work in the activity.
- Number and mean of contributions made.
- Mean of the contributions' size.
- Kind of contributions made.
- Number of replied and refined contributions.
- Depth of the discussion for each task of the problem.
- Number of times the last news was accessed.
- Number of messages sent (classified by kind: planning, coordination, or system).

- Number of instantaneous messages sent.

3. Ubiquitous Computing

Ubiquitous computing [\[10\]](#) is an interaction paradigm that describes different devices which collaborate through networks facilitating, in a non-intrusive way, the tasks that a user wants to perform. These devices are located wherever they are needed, hence the name “ubiquitous computing”, meaning anywhere.

An example of systems that can be encompassed within the paradigm of ubiquitous computing is the AULA (A Ubiquitous Language Appliance) system [\[11\]](#). The AULA system was designed to teach foreign languages using personal digital assistants (PDAs). The students have to produce a collaborative piece of writing by dividing it in a structured way into ideas and aspects. The system can be used in or out of the classroom, and there is a synchronization mechanism for the generated papers.

AULA has a web-based version called A Web-based Language Appliance (AWLA) [\[12\]](#), which has been successfully used in language courses and is complemented by the AIOLE (An Interactive Online Learning Environment) system, a web-based system generator to learn foreign languages.

AWLA (A Web Language Appliance) ([Figure 4](#)) is a wiki-like system with various online language tools that allows the collaborative editing of texts in different languages with teacher correction possibilities and tools to prevent plagiarism by the learners. The whole system can be classified as a personal learning environment (PLE) with the capability to be used within the classroom, anytime and anywhere according to the paradigm of ubiquitous computing, through the use of PDAs.

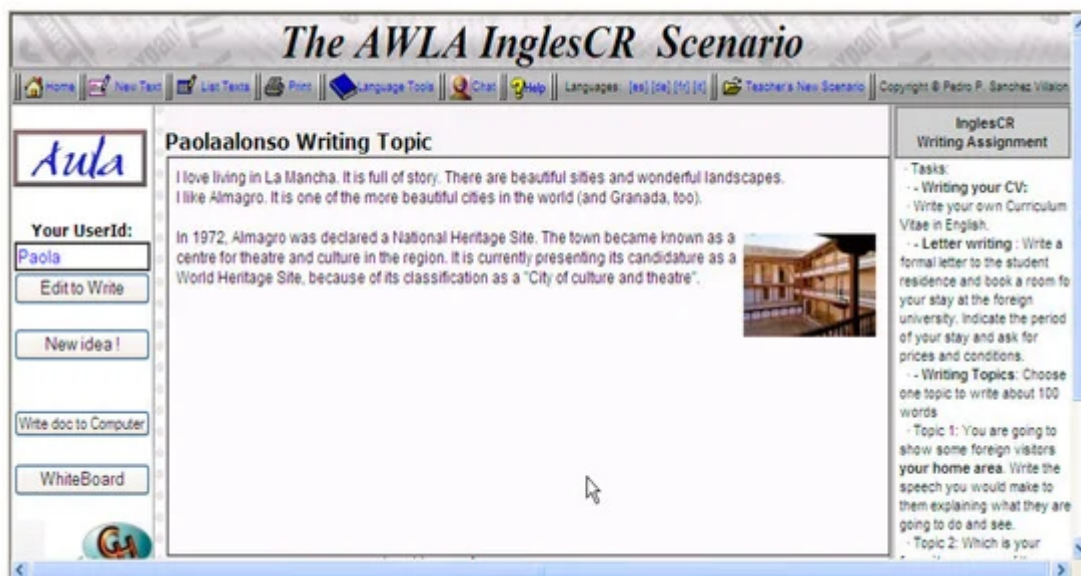


Figure 4. AWLA and AIOLE personal learning environment.

4. Groupware Engineering

The ad hoc implementation of collaborative systems is an arduous task. This is why the automatic or semi-automatic creation of groupware applications is useful for the generalization of their use in classroom. The automatic creation of synchronous applications for both CSCL and Computer Supported Cooperative Work (CSCW) environments is vital to developing useful tools in a reasonable time. In this case, a model-driven engineering (MDE) method has been designed for the development of domain-independent collaborative modelling systems [9]. [Figure 5](#) shows an application generated using the SPACE-DESIGN tool in order to allow the participants (with their picture on the right of the system in F) to generate diagrams in a collaborative way. [Figure 5](#) also shows a shared whiteboard (A) where two participants can add elements from the application domain, in this case logic gates. Entities (B) and relationships (C) can be added and appear on the left in the environment (D, E). The telepointers (G) are used to show the area of the shared whiteboard where each participant is editing. This environment has been developed by defining models that automatically generate the final environment in [Figure 5](#).

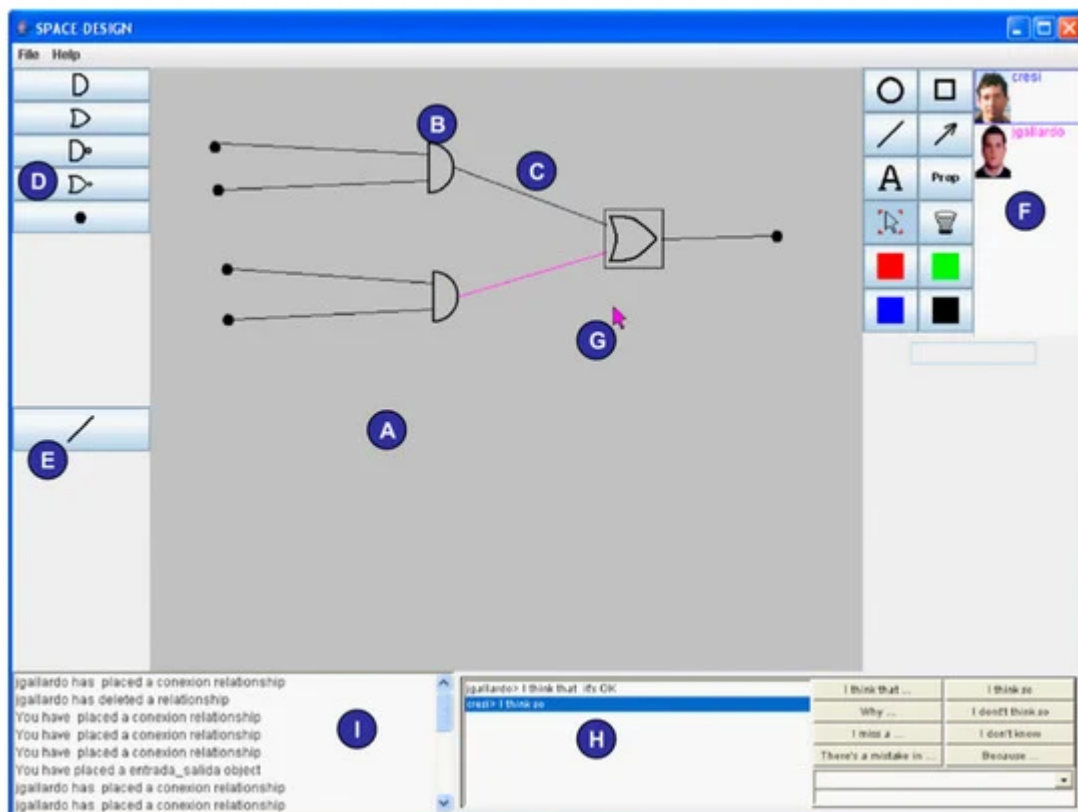


Figure 5. Collaborative environment generated by models.

In order to model collaborative and interactive environments, notations and methodologies are necessary to help describe the systems that will be developed, as well as to automate the processes that generate the collaborative tools. Among the existing proposals, we highlight ConcurTaskTree (CTT) [13] and Collaborative Interactive Application Methodology (CIAM) [14].

CIAM [14] is a methodology that guides the designer to create a conceptual specification of the aspects to be considered within a groupware system in order to architect the system and design the necessary interaction in this kind of groupware system. The different steps to be carried out to obtain the final user interface can be seen in Figure 6.

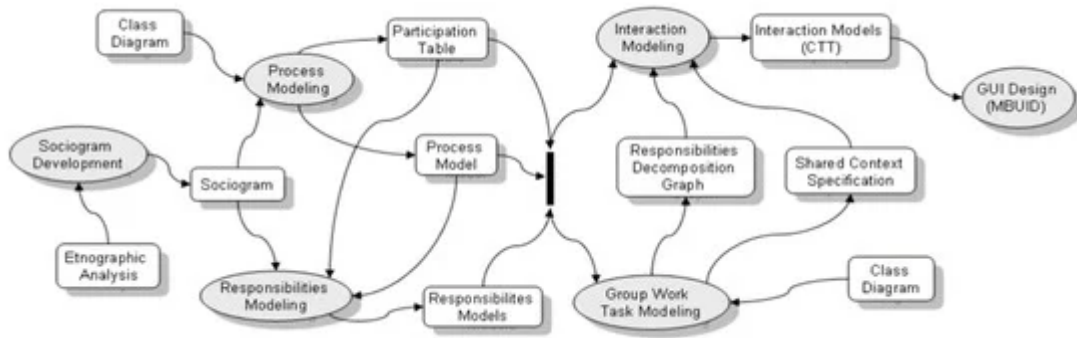


Figure 6. The CIAM proposal and the products obtained in each of its stages.

CIAM (Collaborative and Interactive Application Methodology) is complemented by a notation called CIAN (Collaborative Interactive Application Notation) [14]. A summary of the CIAN is shown in Figure 7. In areas A and B, we can see the icons that represent the nodes that constitute the process model, the relationship types which indicate the different tasks, and the interdependence types. In C, we can see the icons used to represent an interaction task model in CTT notation.

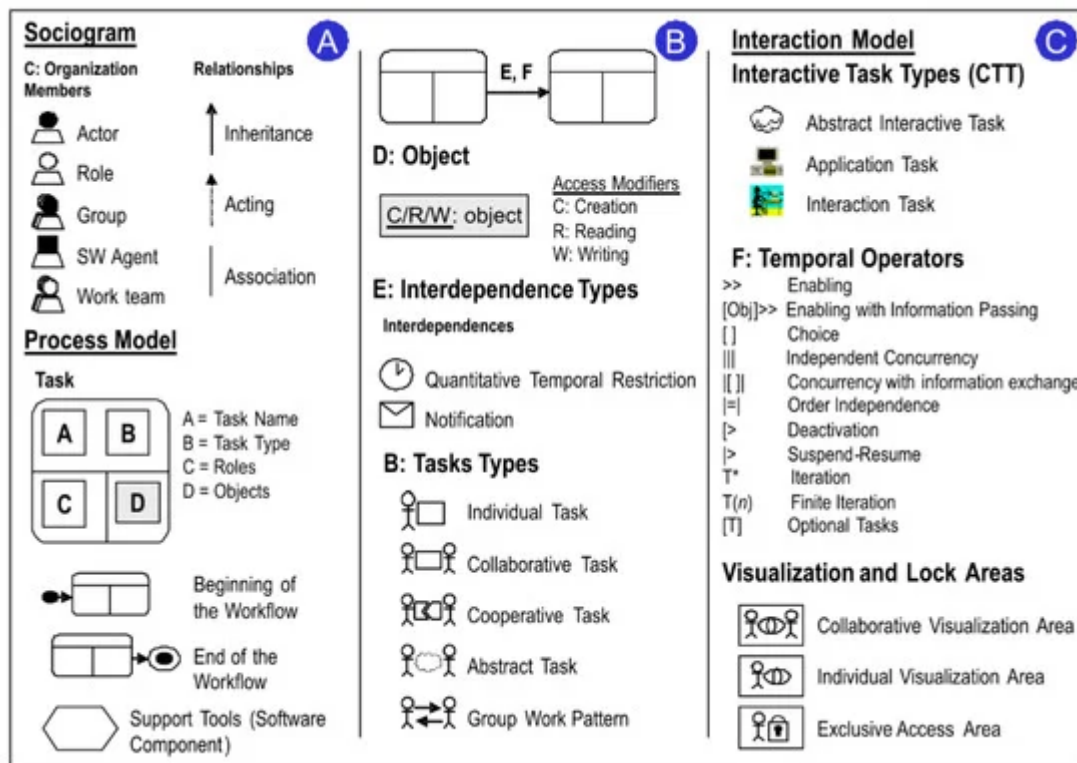


Figure 7. The CIAN notation.

The CIAN design produces a CTT diagram that in turn generates user interfaces in a semi-automatic way. To facilitate this design, a tool called CIAT (Collaborative Interactive Application Tool) [15] is used. With this tool, the whole methodological process can be followed up to the generation by means of model-based user interface design (MBUID).

The CIAN was extended to increase expressiveness in the development of e-Learning applications with the Learn-CIAN expansion and its corresponding Learn-CIAT tool [16], which can be seen in [Figure 8](#).

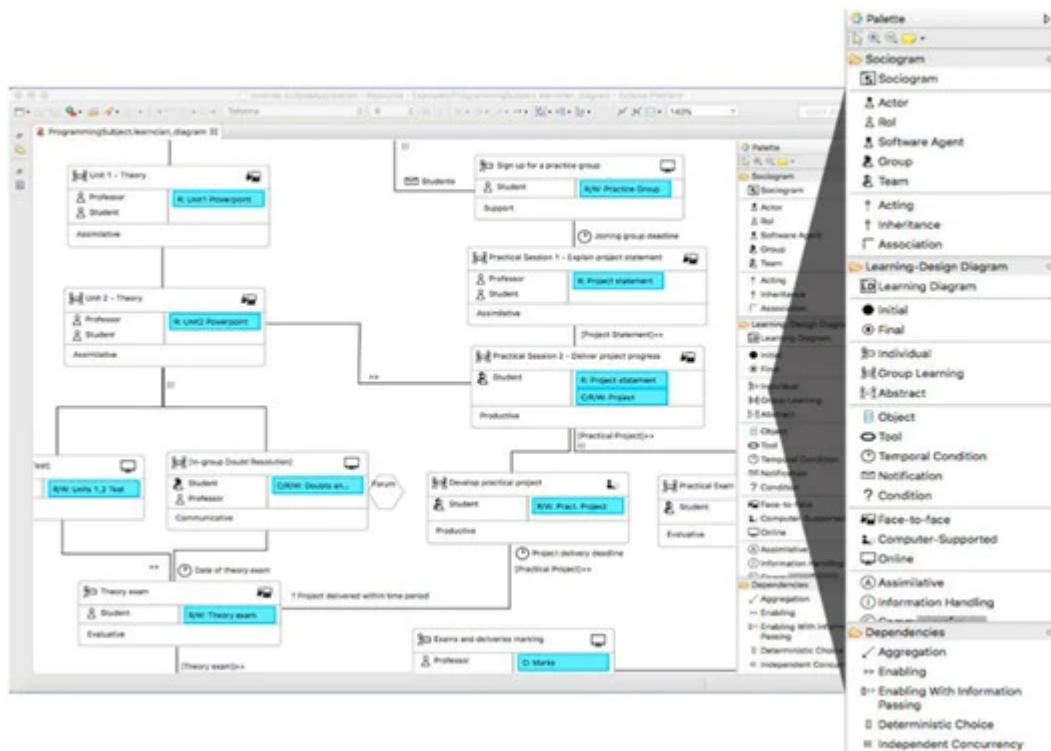


Figure 8. A learning design specified using the Learn-CIAT tool.

5. Usability and User Experience

All the developed systems require usability studies to demonstrate their efficiency in carrying out the tasks that the user or student, as the case may be, wants to perform.

MoLEF (Mobile Learning Evaluation Framework) [17] is a proposed framework in which both the usability and the pedagogical characteristics of an e-Learning environment can be evaluated with a large number of metrics. [Figure 9](#) shows the graphical representations of the technological and pedagogical usability parameters of a mobile application developed by the CHICO group. MoLEF has been used as a basis together with CIAM in the development of Learn-CIAN, which we discussed in the previous chapter.

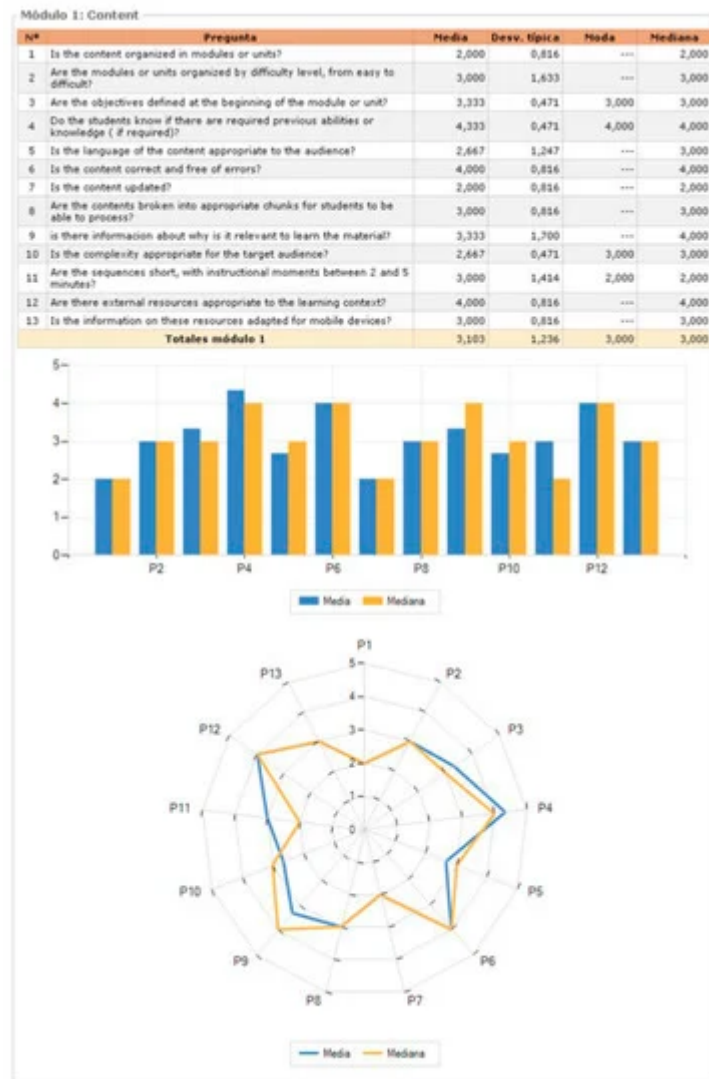


Figure 9. A screenshot of the tool implementing the MoLEF framework.

Classical usability studies based on questionnaires have been completed in recent years using more objective and direct sources of information, among which we highlight the use of eye tracking techniques. The concept of eye tracking refers to a set of technologies that make it possible to record and analyze the way a person looks at a given image (or user interface), providing physiological information on aspects related to the interest, attention, and cognitive effort involved in the visual analysis of the information displayed on the screen [18].

This eye tracking technique can be used in the evaluation of educational software in the field of programming teaching and, in particular, to teach greedy algorithms [19]. This technique has also been used to evaluate awareness characteristics in groupware systems [20] used in the CIAM [21].

The use of eye tracking in the study of multimedia educational materials is a very promising field. In particular, it has been very effective in studies aimed at primary school students, in the field of mathematics teaching [22][23].

The use of eye tracking techniques has been complemented in all the case studies with classical system usability tests. In the MoLEF [17], questionnaires with Likert-type scales are intensively and extensively proposed. The set of applications from Greedex to Greedex-Tab 2.0 [19] is an example of how different user interfaces are improved by system usability tests.

In all the studies, qualitative questionnaires with a Likert scale are used to improve the usability and productivity of the proposed systems [20]. In order to measure the usability and productivity of the CIAM, usability tests were carried out among software engineers comparing these tests with other methodologies through performance of different exercises. Once the exercises were completed by the subjects, a qualitative questionnaire was provided. Each subject was required to answer some questions related to the ease of use, the perceived complexity of each model, the suitability and usefulness of the notation, and specific aspects related to some of the CIAM diagrams. Additionally, some questions about intention to use were provided. With these results, we concluded that CIAM is a notation that has proven useful to help software engineers in the modeling process of interactive and collaborative aspects in groupware applications.

The usability and productivity of different tools were also evaluated by the students, by means of questionnaires in refs. [24][25]. For the Cole Programming system [24], we used usability tests answered by the students on the different collaborative tools used. Analyzing the results, we observed how the use of forums and voting pools seems less widespread when scoring these tools.

The conclusion drawn from the use of eye tracking is that objective usability measures complement and improve classical usability analysis techniques and are of great help in the area of human–computer interaction.

References

1. Ortega, M. Computers in Education: The Near Future, in *Computers and Education in the 21st Century*; Ortega, M., Bravo, J., Eds.; Kluwer Academic Publishers: Dordrecht, The Netherlands; Boston, MA, USA, 2000; pp. 3–16.
2. Di Pietro, G.; Biagi, F.; Costa, P.; Karpiński, Z.; Mazza, J. *The Likely Impact of Covid-19 on Education: Reflections Based on the Existing Literature and Recent International Datasets*; Publications Office of the European Union: Luxembourg City, Luxembourg, 2020.
3. Redondo, M.A.; Bravo, C.; Ortega, M.; Verdejo, M.F. Providing adaptation and guidance for design learning by problem solving. *The DomoSim-TPC approach*. *Comput. Educ.* 2007, 48, 642–657.
4. Redondo, M.Á.; Bravo, C. *DomoSim-TPC: Collaborative Problem Solving to Support the Learning of Domotical Design*. *Comput. Appl. Eng. Educ.* 2006, 4, 9–19.

5. Bravo, C.; Redondo, M.Á.; Ortega, M.; Verdejo, M.F. Collaborative environments for the learning of design: A model and a case study in Domotics. *Comput. Educ.* 2006, 46, 152–173.
6. Bravo, C.; Redondo, M.Á.; Ortega, M.; Verdejo, M.F. Collaborative Distributed Environments for Learning Design Tasks by Means of Modelling and Simulation. *J. Netw. Comput. Appl.* 2006, 29, 321–342.
7. Collazos, C.A.; Gutiérrez, F.L.; Gallardo, J.; Ortega, M.; Fardoun, H.M.; Molina, A.I. Descriptive theory of awareness for groupware development. *J. Ambient Intell. Humaniz. Comput.* 2019, 10, 4789–4818.
8. Duque, R.; Bravo, C.; Ortega, M. A model-based framework to automate the analysis of users' activity in collaborative systems. *J. Netw. Comput. Appl.* 2011, 34, 1200–1209.
9. Duque, R.; Bravo, C.; Ortega, M. An ontological approach to automating collaboration and interaction analysis in groupware systems. *Knowl. Based Syst.* 2013, 37, 211–229.
10. Weiser, M. The Computer for the 21st Century. *Sci. Am.* 1991, 265, 66–75.
11. Paredes, M.; Sánchez-Villalón, P.P.; Ortega, M.; Velázquez-Iturbide, J.Á. Collaborative Composition in a Foreign Language with Handheld Computing and Web Tools. *J. Univers. Comput. Sci.* 2007, 13, 948–958.
12. Villalon, P.P.S.; Ortega, M. AWLA and AIOLE for Personal Learning Environments. *Int. J. Contin. Eng. Educ. Life-Long Learn.* 2007, 17, 418–431.
13. Paternò, F. ConcurTaskTrees: An Engineered Notation for Task Models. In *The Handbook of Task Analysis for HCI*; Diaper, D., Stanton, N.A., Eds.; LEA: Mahwah, NJ, USA, 2004; pp. 483–501.
14. Molina, A.I.; Redondo, M.Á.; Ortega, M. A Methodological Approach for User Interface Development of Collaborative Applications: A case study. *Sci. Comput. Program.* 2009, 74, 754–776.
15. Molina, A.I.; Giraldo, W.J.; Gallardo, J.; Redondo, M.A.; Ortega, M.; García, G. CIAT-GUI: A MDE-Compliant Environment for Developing Graphical User Interfaces of Information Systems. *J. Adv. Eng. Softw.* 2012, 52, 10–29.
16. Molina, A.I.; Arroyo, Y.; Lacave, C.; Redondo, M.A. Learn-CIAN: A visual language for the modelling of group learning processes. *Br. J. Educ. Technol.* 2018, 49, 1096–1112.
17. Navarro, C.X.; Molina, A.I.; Redondo, M.A. Towards a model for evaluating the Usability of m-Learning systems: From a mapping study to an approach. *IEEE Lat. Am. Trans.* 2015, 13, 552–559.
18. Bojko, A. *Eye Tracking the User Experience: A Practical Guide to Research*; Rosenfeld Media: New York, NY, USA, 2013.

19. Arroyo, Y.; Molina, A.I.; Lacave, C.; Redondo, M.A.; Ortega, M. The GreedEx experience: Evolution of different versions for the learning of greedy algorithms. *Comput. Appl. Eng. Educ.* 2018, 26, 1306–1317.
20. Molina, A.I.; Gallardo, J.; Redondo, M.A.; Ortega, M.; Giraldo, W.J.; Orozco, W.J.G. Metamodel-Driven Definition of a Visual Modeling Language for Specifying Interactive Groupware Applications: An empirical study. *J. Syst. Softw.* 2013, 86, 1772–1789.
21. Molina, A.I.; Redondo, M.A.; Ortega, M.; Hoppe, U. CIAM: A Methodology for the Development of Groupware User Interfaces. *J. Univers. Comput. Sci.* 2008, 14, 1434–1446.
22. Navarro, O.; Molina, A.I.; Lacruz, M.; Ortega, M. Evaluation of multimedia educational materials using eye tracking. *Procedia Soc. Behav. Sci.* 2015, 197, 2236–2243.
23. Molina, A.I.; Navarro, O.; Ortega, M.; Lacruz, M. Evaluating Multimedia Learning Materials in Primary Education using Eye Tracking. *Comput. Stand. Interfaces* 2018, 59, 45–60.
24. Jurado, F.; Molina, A.I.; Redondo, M.A.; Ortega, M. Cole-Programming: Shaping Collaborative Learning Support in Eclipse. *IEEE-Rita* 2013, 8, 153–162.
25. Jurado, F.; Redondo, M.A.; Ortega, M. eLearning Standards and Automatic Assessment in a Distributed Eclipse Based Environment for Learning Computer Programming. *J. Comput. Appl. Eng. Educ.* 2014, 22, 774–787.

Retrieved from <https://encyclopedia.pub/entry/history/show/19146>