applied sciences

MDPI

Article

Learn-CIAM: A Model-Driven Approach for the Development of Collaborative Learning Tools

Yoel Arroyo [1,*], Ana I. Molina [1], Miguel A. Redondo [1] and Jesús Gallardo [2]

[1] Escuela Superior de Informática, Universidad de Castilla-La Mancha, Paseo de la Universidad, 4, 13071 Ciudad Real, Spain; AnaIsabel.Molina@uclm.es (A.I.M.); Miguel.Redondo@uclm.es (M.A.R.)
[2] Escuela Universitaria Politécnica de Teruel, Universidad de Zaragoza, Calle Atarazana, 2, 44003 Teruel, Spain; Jesus.Gallardo@unizar.es
* Correspondence: Yoel.Arroyo@uclm.es

Abstract: This paper introduces Learn-CIAM, a new model-based methodological approach for the design of flows and for the semi-automatic generation of tools in order to support collaborative learning tasks. The main objective of this work is to help professors by establishing a series of steps for the specification of their learning courses and the obtaining of collaborative tools to support certain learning activities (in particular, for in-group editing, searching and modeling). This paper presents a complete methodological framework, how it is supported conceptually and technologically, and an application example. So to guarantee the validity of the proposal, we also present some validation processes with potential designers and users from different profiles such as Education and Computer Science. The results seem to demonstrate a positive reception and acceptance, concluding that its application would facilitate the design of learning courses and the generation of collaborative learning tools for professionals of both profiles.

Keywords: MDE; MBUID; CSCL; TEL; pedagogical usability; awareness; authoring tool

Citation: Arroyo, Y.; Molina, A.I.; Redondo, M.A.; Gallardo, J. Learn-CIAM: A Model-Driven Approach for the Development of Collaborative Learning Tools. *Appl. Sci.* **2021**, *11*, 2554. https://doi.org/10.3390/app11062554

Academic Editor: Dov Dori

Received: 31 January 2021
Accepted: 10 March 2021
Published: 12 March 2021

1. Introduction and Motivation

The aim of this work is to establish a series of steps for the modeling of learning flows and their technological support, and to facilitate the generation of tools to support some of the activities defined in the flow. Specifically, we tackle the design and generation of collaborative activities that involve the intervention of several users in the resolution of learning tasks. For this purpose, we have decided to use the Model-Driven Development (MDD) paradigm in view of the benefits that it brings. The use of MDD implies a faster and more economical development process, in which the domain experts play a more important role, being able to focus on the modeling and definition of the content, abstracting the more technical aspects [1,2]. Thus, modeling help users with less technical knowledge in the design of their learning courses and the generation of their support tools, since they do not have to know how to develop them from scratch.

In the literature there are works that might be helpful in the modeling and generation of collaborative learning tasks, such as the works discussed in Section 2 of this paper. However, we have detected that most of them are focused on other domains or do not take into account some aspects that are important in the development of groupware [3,4]. Throughout the development of groupware there are aspects that must be considered such as synchronization, coordination or shared workspaces [5–7]. Moreover, it is also necessary to provide these systems with usable user interfaces and an adequate group conscience or awareness support [8–10]. However, as stated before, our goal is to support the specification of learning activity flows, including asynchronous and synchronous collaborative tasks, and the generation of the latter. Thus, in order to enable the generation of collaborative learning tools, this proposal must also consider some pedagogical aspects [11–13], such as the pedagogical usability [14,15]. The purpose of pedagogical usability is not to qualify

a didactic material as "good" or "bad", but to help users to choose the most suitable alternative for each specific learning situation [16,17].

To this end, a new systematic model-driven methodological approach, called Learn-CIAM, is proposed and described in this paper. Learn-CIAM includes the consideration of aspects of synchronous collaboration, incorporating widgets to support awareness (telepointers, radar views, session panels, etc.) in the automatic generation process of the final user interfaces, and aspects related to learning to guide the design process [3,8,18]. Technologically, Learn-CIAM is supported by an Eclipse-based Computer Aided Software Engineering (CASE) tool, the Learn-CIAT graphical editor, which is also described in this paper. The main purpose of Learn-CIAT is to guide professors through the design of their learning courses as well as through the semi-automatic generation of the collaborative support tools that might be used in their learning activities. Since the set of collaborative learning tools that could be generated is very broad, currently the Learn-CIAM methodological process supports the generation of a subset within them. In particular, it will be possible to generate tools for collaborative textual edition (for the edition of a manuscript, or source code, in programming tasks), Web browsing (for consultation and search of information) and graphical modeling (for real time modeling and simulation).

This paper is structured as follows. Section 2 introduces some related works. Section 3 describes the Learn-CIAM approach, introducing its main components: its conceptual, technological and methodological frameworks. Section 4 describes an application example of the Learn-CIAM methodology. Specifically, the design of a flow of learning activities and the specification and obtaining of a collaborative Java editor to be used in programming courses, and a graphical collaborative tool for learning digital circuits. Section 5 is a brief introduction to the set of validations followed by the Learn-CIAM proposal, describing in detail the last two validations carried out. Finally, Section 6 reflects, on the Learn-CIAM proposal, its strengths, its weaknesses, and possible future work derived from it.

2. Related Works

This work aims to apply the principles of the Model-Driven Engineering (MDE) to address a problem that, until now, had not been addressed comprehensively in the literature: the design of flows of learning activities and the development of collaborative learning tools following a model-based approach.

There are many different proposals in the literature for the modeling of collaborative tools, coming from different areas such as the Human–Computer Interaction (HCI), the Computer Supported Cooperative Work (CSCW) or the Software Engineering (SE) communities. Hence, for instance, among the most relevant contributions we can highlight the ConcurTaskTrees (CTT) notation [19], which facilitates the specification of the most interactive aspects of cooperative tasks; the Groupware Task Analysis (GTA) approach [20], whose conceptual framework proposes a set of representations for the specification of the different groupwork aspects and the environment in which it is carried out; or the Collaborative Usability Analysis (CUA) approach [21], a task analysis technique designed to represent groupwork tasks and carry out the usability evaluations of groupware system. Regarding more recent works, we can also find in the literature other works such as the Model of Coordinated Action (MoCA) method [22], focused on the description of complex collaborative scenarios and environments such as those which have diverse, high-turnover memberships or emerging practices; Domino [23], a framework for hybrid collaboration that involves simultaneous co-located and remote collaboration with phases of both synchronous and asynchronous work that spans multiple groupware applications and devices; Collab4all [24], which addresses accessibility issues when designing accessible and inclusive groupware; the Groupware for Collaborative Virtual Environments (G4CVE) method [25], focused on the development of collaborative virtual environments; Tesperanto [26,27], a model-based methodology for authoring technical documents; the Designing for Awareness in Shared Systems (DASS) framework [28], which gives a structured and comprehensive overview of design considerations for awareness, thus introduc-

ing interaction designers to awareness into a more generalizable and operational design knowledge; Unified Modeling Languages (UML) extensions such as the TOUCHE proposal [29–31], whose conceptual framework incorporates a series of requirement templates that include new information to record specific characteristics of groupware systems; and the Collaborative System Requirement Modeling Framework (CSRMF), inspired by the i* notation [32], which proposes the CSRML notation [9,33]. Nevertheless, most of these works deal with other domains or are focused on the technological aspects of the systems, without addressing the development of collaborative learning scenarios, neither incorporating an adequate and complete awareness support (only the CSRMF and the DASS frameworks do), nor pedagogical usability support (the CUA method only incorporates usability patterns, but not the pedagogical one).

One of the main barriers in the conceptual modeling of learning scenarios rises between Education and Computer Engineering areas. The main inconvenient is that teachers or educators do not usually describe the learning scenarios in as clear and concise a manner as software engineers do, who are forced to transform ideas and projects into the precise and logical word of machines [34]. As solution to this problem, the Visual Instructional Design Languages (VIDLs), the Educational Modeling Languages (EML) and some authoring tools (those built to help users in specifying learning scenarios with some specific terminology and graphic formalism) appeared in the e-learning domain as a tool for both teachers and software and educational multimedia developers. These languages can, therefore, be seen as Domain Specific Modeling (DSM) languages for specifying formal models about instructional designs such as learning scenarios. Some approaches that the MDE paradigm applies were found in the literature: the Learning Design Language (LDL) language [35], a domain-specific language to model collaborative learning activities; the Instructional Modeling Language (IML) [36], a modeling language that is aimed to teach MDE at the university level; the Massive Open Online Courses (MOOC) Authoring Tool (MOOCAT) [37,38], based on Business Processing Modeling Notation (BPMN) (https://www.bpmn.org/, accessed on 11 March 2021), a visual authoring tool for conceiving pedagogical scenarios in a simple way through graphical representation; the CaVa^{DSL} [39], aiming the generation of Virtual Learning Spaces for the use of the responsible of institutional archives or museums; INDIeAuthor [40], which allows authors with no programming knowledge to produce rich digital content for online publication, similar to that of the most common Learning Management Systems (LMS) such as Moodle (https://moodle.org/, accessed on 11 March 2021); or two EMLs such as PoEML [41], a process-oriented EML, and the [42] approach, an activity-oriented proposal based on Physics of Notations (PoN) [43]. Nonetheless, similar to works presented in the first lines of this section, none of them offers an adequate support to collaborative aspects such as awareness, nor to aspects related to the usability during the development of the learning tools or scenarios (neither technological nor pedagogical). As a response to this need, the next section introduces Learn-CIAM, a new methodological approach (and its specification language) that considers all these aspects (collaboration and learning issues).

3. The Learn-CIAM Proposal: Supporting Model-Driven Development of Collaborative Learning Tools

This section introduces Learn-CIAM, a new model-based methodological approach for the design of flows of learning activities and the semi-automatic generation of collaborative learning tools. This approach is based on the three different frameworks that we have designed, which are introduced in the next subsections. They are the conceptual, technological, and methodological framework.

3.1. Conceptual Framework and Specification Language

This section describes the aspects related to the specification and modeling of collaborative learning scenarios and tools supported by Learn-CIAN, that is, the conceptual framework, the notation derived from that framework and the consideration of pedagogical and awareness aspects.

3.1.1. Learn-CIAN Conceptual Framework and Notation

One of the most important components during a MDD process is the graphical notation that will serve as the basis of the automatic generation process. In the literature there are many notations that support the design of flows of activities and/or learning activities. For example, among the most widely used generic notations we could highlight BPMN, a standard set of diagramming conventions for describing business processes, whilst in the learning domain there also exist several notations, such as PoEML, mentioned in the previous section. However, after performing an analysis, several drawbacks have been found. Notations such as BPMN facilitate the modeling of business activity flows, but from a more generic and technical point of view, so we consider that it does not favor the work of users with a non-technical background, as can be the case of teachers/professors. Moreover, PoEML is a specific notation to design learning scenarios. This notation approaches modeling from 13 different perspectives in which the concerns involved in learning units have been separated (such as roles, tools, environment, organization, goal ...) [41]. Furthermore, it includes a large number of elements, as diagrams or icons. The main inconvenient is that the inclusion of such a great number of elements adds cognitive load to the notation, making it difficult to understand and increase the learning curve of the users. In addition, regarding [44], many of these symbols are quite similar, e.g., many rectangles are used for different concepts, which can only be discriminated by colors and the icons inside. Consequently, it is also difficult to use for users with a non-technical background. Furthermore, the visual components that both notations incorporate are insufficient for the purpose of this work. For example, they do not include the possibility of specifying the distinction between different types of learning tasks (either group or individual), nor configure the workspace of the collaborative tools or describe pedagogical aspects, being a prominent and necessary aspect for the aim of this work.

Therefore, first, we decided to create a new graphical notation that would allow the modeling of flows of learning activities which would be easy to use (incorporating an appropriate and manageable set of graphical elements) and which would be understood by more/less technical profiles (i.e., both Educators and Computer Scientists). This notation was called Learn-CIAN [45], and incorporates three types of learning activities depending on the number of users involved: individual, group and abstract (those which contain a new learning flow inside). In addition, the specification of each of these learning activities has been enriched by distinguishing up to three modalities of attendance: face-to-face, online or computer-supported (mixed); and up to eight subtypes of activity: assimilative, information handling, adaptive, communicative, productive, experiential, evaluative or support. These classifications (Table 1) were established according to works previously validated such as the taxonomy proposed by Conole [46] and a revised version of it [47]. In this work, we decided to also include the "support" learning activity subtype, which includes extra tasks that the teacher will organize as support for the students.

When designing a graphical notation like Learn-CIAN, we must identify what this language should express. To this end, the use of a meta-model (the conceptualization of a specific domain) to obtain the final notation would be recommendable. Figure 1 shows the meta-model (conceptual framework or *abstract syntax*) that serves as the basis for specifying the Learn-CIAN modeling language, whilst Figure 2 shows its concrete syntax, which represents the elements of the notation that will appear on the graphical editor (which we have called Learn-CIAT, and which we will describe in later sections), and that will be manipulated by the professors. These icons were designed bearing in mind their usability when applied by users (the multidisciplinary stakeholders involved such as Educators and Computer Scientists).

Table 1. Modality and subtype of a Learn-CIAN learning activity.

Modality	
Face-to-Face	Activities that must be carried out physically, attending the learning place with the rest of those involved
Computer-Supported	Activities that are not mandatory to be carried out in person, linked to digital tools that facilitate learning
Online	Tasks that are done through the Internet, and not in a "face-to-face way"
Subtype	
Assimilative	Passive activities: reading, seeing and hearing
Information Handling	Activities of searching, collecting, and contrasting information through text, audio or video
Adaptive	Activities in which students use some modeling or simulation software to put their knowledge into practice
Communicative	Spoken activities in which students are asked to present information, discuss, debate, etc.
Productive	Activities of an active nature, in which some artifact is generated, such as a manuscript, a computer program, etc.
Experiential	Practical activities, for the improvement of skills or abilities in a given context
Evaluative	Activities aimed at student assessment
Support	Support activities for students

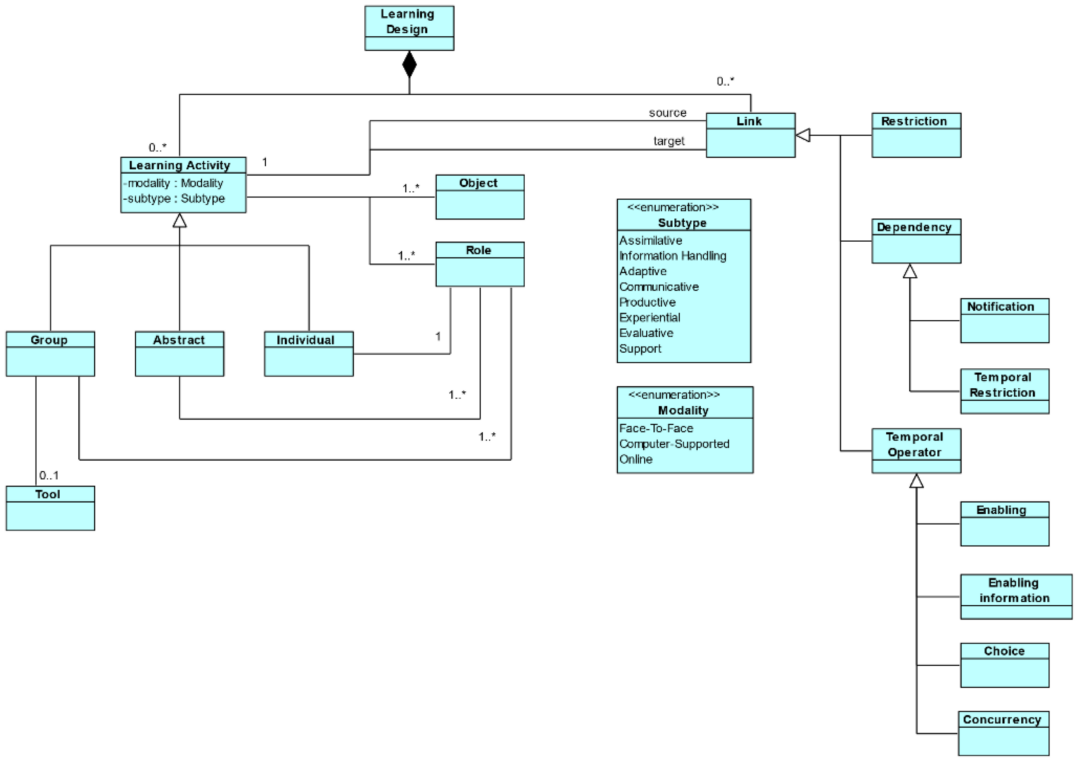

Figure 1. Learn-CIAN notation abstract syntax.

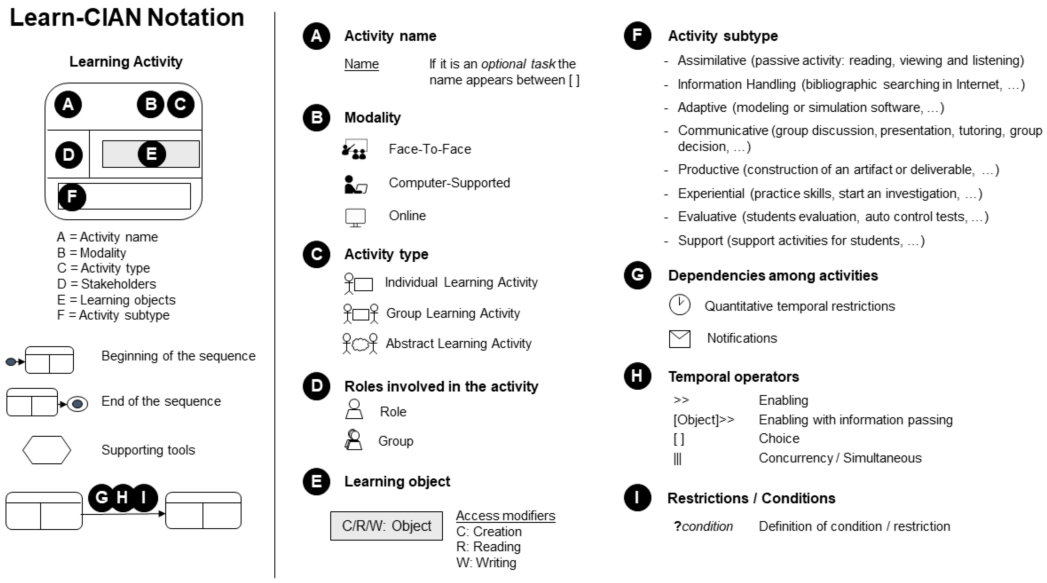

Figure 2. Learn-CIAN notation concrete syntax, adapted from [45].

3.1.2. Workspace and Awareness Aspects

The Learn-CIAN notation is therefore ready to support the design of flows of learning activities. However, it still does not include aspects related to awareness. To this end, we have developed the SpacEclipse plug-in, for the Eclipse IDE. SpacEclipse makes it possible to generate synchronous collaborative modeling systems adapted to many different domains. It incorporates a series of conceptual frameworks that contain, among other things, a broad set of awareness components (telepointers, radar views, session panels, etc.). Thus, its integration with Learn-CIAN (contemplating the generation of more tools than modeling) would allow the automatic generation of collaborative learning workspaces with an adequate awareness support. In this regard, up to three different learning workspaces have been determined in this work, respecting the rules established by the Learn-CIAN notation (Table 1) and the nature of the main tool to be chosen:

- A *graphical modeling* workspace, whose main tool is a synchronous collaborative graphical editor. It must contain a specific domain, i.e., a specific diagram on which students can work collaboratively. According to the rules established by the Learn-CIAN notation, a workspace with these characteristics would correspond to a group computer-supported and productive kind of learning activity.
- A collaborative workspace in which *textual* information is shared, providing, for instance, two types of editors: *plain text* (if the students want to work collaboratively in the elaboration of a manuscript) and *code* (if students want to address the implementation of a programming problem). This type of workspace would be related to group computer-supported and productive learning activities.
- Finally, a *Web browsing* workspace is provided, allowing students to collaborate in the search and consultation of information. A workspace of this kind would be a group computer-supported and information handling learning.

Each of these workspaces can be enriched with a series of widgets and components to configure the final user interface and provide support for communication, coordination and awareness features (Table 2). However, it is important to note that, since the professor must be focused on the design of the flow of learning activities and/or the configuration of the collaborative tool that supports each one of the group activities, this configuration will be optional and not mandatory. Thus, the professors will receive some support recommendation messages indicating the steps they might take for the configuration of the workspace, without being it mandatory for the final generation of the collaborative

tool. In case that they ignore these advices, the final tool will be generated with its default configuration.

Table 2. Collaborative components and workspaces, their configuration and awareness features.

Component	Workspace	Configuration	Awareness
Chat	All	Position/Language/Size/isStructured	Structured/Free
Telepointers	Graphical	-	Colors
Session Panel	All	Position/Language/Size	Colors/Icons/State
Console	Textual (Code)	Position	-
Outline/Radar View	Textual/Graphical	Position	-
Project Explorer	All	Position	-
Properties View	Graphical	Position	-
Turn-taking Panel	Graphical/Textual	Position/Language/Size	Semaphore/Votes
Problems View	Graphical/Textual	Position	-

On the other hand, if they finally decide to modify the default configuration, these components could be added respecting the nature of the component itself and the learning scenario to which it is desired to incorporate. So, a turn-taking panel, for example, makes no sense in a web browsing environment, since it basically performs visualization tasks and does not "produce" an artifact, as it does in a textual workspace. However, it would make sense to include a chat and a session panel to support this group task. In addition, each widget/component has its own configuration and awareness characteristics. To illustrate, a chat can be instantiated on the final user interface in different languages (English or Spanish), different sizes (Large or Small) and it can be free or structured (includes a series of phrases that helps in the decisions), whilst a telepointer has a unique color for each user. Specifically, the same color that the user possessed in the session panel.

All this information must also be conceptualized in a meta-model (Figure 3) with the goal of its incorporation into the Learn-CIAT graphical editor and the reflection of its use in the final user interface of the collaborative tools.

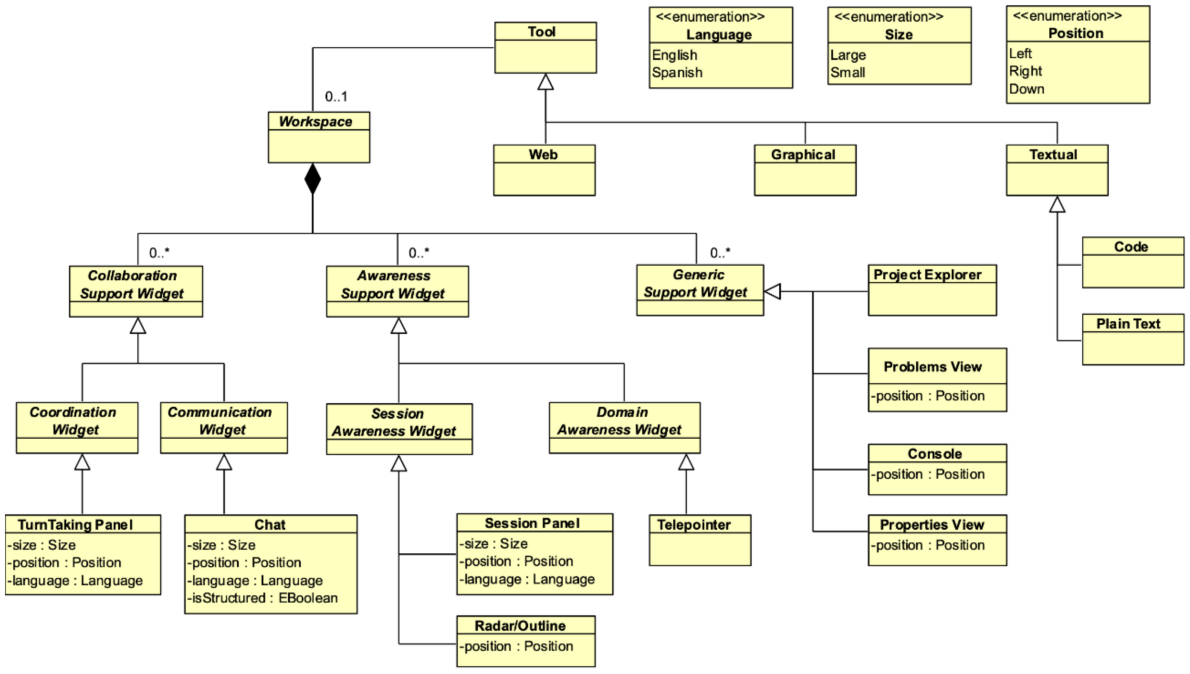

Figure 3. Collaborative tool components, workspace and awareness aspects abstract syntax.

3.1.3. Pedagogical Usability Aspects

The integration of the Learn-CIAN notation including the aspects presented before would allow the automatic generation of collaborative tools with an adequate awareness support. Nonetheless, the aspects related to learning, the so-called pedagogical usability [15], are still not covered. With the aim of guiding the design process, the Learn-CIAN notation should also include a series of learning issues or criteria that could be reflected in both learning flows and user interfaces of the collaborative tools generated.

Thus, we used a set of heuristics included in an evaluation instrument called CECAM (M-learning Applications Quality Evaluation Questionnaire). CECAM has been refined [48] to analyze its validity and reliability, so that it can be considered a fairly reliable instrument. It is a questionnaire for the design and evaluation of mobile-learning applications that proposes a series of dimensions and sub-dimensions for the evaluation of the usability factors that must support these kinds of applications. In the higher-level dimensions, it makes a distinction between pedagogical usability (which includes criteria related to learning factors) and technological usability (that of the user interface focused on mobile computing factors). Because the aim of this work is to design learning courses and generate some desktop collaborative learning tools, only the pedagogical dimension will be considered. In particular, CECAM comprises a total of 56 items, 28 of which refer to pedagogical usability, splitting it into five sub-dimensions: content, task and activities, social interaction, personalization and multimedia, which, in turn, are divided into a series of criteria.

These items (Table 3) have been included in Learn-CIAM. It is noteworthy that there are criteria that could not fit very well in the generation of the collaborative tool initially but could fit in the design and configuration of the learning activities that constitute a course. In other words, and as stated at the beginning, the aim of this work is not only to generate collaborative learning tools, but to provide a set of heuristics that help in the design and configuration of the learning courses. For example, when modeling a collaborative learning tool, the professor would be able to declare a collaborative, computer-supported and productive learning activity, selecting a code editor as tool and including a chat in its workspace. This way, the activity would be allowing communication between the users through the chat (S2 criterion), and the same activity would be presenting a work environment that reflects a real practice of their future professional life (T5 criterion). As a final example, let us imagine a professor that would define a Moodle activity as support of one of the learning tasks. The professor would be able to define an online activity, organizing the modules and material by levels of difficulty (C5 criterion), and including multimedia resources with a duration less of than 7 min (M3 criterion). To summarize, following these heuristics the professor will be guided through the design process of their courses and some of the collaborative learning tools that we already support.

Table 3. Learn-CIAM pedagogical usability criteria.

	Content
C1	The content is organized in modules or units
C2	The learning objectives are defined at the beginning of a module or unit
C3	The activities inform or warn that prior knowledge is required
C4	The explanation of the concepts is produced in a clear and concise manner
C5	The modules or units are organized by levels of difficulty
C6	The activities incorporate links to external materials and resources
	Task and Activities
T1	The activities have been proposed for the purpose of determining the level of learning acquired (questions, exercises, problem solving, etc.)
T2	The activities facilitate understanding of subject content
T3	The activities help improve the students' skills
T4	The activities help students to integrate new information with information studied previously
T5	The activities reflect real practices of future professional life
T6	The activities are consistent with students' abilities (neither too easy nor too difficult)
T7	There are activities that serve to evaluate students (questionnaires, exams, presentations, etc.)

Table 3. *Cont.*

	Social Interaction
S1	The learning system allows the students to work collaboratively
S2	The learning system allows you to communicate with other users (chat, e-mail, forum, etc.)
S3	The learning system allows us to share information (photos, videos or documents related to the task)
S4	The learning system motivates competitiveness among users (achievements, scores, etc.)
	Personalization
P1	The students have freedom to direct their learning
P2	The students have control over the content that they want to learn and the sequence that they want to follow
P3	The activities represent multiple views of the content
P4	The students can personalize their learning through material (videos, texts or audios)
P5	The questionnaires and assessments can be customized to fit the student's profile
	Multimedia
M1	The activities present elements that facilitate learning (animations, simulations, images, etc.)
M2	The multimedia resources have been appropriately selected to facilitate learning
M3	The multimedia resources have a duration of less than 7 min
M4	The multimedia resources have good video, audio, and image quality
M5	The multimedia resources can be downloaded
M6	There is an adequate proportion of multimedia resources

3.2. Technological Framework

This section details the technologies and languages chosen to give computational support to the proposal. First, Eugenia is introduced as the technology used for the development of graphical editors. Then, we list a set of languages for the management of models on Eclipse. Finally, the Learn-CIAT graphical editor is also introduced, which supports the conceptual framework presented before.

3.2.1. Eugenia and Epsilon Core Languages

The Eclipse Modeling Framework (EMF) (https://www.eclipse.org/modeling/emf/, accessed on 11 March 2021) and Graphical Modeling Framework (GMF) (https://www.eclipse.org/gmf-tooling/, accessed on 11 March 2021) are two frameworks to support the implementation of graphical modeling tools on Eclipse. Thanks to the conjunction of these two frameworks, from the definition of an Ecore meta-model (EMF own meta-model language), and a series of models that contain the graphical syntax of the editor, it is possible to generate high quality graphical editors on Eclipse. Moreover, they provide options for the customization of almost all aspects of the generated editor. Therefore, EMF and GMF are a very powerful and flexible set of frameworks. However, greater power and flexibility means a greater increase in development complexity. The implementation of a graphical editor using EMF and GMF results in a little oriented, laborious, and error-prone process, as it requires developers to develop and maintain, by hand, a large number of low-level models interconnected and difficult to understand.

The main goal of Eugenia [49] is to reduce the complexity of the creation of GMF-based models. The EMF/GMF process forces developers to implement and maintain the GMF models manually, making this task considerably more difficult. To address this inconvenience, Eugenia incorporates a set of annotations (https://www.eclipse.org/epsilon/doc/eugenia/, accessed: 11 March 2021) that can be applied to the main Ecore meta-model. This way, Eugenia can execute a series of automatic transformations to generate the necessary models and reach the graphical editor, increasing the GMF level of abstraction.

Eugenia is included as part of Epsilon (Extensible Platform of Integrated Languages for mOdel maNagement) (https://www.eclipse.org/epsilon/, accessed on 11 March 2021) [50], a platform whose main objective is the creation of specific languages for model management tasks, such as transformation, code generation, comparison, fusion, refactoring and

model validation. In the Learn-CIAM context, Eugenia is used for a faster and easier generation of graphical editors. Specifically, it should be used in the generation process of the collaborative modeling tools. Initially, these graphical editors are single user; however, the Epsilon platform incorporates a series of high-level languages that allow the manipulation, validation, and modification of the final generated code, allowing to make the generated graphical editor collaborative. Epsilon is composed of many languages and tools. Thus, the set of languages that must be known by Learn-CIAM end users is the following:

- The *Epsilon Object Language (EOL)*, whose main objective is to provide a reusable set of functionalities to facilitate model management. In the Learn-CIAM context, EOL is required when the user wants to customize a graphical editor or add additional functionality (e.g., reduce the distance between labels and nodes, or embed custom figures to the nodes).
- The *Epsilon Validation Language (EVL),* whose main objective is to contribute to the model validation process. Specifically, it allows the user to specify and evaluate constraints on models which come from different meta-models and modeling technologies. In the Learn-CIAM context, EVL is required when the user needs to implement additional validations to the conceptual framework (e.g., to check whether the specification of a concrete collaborative tool respects some of the Learn-CIAN, the workspace and the pedagogical usability rules).
- Finally, the *Epsilon Generation Language (EGL)*, as the language that transforms the information specified in the models into the desired code. In other words, it is the language used to carry out the M2T (model-to-text) transformations. In the Learn-CIAM context, EGL is required to implement the templates that will guide the automatic generation process of the final collaborative tools. This generation process will be triggered from a specific diagram, previously modeled and validated by the professor using the Learn-CIAT graphical editor, introduced in the next subsection.

3.2.2. Learn-CIAT Graphical Editor

The Learn-CIAT graphical editor (Figure 4) is an important and fundamental part of Learn-CIAM, as it is the technological support for the proposal. As with any Eclipse-based graphical editor, it is composed of a canvas (drawing area) and a palette which contains the nodes and relationships that can be dragged and instantiated over it in the design of each new diagram. Thus, by using Learn-CIAT, it is possible to model and configure all the components introduced previously, i.e., the Learn-CIAN notation in order to model the flows of learning activities (Figure 4a); the workspace and awareness configuration of collaborative tools (Figure 4b); and the criteria related to pedagogical usability (Figure 4c).

The basis for its model-driven generation process, which will be described in the methodological framework, is the set of conceptual frameworks introduced before, interrelated with each other (Figure 5). This meta-model is the main input artifact of the automatic generation process using Eugenia and collects the set of nodes and relationships that it must support (the concrete syntax spread over the palette and canvas of the graphical editor).

3.3. Methodological Framework

This section details the Learn-CIAM methodological framework, which is divided into two phases: (1) the automatic generation of graphical editors; and (2) the modeling, validation and generation of collaborative tools, supported by the Learn-CIAT graphical editor. Since different collaborative tools can be generated (graphical editors, text editors and Web browsers), it is mandatory to highlight that the first phase will only be necessary when the user needs a collaborative graphical support tool. That is, the first phase is necessary to generate the graphical editor with a specific domain that will be modeled, configured and validated by the professor during the second phase of the method as support for one of the group learning activities.

First, we introduce the users involved and the specific roles that they may play. Then, each of the phases into which it is divided are enumerated and described.

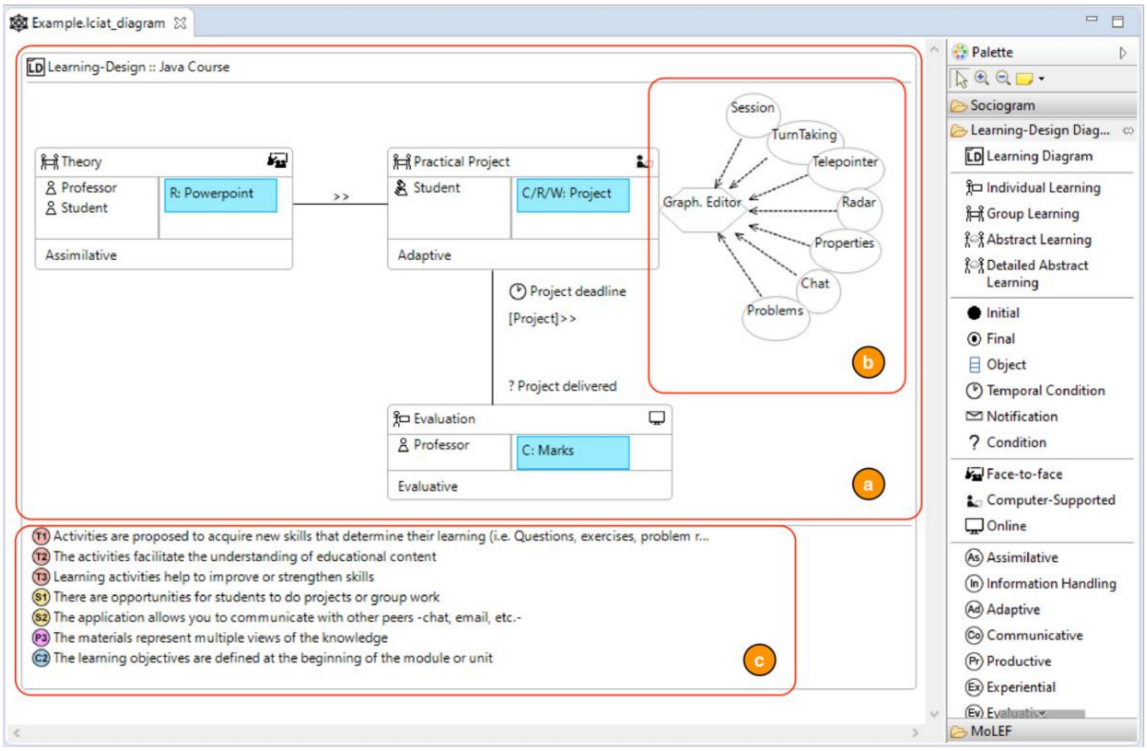

Figure 4. Learn-CIAT graphical editor.

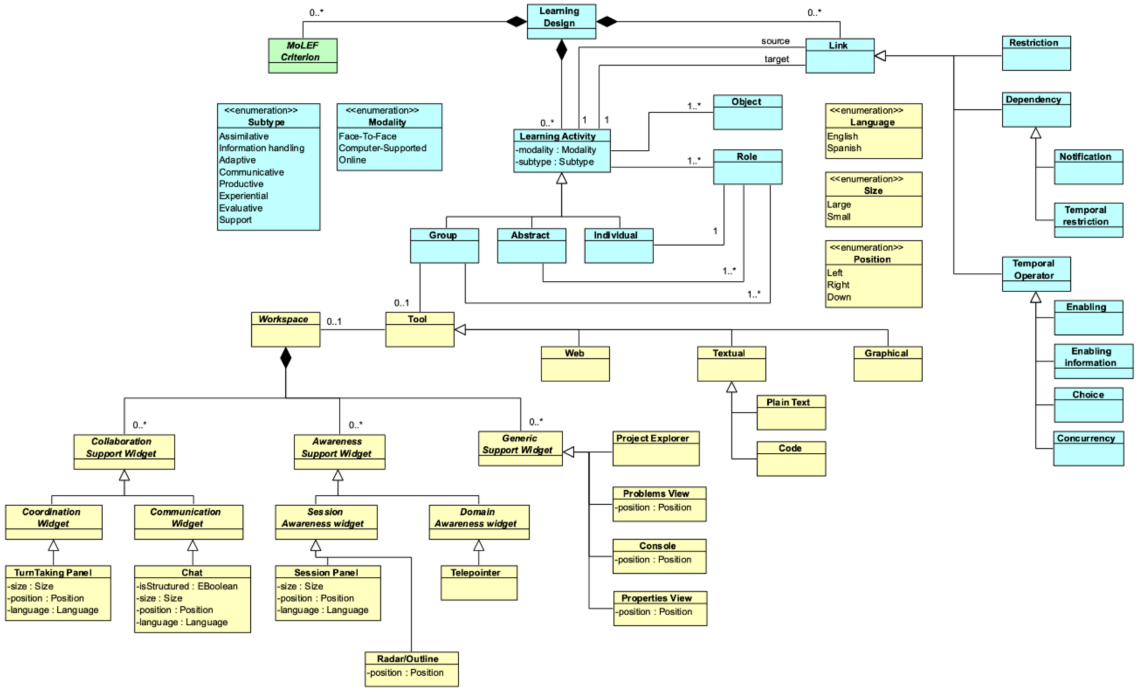

Figure 5. Learn-CIAT graphical editor abstract syntax.

3.3.1. Roles

Throughout the development process there will be three roles that the different users will be able to play. Since the final scope of this work is the design of learning flows and the generation of collaborative tools, end users will be mostly people related to this

context of work (professors and students). The different roles which can be played are described below:

- Domain expert: a user who brings their experience and knowledge of a given context to the development of tools. In the first phase, it is the user in charge of elaborating the domain that is later supported by the collaborative graphical editor. In the second phase, and using the Learn-CIAT graphical editor, it is the user in charge of designing and modeling the flow and/or the collaborative tools to be generated. This role is usually found in most stages of the process and it is linked to the figure of a professor.
- End user: user who will make use of the collaborative tool generated. As it is a collaborative tool, the users will work together to solve a problem previously declared by the domain experts. This role is common in the last phases of the development method and is usually associated with the figure of a student.
- Software engineer: user who has sufficient experience and knowledge of the technology implied in the methodological approach. Thus, they are software engineers who are experts in the field of model-driven development and, specifically, its Eclipse implementation. These kinds of users are spread at various graphical editor development stages and supporting other users at the final development stages of the collaborative tools.

3.3.2. Phases

The methodological proposal consists of two different phases. In the first phase (Figure 6), we enumerate the stages to be followed in the elaboration of an Eclipse-based graphical editor. In the second phase (Figure 7), we list the stages required to model, validate, and generate a collaborative tool. The basis of this second phase is the graphical editor, Learn-CIAT, which was also created and generated by applying the first phase of the method.

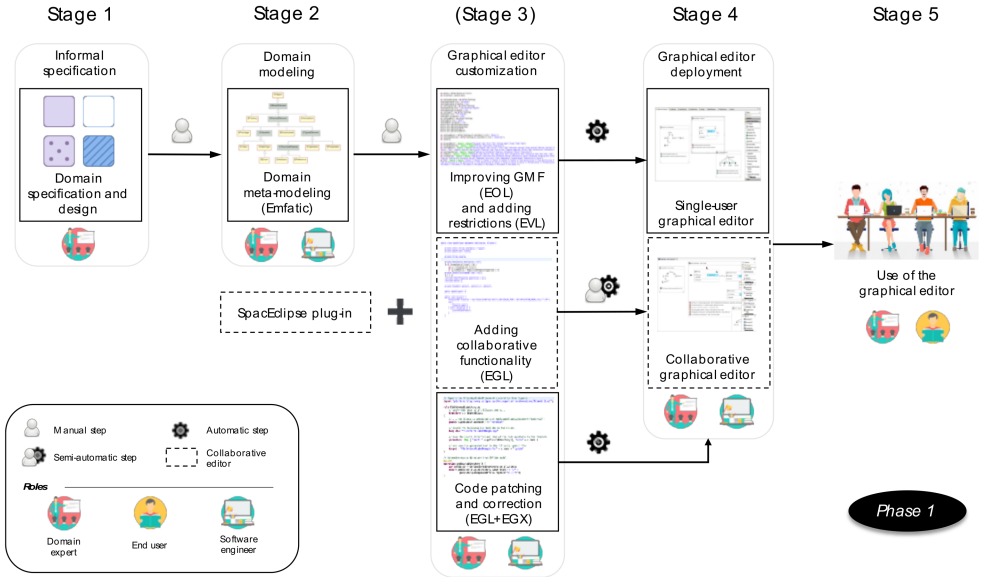

Figure 6. Phase 1: Generation of a single-user/collaborative graphical editor.

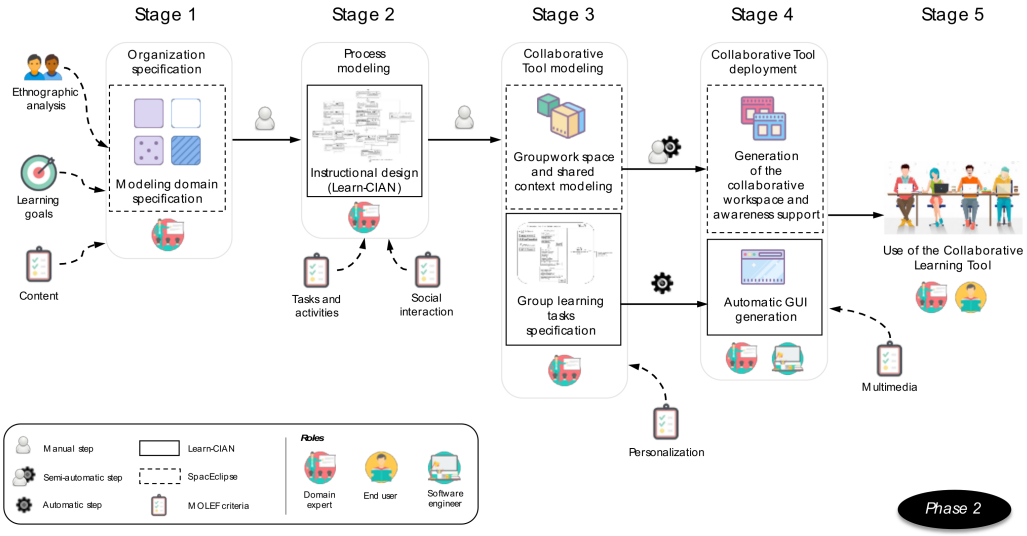

Figure 7. Phase 2: Modeling, validation and generation of collaborative learning tools.

First Phase: Elaboration of a Graphical Editor Using Eclipse Technology

The first phase consists of a total of five stages. This process is similar to that of generating graphical editors, which uses Eugenia technology and Epsilon core languages, except for the third stage, in which it is possible to provide collaborative functionality to the generated single-user editor through its integration with SpacEclipse. The stages and the users involved are listed below.

- Stage 1—Informal specification: in the first stage, the domain experts must generate a first "sketch" of the graphical editor, specifying their working domain. For instance, a graphical editor could be generated for modeling digital circuits, network topologies or use cases, among many other domains.
- Stage 2—Domain modeling: during this stage, the domain experts, with the help of the software engineer, must proceed to model the domain defined previously. For this task, the software engineer must be an expert in the chosen technology, identifying and relating each of the parts of the graphical editor "sketch" with the proper meta-modeling components. Specifically, it must generate the Ecore representation. After that, since Eugenia was the chosen technology, it is also necessary to add the adequate annotations. All this information (Ecore meta-model + Eugenia annotations) must be included in a unique Emfatic file.
- Stage 3—Graphical editor customization: in the third stage, the collaboration between the domain expert and the software engineer is very important. This stage is not mandatory; nonetheless, when the desired editor is more complex than what Eugenia technology can provide/generate by using its basic annotations, it will be necessary to modify the figures and relationships that it generates by default to adapt them to the editor's needs. To this end, it is necessary to generate (or reuse) an EOL template which contains the appropriate modifications. This template will be detected and applied by Eugenia automatically during the generation process. In addition, the restrictions that are automatically generated from the meta-model will sometimes not be sufficient; therefore, it will be necessary to add more manually. With this aim, one or more EVL templates must be generated and they must contain the restrictions or advices that the editor must support additionally. Furthermore, the users can also provide collaboration functionality to the editor. To achieve that, the SpacEclipse plug-in must be used. This plug-in contains the basic code in order to receive single-user editors and provide them with collaborative functionality, generating a collaborative graphical editor for Eclipse automatically. This is possible by simply pressing a button that applies a prepared script, which adds the necessary collaborative functionality to

the editor. Another interesting option provided by the technology is the possibility of modifying the final code generated by Eugenia, using EGL/EGX (EGX files are EGL templates which coordinate the generation of the final code (generating the code in the corresponding path)) templates. This step is usually necessary in those cases in which, despite having made some visual modifications, the generated editor still does not meet expectations (e.g., it may be necessary to bring the nodes closer to the label of one of their attributes).

- Stage 4—Graphical editor deployment: once the graphical editor has been generated and it has been verified that its visual aspect and functionality are the desired ones, the domain expert and the software engineer can proceed to the deployment of a stable version. Basically, they must generate an installer which includes the functionality that assures the correct functioning of the graphical editor in other instances of Eclipse. This installer could be shared with the end users in different ways: physical (USB, HDD, etc.) or web-based (through Eclipse shop or an external URL).
- Stage 5—Use of the graphical editor: finally, once the end users have installed the plug-in in their Eclipse platform, they will be able to start making designs on the graphical editor.

Second Phase: Modeling, Validation and Generation of Collaborative Learning Tools

The basis of the second phase is the Learn-CIAT graphical editor. Using it, the professors will be able to model the main features of the flow of learning tasks and tools to support collaborative activities. Thus, this phase receives a model/diagram as main input elaborated using Learn-CIAT. The complete process is summarized in the following lines.

- Stage 1—Specification of organization, learning objectives and domain: as in the first phase, a domain expert should handle this stage. This user will usually be linked to the figure of a professor, since the main objective is to model flows of learning activities and collaborative learning tools (when considered necessary as support of some of the activities). Before starting the application of the process described in our proposal, the expert should carry out an ethnographic analysis of the end users to whom the system will be directed, and it is necessary a clear understanding of the learning objectives to be achieved. At the same time, they should specify the working domain. Thus, it is worth noting that the requirements elicitation is a process that is considered external to our proposal. However, the obtained information is implicitly considered into the diagrams of our models. First, experts must design through a new "sketch" the flow of a course, that is, the set of learning activities and the order or sequence that constitutes it. This flow may contain, provided the domain expert considers it appropriate, computer-supported group learning activities, which will be used for the automatic generation of the collaborative tools.
- Stage 2—Modeling process: in the second stage, the domain expert must use Learn-CIAT to translate the previous specifications into digital format. In other words, the domain expert must generate a diagram using the graphical editor. This diagram might contain the Learn-CIAN learning flow, and the pedagogical usability criteria that it considers appropriate to achieve the learning objectives initially stated.
- Stage 3—Collaborative tool modeling: this stage can be carried out by the domain expert. So as to model the collaborative tools, Learn-CIAT contains the necessary nodes and relationships to instantiate and configure the computer-supported group activities, as well as the collaborative workspace and its working context, allowing the selection of the tools and awareness features that are considered appropriate. To illustrate, it is possible to model a collaborative program edition supporting task (i.e., a collaborative Java editor), whose workspace/user interface is made up of a session panel, to consult the users who are connected to the session, and a chat, to support the communication between them.
- Stage 4—Collaborative tool generation and deployment: once the group activity has been modeled and validated, the domain expert could generate the final code

automatically, including those tools and awareness components that have been defined. The collaborative tools can be generated by the same domain expert since they will only have to execute the transformation process. After that, the software engineer or domain expert must, once again, oversee the plug-in deployment, as in the first phase.

- Stage 5—Use of the collaborative learning tool: the last stage involves the end users. This figure will usually be linked to students. Similar to the first phase, the end users must install the generated collaborative tool on their Eclipse platforms and solve the determined learning task.

4. Application of the Learn-CIAM Methodology

This section describes an example of application of the Learn-CIAM methodological approach. First, we present the design of a flow of learning activities using the Learn-CIAT graphical editor. Then, we present the generation process of two different collaborative learning tools from this flow: (1) a collaborative Java programming tool; and (2) a graphical collaborative tool for learning digital circuits. For a better understanding, the material and installation instructions were uploaded on GitHub (https://github.com/yarp14?tab=projects, accessed on 11 March 2021).

Figure 8 shows an instance of Learn-CIAT which describes the learning flow of a potential technological course. This course consists of four activities: a first theoretical activity in which the students and the professor participate in person (face-to-face); two practical activities in which the students participate in teams, such as the elaboration of a programming project and the elaboration of a digital circuit modeling project. Each of these activities can be supported by a different collaborative tool: the "Programming Project" (Figure 8b), supported by a collaborative Java editing tool; and the "Digital Circuit Project" (Figure 8c), supported by a collaborative graphical modeling tool; finally, there is an assessment activity in which the professor evaluates the work of the students.

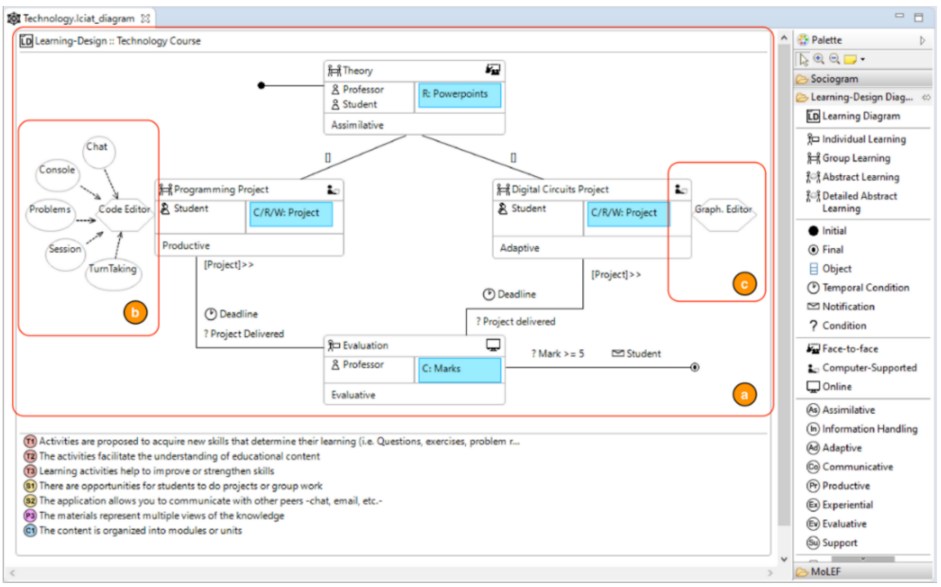

Figure 8. An example of flow of learning activities modeled using Learn-CIAT.

As a starting point, and as an example of the generation of collaborative tools from a learning diagram, we begin by describing the generation of the support tool for the "Programming Project" activity. This tool is a text-based collaborative tool to learn Java. Therefore, the first phase of the Learn-CIAM methodological framework should be ignored. Then, we describe the process required for its generation, which corresponds with the second phase of the method:

- In the first stage, the domain expert (the professor) must define the learning objectives, the flow of learning activities to be carried out (Figure 8a) and, if they consider

it necessary, the collaborative tool to be generated. In this example, a text-based collaborative tool for learning Java will be generated (Figure 8b).

- In the second stage, the professor must transform this information to a digital format. Specifically, they must use the Learn-CIAT graphical editor to generate a diagram in Learn-CIAN notation, similar to that in Figure 8. The third stage is very important, especially when the professor considers that a support tool is essential for one of the group learning activities. In this stage, the professor must define the tool to be generated (one of the ones that we already support: graphical editor, text editor or Web browser). In order to do so, they must select it in the activity (EGL Tool), mark it as it is going to be generated (EGL Transformation) (Figure 9a), and configure its workspace (as stated before, this configuration is not mandatory). To help and guide the professor through this process, we have incorporated a series of validations (warnings and errors) defined in the EVL language (Figure 9b). To launch a model/diagram validation, the professors only need to press a button in the environment. In Figure 9, for instance, Learn-CIAT has detected up to two errors and two recommendations after the validation process of the model: (1) an error showing that a user is trying to add telepointers to a code editor, when these can only be added to graphical modeling environments (Table 2); (2) a warning showing that the computer-supported group activity named "Programming Project" has been declared as assimilative, when these types of tasks should be of a productive nature (Table 1); and (3) lastly, different warnings showing that the professor can configure the widgets if they want (e.g., to introduce the desired position for the session panel of the code editor). In terms of pedagogical usability, since the learning flow designed does not fulfill criterion S2: "*The learning systems allow the communication with other uses (chat, e-mail, forum, etc.)*" (Table 3), there is one error added to the diagram. This error is displayed since the tool selected to be generated, a code editor, does not incorporate at least one chat into its workspace. Furthermore, the user can correct these errors or warnings automatically by pressing a button which will initiate the corresponding quick fix (if it was declared in the EVL template).
- In stage 4, the professor can initiate the automatic collaborative tool generation process (Figure 10). To this end, we implemented a series of templates written in EGL. Specifically, seven EGL templates have been prepared. These templates are responsible for generating, in a dynamic manner, the Java classes that contain the user interface and the functionality of the final tool, with respect to the configuration of the Learn-CIAT validated model. At the same time, in order to coordinate the generation of each one of the classes, it was necessary to implement an additional EGX template, which is responsible for the generation of each Java class in the path indicated within the SpacEclipse plug-in. This way, the user only needs to focus on modeling and validating the desired system, as it will be able to generate learning tools such as the one shown in Figure 11, by simply pressing a button. It is also important to note that, in this stage, the Software Engineer must launch the database and the server included with SpacEclipse, where the professor must access and create a new learning session. This session will be accessed later by the end users (the students) to start working collaboratively. Figure 11 shows a collaborative code editor (in Java language) surrounded by the following set of components: a session panel (1), a project explorer (2), a chat (3), a console (4), a turn-taking panel (5) and a problem view (6), each of them with their own configuration. For instance, the *chat* has been placed at the bottom of the editor, it is large, and it is translated into English, while the *session panel* is large, has been placed on the left of the editor, and has also been generated in this language. In addition, each component incorporates its own awareness characteristics. Thus, for example, the *turn-taking panel* includes a semaphore, located at the top right, which changes color according to the user's request: take the turn (in green) or not take the turn (in red), and a voting system at the bottom, which indicates whether this change is accepted.

- In the final stage, either the professor or the Software Engineer must export the collaborative tool generated and share it with the students. Then, the students must proceed to install it in their environments and access the corresponding collaborative learning session.

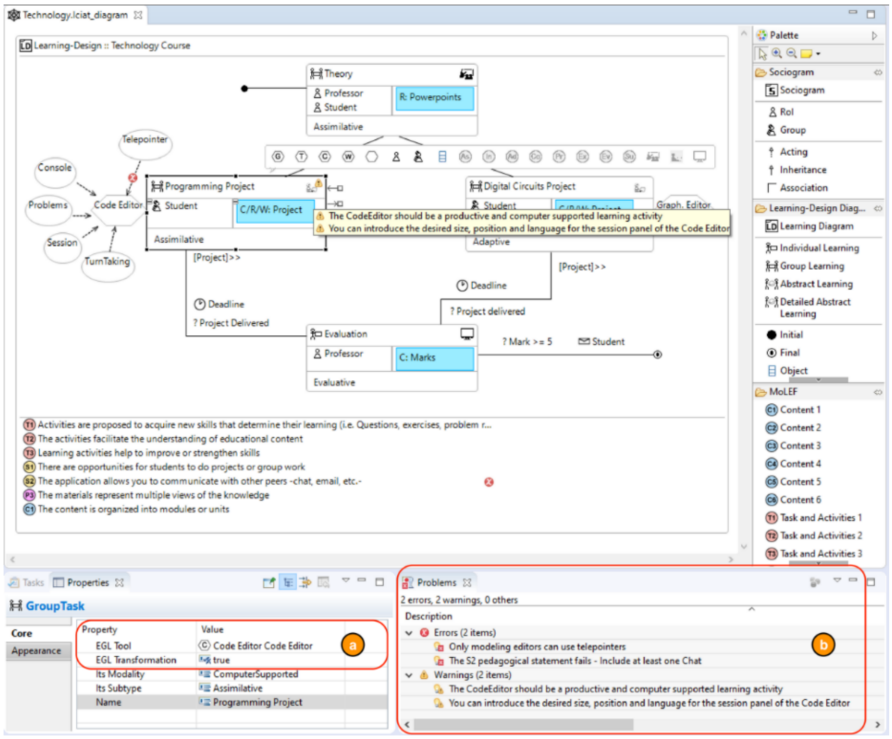

Figure 9. A validation process using the Learn-CIAT graphical editor.

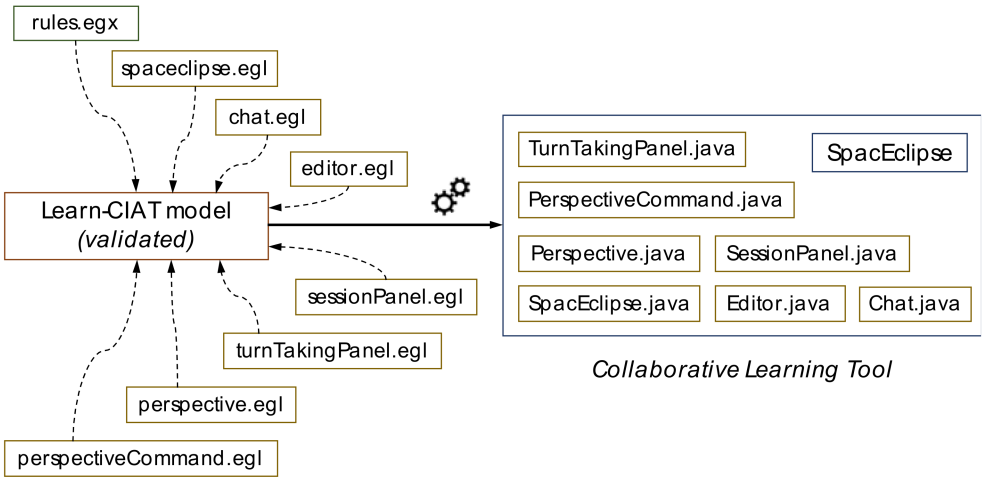

Figure 10. Collaborative learning tool generation process.

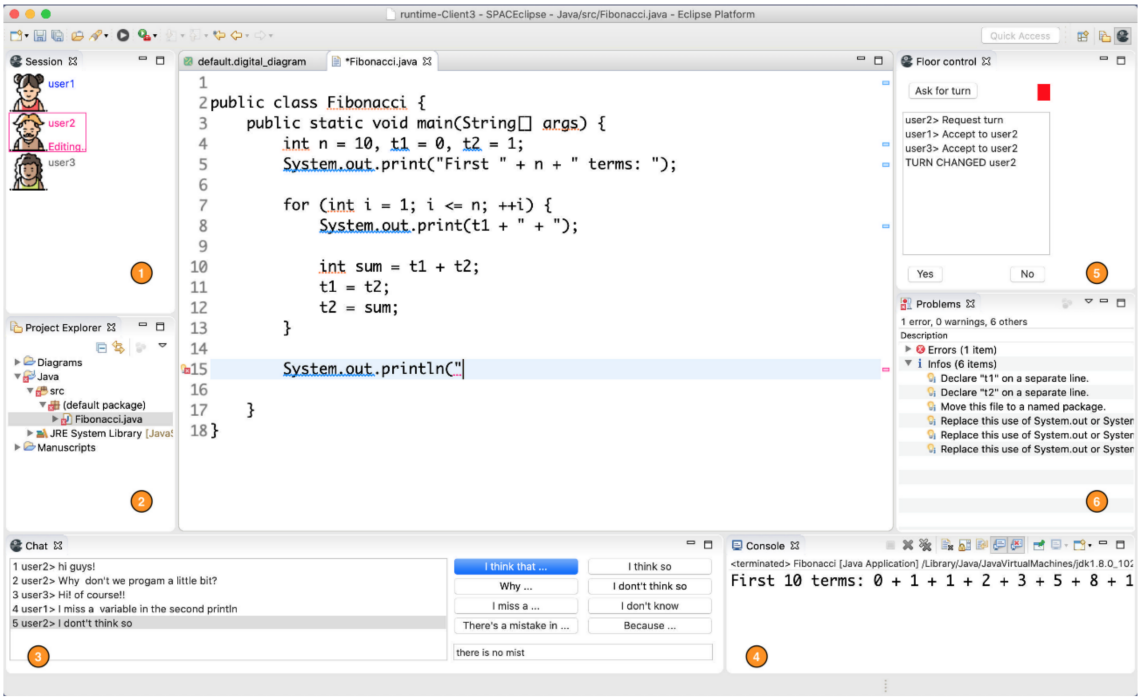

Figure 11. The collaborative Java programming tool generated.

In the second-generation example, since the final objective is the creation of a collaborative modeling tool for learning digital circuits (Figure 8c), it is necessary to follow the steps of both phases of the method. First, it is necessary to follow the steps of the first phase for the generation of the graphical editor with this domain (Section 3.3.2):

- In the first stage, the domain expert (professor) must define the visual aspect of the graphical editor. In this example: digital circuit diagrams.
- In the second stage, the professor must share the domain with the Software Engineer who will proceed to its implementation in Emfatic code. At this point, it is possible to generate the single-user graphical editor automatically. However, as it is intended to provide it with collaborative functionality, it is also necessary to execute the third stage.
- To make the graphical editor a collaborative one, the professor must follow the next steps: (1) import the SpacEclipse project into the workspace; and (2) apply a script. This script is already prepared for every graphical editor, regardless of the domain, in the form of a patch named *CollaborativeDiagramEditor.patch*, adding the Java collaborative functionality that will be added to the SpacEclipse instance. This script can be applied/disapplied by the professor just by clicking on the "patches" folder thanks to the Eugenia technology. This facilitates the work to be done by the professors, as they do not need to know anything about the most technical aspects.
- Now, the collaborative graphical editor is prepared. Then, the professor only needs to follow the steps of the second phase of the method, as in the first example. The difference is that, in this case, the professor needs to select a graphical editor to be generated, not a code editor, and make sure that the graphical editor previously generated is in the workspace next to the learning diagram (at the same root path). Figure 12 shows the collaborative digital circuit modeling editor generated.

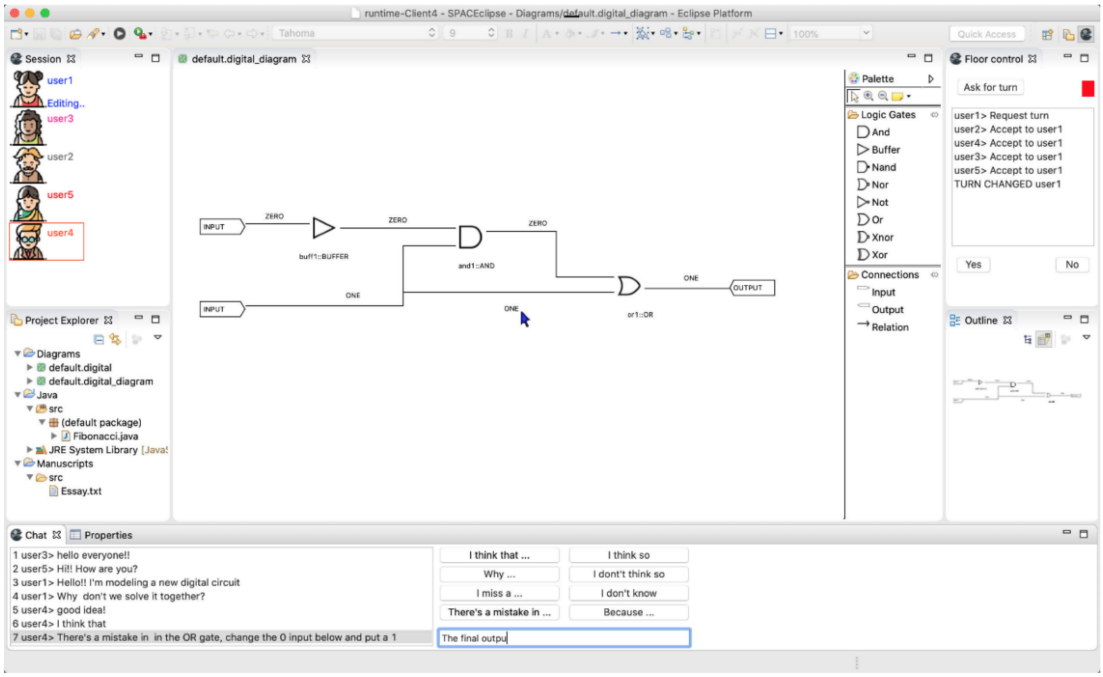

Figure 12. The collaborative graphical modeling digital circuit tool generated.

5. Validation

The Learn-CIAM proposal has undergone three different evaluations. First, the aspects related to the pedagogical usability design included in the proposal underwent an evaluation process [48]. In this evaluation, two mobile-learning applications were implemented following those pedagogical criteria with satisfactory results. Second, the Learn-CIAN notation was validated with two groups of instructional designers coming from different fields: Education (less technical) and Computer Science (more technical) [45]. Again, the results obtained denote positive perceptions of its use. Finally, this paper presents a new validation carried out on Learn-CIAM. This experience consists of two different experiments. In the first one, we carried out a demonstration and an exchange of opinions with a group of experts while, in the second one, the group of experts used the Learn-CIAT tool to model and validate a learning course and generate a collaborative tool to support one of the activities.

5.1. Plannig and Execution of the Experiments

This section describes the experiment carried out to provide empirical evidence with regard to the potential suitability, user-friendliness and completeness of the Learn-CIAM approach. This validation involved potential designers and users of the proposal, with different profiles: experts in Education (ED) and Computer Science (CS).

5.1.1. Research Questions

The research was conducted to assess the opinion and subjective perception of potential users of the proposed framework regarding its main components (methodological framework, specification language and conceptual framework and, finally, the technological support). This objective is embodied in the following research question:

RQ. What is the subjective perception of experts in the areas of Education and Computer Science regarding the *suitability*, *completeness*, *usefulness* and *ease of use* of the Learn-CIAM proposal and its main components?

Since the research question under consideration may be a broad one, we propose a set of sub-questions:

RQ1. What is the subjective perception of potential users regarding the *need* to apply a process of modeling and generation of collaborative learning tools?
RQ2. What is the subjective perception of potential users regarding the *convenience of applying the model-based development paradigm* to approach the development of collaborative learning tools?
RQ3. What is the subjective perception of potential users regarding the selection of *Eclipse* as environment to support the proposed method?
RQ4. What is the subjective perception of potential users regarding the *proposed* Learn-CIAM *process* (its phases and stages and the use of high-level languages)?
RQ5. What is the subjective perception of potential users regarding the support offered by Learn-CIAM to *non-technical users*?
RQ6. What is the subjective perception of potential users regarding the *suitability, complexity* and *completeness* of modeling elements supported by the Learn-CIAN notation?
RQ7. What is the subjective perception of potential users regarding the *suitability* and *completeness* of the modeling elements related to *awareness* and *pedagogical usability* supported by the proposal?
RQ8. What is the subjective perception of potential users regarding the *difficulty of applying* the modeling, validation and generation *stages* of collaborative learning tools supported by Learn-CIAT?
RQ9. What is the subjective perception of potential users regarding *the quality of the products generated* as a result of applying Learn-CIAM, in terms of *ease of use* and *learning*, as well as *support for communication, coordination* and *awareness mechanisms*?

As can be seen, research questions RQ1–RQ5 refer to aspects related to the methodological proposal (Learn-CIAM), questions RQ6 and RQ7 focus on aspects related to the notation (Learn-CIAN) and the conceptual framework and, finally, questions RQ8 and RQ9 address aspects related to the use of Learn-CIAT and the quality of the products generated as a result of applying the method. The relation between these research questions and the items of the questionnaire are summarized in both tables of the Appendix A.

5.1.2. Participants

In the first experiment, several experts from the School of Informatics and the Faculty of Education of Ciudad Real (University of Castilla-La Mancha, Spain) participated. Specifically, a total of 13 experts, 11 of whom had a more technical profile (Computer Science, CS), and 2 a less technical profile (Education, ED). In the second experiment, a total of 15 experts participated, 11 of whom were from CS and 4 from ED.

As mentioned above, the main difference between the two experiences is that, while in the first one the subjects did not make use of the application and did not directly apply the method, in the second one they tackled a modeling and generation problem from scratch using Learn-CIAT. The participants in the two evaluations were different.

5.1.3. Method

We performed this experiment following the experimental process of [51]. We formulated the experimentation goal, using the Goal–Question–Metric (GQM) template, as follows:

To ***analyze*** the Learn-CIAM proposal and its main components (methodological, specification support and technological)
with the purpose of knowing the subjective perception
with regard to the suitability, completeness, usefulness and ease of use
from the point of view **of** Computer Science (CS) and Education (ED) experts
within the context of the application a model-driven approach to the generation of collaborative learning tools.

Figure 13 presents the experimental design followed in both experiments. The sample obtained is small in size as it is a preliminary evaluation. We are planning to carry out

a more complete evaluation in the future by recruiting more experts. The experience consisted of three phases:

- Introductory presentation: first, there was a presentation in which the methodological framework was introduced as well as the concepts and technologies involved.
- Demonstration/Execution: in the first experiment, the demonstration consisted of modeling and validating a learning course using Learn-CIAT (the same of Figure 9), to end with the generation of a collaborative programming system (the same of Figure 11) to support one of the activities of the course. Moreover, some pedagogical usability and awareness criteria were included in order to show the possibilities provided by the method and the capacities of the Learn-CIAT graphical editor. At the same time, during the whole process, there was an exchange of opinions between the speaker and the experts to find possible defects and improvements. In the second experiment, the group of experts had to use the Learn-CIAT tool to model, validate and generate the same scenario as in the demonstration of the first experiment. This is a more real case study, in which the users gain greater insight so that the subjective information that they provide becomes more realistic and useful.
- Questionnaire to the experts: finally, the experts were provided with a questionnaire which enabled them to evaluate each of the components of Learn-CIAM (methodological, conceptual and technological frameworks). First, they had to fill in a section with which to delimit their profile, differentiating between professors and non-professors (researchers), as well as their expertise, differentiating between technical and non-technical profiles (Education/Pedagogy and Software Engineers). The latter, in turn, could be specialized in three more specific branches, without being exclusive: model-driven development experts, Eclipse development environment experts, or instructional design experts. The questions were distributed concerning the subjective perception of the usefulness, ease of use, usability, manageability, etc. of the three main frameworks (Appendix A, Tables A1 and A2): methodological (MET), conceptual (CON) and technological (TEC). The methodological framework assessment consisted of 6 questions, as did the conceptual framework, while the technological framework consisted of 8. Each of the questions was scored using a Likert scale of 1 out of 5 points, with 5 being the highest value, denoting "full agreement" with the corresponding statement. In turn, each of these sections contained a blank box where the expert could argue any comment freely.

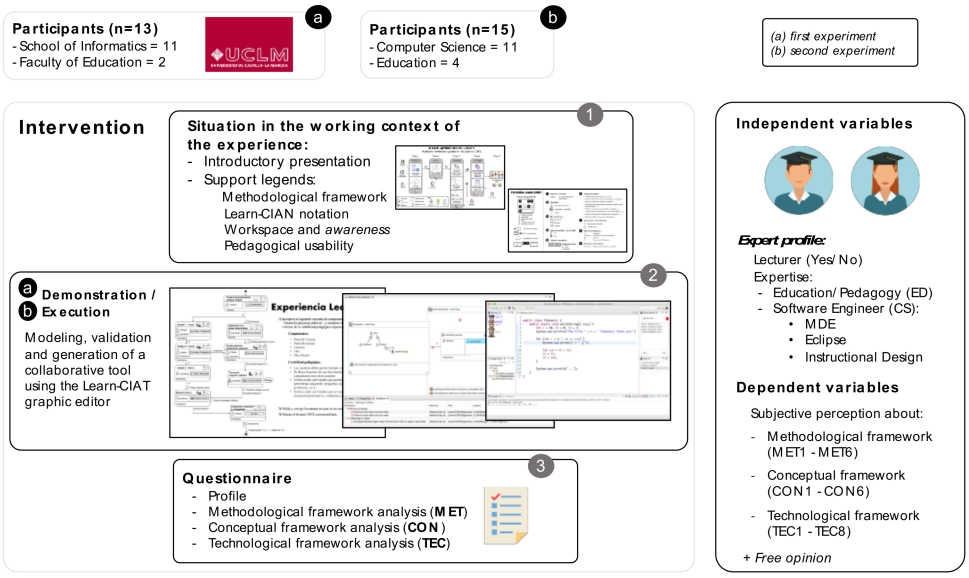

Figure 13. Experimental design of the Learn-CIAT experience.

5.2. Results of the First Experiment

The results were generally satisfactory. We compared technical (CS) with non-technical (ED) profiles. Table A1 in the Appendix A summarizes the results of this experiment, as well as the descriptive statistics (mean, median and mode) of both profiles. Next, we proceed to highlight the results of the items of the questionnaire which are summarized in Figure 14.

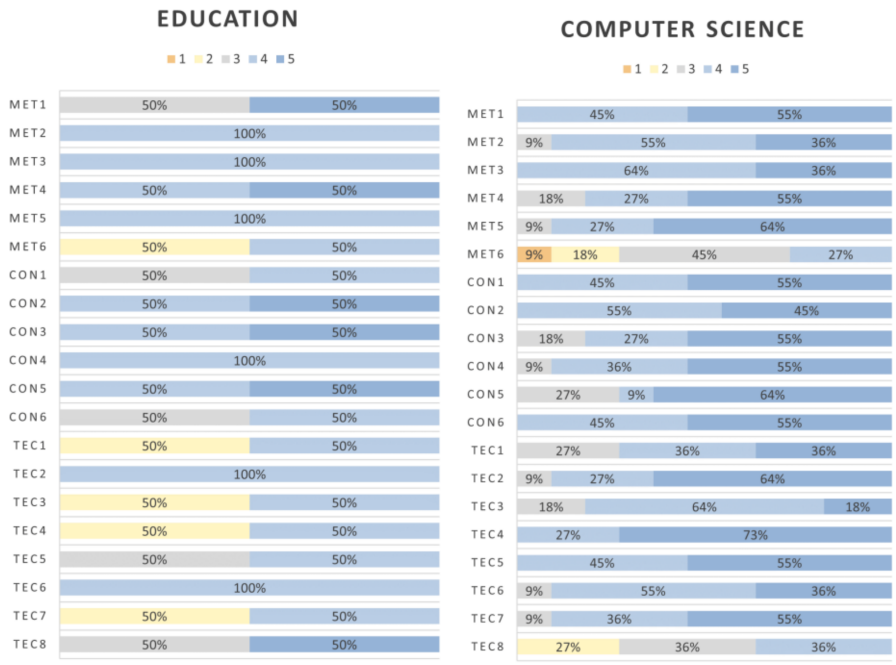

Figure 14. Distribution of results of the most relevant items from the first experiment.

In the first question (MET1), the participants were asked about the suitability and usefulness of proposing a new methodological framework to model, validate and generate collaborative tools. In response, approximately half of both ED and CS experts, 50% and 55% of the respondents, respectively, rated it as five, considering it very useful to propose a method such as Learn-CIAM (**RQ1**). This denotes a possible predisposition to its utility for the part of both profiles. On the other hand, addressing the second aspect of the methodological framework (MET2), the respondents were asked whether following a Model-Driven Development paradigm, as opposed to a traditional ad-hoc development paradigm, was easier for them to use. In this sense, 100% of the respondents with an ED profile fully agreed, while 91% of the respondents with a CS profile agreed, from which 36% fully agreed. These results are positive from both profiles, and it should be noted that the respondents with a less technical (ED) profile were in full agreement (**RQ2**). Regarding the third question (MET3), in which the participants were asked about the adequacy of the election of the Eclipse platform as support for the method, all the respondents with an ED profile agreed, whereas 64% of the respondents with a CS profile agreed, from which 36% fully agreed (**RQ3**). These results denote that both profiles consider as adequate the election of the Eclipse MDD approach as a support for the methodology, the technical experts being more critical about this choice. Furthermore, regarding the questions about the subjective perception of the proposed method, its division in different phases and stages, and the use of high-level languages, the answers were positive and none of the respondents from both profiles disagreed (**RQ4**). Lastly, in relation with the last question about the methodological framework (MET6), which asked whether they considered that a user with less advanced computer skills would be able to apply and generate collaborative tools, 50% of the ED respondents agreed, while the other 50% disagreed, scoring a two. On this same item, among the respondents with a CS profile, there was a bigger variation

in the results, with only 27% agreeing, 18% disagreeing and 9% completely disagreeing. These results indicate a greater disparity of opinion; however, they do not denote very negative results as half of the ED experts agreed. In addition, among the CS experts, a greater part agreed than the sum of those who did not agree with this statement (**RQ5**).

In relation to the conceptual framework, we asked about the Learn-CIAN notation (CON2), the conceptual core of the Learn-CIAM methodology, it being worth noting how 100% of the experts of both profiles considered it to be adequate for modeling flows of learning activities, with 50% of the respondents with an ED profile considering it to be very appropriate as well as 45% of the respondents with a CS profile. Additionally, in relation to the visual strain of the notation that this could cause (CON3), again the respondents gave good scores, since more than half of the experts in both groups fully agreed, with 50% (ED) and 55% (CS) respectively, pointing out that the Learn-CIAN notation has a sufficient and manageable number of visual elements (**RQ6**). Regarding the questions about the subjective perception of the potential users with respect to the suitability and completeness of the modelling elements related to awareness and pedagogical usability supported by the proposal, the answers were positive (**RQ7**), with the majority of the participants considering the number of awareness and pedagogical usability components as adequate and sufficient.

In relation to the technological framework, regarding the difficulty of applying the modeling, validation and generation stages of collaborative learning tools supported by Learn-CIAT (**RQ8**), a greater disparity in the results can be seen, denoting certain doubts in this regard. Doubts arise specially on the self-explanatory nature of the Learn-CIAT user interface (TEC8), CS experts being most critical about it. However, it is noteworthy that 50% of the ED experts were in full agreement with this statement. Furthermore, with regard to the quality of the products generated as a result of applying Learn-CIAM (**RQ9**), the results were positive in general. None of the CS experts complained about its quality, and only half of the ED experts disagreed on aspects about the user interface (TEC4) and the mechanisms to regulate the shared workspaces (TEC7). This may suggest, again, that experts with a more technical profile are more familiar with these kinds of tools.

In view of the results highlighted above, it is noteworthy how both profiles show interest in the methodological approach (MET1) and consider the Learn-CIAN notation adequate (CON2), the latter being the most important conceptual part of Learn-CIAM. At the same time, we observe a greater variation of the results of the capacity of the users with a less technical profile to use this proposal, denoting their doubts, but without presenting bad results (variables MET6 and TEC8) since most of them scored a three or higher.

Another important aspect of this experience to consider, beyond the previous results, was the comments, complaints, recommendations, and improvements proposed by the experts. With respect to the methodological framework, two variables were referred to. Hence, in view of question MET3, two users agreed that limiting the solution to the Eclipse environment introduces a strong dependence, although they see it as reasonable. While it is true that the first phase of the methodological framework (the generation of graphical editors) is limited to the Eclipse platform by Eugenia technology, the second phase would be able to generate collaborative tools on other platforms besides Eclipse. Therefore, this limitation occurs only in the first phase of the proposed method. On the other hand, according to variable MET6, up to six people said that, despite agreeing with the statement, it might be necessary to support these types of users in some of the stages.

In the section related to the conceptual framework, there were, again, two questions on which the experts made comment. In response to question CON4, one expert recommended adding a new widget that showed the statement of the task. With respect to question CON6, another expert indicated that some pedagogical usability criteria could be omitted or simplified. The latter has already been discussed in this paper: not all the criteria selected initially could be technologically supported. Nevertheless, despite being criteria that may not serve to guide users through the construction of collaborative tools, they are useful for the design of learning courses (imagine that a professor does not need to

generate a collaborative tool but does need to design the flow of activities to be carried out in the course).

The technological framework received the most comments from the experts. To illustrate, in response to question TEC2, one of the participants mentioned that it would be desirable that the validation process be fully automatic after a change has been made to the diagram. Currently, the chosen technology requires this small process to be carried out manually. Even so, the users only have to press a button each time they want to check the status of the diagram. On the other hand, question TEC6, was one of the most discussed. Two experts mentioned the possibility of including an audio/video chat. Another expert indicated, as an improvement to the awareness features of the chat, that colors should be included in the phrases of the users (the same one that they have in the session panel). Another expert indicated that, nowadays, these types of structured chats are somewhat outdated with respect to communication tools such as WhatsApp, Telegram, etc. Regarding these, although it is true that the chat which is already implemented does not have the power of such tools, it is in the same workspace, which prevents a clear loss of context. Question TEC7 was also mentioned; according to two of the experts, it would be advisable to incorporate simultaneous editing functionality instead of controlling the flow through a turn-taking panel. Thanks to Learn-CIAM, this would not be a problem as these systems or functionality could also be generated automatically if desired. Finally, considering question TEC8, the majority of the participants indicated that it is sufficiently self-explanatory, however five of them agreed about recommending an introductory workshop so that the users (especially those with a non-technical profile) can use the tool correctly and best take advantage of it. Furthermore, as a recommendation, one of the experts suggested that the legends used in this experience could also be included in the tool as manuals.

To conclude, we would like to emphasize that one of the experts made a generic comment on the proposal to indicate that they considered the generation of learning tools to be very interesting and necessary, beyond the current paradigm in which the only teaching aid is based on LMS content managers, affirming that the use of system generators that do something different from content management is very important.

5.3. Results of the Second Experiment

In this experiment, we used the same questionnaire as in the first one, observing now the following results (Appendix A, Table A2). Again, we proceed to highlight the results of the items of the questionnaire, summarized in Figure 15.

On this occasion, regarding the usefulness of proposing a new methodological framework to model, validate and generate collaborative tools (MET1), 75% of the ED and 100% of the CS experts considered it to be useful; 64% of the latter considered it to be really useful (**RQ1**). On the second question (MET2), in relation to following a Model-Driven Development paradigm, half of the respondents with an ED profile fully agreed, compared to 82% of the respondents with a CS profile who fully agreed (**RQ2**). When asked about the adequacy of the election of the Eclipse platform as support for the method (MET3), 50% of the respondents with an ED profile agreed, while 63% of the respondents with a CS profile agreed, from which 45% fully agreed, considering the election of the Eclipse platform solutions as support for the methodology adequate (**RQ3**). Furthermore, in relation with the questions about the subjective perception of Learn-CIAM, its division in different phases and stages, and the use of high-level languages, the answers were positive (**RQ4**) and only 18% of the CS experts disagreed on the division of the method in different phases and stages (MET4). Lastly, regarding the last question related to the methodological framework (MET6), in which it was asked whether they considered that a user with less advanced technical skills would be able to follow the steps of Learn-CIAM to generate collaborative tools, 75% of the ED experts would not know if a user without sufficient computer knowledge would be able to follow the steps and use the tool correctly, when scoring it with a three. On this same item, among the CS experts, there was more variation in the results, with 45% of them saying that they would not be able to generate collaborative

learning tools, compared to 36% who agreed. These results denote doubts about the idea that a user with less technological experience can generate collaborative tools without help. However, we should highlight the marks of the ED experts as no one rated them below three (**RQ5**).

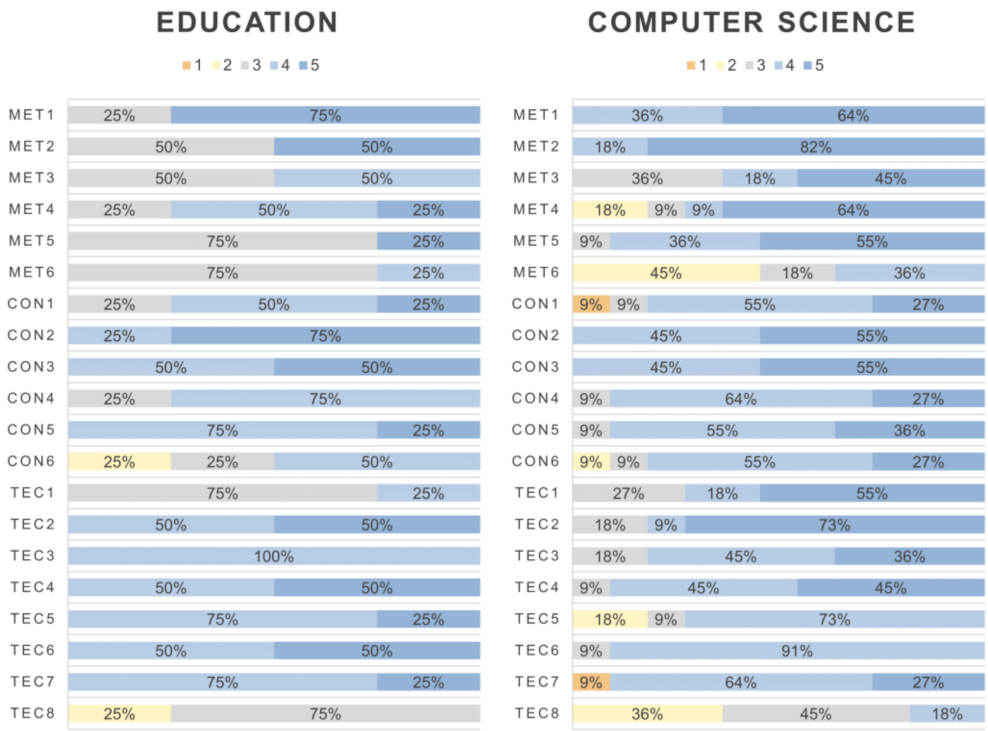

Figure 15. Distribution of results of the most relevant items from the second experiment.

According to the conceptual framework, about the Learn-CIAN notation (CON2), 100% of both ED and CS experts indicated that they considered it adequate or very adequate. While on the visual strain of this notation (CON3), again, 100% of the participants in both profiles gave it a score of four or five, which indicates that they agreed or strongly agreed with the number of elements chosen (**RQ6**). These results are very positive for such an important component as the Learn-CIAN notation within the Learn-CIAM methodological proposal. In addition, regarding the questions about the subjective perception of the potential users with respect to the suitability and completeness of the modeling elements related to awareness and pedagogical usability supported by the proposal, the answers were positive (**RQ7**) and only 25% of the ED experts and 9% of the CS experts disagreed on the adequacy of the pedagogical usability criteria included (CON6).

With regard to the technological framework, with respect to the difficulty of applying the modeling, validation and generation stages of collaborative learning tools supported by Learn-CIAT (**RQ8**), again, as in the first experiment, a greater disparity in the results can be seen. On this occasion, on the modeling stage using Learn-CIAT (TEC1) ED experts show doubts and 75% did not know what to answer (scored three). In addition, the self-explanatory nature of the Learn-CIAT tool (TEC8) was criticized by both profiles, as 75% of the ED experts did not know what to answer, as well as 45% of the CS experts. Furthermore, regarding the quality of the products generated as a result of applying Learn-CIAM (**RQ9**), the results were positive, as none of the CS experts complained about its quality, and only half the ED experts disagreed on aspects about the mechanisms offered to regulate the shared workspaces (TEC7).

Thus, it is noteworthy how both profiles, once again, after using the tool, show interest in Learn-CIAM (MET1), and consider the Learn-CIAN notation (CON2) to be adequate. In turn, as in the first experiment, certain doubts can be observed regarding the capacity of

the users with a less technical profile to apply Learn-CIAM without help. However, we do not consider these as alarming results, since most of them scored three, in some cases agreed, and only 25% of the users with a less technical profile (ED) did not agree (variables MET6 and TEC8).

As in the first experiment, the experts noted a few comments and proposals for improvement that we could pick up on for future work. Thus, according to the methodological framework, two variables were discussed: MET3 and MET6. With respect to MET3, two respondents argued that it could be a good idea to use more current technologies than those chosen on the Eclipse platform, such as web and/or cloud technologies. On the other hand, regarding MET6, three experts discussed it, indicating that a user without advanced computer knowledge would need at least an introductory course or tutorial. These comments are quite similar to those made by the experts in experiment 1.

According to the conceptual framework, there were three variables contrasted by the experts. About CON1, one of the participants said that the chosen pedagogical usability criteria were very subjective. Related to this comment, it is important to remember that the main objective of Learn-CIAM is to help and guide professors through the design of their learning courses and, if they consider it necessary, the generation of support tools for some of the group activities. Thus, the pedagogical usability criteria are considered as a "checking" tool for both learning courses and collaborative learning tools, without being mandatory for the final generation of the systems. Thus, they are intended to guide the professor through the definition of the learning objectives, hence their subjectivity. Another of the variables questioned was CON4, about which an expert indicated that the structured chat would be useless those times. On the other hand, also in this framework, the most debated question was variable CON6, about which four of the experts agreed that they saw too many criteria of pedagogical usability and did not know very well if it could be validated and reflected in the tools.

Finally, the technological framework, as in the first experiment, was again the most discussed. Thus, four variables were commented on by the experts. With respect to question TEC5, one expert also suggested adding the users' colors to the chat in order to improve awareness of it. Regarding variable TEC6, an expert indicated that the chat looked a bit old (we imagine that he referred to the structured chat), which is quite similar to the comment of another expert about question CON4. As for variable TEC7, one expert stated that he would like to work with a real-time system. We are not very sure what the expert meant, as the collaborative learning tools generated on Learn-CIAM are synchronous. Finally, there were four participants who criticized variable TEC8, saying that the users with less technical profiles need some kind of prior help, or some prior workshop. In addition, two of them indicated that they found the user interface to be very good and they considered it as a good approach but said that the users with less technical profiles need some type of prior help in the form of either a workshop or an introductory course.

5.4. Reflection

The results discussed in the previous subsections were generally positive. According to the scores, both profiles indicate agreement with the proposal, the group of experts with a more technical profile being the one that granted higher scores. This may be mainly due to the fact that this group of users is more accustomed to carrying out this type of work. However, the group of educators was also predisposed to work with the tool, as long as a previous informative workshop was given. One aspect that we could appreciate in both experiments is that the ED experts answered with more threes than the CS experts, which may mean that, lacking advanced computer knowledge, they do not really know whether it is easy to use, useful, etc. In turn, the CS experts show a wider variation in the results. The latter may be since these experts have more knowledge about this field than the ED experts and are, therefore, more critical. In addition, the CS experts know more tools and methods, which allows them to compare the proposal presented with others that they already know.

On the other hand, this experience has also made it possible to compile a series of recommendations and/or improvements, mainly related to the technological framework, to be considered as future work. Overall, most of the experts criticized the same aspects in both experiments. Thus, for instance, some participants indicated that it would be convenient to add a new widget that shows the statement of the activity to be carried out by the students, further automate the process of validating the diagrams, or even incorporate audio/video chats, among others. Nevertheless, since the main objective of this work is to propose a new systematic and formalized method that facilitates the semi-automatic generation of the collaborative learning tools, and not 100% functional collaborative tools, initially these recommendations, some of which are very interesting, will be considered in future work. For example, by incorporating new learning tools as output of the Learn-CIAM process, or by improving the tools already implemented including the recommendations of some of the experts.

5.5. Study Limitations

There are some limitations and threats to the experimental validity of the evaluation carried out.

One issue which could affect the *conclusion validity* of this study is the *sample size* in both experiments. We are aware that more experimentation is needed in the future, so that the results can be more generalizable. On the other hand, the proportion of education experts in both experiments is lower than that of computer science experts. Future work is planned to carry out evaluations involving the participation of a substantially larger number of experts from both profiles. In relation to *the reliability of the measure*, the use of subjective perception questionnaires means that the results obtained may be biased. In any case, the fact that the participants were of adults involved in improving their research and teaching activities, makes us think about their involvement, impartiality and seriousness when giving their answers.

In terms of *external validity*, the material and tasks used in both experiments were designed with time constraints in mind. Regarding the *experimental task*, only one task was posed, and of a manageable complexity. In order to have a more complete perception of the proposal, it would be advisable for the subjects to carry out different specification and generation activities, with different levels of difficulty.

As far as *internal validity* is concerned, aspects that could have threatened it were addressed and minimized as far as possible. The activities were carried out with subjects who had never done a similar experiment before, and the subjects who participated in each of the experiments were different, thus avoiding *persistence* and *learning effects.* The average duration of the experiments was 2 h in the first one and 3 h in the second (including documentation learning and oral explanations), and no *fatigue effects* appeared. In order to assess this aspect, participants were asked about it at the end of the activity. As noted above, the participants were highly committed to this experiment, being mostly researchers and teachers, so the results could potentially be of benefit to them. Therefore, the *subject motivation* was high.

6. Conclusions

The accomplishment of this work has allowed us to define Learn-CIAM, a new methodological proposal as support to the modeling of flows of learning activities and to the generation of collaborative learning tools as support. In particular, it brings the possibility to generate collaborative tools from different domains (i.e., graphical editors, textual editors-plain text or code-, or Web browsers), using a set of technologies and high-level languages framed within the Eclipse solutions. As the main novelty with respect to other existing methods and previous works, it takes into account some components related to learning environments and synchronous collaborative systems, such as pedagogical usability and the awareness support. Additionally, it applies a model-driven development process for tool generation. The use of conceptual frameworks as input artifacts favors, among other

things, their reuse in the resolution of other problems, regardless of the specific scope of work for which they were originally intended. Hence, for instance, it is considered that other tools could be generated more easily and in a more direct manner. To do this, it would be enough to model this tool and incorporate it into the conceptual framework of Learn-CIAM that has been presented.

On the other hand, in order to guarantee the validity of the Learn-CIAM methodology, it has undergone several validation processes with users and designers from different profiles such as Education and Computer Science. In the context of this work, a new validation has been carried out with the aim of evaluating the subjective perception and the degree of acceptance of the proposal. This validation consisted of two different experiments. First, we carried out a demonstration of application of the method. Then, the participants used Learn-CIAT to model a flow of learning activities and to generate one collaborative learning tool. The objective was to obtain a first impression from the potential end users. The results indicate a positive acceptance, concluding that its application would facilitate the generation of collaborative learning tools according to both profiles. Nonetheless, future work has also been drawn from the comments and complaints of the experts, as well as from the work done. We consider it necessary to increase the number of learning tools provided by Learn-CIAM. Currently, for instance, only Java code and plain text collaborative editors can be generated. In the future, it will be interesting to add support for more programming languages (preferably the most leading-edge) and to increase the capabilities of the textual editor. In addition, if we attend to several of the users' complaints after the last validation, it would be interesting to add a new widget in which it would be possible to show the statement of the learning activity to be carried out by the students, or even to incorporate audio/video chats, among others. Finally, we also consider it necessary to continue performing validations on the proposal with a larger sample of potential users.

Author Contributions: Conceptualization, Y.A., A.I.M., M.A.R. and J.G.; methodology, Y.A., A.I.M., M.A.R. and J.G.; software, Y.A. and J.G., validation, Y.A. and A.I.M.; formal analysis: Y.A.; investigation, Y.A.; writing-original draft preparation, Y.A.; writing-review and editing, Y.A., A.I.M., M.A.R. and J.G.; supervision, A.I.M., M.A.R. and J.G.; funding acquisition, M.A.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been funded by the project TIN2015-66731-C2-2-R of the Ministry of Science, Innovation and Universities.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this research are available on request from the corresponding author.

Acknowledgments: This work has been partially funded by the project TIN2015-66731-C2-2-R of the Ministry of Science, Innovation and Universities. Moreover, the authors of this paper would like to thank the experts who participated in the Learn-CIAT experience for their collaboration.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Means, median, mode of experiment 1 and comparison between Education (ED) and Computer Science (CS) profiles.

ID (RQ)	Statement	Mean	Median	Mode	Likert									
					Education (ED)					Computer Science (CS)				
					1	2	3	4	5	1	2	3	4	5
MET1 (**RQ1**)	Do you consider the idea of proposing a method to support the modeling, validation and generation of collaborative learning tools to be useful?	4.46	5.00	5	-	-	50%	-	50%	-	-	-	45%	55%
MET2 (**RQ2**)	Do you think that following a Model-Driven Development paradigm makes the method more understandable and easier to use than a traditional ad-hoc programming development method?	4.23	4.00	4	-	-	-	100%	-	-	-	9%	55%	36%
MET3 (**RQ3**)	Do you consider the choice of the Eclipse Model-Driven Development approach as a support for the method to be appropriate?	4.31	4.00	4	-	-	-	100%	-	-	-	-	64%	36%
MET4 (**RQ4**)	Do you consider adequate the division of the methodological proposal into different phases and stages?	4.38	5.00	5	-	-	-	50%	50%	-	-	18%	27%	55%
MET5 (**RQ4**)	Do you think that the combination of technologies and high-level languages presented as support for the method is adequate?	4.46	5.00	5	-	-	-	100%	-	-	-	9%	27%	64%
MET6 (**RQ5**)	Do you think that a user without advanced computer knowledge could be able to generate collaborative learning tools by following these steps?	2.92	3.00	3	-	50%	-	50%	-	9%	18%	45%	27%	-
CON1 (**RQ7**)	Do you think that the aspects considered in the meta-models (domain, workspace, awareness and pedagogical usability) are adequate?	4.38	4.00	4	-	-	50%	50%	-	-	-	-	45%	55%
CON2 (**RQ6**)	Do you consider the Learn-CIAN notation adequate for modeling flows of learning activities?	4.46	4.00	4	-	-	-	50%	50%	-	-	-	55%	45%
CON3 (**RQ6**)	Do you think that the number of visual elements in the Learn-CIAN notation is sufficient?	4.38	5.00	5	-	-	-	50%	50%	-	-	18%	27%	55%

Table A1. *Cont.*

ID (RQ)	Statement	Mean	Median	Mode	Likert									
					Education (ED)					Computer Science (CS)				
					1	2	3	4	5	1	2	3	4	5
CON4 (**RQ7**)	Do you consider the set of components/widgets and awareness elements that were chosen for the specification of the workspaces to be adequate?	4.38	4.00	4	-	-	-	100%	-	-	-	9%	36%	55%
CON5 (**RQ7**)	Do you think that the number of components/widgets and awareness elements that were chosen for the specification of workspaces is sufficient?	4.38	5.00	5	-	-	-	50%	50%	-	-	27%	9%	64%
CON6 (**RQ7**)	Do you think that the set of criteria selected for specifying aspects related to pedagogical usability is adequate?	4.38	4.00	4	-	-	50%	50%	-	-	-	-	45%	55%
TEC1 (**RQ8**)	Do you find the modeling process provided by the Learn-CIAT support tool simple?	3.92	4.00	4	-	50%	-	50%	-	-	-	27%	36%	36%
TEC2 (**RQ8**)	Do you find the validation process provided by the Learn-CIAT support tool simple?	4.46	5.00	5	-	-	-	100%	-	-	-	9%	27%	64%
TEC3 (**RQ8**)	Do you find the generation process provided by the Learn-CIAT support tool simple?	3.85	4.00	4	-	-	50%	50%	-	-	-	18%	64%	18%
TEC4 (**RQ9**)	Do you find the user interface of the generated tools easy to use and learn?	4.46	5.00	5	-	-	50%	50%	-	-	-	-	27%	63%
TEC5 (**RQ9**)	Do you think that the collaborative tools generated offer adequate mechanisms to perceive the rest of the users who are working in the same session?	4.38	4.00	4	-	-	-	50%	50%	-	-	-	45%	55%
TEC6 (**RQ9**)	Do you consider the mechanisms offered by the generated collaborative tools to establish textual communication between the users to be adequate?	4.23	4.00	4	-	-	-	-	100%	-	-	9%	55%	36%
TEC7 (**RQ9**)	Do you consider the mechanisms that they offer to regulate the use of the shared workspace between the users of the same session to be adequate?	4.23	4.00	5	-	50%	-	50%	-	-	-	9%	36%	55%
TEC8 (**RQ8**)	To carry out this activity, a description of the main components was provided. Without this help, do you consider the user interface of the graphical editor sufficiently self-explanatory so that the user does not have difficulty in using it?	3.23	3.00	3	-	-	50%	-	50%	-	27%	36%	36%	-

Table A2. Means, median, mode of experiment 2 and comparison between ED and CS profiles.

ID (RQ)	Statement	Mean	Median	Mode	Likert									
					Education (ED)					Computer Science (CS)				
					1	2	3	4	5	1	2	3	4	5
MET1 (**RQ1**)	Do you consider the idea of proposing a method to support the modeling, validation and generation of collaborative learning tools to be useful?	4.60	5.00	5	-	-	25%	-	75%	-	-	-	36%	64%
MET2 (**RQ2**)	Do you think that following a Model-Driven Development paradigm makes the method more understandable and easier to use than a traditional ad-hoc programming development method?	4.60	5.00	5	-	-	50%	-	50%	-	-	-	18%	82%
MET3 (**RQ3**)	Do you consider the choice of the Eclipse Model-Driven Development approach as a support for the method to be appropriate?	3.93	4.00	3	-	-	50%	50%	-	-	-	36%	18%	45%
MET4 (**RQ4**)	Do you consider adequate the division of the methodological proposal into different phases and stages?	4.13	5.00	5	-	-	25%	50%	25%	-	18%	9%	9%	64%
MET5 (**RQ4**)	Do you think that the combination of technologies and high-level languages presented as support for the method is adequate?	4.20	4.00	5	-	-	75%	-	25%	-	-	9%	36%	55%
MET6 (**RQ5**)	Do you think that a user without advanced computer knowledge could be able to generate collaborative learning tools by following these steps?	3.00	3.00	2	-	-	75%	25%	-	-	45%	18%	36%	-
CON1 (**RQ7**)	Do you think that the aspects considered in the meta-models (domain, workspace, awareness and pedagogical usability) are adequate?	3.93	4.00	4	-	-	25%	50%	25%	9%	-	9%	55%	27%
CON2 (**RQ6**)	Do you consider the Learn-CIAN notation adequate for modeling flows of learning activities?	4.60	5.00	5	-	-	-	25%	75%	-	-	-	45%	55%
CON3 (**RQ6**)	Do you think that the number of visual elements in the Learn-CIAN notation is sufficient?	4.53	5.00	5	-	-	-	50%	50%	-	-	-	45%	55%
CON4 (**RQ7**)	Do you consider the set of components/widgets and awareness elements that were chosen for the specification of the workspaces to be adequate?	4.07	4.00	4	-	-	25%	75%	-	-	-	9%	64%	27%

Table A2. *Cont.*

ID (RQ)	Statement	Mean	Median	Mode	Likert									
					Education (ED)					Computer Science (CS)				
					1	2	3	4	5	1	2	3	4	5
CON5 (**RQ7**)	Do you think that the number of components/widgets and awareness elements that were chosen for the specification of workspaces is sufficient?	4.27	4.00	4	-	-	-	75%	25%	-	-	9%	55%	36%
CON6 (**RQ7**)	Do you think that the set of criteria selected for specifying aspects related to pedagogical usability is adequate?	3.80	4.00	4	-	25%	25%	50%	-	-	9%	9%	55%	27%
TEC1 (**RQ8**)	Do you find the modeling process provided by the Learn-CIAT support tool simple?	4.00	4.00	3	-	-	75%	25%	-	-	-	27%	18%	55%
TEC2 (**RQ8**)	Do you find the validation process provided by the Learn-CIAT support tool simple?	4.53	5.00	5	-	-	50%	50%	-	-	-	18%	9%	73%
TEC3 (**RQ8**)	Do you find the generation process provided by the Learn-CIAT support tool simple?	4.13	4.00	4	-	-	-	100%	-	-	-	18%	45%	36%
TEC4 (**RQ9**)	Do you find the user interface of the generated tools easy to use and learn?	4.40	4.00	4	-	-	-	50%	50%	-	-	9%	45%	45%
TEC5 (**RQ9**)	Do you think that the collaborative tools generated offer adequate mechanisms to perceive the rest of the users who are working in the same session?	3.73	4.00	4	-	-	-	75%	25%	-	18%	9%	73%	-
TEC6 (**RQ9**)	Do you consider the mechanisms offered by the generated collaborative tools to establish textual communication between the users to be adequate?	4.07	4.00	4	-	-	-	50%	50%	-	-	-	9%	91%
TEC7 (**RQ9**)	Do you consider the mechanisms that they offer to regulate the use of the shared workspace between the users of the same session to be adequate?	4.07	4.00	4	-	-	-	75%	25%	9%	-	-	64%	27%
TEC8 (**RQ8**)	To carry out this activity, a description of the main components was provided. Without this help, do you consider the user interface of the graphical editor sufficiently self-explanatory so that the user does not have difficulty in using it?	2.80	3.00	3	-	25%	75%	-	-	-	36%	45%	18%	-

References

1. Schmidt, D.C. Model-Driven Engineering. 2006. Available online: http://www.computer.org/portal/site/computer/menuitem.e533b16739f5 (accessed on 28 January 2021).
2. Bucchiarone, A.; Cabot, J.; Paige, R.F.; Pierantonio, A. Grand challenges in model-driven engineering: An analysis of the state of the research. *Softw. Syst. Model.* **2020**, *19*, 5–13. [CrossRef]
3. 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. [CrossRef]
4. Ellis, C.A.; Gibbs, S.J.; Rein, G. Groupware: Some issues and experiences. *ACM Commun.* **1991**, *34*, 39–58. [CrossRef]
5. González, V.M.; Mark, G. Managing currents of work: Multi-tasking among multiple collaborations. In Proceedings of the 9th European Conference on Computer-Supported Cooperative Work, Paris, France, 18–22 September 2005; pp. 143–162. [CrossRef]
6. Sire, S.; Chatty, S.; Gaspard-Boulinc, H.; Colin, F.R. How Can Groupware Preserve our Coordination Skills? Designing for Direct Collaboration. In *INTERACT '99*; IOS Press: Edinburgh, UK, 1999; pp. 304–312. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.74.6271&rep=rep1&type=pdf (accessed on 11 March 2021).
7. Agredo-Delgado, V.; Ruiz, P.H.; Mon, A.; Collazos, C.A.; Fardoun, H.M. *Towards a Process Definition for the Shared Understanding Construction in Computer-Supported Collaborative Work*; Springer: Cham, Germany, 2020; pp. 263–274.
8. Dourish, P.; Bellotti, V. Awareness and coordination in shared workspaces. In Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work—CSCW '92, Toronto, ON, Canada, 31 October–4 November 1992; pp. 107–114. [CrossRef]
9. Teruel, M.A.; Navarro, E.; López-Jaquero, V.; Montero, F.; González, P. A comprehensive framework for modeling requirements of CSCW systems. *J. Softw.: Evol. Process* **2017**, *29*, e1858. [CrossRef]
10. Antunes, P.; Herskovic, V.; Ochoa, S.F.; Pino, J.A. Reviewing the quality of awareness support in collaborative applications. *J. Syst. Softw.* **2014**, *89*, 146–169. [CrossRef]
11. Silva, L.; Mendes, A.J.; Gomes, A. Computer-supported Collaborative Learning in Programming Education: A Systematic Literature Review. In Proceedings of the 2020 IEEE Global Engineering Education Conference (EDUCON), Porto, Portugal, 27–30 April 2020; pp. 1086–1095. [CrossRef]
12. Kabza, E.M.; Berre, A.J.; Murad, H.; Mansuroglu, D. Evaluating pedagogical practices supporting collaborative learning for model-based system development courses. In Proceedings of the Norsk IKT-Konferanse for Forskning og Utdanning, Digital Conference, 24–25 November 2020; no. 4. Available online: http://www.nik.no/ (accessed on 27 January 2021).
13. Stahl, G. A decade of CSCL. *Int. J. Comput. Collab. Learn.* **2015**, *10*, 337–344. [CrossRef]
14. Chong, F. The Pedagogy of Usability: An Analysis of Technical Communication Textbooks, Anthologies, and Course Syllabi and Descriptions. *Tech. Commun. Q.* **2015**, *25*, 12–28. [CrossRef]
15. Sales Júnior, F.M.; Ramos, M.A.; Pinho, A.L.; Santa Rosa, J.G. Pedagogical Usability: A Theoretical Essay for E-Learning. *Holos* **2016**, *32*, 3–15. [CrossRef]
16. Nokelainen, P. An empirical assessment of pedagogical usability criteria for digital learning material with elementary school students. *Educ. Technol. Soc.* **2006**, *9*, 178–197. Available online: https://www.j-ets.net/ETS/journals/9_2/15.pdf (accessed on 14 February 2019).
17. Nielsen, J. *Evaluating Hypertext Usability*; Springer: Berlin/Heidelberg, Germany, 1990.
18. Ma, X.; Liu, J.; Liang, J.; Fan, C. An empirical study on the effect of group awareness in CSCL environments. *Interact. Learn. Environ.* **2020**, 1–16. [CrossRef]
19. Paternò, F. ConcurTaskTrees: An Engineered Notation for Task Models. In *The Handbook of Task Analysis for Human-Computer Interaction*; CRC Press: Boca Raton, FL, USA, 2003; pp. 483–503.
20. Van Der Veer, G.C.; Lenting, B.F.; Bergevoet, B.A. GTA: Groupware task analysis—Modeling complexity. *Acta Psychol.* **1996**, *91*, 297–322. [CrossRef]
21. Pinelle, D. *Improving Groupware Design for Loosely Coupled Groups*; University of Saskatchewan: Saskatoon, SK, Canada, 2004.
22. Lee, C.P.; Paine, D. From the Matrix to a Model of Coordinated Action (MoCA). In Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing—CSCW '15, Vancouver, BC, Canada, 14–15 March 2015; pp. 179–194. [CrossRef]
23. Neumayr, T.; Jetter, H.-C.; Augstein, M.; Friedl, J.; Luger, T. Domino. *Proc. Acm Hum.-Comput. Interact.* **2018**, *2*, 1–24. [CrossRef]
24. Luque, L. *Collab4All: A Method to Foster Inclusion in Computer-Supported Collaborative Work*; Universidade de Sao Paulo: Sao Paulo, Brazil, 2019.
25. Miranda, F. G4CVE: A framework based on CSCW for the development of collaborative virtual environments. In Proceedings of the 7th Workshop on Aspects of Human-Computer Interaction for the Social Web, Sao Paulo, Brazil, 9–14 July 2017; pp. 10–17. Available online: https://sol.sbc.org.br/index.php/waihcws/article/view/3869 (accessed on 20 January 2021).
26. Blekhman, A.; Dori, D. 3.5.2 Tesperanto—A Model-Based System Specification Methodology and Language. *Incose Int. Symp.* **2013**, *23*, 139–153. [CrossRef]
27. Blekhman, A.; Wachs, J.P.; Dori, D. Model-Based System Specification with Tesperanto: Readable Text from Formal Graphics. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *45*, 1448–1458. [CrossRef]

28. KNiemantsverdriet, K.; Van Essen, H.; Pakanen, M.; Eggen, B. Designing for Awareness in Interactions with Shared Systems. *ACM Trans. Comput. Interact.* **2019**, *26*, 1–41. [CrossRef]
29. Penichet, V.M.R.; Lozano, M.D.; Gallud, J.A.; Tesoriero, R.; Rodríguez, M.L.; Garrido, J.L.; Noguera, M.; Hurtado, M.V. Extending and supporting featured user interface models for the development of groupware applications. *J. Univers. Comput. Sci.* **2008**, *14*, 3053–3070. Available online: http://www.jucs.org/jucs_14_19/extending_and_supporting_featured (accessed on 19 January 2021).
30. Penichet, V.M.R.; Lozano, M.D.; Gallud, J.A.; Tesoriero, R. User interface analysis for groupware applications in the TOUCHE process model. *Adv. Eng. Softw.* **2009**, *40*, 1212–1222. [CrossRef]
31. Penichet, V.M.R.; Lozano, M.D.; Gallud, J.A.; Tesoriero, R. Requirement-based approach for groupware environments design. *J. Syst. Softw.* **2010**, *83*, 1478–1488. [CrossRef]
32. Yu, E.S.K. Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In Proceedings of the 3rd IEEE International Symposium on Requirements Engineering, Annapolis, MD, USA, 5–8 January 1997; pp. 226–235. Available online: http://www.cs.toronto.edu/pub/eric/RE97.pdf (accessed on 21 January 2021).
33. Teruel, M.A.; Navarro, E.; López-Jaquero, V.; Montero, F.; González, P. CSRML Tool: A Visual Studio Extension for Modeling CSCW Requirements. In Proceedings of the 6th International i* Workshop, Valencia, Spain, 17–18 June 2013; pp. 122–124. Available online: http://ceur-ws.org/Vol-978/paper_21.pdf (accessed on 19 January 2021).
34. Botturi, L.; Stubbs, S.T. *Handbook of Visual Languages for Instructional Design*; Information Science Reference: Hershey, PA, USA, 2011.
35. Ferraris, C.; Martel, C.; Vignollet, L. LDL for Collaborative Activities. In *Instructional Design*; IGI Global: Hershey, PA, USA, 2011; pp. 403–430.
36. Rapos, E.; Stephan, M. IML: Towards an Instructional Modeling Language. In Proceedings of the MODELSWARD 2019, Prague, Czech Republic, 20–22 February 2019; pp. 417–425. Available online: https://www.scitepress.org/Papers/2019/74852/74852.pdf (accessed on 19 January 2021).
37. Bakki, A.; Oubahssi, L.; George, S.; Cherkaoui, C. MOOCAT: A visual authoring tool in the cMOOC context. *Educ. Inf. Technol.* **2018**, *24*, 1185–1209. [CrossRef]
38. Bakki, A.; Oubahssi, L.; George, S.; Cherkaoui, C. A Model and Tool to Support Pedagogical Scenario Building for Connectivist MOOC. *Technol. Knowl. Learn.* **2020**, *25*, 899–927. [CrossRef]
39. Martini, R.G.; Henriques, P.R. Automatic generation of virtual learning spaces driven by CaVa DSL: An experience report. *ACM Sigplan Not.* **2017**, *52*, 233–245. [CrossRef]
40. Perez-Berenguer, D.; Garcia-Molina, J. INDIeAuthor: A Metamodel-Based Textual Language for Authoring Educational Courses. *IEEE Access* **2019**, *7*, 51396–51416. [CrossRef]
41. Caeiro, M.; Llamas, M.; Anido, L. PoEML: Modeling learning units through perspectives. *Comput. Stand. Interfaces* **2014**, *36*, 380–396. [CrossRef]
42. Ruiz, A.; Panach, J.I.; Pastor, O.; Giraldo, F.D.; Arciniegas, J.L.; Giraldo, W.J. Designing the Didactic Strategy Modeling Language (DSML) From PoN: An Activity Oriented EML Proposal. *IEEE Rev. Iberoam. De Tecnol. Del Aprendiz.* **2018**, *13*, 136–143. [CrossRef]
43. Moody, D.L. The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Trans. Softw. Eng.* **2009**, *35*, 756–779. [CrossRef]
44. Figl, K.; Derntl, M.; Rodríguez, M.C.; Botturi, L. Cognitive effectiveness of visual instructional design languages. *J. Vis. Lang. Comput.* **2010**, *21*, 359–373. [CrossRef]
45. 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. [CrossRef]
46. Conole, G. Describing learning activities: Tools and resources to guide practice. In *Rethinking Pedagogy for a Digital Age: Designing and Delivering Elearning*; Routledge: New York, NY, USA, April 2007; pp. 101–111. [CrossRef]
47. Marcelo, C.; Yot, C.; Mayor, C.; Sanchez Moreno, M.; Rodriguez Lopez, J.M.; Pardo, A. Learning activities in higher education: Towards autonomous learning of students? *Rev. Educ.* **2014**, *363*, 334–359. [CrossRef]
48. 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. [CrossRef]
49. Kolovos, D.S.; García-Domínguez, A.; Rose, L.M.; Paige, R.F.; Kolovos, D. Eugenia: Towards disciplined and automated development of GMF-based graphical model editors. *Softw. Syst. Model.* **2015**, *16*, 229–255. [CrossRef]
50. Kolovos, D.S.; Paige, R.F. The Epsilon Pattern Language. In Proceedings of the 2017 IEEE/ACM 9th International Workshop on Modelling in Software Engineering, MiSE 2017, Buenos Aires, Argentina, 21–22 May 2017; pp. 54–60. [CrossRef]
51. Wohlin, C.; Runeson, P.; Höst, M.; Ohlsson, M.C.; Regnell, B.; Wesslén, A. *Experimentation in Software Engineering*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 9783642290.