

Article

Educational Cyber–Physical Systems (ECPSs) for University 4.0

Laurent Gallon ^{1,*} , Khoulood Salameh ² , Richard Chbeir ³ , Samia Bachir ⁴ and Philippe Aniorté ⁵ 

¹ Université de Pau et des Pays de l'Adour, E2S, LIUPPA, 40000 Mont-de-Marsan, France

² Computer Science and Engineering Department, American University of Ras Al Khaimah, Ras Al Khaimah 72603, United Arab Emirates; khoulood.salameh@aurak.ac.ae

³ Université de Pau et des Pays de l'Adour, E2S, LIUPPA, 64600 Anglet, France; richard.chbeir@univ-pau.fr

⁴ Research and Innovation Department, Capgemini Engineering, 92190 Meudon, France; samia.bachir@univ-pau.fr

⁵ Indicatic-AIP, National Institute on ICT, Panamá City 0801, Panamá; indicatic_director@senacyt.gob.pa

* Correspondence: laurent.gallon@univ-pau.fr

Abstract: University 4.0 represents the adaptation of the Education 4.0 paradigm to the university context. The core principle is the automated supervision of the entire student learning process by an AI-driven computer assistant, allowing for timely adjustments based on the student's progression. Critical to this process is the assistant's ability to collect comprehensive information on all student activities within the curriculum, particularly overseeing pedagogical activities in real time to make necessary adaptations. This utilizes Educational Cyber–Physical Systems (ECPSs) to gather all relevant data and extract appropriate information effectively. This article examines a specific case of practical work involving students at two distinct geographical locations collaborating in a blended learning environment. A specialized ECPS is deployed to collect data from equipment at both sites, enabling the modeling of the pedagogical sequence, the cyber–physical system, and the data necessary for monitoring student progress in real time.

Keywords: cyber–physical systems; Education 4.0; hybrid teaching



Citation: Gallon, L.; Salameh, K.; Chbeir, R.; Bachir, S.; Aniorté, P. Educational Cyber–Physical Systems (ECPSs) for University 4.0. *Information* **2024**, *15*, 790. <https://doi.org/10.3390/info15120790>

Academic Editor: Lesheng Jin

Received: 26 September 2024

Revised: 19 November 2024

Accepted: 20 November 2024

Published: 9 December 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The COVID-19 crisis significantly disrupted traditional teaching practices. Over several months, France, like several countries, experienced three lockdowns. During the initial phase, teaching sequences transitioned from face-to-face to fully online modes, employing both synchronous and asynchronous methods, often in combination [1]. In subsequent lockdowns, the methodology evolved into blended formats—combining face-to-face and remote learning—to facilitate a gradual return to conventional classroom settings while adhering to attendance limits to curb virus transmission.

To illustrate this, let us consider our technological institute, where practical work is a pivotal component of the curriculum. During periods of blended learning, it was crucial to enable remote students to participate in practical sessions and interact with the equipment. As part of the network and telecommunications department, we leveraged technical solutions to facilitate remote connections and configure equipment. Nevertheless, a significant challenge was to effectively supervise the work of students who were not physically co-located.

This article demonstrates how our engagement with the University 4.0 initiative, which applies the Education 4.0 paradigm to the university context, has enabled us to facilitate and automate supervision. More broadly, we show how it is possible to collect information from an Educational Cyber–Physical System (ECPS), regardless of geographical constraints, to automate the supervision of teaching sequences. This ECPS is mandatory for gathering all the data needed to characterize the development of the teaching sequence and the participants at each geographical site. Utilizing these data, an intelligent assistant can

make informed decisions based on student's progress, adjusting the teaching sequence in real time.

The structure of this article is as follows. Section 2 provides a concise review of the current state of the art in modeling cyber-physical systems (CPSs) for educational purposes. Section 3 describes the architecture of the University 4.0 concept. Section 4 details the modeling principles used to depict both the hardware (cyber-physical aspects) and the educational (pedagogical) components of student activities. Section 5 offers a practical illustration of the concepts discussed in the previous sections. Finally, Section 6 concludes the article and outlines future perspectives.

2. State of the Art

To the best of our knowledge, ECPSs have not yet been extensively modeled or standardized in the literature, nor has the Internet of Everything (IoE) paradigm. Our proposal, therefore, draws on both the modeling of the Internet of Things (IoT) and cyber-physical systems (CPSs), as well as on the modeling of educational aspects. In the following subsections, we will explore some references and provide insights to better understand our modeling choices.

2.1. IoT Modeling

Extensive research has been conducted on IoT modeling. We will focus on those studies most relevant to our concerns. From an architectural standpoint, the IoT ARM [2] serves as a reference model widely adopted in numerous studies. It outlines the primary concepts of an IoT architecture, detailing both the hardware and software aspects of "things". Unfortunately, it does not address the IoE, which includes important concepts such as humans and data. Opportunistic IoT services [3] concentrate on modeling IoT services. They usually employ Petri Nets to describe service properties, enabling simulation and formal verification. This allows for controlling the correctness of services in dynamic contexts and verifying the model's conformance to service specifications. SysML4IoT [4] is a SysML profile designed to provide a framework for modeling IoT systems and verifying their properties. In this framework, each component, whether hardware or software, is treated as a stereotype. This approach has been further extended in [5] to accommodate adaptive IoT systems within an MDE framework.

All these works address several concepts required for our approach, except for IoE dimension, particularly concerning humans and data. Therefore, our proposal builds upon these frameworks, especially IoT ARM and SysML4IoT, while incorporating additional IoE concepts.

2.2. CPS Modeling

Most research on CPS is related to the IoT. However, few studies address the architecture of these systems. To the best of our knowledge, no standard or reference model currently exists for CPS architecture. Nonetheless, several modeling projects have been undertaken in recent years.

In CAPS [6], Muccini et al. proposed an architecture framework for modeling and simulating situational-aware CPS. This framework integrates behavioral, hardware, and spatial perspectives based on three distinct modeling views. Notably, the emphasis is put more on device modeling than on the behavioral aspects of CPS. In [7], the authors proposed a CPS metamodel along with a feature-based ontology for CPS. Their work builds on the IoT ARM framework [2] for CPS modeling. The ontology adopts a multi-paradigm approach to represent all system aspects, including constituent concepts, non-functional requirements, application domain, and architecture. As they noted, this approach is not intended for domain-specific modeling. In [8], the National Institute of Standards and Technology proposed a framework for modeling CPS. Specifically, the corresponding authors introduced an analysis method and established a vocabulary for describing it. The framework includes a representation of data exchange, which is vital for CPS operation.

However, while this work is compelling, it is limited to a multi-layer representation of the system and does not provide detailed insights into the CPS architecture. In [9], the authors conceptualized the CPS as components and connectors, a method that explicitly delineates communication and dependencies within the modeling process. Finally, in [10], the authors introduced ThingML, a domain-specific language (DSL) designed to model the architecture and communications between CPS components.

In conclusion, the distinction between IoT and CPS is often ambiguous in CPS modeling literature. The software dimension—encompassing data collection, information building, analysis, and decision-making capabilities—is seldom addressed. Our work considers these aspects, incorporating both the architectural dimension based on the IoE paradigm [11] and the “cyber” component, which enables the comprehensive description and modeling of CPS behavior.

2.3. Modeling Educational Purposes

In the early 2000s, significant work was conducted on the Educational Modeling Language (EML) with the aim of enabling the description of learning units and processes. The literature includes several studies focusing on

- Learning design based on activity modeling language (e.g., E2ML [12], ColeML [13]),
- Learning design based on content structuring language (e.g., PoEML [14], SCORM [15]),
- Learning design that integrates both activity and content structuring languages (e.g., IMS LD [16]), and
- Works focused on evaluation modeling (e.g., IMS QTI [17]).

All these seminal works, which have become benchmarks, were developed before the emergence of the IoT and IoE paradigms. They do not account for cyber–physical systems, particularly the physical interactions among stakeholders or between stakeholders and objects. These studies primarily focus on LMS-based teaching (e.g., Moodle) and often overlook modalities such as face-to-face or blended learning.

In more recent studies, connected cyber–physical learning environments have been explored, based on the IoT paradigm. These can be classified as follows:

- Learning/teaching of IoT: In this category, IoT is the subject of learning (e.g., [18,19]).
- Learning/teaching by IoT: IoT is utilized here as a tool to acquire other knowledge or skills. For more details, the reader can refer to [20,21].
- Learning/teaching based on IoT: Here, IoT is employed to collect data on the educational CPS (learning analytics) to monitor, assess, and manage the learning process. Examples of such works are detailed in [22,23].

While this list is not exhaustive, a major drawback of the referenced works is their failure to incorporate the IoE, thereby excluding people and data as critical components of the system from which data can be collected and upon which actions can be based. As detailed later, our approach utilizes IoE components and collected data to monitor and control educational processes, placing us within the third category: learning/teaching based on IoT but grounded in the recent IoE paradigm.

3. An Overview of University 4.0 and ECPS

University 4.0 can be defined as *the intelligent integration of learning and teaching processes, devices equipped with networked sensors and software, educational stakeholders, and related data. This integration is used to monitor, analyze, plan, control, and respond to achieve improved educational and societal outcomes within the educational context* [24]. ECPS enables the seamless integration of intelligent components. Every ECPS incorporates a specific element of the University 4.0 framework. In other words, the structure of University 4.0 relies on a set of cooperative ECPS.

Thus, ECPS is defined as *an educational complex system consisting of physical and cyber components that utilize control loops to effectively manage educational processes based on the needs and expectations of educational stakeholders* [24,25]. These components are linked to the

Internet of Everything (IoE) since they can make use of interconnected devices (IoT), data, services, and people (educational stakeholders). ECPS implements MAPE (Monitor, Analyze, Plan, Execute) control loops (see Figure 1) to supervise and self-adapt the curriculum, course structures, and pedagogical scenarios based on the students' journey. These loops form the dynamic and adaptive components of the University 4.0 architecture.

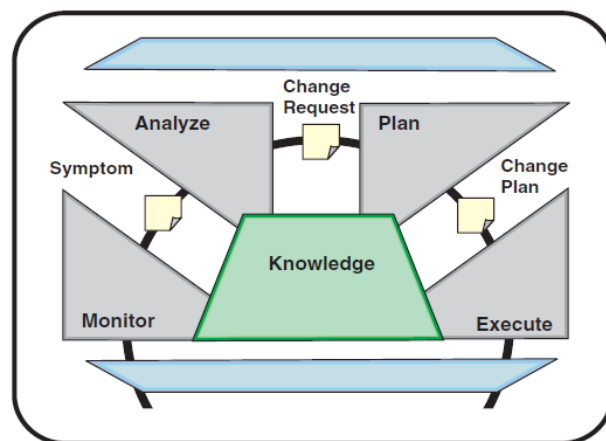


Figure 1. MAPE control loop from IBM's definition [26].

As we are relying on the IoE definition, it is important to clearly identify the “humans” (people) and their roles in the ECPS. With regard to the teaching sequences, the educational stakeholders involved are the learners (who follow and carry out the teaching sequence using the ECPS), the teachers (who follow the progress of each learner in the teaching sequence, thanks to supervision information given by the ECPS), and potentially assistants (technical human resources to help the learners). From the curriculum management point of view, those responsible for all or part of the curriculum (specialty managers, year managers, diploma managers, etc.) are also involved.

As shown in Figure 2, the generic model of an ECPS is composed of several layers, which allows, based on data collected from the physical and cyber educational environment (IoE), to make decisions about adapting the educational process and to apply the actions related to these decisions to that same cyber-physical environment. These layers are as follows.

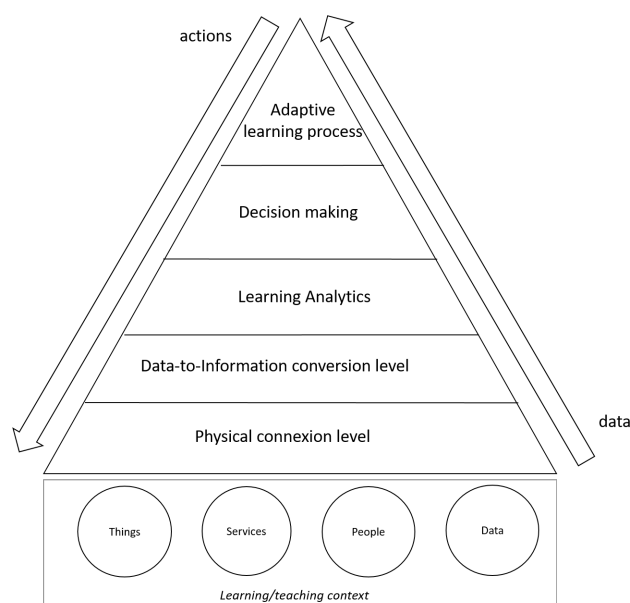


Figure 2. Generic ECPS model.

- Interconnection to the physical world (Layer 1),
- Collection, cleaning, and homogenization of data, followed by extraction/construction of relevant information (Layer 2),
- Information analysis (Layer 3),
- Decision-making and the generation of actions to be carried out (Layer 4), which relies on both information analysis and the adaptive model of the process (Layer 5). This layer can be considered the intelligent component of the ECPS, encompassing AI.

It is important to note that ECPS models may require inter-model data exchange. This exchange is moderated by Layer 4.

ECPS can be instantiated into three categories according to the temporal granularity of the activities they oversee:

- *Classroom ECPS*: Pedagogical activities over short periods, ranging from a few minutes to several hours (practical work sessions, lectures, tutorials, etc.),
- *Course ECPS*: Pedagogical activities over medium durations, spanning over several days to a few weeks (a project, a teaching module, etc.),
- *Curriculum ECPS*: Pedagogical activities over extended periods, ranging from several weeks to several months (a semester, an academic year, etc.).

A *Classroom ECPS* enables the capture of students' progress in various pedagogical activities, with a real-time temporal granularity. The instantiation of the generic model into *Classroom ECPS* is depicted in Figure 3a. Data are collected from the components of both the physical and cyber environments of the supervised pedagogical activity. *Classroom ECPS* supervises practical work activities; for example, traces of student manipulations on the equipment used can be gathered. It is also possible to capture audio exchanges, social media interactions, or video conference communications among students to observe their interactions and assess their collaboration. Furthermore, if online students utilize specific remote devices, such as telepresence robots [27], the movements of these robots within the practical workroom can be tracked to ensure these students have adequately viewed certain equipment or procedures when necessary from a pedagogical perspective or to characterize interactions with their peers.

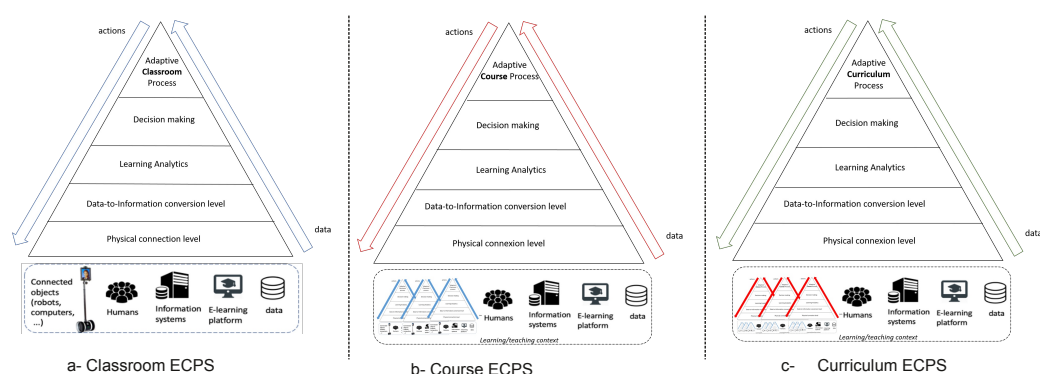


Figure 3. ECPS instances.

A *Course ECPS* enables the capture of students' progress across a set of activities related to the same theme (teaching module). Therefore, it utilizes data from the IoE environment (such as attendance records) but primarily from *Classroom ECPS*, which supervises each activity within the theme. The instantiation of the generic model into *Course ECPS* is depicted in Figure 3b.

A *Curriculum ECPS* captures the progression of students across the entirety of their academic journey (semesters, years, degree). It utilizes data from the IoE environment (such as attendance records, grades, etc.), as well as from *Course ECPSs* that oversee each thematic teaching module. It may also use data from other *Curriculum ECPSs*, depending on the architecture's definition. For example, if a different *Curriculum ECPS* supervises each

academic semester, the *Curriculum ECPS* that monitors and evaluates the students' progress over the entire school year will rely on the two *Curriculum ECPSs* overseeing the first and second semesters, rather than on the *Course ECPS* of each semester. Each *Curriculum ECPS* supervising a semester will in turn collect data from the *Course ECPS* of that semester. The instantiation of the generic model into Curriculum ECPS is depicted in Figure 3c.

In the remainder of this paper, we will focus primarily on Classroom ECPS, which constitutes our case study.

4. Modeling ECPS

This section outlines the process of creating an ECPS model using a Model-Driven Engineering (MDE) approach that involves meta-modeling [28]. It is noteworthy that a meta-model is a modeling language that is used to define a specific domain model. In this study, a Domain-Specific Language (DSL) has been adopted as a meta-model. We implemented the traditional “Y development cycle” methodology from the Model-Driven Architecture (MDA) paradigm [28], as depicted in Figure 4. The objective is to produce Platform Specific Models (PSMs), such as ECPS models, by utilizing Platform-Independent Models (PIMs) that are tailored to the application domain, such as education, and Platform Description Models (PDMs), such as CPS models. There are three primary benefits of this methodology:

- PIM systems are specifically engineered to endure and adapt to frequent alterations in platform technology. For instance, the identical PIM of a teaching sequence (which is modeled by the teacher responsible for the sequence) can be utilized for different PDMs, such as various types of CPS environments (e.g., several practical rooms or several distant students with their own cyber–physical environment) or a recently improved CPS environment with advanced technology. The person responsible for setting up the CPS (engineer, assistant, or teacher) can then configure each CPS to collect the data needed to supervise the teaching sequence.
- The PDM system in a CPS environment can facilitate alternative educational sequences, hence accommodating various PIMs.
- PIM and PDM models are important communication tools when the designer of the two models is not the same: the PDM designer will be able to modify his CPS to collect all the supervision data used in the PIM model, and the PIM designer can adapt their teaching sequence, in particular the steps for validating the student's progress, according to the data available in the CPS. We feel that this is a very important element in the day-to-day use of our proposal.

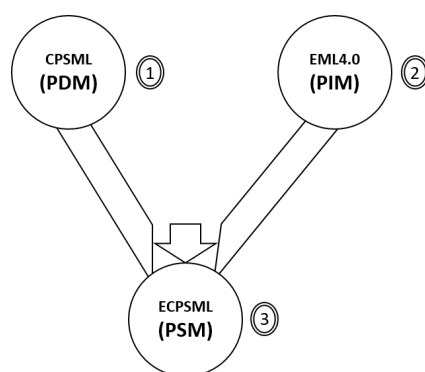


Figure 4. Y development cycle from MDA.

In the following sections, we detail the meta-models employed in our study for PIMs, PDMs, and PSM [24,29].

4.1. CPS Models Using CPSML

We propose CPSML (Cyber-Physical System Modeling Language), a specialized modeling language that aims to simplify the creation of CPS environments, regardless of the domain model. CPSML is a SysML profile [30], which can be seen as an extension of SysML. We improve the block diagram by providing detailed explanations of the block concept using several stereotypes, as seen in Figure 5 (represented by grey boxes). To more accurately describe the Internet of Everything (IoE) environment, we introduce the following stereotypes: *Humans* (students, teachers, administrators, or other individuals involved in the learning and teaching processes), *Objects* (devices that sense data or interact with the environment, such as computers, servers, telepresence robots, phones, etc.), *Services* (VPN services, authentication services, and Discord channels), and *Data* (data flows originating from elements of the CPS, including resources available on Technology-Enhanced Learning systems, and Discord activity logs).

In addition, we extend the activity diagram to include Autonomic Computing (AC) elements, as described in [26]. This extension specifically defines several types of operations that are associated with controlling, analyzing, and utilizing the computing capabilities of CPSs. To be more explicit, we are introducing the archetypes of *Monitoring Action*, *Analyzing Action*, and *Event*.

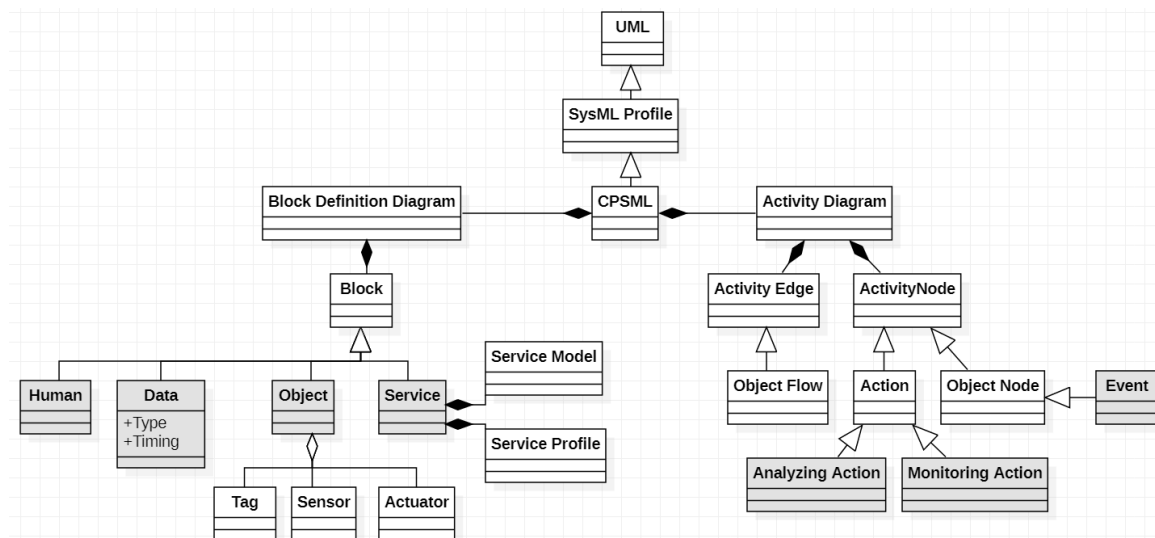


Figure 5. CPSML.

4.2. Teaching Sequence Models Using EML4.0

To represent the domain-specific aspect (teaching sequences), a language that combines both behavioral and structural elements is required. To achieve this, we adopt an approach called Educational Modeling Language for University 4.0 (EML4.0). It is built upon IMS-LD (Instructional Modeling System–Learning Design) [16], which is an IMS standard. We have expanded it by incorporating many archetypes, as illustrated in Figure 6: *Learning Activity*, *Learning Object*, *Learning Service*, and *TELS* (such as ITS, MOOC, and Moodle).

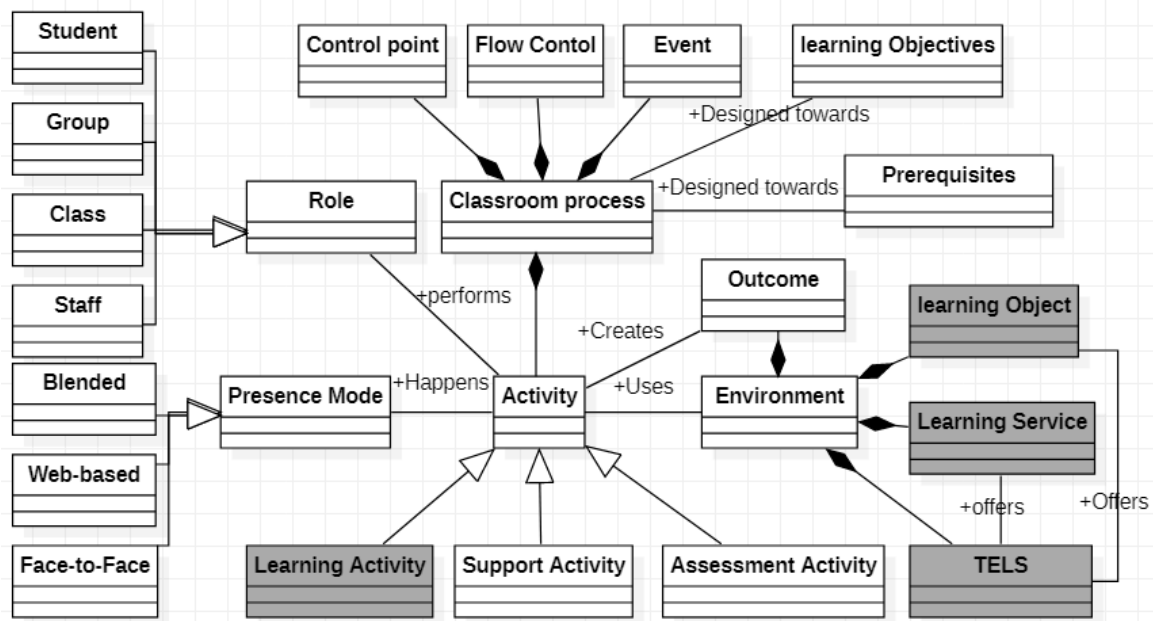


Figure 6. EML4.0.

4.3. ECPS Models Using ECPSML

The ECPS model depicted in Figure 7 is derived from the combination of two models: the CPS model and the educational model, following the Y-process outlined in Figure 4. Similar to the CPS concept, it is a profile in SysML. It combines structural principles from both CPS and pedagogical meta-models. There are two distinct types of objects: *technological objects*, which are developed from the CPS model, and *pedagogical objects*, which are derived from the pedagogical model. From a behavioral standpoint, the *learning activity* can be classified as an action within a SysML profile. The generation of ECPS models is managed using ATL [31]. Examples of some of the rules we employed are given in Figure 8.

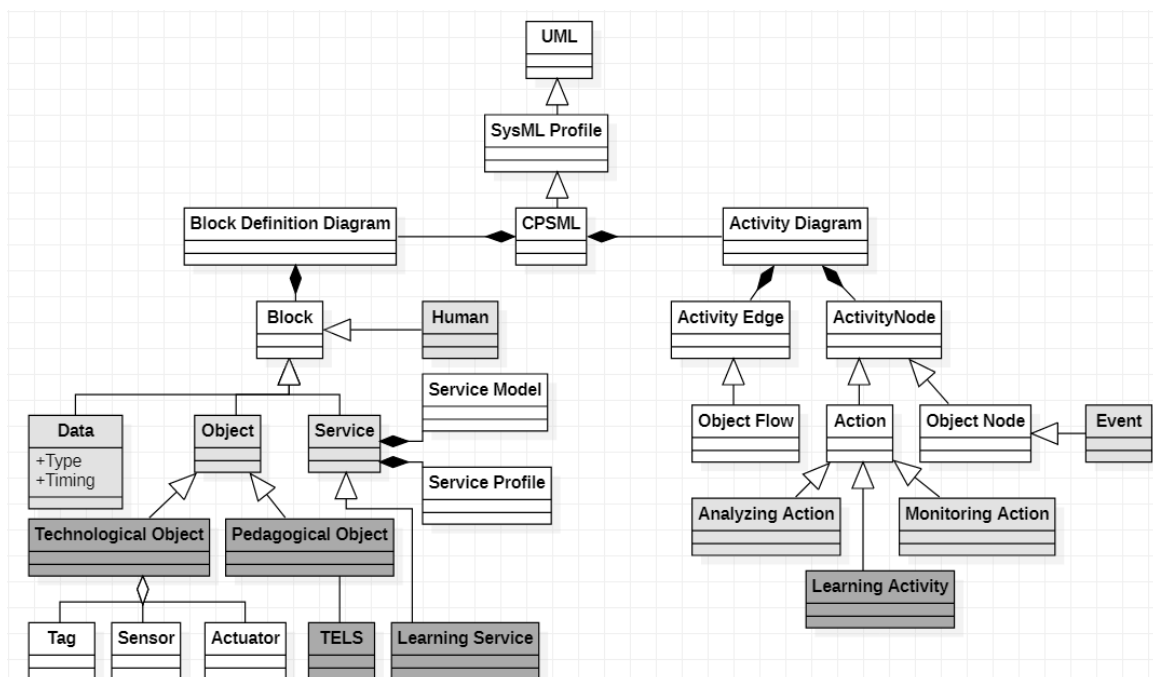


Figure 7. ECPSML.


```

1 -- @path CPSMM=/CPSML_v2/model.profile.di
2 -- @path EMLMM=/ECPS-v2/MetaModels/EML4_MetaModel.ecore
3 -- @path ECPSMM=/ECPS-profile/model.profile.di
4
5 module MyRules;
6 create OUT : ECPSMM from IN : CPSMM, IN1 : EMLMM;
7 rule Obj2tObj {
8   from
9     s1 : CPSMM!Object
10  to
11    t : ECPSMM!TechnologicalObject (
12      name <- s1.name
13    )
14 }
15 rule Obj2pObj {
16   from
17     s11 : EMLMM!LearningObject
18  to
19    t1 : ECPSMM!PedagogicalObject (
20      name <- s11.name
21    )
22 }
23 }

```

Figure 8. Examples of some ATL rules.

5. Case Study

We conducted an experiment during the second and third lockdowns in France, spanning from January to June 2021. The study was hosted at the Technology Institute of Pays de l'Adour, specifically in the Department of Networks and Telecommunications, located in Mont-de-Marsan.

The experiment focused exclusively on the IP telephony course. There were 54 students and 2 teachers involved. The course was delivered in a hybrid mode, with half of the students attending in person and the other half participating remotely. The content of the lecture and tutorial sessions, not being directly relevant to our study, will not be discussed here; our focus will solely be on the practical work sessions. In these sessions, students worked in groups of no more than fourteen, under the supervision of one teacher. This setup allowed for a maximum of seven students on-site and seven students online, working in pairs concurrently (each in-person student paired with one remote student). The entire teaching sequence included four practical work sessions and one final assessment session, also conducted as practical work, as illustrated in Figure 9.

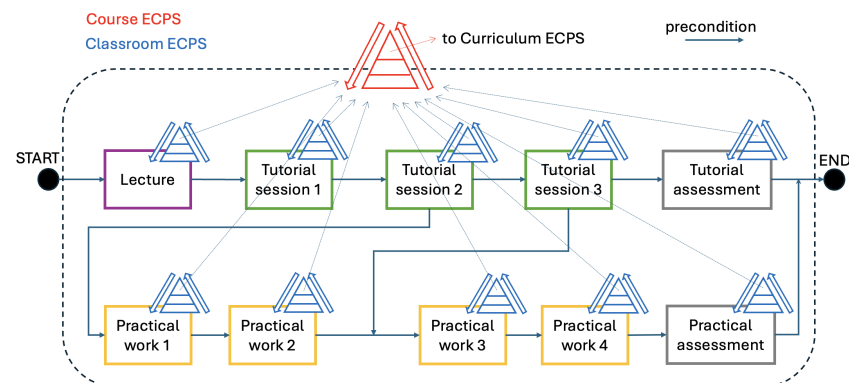


Figure 9. ToIP course architecture with a dedicated ECPS.

The main objective of the practical sessions was to establish a private telephony network (including phones, a server, and network devices), configure various services (e.g., messaging service), and establish a connection through a public network (e.g., Internet, ISDN). Figure 10 depicts the infrastructure built for each pair during each practical session. This setup was repeated seven times, as there were a maximum of seven pairs. It is important to note that the remote component of each ECPS pair may have varied significantly for each distance-learning student, depending on their network environment and computer devices. This remote setup included a set of tools (VPN, SSH, web interfaces,

TFTP) that enable the student to interact with the equipment in the practical room (connect, observe, and configure), as well as a phone (softphone) installed locally on the student's computer at home. The local part of the ECPS, located in the practical room, consisted of an Asterisk server, IP phones, and network infrastructure (switch, PoE, public phone network simulator). Communication between peers, as well as between remote students and the teacher, was facilitated using Discord channels. All documents necessary for conducting the experiments were available on a Moodle platform (<https://elearn.univ-pau.fr/>, accessed on 1 September 2024).

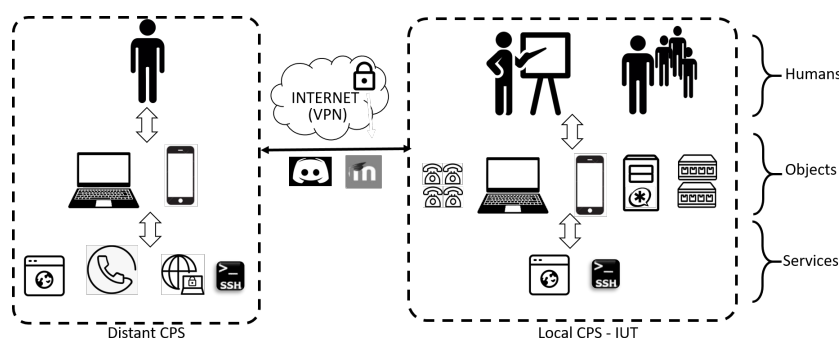


Figure 10. ToIP practical work ECPS setup for each pair.

The ECPS configuration used (comprising the remote, local, and communication parts) remained consistent across each practical session. However, the pedagogical content varied from one session to another, reflecting pedagogical progression, and even within the same session. Specifically, the teachers had set different objectives for the remote-learning students compared to the face-to-face students. Consequently, this led to two distinct topics for each practical session. Therefore, in terms of modeling, we had two different teaching models for each session, with varying models across the sessions. Ultimately, eight distinct teaching models were implemented using the same CPS setup. Focusing on one practical session, we can have up to seven different CPSs (if the distant CPS part is different for each distant student), and two different teaching models, making a total of fourteen potentially different ECPSs. Our example is therefore relatively complex in its configuration and can pose the problem of the availability of data on the ECPS set, as well as its quality, as we will discuss later.

5.1. Modeling Various Aspects of the System

As described above, our goal was to model the platform-dependent aspects (CPS) and the platform-independent aspects (teaching sequence) separately, before combining them to create a comprehensive model of the practical session. Based on this model, it was then possible to automatically generate dashboards for the real-time monitoring of student progress within the teaching activities conducted in the modeled practical room.

5.1.1. Equating PDM Aspects with the CPS Model

In this section, we show how we modeled the PDM perspective, which includes the main components of the IoE educational environment and the various potential actions (monitoring, analyzing, etc.) performed within it. This modeling was conducted by the educational engineer, who is responsible for the operation and maintenance of the practical room. It is important to note that teachers do not perform this modeling.

To achieve this, one can use our CPSML language, implemented in an Eclipse component (Papyrus). The results are depicted in Figure 11. Each type of IoE component within the CPS is distinguished by a different color: objects in yellow, services in green, data in grey, and humans in red. Entities in the remote part of the CPS are labeled “distant”, while those in the local part are labeled “present”. Additionally, white blocks indicate elements inherited from SysML.

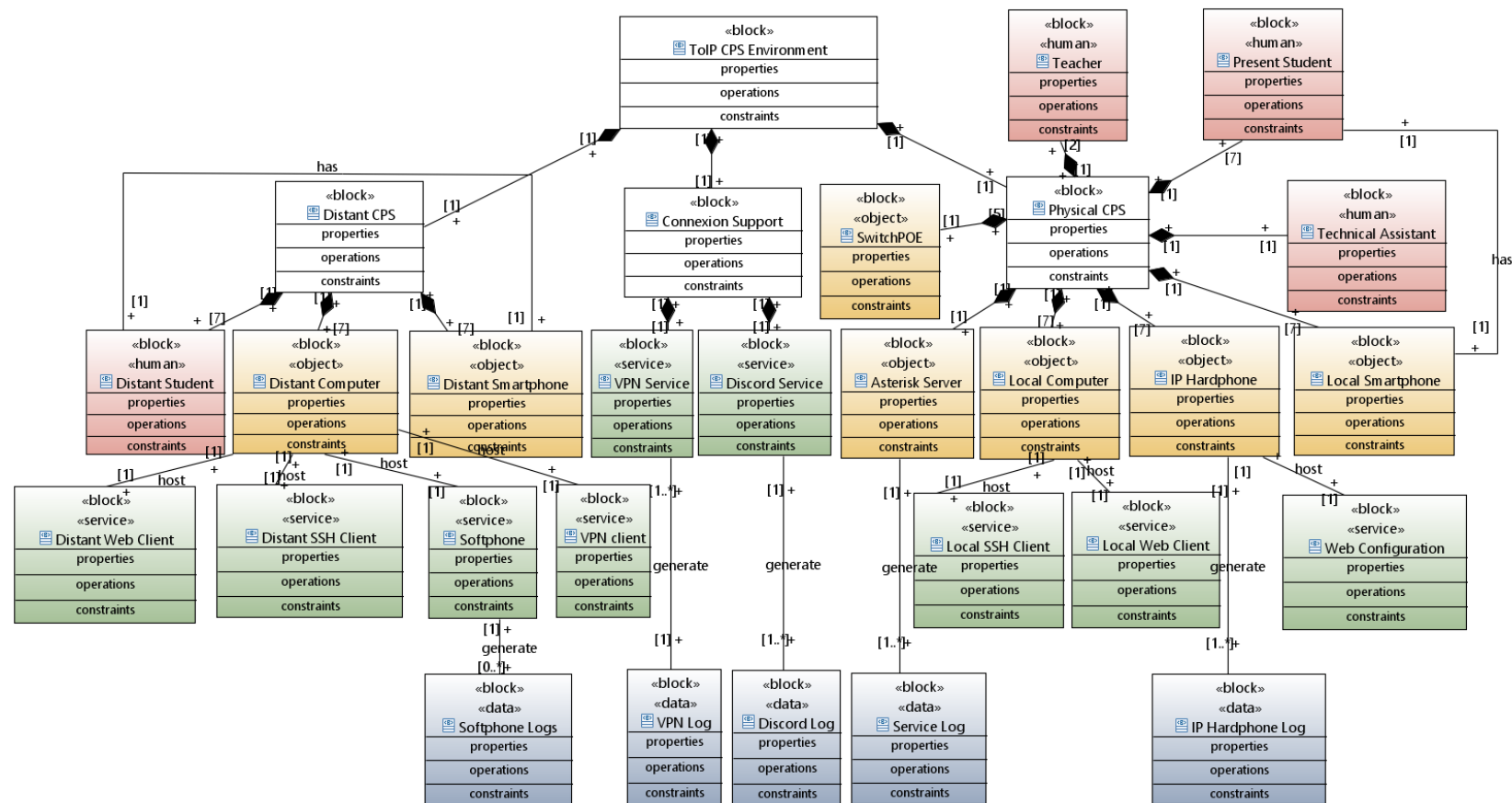


Figure 11. CPS model.

The dynamic aspects of the system involve supervising the data generated by various CPS entities to monitor students' progress throughout the teaching sequence. Figure 12 shows the activity diagram (behavioral aspects) of the implementation of the MAPE control loops to continuously collect data from the CPS components. It is important to note that we are modeling all the data (logs) we intend to collect, including their format. The implementation of this data collection, driven by the MAPE control loop code, is highly dependent on the hardware and is not directly related to the teaching sequence. Thus, it is within the pedagogical model (EML4.0) that we specify how these data should be utilized. This approach ensures that the modeling of dynamic aspects remains independent from the pedagogical aspects, aligning with the Y model concept.

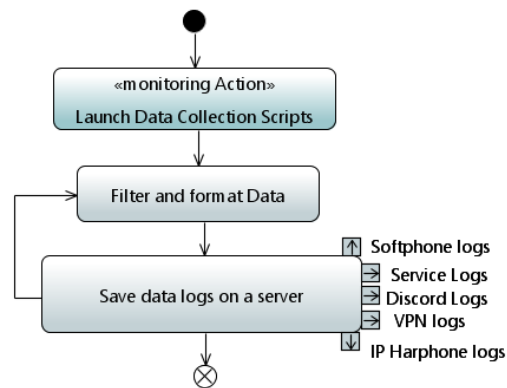


Figure 12. MAPE CPS model.

5.1.2. Aligning PIM Aspects with Pedagogical Models

In this section, we show how to model from the PIM perspective, focusing on the pedagogical aspects of the system. More precisely, we examine pedagogical processes and learning activities. This part of the modeling is undertaken by the teachers. They are not required to understand the technological details of the practical room; instead, they need to identify the data necessary to monitor teaching activities and describe both the structural and behavioral aspects of the sequence. Figure 13 shows a manual draft of a learning scenario created by the teachers (written in French). The left side details the steps of student progression, while the right side identifies the indicators (data) that can be used to monitor each step, i.e., determining whether each step has been accomplished. These indicators form a link with the CPS model.



Figure 13. Manual draft of the learning scenario.

The pedagogical process is illustrated in Figure 14. The sequential flow of different learning activities is specified using control flows. Control points are established to monitor students' progress during the session. These points consist of a set of indicators (events) that are validated through the implementation of MAPE control loops in the CPSML model. A learning activity is considered accomplished only if all indicators at the associated control point are confirmed as true.

Similarly, the learning environment is described in Figure 15. It encompasses learning objects, learning services, and TELs upon which the teaching sequence is based.

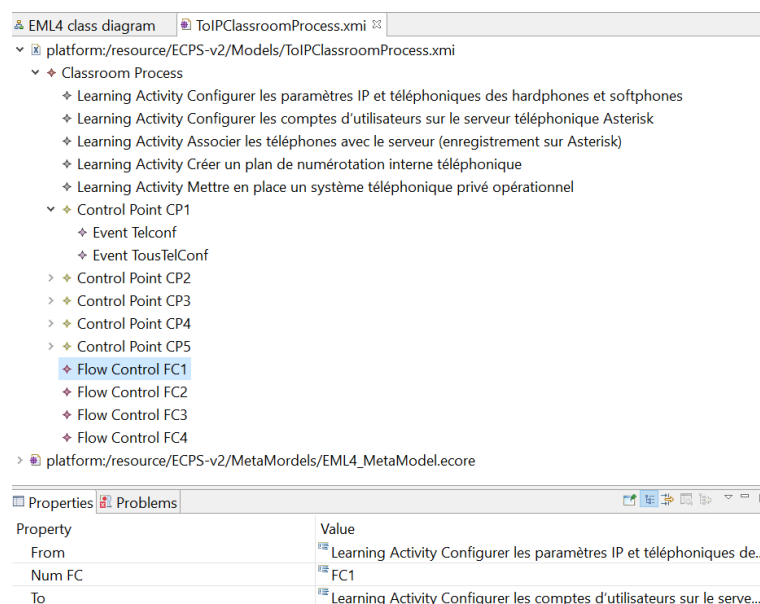


Figure 14. Learning activities and controls (EML4.0-compliant).

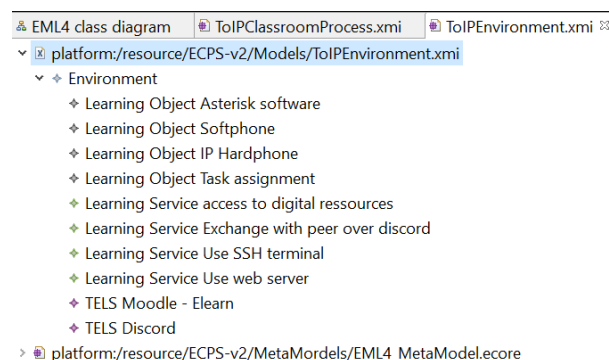


Figure 15. Modeled learning environment through learning objects, learning services, etc. (conforms to EML4.0).

5.1.3. ECPS Model

As previously mentioned, our ECPS model corresponds to a PSM model within the Model-Driven Architecture (MDA) framework. To achieve this, model transformations are necessary, which integrate the CPSML model with the EML4.0 model. Figure 16 illustrates a segment of the model derived from applying two ATL rules to these models. The CPSML model addresses technological issues, while the EML4.0 model focuses on pedagogical aspects. In the construction of the ECPS model, each concept from the two preceding models is aligned to ensure that it exhibits both the technological and pedagogical properties in the resultant model.

```

ECPSGeneratedModel.xmi
<?xml version="1.0" encoding="ISO-8859-1"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns="www.ecps.com">
  <TechnologicalObject name="Distant Computer"/>
  <TechnologicalObject name="Local Computer"/>
  <TechnologicalObject name="Distant Smartphone"/>
  <TechnologicalObject name="Local Smartphone"/>
  <PedagogicalObject name="Asterisk Server"/>
  <PedagogicalObject name="IP Hardphone"/>
  <PedagogicalObject name="Task assignment"/>
</xmi:XMI>

```

Figure 16. Automatic transformation of the ECPS model via ATL rules.

Figure 17 shows the block diagram of the resulting ECPSML model, while Figure 18 shows the activity diagram. In the latter, each type of activity is distinguished by a different color: green for learning activities, grey for monitoring activities, and yellow for analysis activities. Notably, among the monitoring activities, one involves the collection of data on the components of the CPS (shown at the bottom of the figure), while another pertains to updating the dashboards that report on the student's progress in the teaching sequence (shown at the top of the figure).

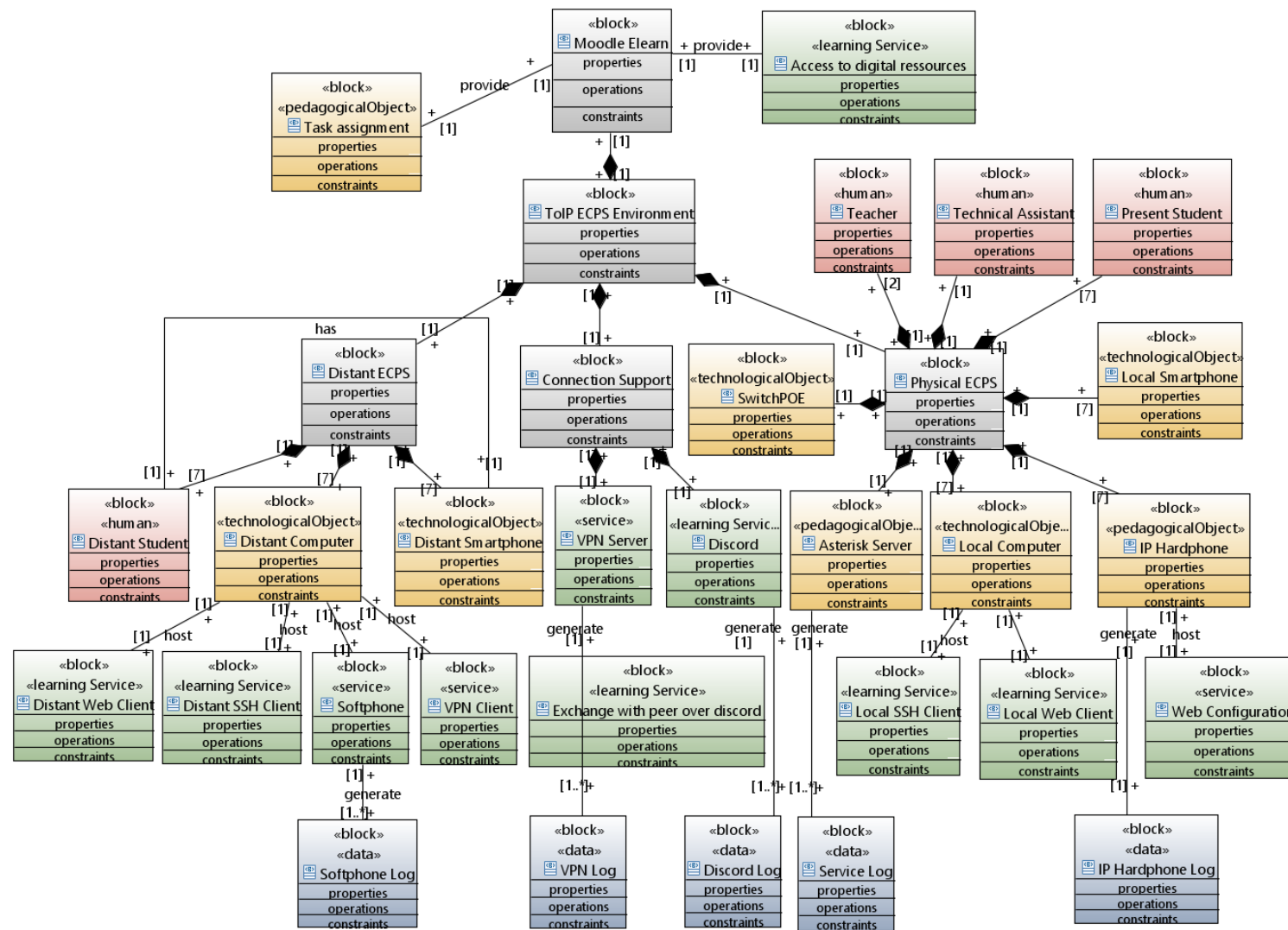


Figure 17. ECPS model.

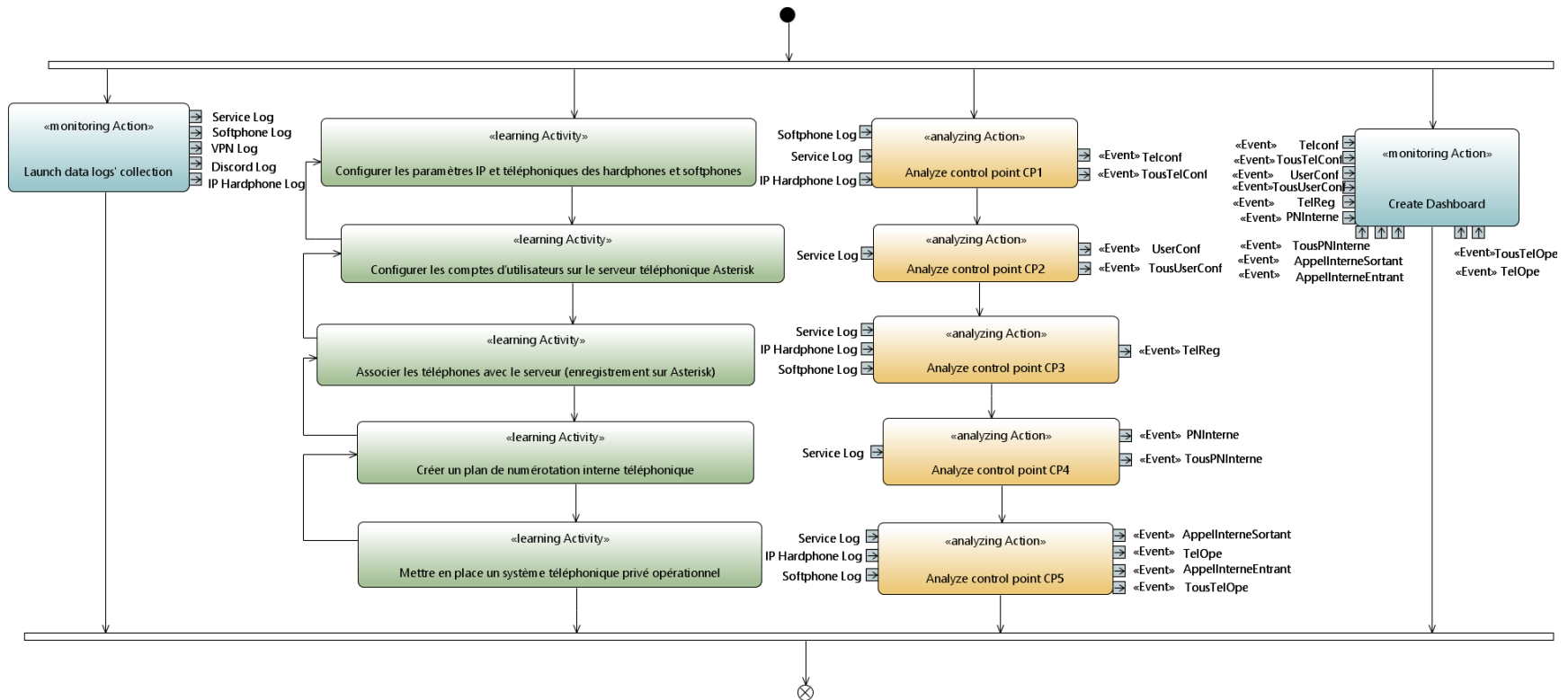


Figure 18. ECPS activity diagram.

5.2. Runtime Application of ECPSML

First of all, it is important to note that the ECPSML model is designed for implementation in an ECPS classroom to automatically supervise the practical work session at runtime. Figure 19 details this implementation over the structure of the Classroom ECPS, highlighted in red.

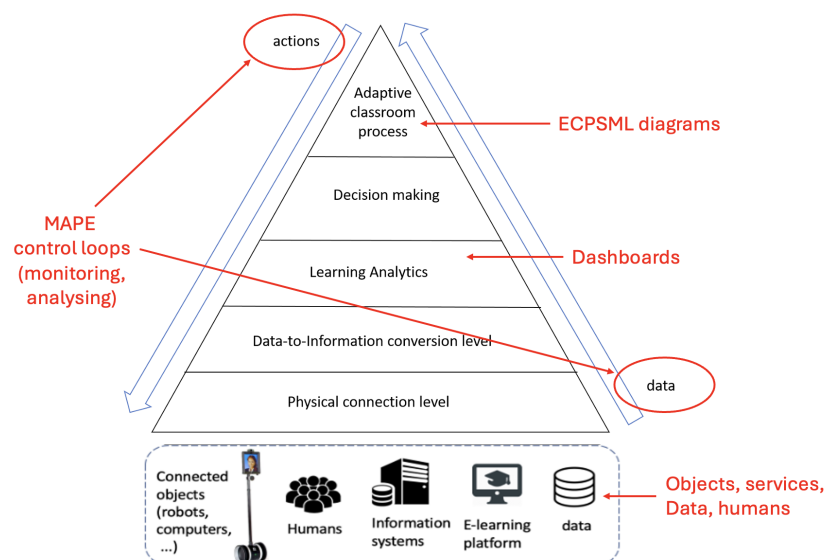


Figure 19. Implementation of the ECPSML model in a ClassroomECPS.

Dashboards are crucial for visualizing students' progress throughout the teaching sequence. In our experiment, we developed three dashboards: two for teachers and one for students.

The first dashboard, dedicated to teachers, indicates the availability of all necessary data for supervising each part of the CPS. Figure 20 displays the part of this dashboard dedicated to the telephony server. Each line labeled "Asterisk *" represents a student's peer CPS. Lines in black indicate that the corresponding peer CPS was not used during the session. Each column corresponds to a data point to be collected. A green box signifies that the data are available in the corresponding CPS. A red box indicates that the system is currently unable to collect the data. In such cases, all steps that require these data for validation cannot be verified.

Let us take the example of control point CP1 in Figure 18. The corresponding activity takes Softphone logs, IP Hardphone logs, and Service logs as inputs. Service logs are made up of SauveTFTP data, to see whether the students have successfully deposited the configuration files in the TFTP directory for IP hardware phones, and ChanSIPlog data, to find out whether the phones are trying to authenticate themselves to the server, which indicates that they have been correctly configured (ChanSIPlogs data give all messages exchanged between the telephony server and the phones). Finally, the outputs of this activity (Telconf = at least one phone is configured; TousTelConf = all phones are configured) are used to validate step 1 of the teaching sequence (see Figure 21).

As evident from Figure 20, there were varying configurations among different student groups. For the FA group, all data were available. However, for other groups, several data points were unavailable, and sometimes, all of the CPS data was entirely unavailable (line entirely in red). Data unavailability often occurs by group because it is collected from the same equipment. If this equipment malfunctions or is turned off, the entire data group becomes unavailable (e.g., a server from which data are retrieved across several log files, or a VPN connection that fails, preventing data retrieval from the remote part of the CPS).

TP3	Bashsyslog	Chan SIP log	Dahdi	Fullog	Name	PBX log	Plandenum	Registrar	SauveTFTP
FA									
Asterisk 1 (201)									
Asterisk 3 (203)									
Asterisk 4 (204)									
Asterisk 5 (205)									
Asterisk 6 (206)									
Asterisk 7 (207)									
Asterisk 8 (208)									
G1									
Asterisk 1 (201)									
Asterisk 3 (203)									
Asterisk 4 (204)									
Asterisk 5 (205)									
Asterisk 6 (206)									
Asterisk 7 (207)									
Asterisk 8 (208)									
G2									
Asterisk 1 (201)									
Asterisk 3 (203)									
Asterisk 4 (204)									
Asterisk 5 (205)									
Asterisk 6 (206)									
Asterisk 7 (207)									
Asterisk 8 (208)									
G3									
Asterisk 1 (201)									
Asterisk 3 (203)									
Asterisk 4 (204)									
Asterisk 5 (205)									
Asterisk 6 (206)									
Asterisk 7 (207)									
Asterisk 8 (208)									

Figure 20. CPS data dashboard availability—green for available, red for unavailable, black for not used.

General / TP Asterisk ☆ 🔊											
Récapitulatif Étapes											
serveur ▾	etape1 ▾	etape2 ▾	etape3 ▾	etape4 ▾	etape5 ▾	etape6 ▾	etape7 ▾	etape8 ▾	etape9 ▾	etape11 ▾	etape12 ▾
asterisk1	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé
asterisk3	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé
asterisk4	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé
asterisk5	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé
asterisk6	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé
asterisk7	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé
asterisk8	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé	Non Validé
Nombre d'indicateurs consécutifs validés						Nombre d'indicateurs validés					
asterisk1					0	asterisk1					0
asterisk3					0	asterisk3					0
asterisk4					0	asterisk4					0
asterisk5					0	asterisk5					0
asterisk6					0	asterisk6					0
asterisk7					0	asterisk7					0

Figure 21. Teacher dashboard.

The second teacher dashboard (Figure 21) displays a table charting each student pair’s progress through the stages of the learning sequence. Each row represents a peer pair (one face-to-face student and one distant student), while each column corresponds to a step in the teaching sequence. A red box indicates that the corresponding step has not been validated by the student pair. Conversely, a green box signifies that all indicators associated with that step’s validation have been checked, and the stage is thus considered completed.

The student dashboard (Figure 22) features the same elements as the second teacher dashboard, though students can only view their progress line. Additionally, it includes a few extra indicators not detailed here.

Dashboard update routines are directly linked to the MAPE control loops. In our experiment, these loops were continuously active, allowing all indicators to be validated at any time during the teaching sequence. This adaptability catered to the diverse strategies

students employed to progress through the sequence. For instance, some students completed certain steps before finishing prior ones. Observing these patterns provided deeper insights into their approach to the teaching sequence, as well as their understanding and engagement with the provided materials.

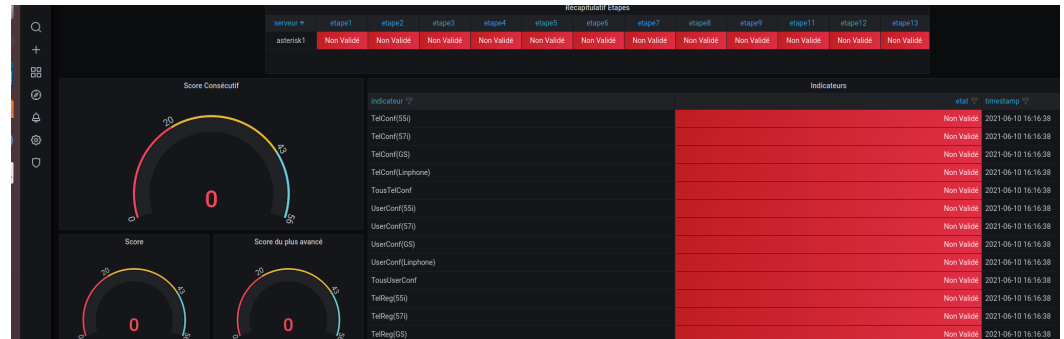


Figure 22. Student dashboard.

5.3. Discussion on the Case Study

This case study was carried out with four different groups of students (FA, G1, G2, G3) and two different teachers. It enabled us to monitor the progress of the students and to quickly identify the “bottlenecks” that are holding them back and preventing them from completing the work. Note that we recorded the evolution over time of the indicators from the different activities given in Figure 20. We were then able to replay the evolution of the students in the sequence in order to better understand their strategies and the mistakes they made. It is also possible to set up a reflective activity for the students, during which they replay their own activity, to understand why they did not succeed in validating certain stages.

This example, which may seem simple at first glance, generated up to 14 different ECPS to be managed simultaneously by the same teacher. This requires a detailed knowledge of the teaching sequence and the technical reasons that could explain the non-validation of the steps. In addition, it is difficult to obtain the total availability of data on such a number of ECPS. As a result, the teacher may be blind to the progress of students in certain stages of the sequence, i.e., have no information on whether or not the latter has been validated. We solved this problem by multiplying the data sources, allowing us to validate or not validate the indicators. For example, regarding the registration of a phone, we can go and check in the logs of this phone or check in the server logs if the phone has registered. Thus, if one of the two data sources is not available, then the second still allows us to obtain the information. Such a strategy requires a lot of configuration time before being able to carry out the teaching sequence, but it ensures a certain level of quality of our data.

6. Discussion

In this article, we introduce the notion of ECPS and propose an IDM-based methodology for modeling and implementing them. We show that it is possible to break down the modeling into two parts, one devoted to the physical platform aspects (CPS and IoE paradigms) and the other to the educational aspects. We propose two modeling languages (DSLs) dedicated to these two parts. Finally, we use a Y-process to obtain the complete ECPS model. An example illustrates how this approach can be used in practice.

Our main contribution concerns three points: the definition and the architecture of the ECPS, which to our knowledge has not been addressed in the literature, the definition of specific DSLs for the CPS and pedagogical activity models, and finally the use of the y-process, which to our knowledge is rarely used.

We believe our proposal offers several distinct advantages:

- It leverages the recent IoE paradigm, incorporating humans, data, and services as integral components of the cyber-physical system. This integration allows us to

model interactions with human elements—learners (e.g., dynamically modifying the learning sequence in real-time) and instructors (e.g., directing teachers to assist students facing difficulties).

- The MDE dimension of our proposal enhances the interoperability and reusability of the developed models. Consequently, a pedagogical sequence can be deployed across various ECPSs (e.g., practical learning environments), and a single ECPS can accommodate multiple pedagogical sequences.
- Our approach is agnostic to the mode of teaching presence. It seamlessly incorporates face-to-face, remote, and blended learning environments and can manage learning spaces composed of multiple interconnected physical locations.
- It addresses the various granular levels present in university curricula—from individual teaching activities to modules, teaching units, semesters, and academic years—allowing for the comprehensive modeling of all teaching sequences.
- The CPS and learning models are distinct and manageable by different individuals who need not be experts in both domains.
- Student progress supervision within the learning sequence is driven by metrics from MAPE loops specific to the CPS but reusable across any teaching sequence that utilizes the same CPS, thus bridging the physical and cyber components of the CPS.
- Our supervision extends beyond physical spaces; it can also encompass cyber dimensions. For instance, a Discord bot can be employed to evaluate interactions among remote students within dedicated channels.

However, several drawbacks present opportunities for further enhancements of our work:

- Currently, our tool for modeling teaching sequences is text-based and requires in-depth knowledge of the model before it can be effectively used by educators. We plan to develop a more user-friendly graphical version that simplifies usage without extensive technical knowledge.
- Additionally, in our experiment, the construction of dashboards used to monitor student progress is predetermined. Enabling teachers to customize dashboard views could enhance utility; this could be achieved by introducing a third model dedicated to dashboard construction, which, when integrated with the ECPS and teaching sequence models, would allow for the automated assembly of these dashboards.
- At present, adaptations to the teaching sequence are static and predefined within the teaching model. Future developments will explore the implementation of an intelligent assistant capable of making runtime adjustments to the sequence.
- Moreover, our experiment has been confined to the classroom level of our architecture. Going forward, we will expand our focus to include the course and curriculum levels, particularly exploring interactions among various ECPS levels to facilitate the comprehensive supervision of the entire student curriculum.

Author Contributions: Conceptualization, L.G., K.S., R.C., S.B. and P.A.; Methodology, L.G., K.S., R.C., S.B. and P.A.; Software, L.G., S.B. and P.A.; Validation, L.G., K.S., R.C. and S.B.; Formal analysis, L.G., R.C. and S.B.; Investigation, L.G., K.S., S.B. and P.A.; Resources, L.G., R.C., S.B. and P.A.; Data curation, L.G., K.S., R.C. and S.B.; Writing – original draft, L.G. and S.B.; Writing – review & editing, L.G., K.S., R.C., S.B. and P.A.; Visualization, L.G., K.S., R.C. and S.B.; Supervision, L.G., K.S., R.C. and P.A.; Project administration, L.G. and P.A.; Funding acquisition, L.G.—and P.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been funded for three years by the French “Conseil Départemental des Landes”. The authors wish to express their gratitude for this invaluable support.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: Author Samia Bachir was employed by the company Capgemini Engineering. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Educause: The Difference Between Emergency Remote Teaching and Online Learning. Available online: <https://er.educause.edu/articles/2020/3/the-difference-between-emergency-remote-teaching-and-online-learning> (accessed on 12 June 2024).
2. Bauer, M.; Bui, N.; De Loof, J.; Magerkurth, C.; Nettsträter, A.; Stefa, J.; Walewski, J.W. Iot reference model. In *Enabling Things to Talk*; Springer: Berlin/Heidelberg, Germany, 2013.
3. Fortino, G.; Russo, W.; Savaglio, C.; Viroli, M.; Zhou, M. *Modeling Opportunistic Iot Services in Open Iot Ecosystems*; WOA: Schleswig-Holstein, Germany, 2017.
4. Costa, B.; Pires, P.F.; Delicato, F.C.; Li, W.; Zomaya, A.Y. Design and analysis of iot applications: a model-driven approach. In Proceedings of the 14th IEEE International Conferences on Dependable, Autonomic and Secure Computing, Auckland, New Zealand, 8–12 August 2016.
5. Hussein, M.; Li, S.; Radermacher, A. Model-Driven Development of Adaptive Iot Systems. MODELS (Satellite Events). 2017. Available online: https://ceur-ws.org/Vol-2019/?utm_source=chatgpt.com (accessed on 1 September 2024).
6. Muccini, H.; Sharaf, M. Caps: Architecture description of situational aware cyber physical systems. In Proceedings of the 2017 IEEE International Conference on Software Architecture (ICSA), Gothenburg, Sweden, 3–7 April 2017.
7. Tekinerdogan, B.; Rakshit, M.; Al Ali, R.; Mauro, I.; Eva, N.; Ken, V.; Barisic, A. A feature-based ontology for cyber-physical systems. In *Multi-Paradigm Modelling Approaches for Cyber-Physical Systems*; Academic Press: Cambridge, MA, USA, 2021; pp. 45–65.
8. Cyber Physical Systems Public Working Group. *Framework for Cyber-Physical Systems, Release 1.0*; National Institute for Standards and Technology: Gaithersburg, MD, USA, 2016; pp. 1–266.
9. Ringert, J.O.; Rumpe, B.; Wortmann, A. From software architecture structure and behavior modeling to implementations of cyber-physical systems. *arXiv* **2014**, arXiv:1408.5690.
10. Harrand, N.; Fleurey, F.; Morin, B.; Husa, K.E. Thingml: A language and code generation framework for heterogeneous targets. In Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, Saint-Malo, France, 2–7 October 2016.
11. Cisco. Digital Transformation with the Internet of Everything. 2016. Available online: https://www.cisco.com/c/dam/m/en_us/ioe/digital-transformation-stories/digital-transformation-with-the-internet-of-everything.pdf (accessed on 9 July 2024).
12. Botturi, L. E2ml-educational environment modeling language. In *EdMedia+ Innovate Learning*; Association for the Advancement of Computing in Education (AACE): Asheville, NC, USA, 2003.
13. Martel, C.; Vignollet, L.; Ferraris, C.; Durand, G. Ldl: A language to model collaborative learning activities. In Proceedings of the 2006 World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA'2006), Orlando, FL, USA, 26–30 June 2006.
14. Caeiro-Rodriguez, M. POEML: A separation of concerns proposal to instructional design. In *Handbook of Visual Languages for Instructional Design: Theories and Practices*; IGI Global: Hershey, PA, USA, 2008.
15. Bohl, O.; Scheuhase, J.; Sengler, R.; Winand, U. The sharable content object reference model (SCORM)—A critical review. In Proceedings of the International Conference on Computers in Education, Auckland, New Zealand, 3–6 December 2002; Volume 2, pp. 950–951.
16. Koper, R.; Olivier, B. Representing the learning design of units of learning. *J. Educ. Technol. Soc.* **2004**, *7*, 97–111.
17. Abelló, A.; Rodríguez, M.E.; Urpí, T.; Burgués, X.; Casany, M.J.; Martín, C.; Quer, C. LEARN-SQL: Automatic Assessment of SQL Based on IMS QTI Specification. In Proceedings of the 2008 Eighth IEEE International Conference on Advanced Learning Technologies, Santander, Spain, 1–5 July 2008; pp. 592–593.
18. Quezada-Sarmiento, P.-A.; Enciso, L.; Washizaki, H.; Hernandez, W. Body of knowledge on iot education. In Proceedings of the WEBIST, Seville, Spain, 18–20 September 2018; pp. 449–453.
19. Burd, B.; Barker, L.; Divitini, M.; Perez, F.A.F.; Russell, I.; Siever, B.; Tudor, L. Courses, content, and tools for Internet of things in computer science education. In Proceedings of the 2017 ITiCSE Conference on Working Group Reports, Bologna, Italy, 3–5 July 2017; Association for Computing Machinery: New York, NY, USA, 2018; pp. 125–139.
20. Yang, Y.; Yu, K. Construction of distance education classroom in architecture specialty based on Internet of things technology. *Int. J. Emerg. Technol. Learn. (IJET)* **2016**, *11*, 56–61. [[CrossRef](#)]
21. Moreira, F.T.; Magalhaes, A.; Ramos, F.; Vairinhos, M. The power of the Internet of things in education: an overview of current status and potential. In Proceedings of the Conference on Smart Learning Ecosystems and Regional Development, Aveiro, Portugal, 22–23 June 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 51–63.
22. Takpor, T.; Atayero, A.A. Integrating Internet of things and ehealth solutions for students' healthcare. In Proceedings of the World Congress on Engineering, London, UK, 1–3 July 2015; World Congress on Engineering: London, UK; Volume 1.
23. Palma, D.; Agudo, J.; Sanchez, H.; Macias, M. An Internet of things example: Classrooms access control over near field communication. *Sensors* **2014**, *14*, 6998–7012. [[CrossRef](#)] [[PubMed](#)]
24. Bachir, S. Towards University 4.0: A Model Driven Engineering Method to Design Educational Cyber Physical Systems. Ph.D. Thesis, University of Pau and Pays de l'Adour, Pau, France, January 2022.
25. Lee, J.; Bagheri, B.; Kao, H.-A. A cyberphysical systems architecture for industry 4.0-based manufacturing systems. *Manuf. Lett.* **2015**, *3*, 18–23. [[CrossRef](#)]

26. *Autonomic Computing. An Architectural Blueprint for Autonomic Computing*; IBM White Paper; IBM: Armonk, NY, USA, 2006; Volume 31, pp. 1–6.
27. Gallon, L.; Abénia, A.; Dubergey, F.; Négui, M. Using a Telepresence Robot in an Educational Context. In Proceedings of the 15th International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS), Las Vegas, NV, USA, 28 July–1 August 2019.
28. Bezivin, J. On the unification power of models. *Softw. Syst. Model.* **2005**, *4*, 171–188. [[CrossRef](#)]
29. Bachir, S.; Gallon, L.; Aniorte, P.; Abenia, A. A Model Driven Method to Design Educational Cyber Physical Systems. In Proceedings of the 16th International Conference on Software Technologies (ICSOFT 2021), Online, 6–8 July 2021.
30. OMG Systems Modeling Language (OMG SysML). Available online: <https://sysml.org/.res/docs/specs/OMGSysML-v1.4-15-06-03.pdf> (accessed on 12 June 2024).
31. Allilaire, F.; Idrissi, T. Adt: Eclipse development tools for atl. In Proceedings of the Second European Workshop on Model Driven Architecture (MDA) with an Emphasis on Methodologies and Transformations, Canterbury, UK, 7–8 September 2004.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.