



HAL
open science

Towards University 4.0 : A Model-Driven-Engineering Method to Design Educational Cyber Physical Systems

Samia Bachir

► **To cite this version:**

Samia Bachir. Towards University 4.0: A Model-Driven-Engineering Method to Design Educational Cyber Physical Systems. Technology for Human Learning. Université de Pau et des Pays de l'Adour, 2022. English. NNT : 2022PAUU3001 . tel-03643089

HAL Id: tel-03643089

<https://theses.hal.science/tel-03643089v1>

Submitted on 15 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR

ÉCOLE DOCTORALE DES SCIENCES EXACTES ET LEURS
APPLICATIONS



Towards University 4.0 : A Model Driven Engineering Method to Design Educational Cyber Physical Systems

Prepared by

Samia Bachir

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy
in Computer Science

Examination Committee :

Pr. Hernan Astudillo, Universidad Técnica Federico Santa Maria (Reviewer)

Pr. Sébastien Iksal, Le Mans Université (Reviewer)

Pr. Agnès Front, Université de Grenoble Alpes (Examiner)

Pr. Anne Lehmans, Université de Bordeaux (Examiner)

Pr. Philippe Aniorté, Université de Pau et des Pays de l'Adour (Advisor)

Dr. Laurent Gallon, Université de Pau et des Pays de l'Adour (Co-Advisor)

January 5, 2022

To my Mother who never saw this adventure ...

Acknowledgement

Je suis une passionnée de montagne et si on me demande de décrire le parcours d'une thèse en un mot, j'aurai répondu il s'agit d'une ascension . En effet, on perd le souffle un peu en fin de chemin mais tout se joue dans le mental. Aujourd'hui, je peux dire, sans prétention aucune, que je suis au sommet de la montagne et je regarde avec beaucoup d'admiration tout ce chemin parcouru dont je suis fière. Ce chemin qui parfois était un peu long, difficile, mais plein de beaux challenges. Durant ces dernières années, j'ai beaucoup appris sur le métier de recherche et enseignement mais j'ai appris aussi à me reconnaître et à m'identifier au milieu de toute cette biodiversité. Je crois qu'avoir une thèse ne se limite pas à avoir uniquement un diplôme. Bien au contraire, c'est une très belle aventure durant laquelle on découvre plein de belles choses et surtout, on se découvre.

Maintenant j'ai envie de dire un Grand Merci à certaines personnes. La liste est vraiment longue. Car je ressens une immense gratitude au fond de moi et j'ai envie de dire Merci à chaque personne qui a croisé mon chemin. Je sais qu'il y a certaines personnes qui je ne remercierais jamais assez. Elles se reconnaîtront, surtout celles qui n'attendront pas un Merci de ma part. Je tiens d'abord à exprimer ma sincère gratitude aux membres de mon Jury pour avoir bien voulu participer à l'évaluation de mes travaux de thèse. Je remercie particulièrement mes deux rapporteurs Pr. Sébasiten Iksal et Pr. Hernan Astudillo pour leurs évaluations fortement appréciées. Je vais essayer de traduire pour Pr Hernan, Pr Hernan, Thank you so much for your feedback and time. I am looking forward to meeting you in person and exchanging with you. Merci à Pr. Anne et Pr. Agnès pour toutes leurs remarques. Je tiens à remercier CD40 pour avoir financé les trois premières années de ma thèse et avoir participé à plusieurs échanges organisés à l'IUT avec les autres doctorants. Je remercie Le Mans université de m'avoir accepté en tant que ATER en étant doctorante depuis Septembre 2020.

Maintenant, je remercie très sincèrement mes directeurs de thèse sans qui je ne serais pas arrivée ici aujourd'hui. C'était vraiment un plaisir de travailler avec vous deux. Philippe, Merci d'avoir accepté la co-direction de ma thèse. Merci pour ta grande confiance dans chaque étape, Merci pour ton temps, que je sais précieux, merci pour tous les conseils et retours, Merci d'avoir cru en moi, Merci du fond du cœur. Laurent, en préparant ces mots, j'ai eu les larmes aux yeux car je sais que je ne te remercierais jamais assez. Je ne sais pas quoi te dire et par où commencer. Merci de m'avoir éclairée quand je ne voyais pas bien les choses. Merci d'être toujours à côté pour écouter mes doutes. Je me rappellerais toujours du jour où tu m'avais dit : je ne suis pas ton directeur,

nous sommes des collaborateurs ! Merci pour la grande confiance et Merci d'être le leader dont tout le monde rêve de travailler avec. Philippe et Laurent, Merci du fond du cœur. Je remercie également Angel Abenia, chercheur associé au LIUPPA avec qui nous n'avons eu que des échanges constructifs. Merci à Ernesto Exposito, professeur des universités pour tous les retours et échanges constructifs que nous avons eus avec lui. Merci aussi à tous les membres chercheurs et personnels de mon laboratoire LIUPPA. Merci à tous les enseignants, enseignants-chercheurs et personnels de l'IUT de Mont de Marsan. Vous êtes vraiment formidables. Un grand Merci à chacune et chacun d'entre vous pour votre bienveillance et amitié. Merci également à toute l'équipe de mon école doctorale ainsi que tous les formateurs.

Merci à mes stagiaires Valentin, Mathieu et Mattéo !

Merci à Anne-Marie d'avoir corrigé mon anglais !

Maintenant que je suis Lavalloise temporairement, je souhaite remercier du fond du cœur les deux équipes enseignantes et de recherche à l'IUT de Laval. Merci de m'avoir fourni les conditions favorables pour pouvoir finir ma thèse en étant ATER. Je tiens à remercier spécialement Sébastien Iksal, en tant que collègue maintenant, et Rémi Venant pour tous les supports de cours qu'ils m'ont fournis au début de mon contrat, de leur confiance et de leur bienveillance. Merci à Pierre Laforcade d'être présent avec nous aujourd'hui, Merci pour les discussions riches que j'ai eues avec toi lors d'un de mes CST. Je souhaite également remercier spécialement les personnels de l'IUT de Laval ; Thierry Amiard et Nathalie Marchand, Neolle Trottier pour leur gentillesse, aide et assistance dans toutes les démarches administratives. Merci à tous mes amis Lavallois, j'ai ici une pensée particulière pour les doctorants ! Merci également à tous mes étudiants avec qui j'ai beaucoup appris et avec qui j'apprends toujours. Merci du fond du cœur à mes deux familles professionnelles montoise et lavalloise. Quelle chance d'être à la fois landaise et mayennaise à la fin de ma thèse.

Maintenant, je passe dire Merci à mes amis et ma grande famille. Je tiens à remercier tous les doctorants et docteurs montois sans exception. Certains parmi eux sont devenu amis proches voire très proches, d'autres des firemen/women. Un grand Merci particulièrement à Jamal, Timothée et Morandise pour leurs vraies amitiés et grand cœur ! Merci d'être vous-même ! Jamal, Merci pour toute ton énergie positive, tous les conseils et les bons moments que nous avons passés ensemble. Merci d'être la plus proche de tous les temps. Merci pour bien beaucoup d'autres choses ! Timothée, Merci pour ton grand cœur, tous les débats philosophiques que j'ai eus avec toi dès mon arrivée ! Merci d'être à côté quand j'avais besoin d'aller aux urgences à 1h du matin, Merci de m'avoir

déménagé à Laval. Je sais que tu reviendras pour me voir ! Morandise, Merci pour toutes les grandes et petites attentions. Je n'oublierai jamais mon trentième anniversaire. Merci pour tous les efforts que tu fais pour animer la vie doctorale à Mont de Marsan. Je sais que tu reviendras toi aussi me voir à Laval. Merci à tous les trois pour votre amitié sincère. Je ne peux pas citer tout le monde mais Merci à vous tous vraiment, pour tous les bons moments que j'ai passés avec vous, Thomas ; Merci aussi pour les débats philosophiques, Charlotte ; Merci pour ta délicatesse, Hamed, William, Tracy, Wadah, Mamadou, Nouha, Raphaëlle, Lea et tous les autres nouveaux doctorants. Je vous souhaite un très bon courage dans votre parcours. Je souhaite remercier aussi tous mes amis randonneurs avec qui j'ai eu la chance d'admirer les Pyrénées Atlantiques durant ces dernières années. Je pense particulièrement à Etienne, Gisèle et Marianne ainsi que tous les membres de la section randonnée de l'ASPTT montoise. Grand Merci également à mes voisins de la rue de côteau à Mont de Marsan qui m'ont offert de bons moments de détente.

Je finis par remercier ma famille de cœur Française ; Guyto et Jaja, alors là que dire ! Merci pour tout ! Vous savez très bien que je ne vous remercierai jamais autant ! Jaja, Je sais que tu n'attends pas de Merci mais Merci pour ta douceur et suivi continu ! Merci pour tous les plats que tu m'as préparés et continuera à le faire malgré les contestations de Guyto. Merci d'avoir corrigé toutes mes lettres de motivation et autres. Guyto, Merci pour ta joie malgré la douleur. Merci pour ton humour, Merci de m'avoir aidé à améliorer mon français également ! Merci à tous les deux pour votre sensibilité, affection et grandeur d'âme !

Grand Merci à ma famille en Tunisie ! Mon papa, mes sœurs, mes frères, vous savez très bien que sans vous, je ne serais pas arrivée où je suis aujourd'hui ! Vous êtes ma force motrice et tout mon bonheur. Merci à ma grande famille et à tous mes amis en Tunisie et partout dans le monde. Merci à toutes les personnes présentes virtuellement avec moi aujourd'hui. Merci à tous d'avoir partagé ce moment virtuel mais vrai avec moi.

Je finis par dédier cette thèse à deux personnes qui ne sont pas sur la terre et qui sont parties très tôt de ma vie : ma maman qui aucun mot ne peut guère exprimer quoi que ce soit ! La deuxième personne c'est mon premier codirecteur de thèse Vanéa Chiprianov pour qui je ressens une immense gratitude. Ces deux personnes me manquent beaucoup mais je sais qu'ils sont là quelque part, peut-être parmi nous aujourd'hui.

Virtuellement, Le 05/01/2022, Samia.

Contents

GENERAL INTRODUCTION	18
Context and Problem Statement	18
Industry 4.0	20
Autonomic Computing	21
Technology Enhanced Learning Environments	22
Objectives and Research Questions	24
Contributions	24
Thesis Structure	25
I RELATED WORKS	27
Part I Introduction	28
1 Connected Environment Modeling	29
1.1 Internet of Things (IoT)	30
1.1.1 Definition	30
1.1.2 Existing Approaches for Modeling Internet of Things	31
IoT Architecture Reference Model	32
Opportunistic IoT Service	33
SysML4IoT	34
1.1.3 Discussion	35
1.2 Cyber Physical Systems (CPS)	36
1.2.1 Definition	36

1.2.2	Existing Approaches for Modeling Cyber Physical Systems	38
	CAPS	38
	Digital Twins and CPS	39
	A Feature-Based Ontology for CPS	39
	CPS framework	39
1.2.3	Conclusion	40
2	Educational Modeling Languages (EML)	41
2.1	Introduction	41
2.2	Existing approaches for Modeling Educational Environments	42
2.2.1	CoUML	44
2.2.2	E2ML	44
2.2.3	ColeML	45
2.2.4	LDL	45
2.2.5	PoEML	45
2.2.6	SCORM	46
2.2.7	LOM	46
2.2.8	IMS-LD	47
2.2.9	IMS-QTI	48
2.3	Discussion	48
2.4	Existing approaches for Educational and Connected Environments	49
2.5	Conclusion	52
	Part I Conclusion	54
II	TOWARDS UNIVERSITY 4.0 : Educational Cyber Physical Systems (ECPS)	57
	Part II Introduction	58

3	Educational Cyber Physical Systems Architecture	59
3.1	Software Architecture	60
3.2	ECPS Definition	61
3.3	ECPS Generic Architecture	63
3.4	Different Levels of ECPS	66
3.4.1	Classroom ECPS	66
3.4.2	Course ECPS	68
3.4.3	Curriculum ECPS	69
3.5	ECPS Coordination	70
4	ECPS Modeling	72
4.1	Model Driven Engineering (MDE)	72
4.1.1	Domain Specific Languages (DSL)	75
4.1.2	Model Transformations	76
4.2	ECPS Design Method (ModelECPS)	77
4.3	Eclipse Modeling Framework (EMF)	80
4.4	Profile Extension Mechanism	81
4.5	Cyber Physical Systems Modeling Language (CPSML)	82
4.5.1	SysML	82
4.5.2	CPSML Meta-Model	84
4.5.3	CPSML Concrete Syntax	87
4.5.4	CPSML Modeling Tool	88
4.6	Educational Modeling Language (EML4.0)	89
4.6.1	IMS-LD	90
4.6.2	EML4.0 Meta-Model	90
4.6.3	EML4.0 Modeling Tool	93
4.7	Educational Cyber Physical Systems Modeling Language (ECPSML)	93
4.7.1	ECPSML Meta-Model	94
4.7.2	ECPSML Modeling Tool	99

4.7.3	ModelECPS - Modeling Process	100
Part II	Conclusion	102
III	VALIDATION OF OUR APPROACH - Towards University	
4.0		103
5	Case study - Covid-19 Educational Context	104
Part III	Introduction	104
5.1	Connected Aspect Modeling (CPSML Model)	107
5.2	Pedagogical Aspect Modeling (EML4.0 Model)	110
5.3	ECPS Model	114
5.4	ECPS on Run-Time	118
5.4.1	Learning Service Implementation	118
5.4.2	Learning Scenario Implementation	122
5.5	Case Study Limitations	126
Part III	Conclusion	128
GENERAL CONCLUSION		130
Fulfilling the Objectives and Contributions		130
Perspectives		133
Short-term Perspectives		133
Long-term Perspectives		134
List of publications		136

List of Figures

1	Autonomic Computing Control Loop Process from [41]	21
1.1	IoT ARM Domain Model from [24]	32
1.2	IoT Service Model from [52]	34
1.3	CPS Development Timeline from [90]	37
2.1	IMS LD from [63]	47
2.2	Learning Approaches from [115]	50
2.3	The Main Components of Cyber Physical Systems (revised from [45] and [14])	55
3.1	The Generic Layered Architecture of Educational Cyber Physical Systems	63
3.2	Educational Cyber Physical Systems in a Classroom Process	67
3.3	Educational Cyber Physical Systems in a Course Process	68
3.4	Educational Cyber Physical Systems in a Curriculum Process	69
3.5	Educational Cyber Physical Systems Coordination Between the Different Learning Processes	71
4.1	The World and The Machine from [26]	73
4.2	Modeling and (Meta)MetaModeling from [26]	74
4.3	Basic Concepts of Model Transformation from [43]	76
4.4	Modeling Process for the ModelECPS method	79
4.5	Ecore Model from [10]	80
4.6	Profile Definition Mechanisms from [11]	82
4.7	SysML Taxonomy from [9]	83

4.8	CPS Modeling Language	85
4.9	Part of the SysML Concrete Syntax	87
4.10	CPS Modeling Language Implemented in Papyrus as a SysML Profile	88
4.11	CPSML Stereotypes Integration in Papyrus	89
4.12	EML 4.0 MetaModel Implemented in EMF based on its Ecore MetaMetaModel	91
4.13	ECPS Modeling Language Designed as a SysML Profile	94
4.14	Part of the ATL Transformation Rules	97
4.15	ModelECPS Transformation Process	98
4.16	ECPS Modeling Language Implemented in Papyrus as a SysML Profile	99
4.17	ModelECPS Modeling Process	101
5.1	Infrastructure of the ToIP Course	106
5.2	Example of Task Sharing Between the Remote and Present Students (in French)	107
5.3	The Block Diagram of the CPSML Model (confirms to CPSML) - Object (yellow) - Service (green) - Human (red) - Data (grey) - SysML block (white)	108
5.4	A CPSML Activity Diagram for a MAPE Control Loop	110
5.5	Manual Draft of the Learning Scenario	111
5.6	Modeled Learning Activities, Control Points, Control Flows and Events (con- firms to EML4.0)	112
5.7	The Modeled Learning Environment Through Learning Objects, Learning Ser- vices and TELS (confirms to EML4.0)	112
5.8	Automatic Generation of a Part of the ECPSML Model Through the Trans- formation Rules given in Figure 4.14	114
5.9	The ECPS Block Diagram Generated Directly From Papyrus Corresponding to the Defined Concept Alignment	115
5.10	The ECPS Activity Diagram Generated Directly From Papyrus Corresponding to the Defined Concept Alignment	117
5.11	Example of a Discord Log (Start Session Log)	119
5.12	Discord Metrics	121

5.13 All Collected Educational Logs from the Asterisk Server	122
5.14 Generated Students' Logs Format	123
5.15 Example of a Python Code to Check the Validation of a Learning Activity .	124
5.16 Activities Validation and Their Corresponding Servers	125
5.17 Indicators' Validation and Their Corresponding Servers	125
5.18 Number of the Validated Indicators	126

List of Tables

- 1.1 Comparative Study Between IoT Modeling Approaches 35
- 2.1 Positioning of EML Regarding University 4.0 characteristics 49
- 4.1 Concept Alignment Between the Different DSL (EML4.0, CPSML, ECPSML) 96
- 5.1 General Context of the Case Study 105

Abstract

The Internet of Everything (IoE) is gaining a growing interest in scientific and industrial communities to have expanded the Internet of Things (IoT) paradigm. In fact, several domains have been through a digital transformation evolution, for instance Industry 4.0, however, a lot still needs to be done regarding the educational domain especially at university. Instructional design is a major concern among the TEL (Technology Enhanced Learning) community. The covid-19 health crisis has dug deeper into several of its challenges. The objective of Industry 4.0 is to foster the integration of complex physical machinery and devices based on Cyber Physical Systems (CPS) and IoT. A set of sensors, actuators and software components are used to monitor, analyze, and control smart factories through control loops (MAPE). In this thesis, we investigate how these concepts could contribute in shaping and engineering the next revolution at university (University 4.0). To do so, we have conducted an exploratory approach in order to, first give a definition of what we call Educational Cyber Physical Systems (ECPS) to build it upon ideas from the literature. This concept is barely addressed in the TELE domain. We apply this concept and its utilization in different educational levels (Classroom, Course, Curriculum). Then, we propose a model-driven method (ModelECPS) to design these systems founded on a modeling process and three Domain Specific Languages (DSL). The modeling process is issued from the Object Management Group (OMG) standards that define the Y process based on three modeling pillars of the MDE method. The platform Independent Model (PIM) represents the educational meta-model, called EML4.0. The Platform Definition Model (PDM) is the connected environment meta-model, called CPSML, that extends the modeling “things” into modeling “everything”. Finally, the ECPSML defines the meta-model that describes the educational connected environment and thus the Platform Specific Model (PSM). Its design is based on concept alignment between both PDM and PIM models and is semi-automatically generated through applying ATL rules (Model-to-Model transformation). A utilization modeling process for the method is given in order to guide the different stakeholders in designing such systems (ECPS). To illustrate our proposal, we conducted a case study on a real-life ECPS

classroom during the covid-19 health crisis. We describe the required models designed in this process and supervise students' progress during the practical sessions.

Résumé

L'Internet of Everything (IoE), extension du paradigme Internet of Things (IoT), est de plus en plus étudié et utilisé par la communauté scientifique et industrielle. Plusieurs domaines ont connu une évolution notable en matière de transformation numérique grâce à l'intégration de ce dernier, comme par exemple l'Industrie 4.0, mais il reste encore beaucoup à faire dans le domaine de l'éducation, notamment à l'université. La conception pédagogique est une préoccupation majeure de la communauté EIAH (Environnements Informatiques pour l'Apprentissage Humain). La crise sanitaire de la covid-19 a fait ressortir plusieurs défis à cet égard. L'objectif de l'Industrie 4.0 est de permettre l'intégration de machines et de dispositifs physiques complexes (Systèmes Cyber-Physiques (CPS) basés sur le paradigme IoT). Un ensemble de capteurs, actionneurs et composants logiciels sont alors utilisés pour monitorer, analyser et contrôler des usines intelligentes, en appliquant des boucles de contrôles autonomiques (MAPE: Monitor, Analyze, Plan, Execute). Ces boucles de contrôles permettent d'automatiser tout ou partie des processus métiers. Dans cette thèse, nous étudions comment les concepts utilisés dans l'Industrie 4.0 pourraient contribuer à l'ingénierie de la prochaine révolution numérique universitaire (Université 4.0). Pour ce faire, nous menons une approche exploratoire afin de définir ce que nous appelons les Systèmes Cyber Physiques Educatifs (ECPS) en nous appuyant sur des travaux de la littérature. Nous appliquons ce concept à différents niveaux d'enseignement (séance pédagogique, module, curriculum). Ensuite, nous proposons une méthode de conception de ces systèmes basée sur un processus de modélisation (ModelECPS) et trois Langages de modélisation Spécifiques au Domaine (DSL). La modélisation suit un processus en Y, issu des normes de l'Object Management Group (OMG). Elle se base sur trois modèles piliers de l'Ingénierie Dirigée par les Modèles (IDM) : le modèle indépendant de la plateforme (PIM) qui représente le méta-modèle éducatif, appelé EML4.0; le modèle de définition de la plateforme (PDM), qui est le méta-modèle d'environnement connecté (CPSML), et qui étend la modélisation des "Things" à la modélisation de "Everything"; enfin, le modèle ECPSML qui définit le méta-modèle qui décrit l'environnement connecté éducatif et donc la plateforme spécifique au domaine (PSM).

La conception du modèle ECPSML est basée sur un alignement des concepts entre les modèles PDM et PIM et est semi-automatiquement générée par l'application de règles ATL définies pour mettre en place une transformation de modèle à modèle. Pour finir, un processus d'utilisation de la méthode est proposé afin de guider les différentes parties prenantes dans la conception de tels systèmes (ECPS). Pour illustrer notre proposition, nous avons réalisé une étude de cas sur une activité de travaux pratiques pendant la crise sanitaire. Nous décrivons l'ensemble des modèles correspondants à cette activité, ainsi que les boucles MAPE utilisées afin de monitorer et superviser les étudiants pendant la réalisation des travaux pratiques.

GENERAL INTRODUCTION

Context and Problem Statement

The world is changing, and it is changing fast. Technological progress has overwhelmed many sectors of our everyday-lives. We are witnessing an explosion of digital learning systems. Among other research domains, TELE (Technology Enhanced Learning Environments) have attracted increasing interest and attention of researchers, through its multidisciplinary aspect, and becomes an important research topic that tends to specify various sub-domains. In fact, this domain fosters traditional practice of instruction by providing a well defined knowledge and support to the learners. It also addresses the openness of education and life-long learning through various systems sometimes without the inter-mediation of a teacher.

We are particularly interested in addressing educational challenges at university where both drop out and failure rates are importantly registered [20]. This is due to various reasons. For instance, massive classrooms sometimes lead to an absence of personalization and adaptation of the educational processes. Moreover, many research efforts have been conducted to develop TELS (Technology Enhanced Learning Systems) that are specialized in addressing a particular pedagogy or in developing particular competencies. However, these efforts are carried out in an ad-hoc manner with no regards to a reference architecture that could gather their heterogeneities and objectives in order to build a unified, well-founded and adaptable educational environment.

In the literature, teaching and learning are defined as knowledge work. With the emergence of smart things and services, the educational processes shape knowledge transfers between connected humans using services, physical objects that together generate a massive amount of data. In general, research on connected environments often focuses on specific domains which may result in a development gap between different domains. For instance, interest in Industry 4.0 exists because of the mass consumption society we live in. For that reason, the industrial field has expeditiously moved through several revolutions to attend Industry 4.0 [61].

From a historical perspective, three industrial generations preceded this revolution (Industry 4.0). From the introduction of mechanical production facilities (Industry 1.0), to

the electrification and the division of labor (Industry 2.0), to the automation of production processes (Industry 3.0), throughout the centuries, the idea of smart manufacturing appears to deliver fundamental improvements to the industrial processes involved in manufacturing, engineering, material usage, supply chain and life cycle management ([48], [61]).

Accordingly and from a historical perspective as well (described in [53] and [57]), education and thus the university has been through different revolutionary phases. It is estimated that University 1.0 refers to the fact that the educational system was limited to privileged people, associated often with religious movements in former eras. In a second stage, the university had become publicly institutional and available to everyone. Then, with the technological progress, the university has reached a third stage, taking an important place in the digitization of educational systems due to the emergence of TELS (e.g. Moodle, MOOCs, ITS, etc.) after the expansion of the electronic learning, since the beginning of the twenty-first century.

Moreover, the evolution of information technology has revolutionised the way people and objects behave in everyday lives due to the emergence of Internet of Things, especially in the last decade, qualified as the fourth wave in [15]. This was preceded by three other phases. The first was categorized by the arrival of computers in the late 1960s where one computer was allocated to several people. Then, a second wave raised after a decade which refers to single-person use of a computer. After the appearance of embedded and ubiquitous computing, a third wave occurred to designate many computers to one person.

In this thesis, we drive an exploratory approach in order to define the next generation of university by studying what would be behind the "University 4.0", build upon ideas from elements of our background in order to propose our vision and the related concepts. Our purpose is not to propose an educational system in itself. The aim is rather to define an environment where different TELS could be used to achieve learning objectives. The use of connected resources in learning and teaching generates a highly amount of data. Thus, control over these data would allow a better monitoring of the educational processes. Regarding this, we present in the following the theoretical background (Industry 4.0, Autonomic Computing, and TELE) that has lead us into this exploratory approach. For each concept, we retain

some key elements or principles regarding our proposals.

Industry 4.0

Industry 4.0 was first raised among the German industrial community in 2011. It is a complex initiative that embraces several overlapping areas [67]. It was defined by Hermann et al. [61] as the integration of complex physical machinery and devices with networked sensors and software, used to predict, control and plan for better business and societal outcomes. In such manufacturing environment, smart machines, installations, work pieces and other components exchange data and information in real time which represents a shift from rigid, centralised factory control systems to decentralised intelligence. This revolution is founded on a set of technologies which together promote the fourth revolution, which is currently happening in our world. Internet of Things (IoT), Cyber Physical Systems (CPS), Cloud Computing, etc. are the basis of Industry 4.0. New business models and new models for cooperation constitute the real added value of Industry 4.0 [46].

Hermann et al. [61] identified four design principles of Industry 4.0: (1) An interconnection aspect, Internet of Everything, to support collaboration between and within different stakeholders (human-human, human-machine, and machine-machine [117]). (2) Information transparency will be insured because a virtual copy of the physical world is regularly generated. (3) Decentralised decisions are enabled by CPS which allow the monitoring and control of the physical world autonomously through their embedded computers, sensors and actuators [77]. (4) Humans are supported with a technical assistance in order to intervene when needed.

CPS are among the key elements in the field of Industry 4.0. According to [61], these are systems that create a fusion between virtual and physical worlds. They monitor and control the physical processes. A 5C architecture was proposed in [78] where authors specify two main functional components of CPS. The first one consists of the advanced connectivity that ensures real-time data acquisition from the physical world and information feedback from the cyber space. The second functionality resides in intelligent data management, analytics and computational capability that together construct the cyber space. Various features were

identified in [56] in order to make the difference between CPS and conventional systems which basically underline the time-sensitive combination of the cyber and the physical, and their connected-ness in CPS.

From Industry 4.0, we retain that among the key elements behind the "4.0", there are technological evolution related particularly to the IoT (Internet of Things) and CPS (Cyber Physical Systems) in a physical infrastructure level. These two technologies are gaining increasing attention and have overwhelmed several domains besides. Many related definitions and architecting attempts are presented in this thesis in order to study their contributions at an educational level.

Autonomic Computing

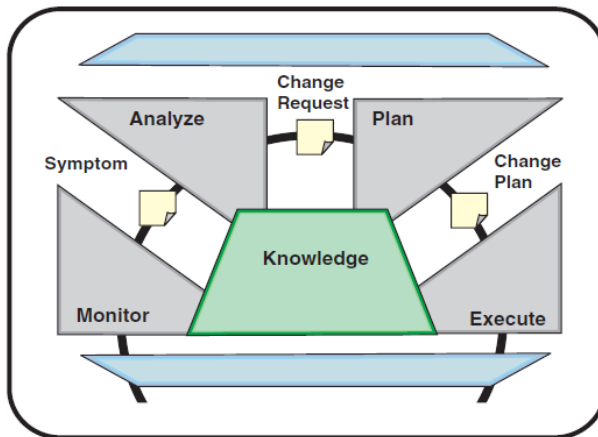


Figure 1: Autonomic Computing Control Loop Process from [41]

Inspired by the autonomic nervous system, IBM researchers [41] are often acknowledged with initiating the Autonomic Computing (AC) blueprint. As defined in [94], AC is considered as a holistic vision of self-managing capabilities in a system. These capabilities cover different aspects of self-management implemented by MAPE control loops (Monitor, Analyse, Plan and Execute), as shown in Figure 1, which collect details from the system and act accordingly, as stated in [41]. Self-configuring AC capacity allows the dynamic adaptation to changes in the environment. The self-healing capacity ensures the performance of a system by detecting the system malfunction. The automated reallocation of resources is

handled with a self-optimising capability. Security issues are supported by a self-protecting capability. These capabilities are implemented through different autonomic managers.

In the reference architecture of the IBM blueprint [41], it is stated that different layers manage data through the knowledge source. Standardisation efforts are still challenging regarding data management. Control loops could be incorporated within a particular managed resource in order to allow the execution of self-managing capacities (self-configuring, self-healing, self-optimising, self-protecting). The integration of autonomic capacities within Cyber Physical Systems emerged through an intelligent aspect with feedback loops where physical processes affect computations and vice versa.

From the AC perspective, we retain the MAPE process that could be incorporated during an educational process (e.g. real classroom session) or even within a particular learning system in order to investigate the teaching and learning processes and act accordingly to students' behaviors. The smart capacity to adapt learning and teaching processes relies on a highly generated data from the connected educational environment in a given learning and teaching context.

Technology Enhanced Learning Environments

As stated above, TELE could be seen as the key element of University 3.0. A clear and complete consensual definition of Technology Enhanced Learning (TEL) has not yet been reached. As noted in the literature ([29], [50]), TEL is an interdisciplinary field involving various domains of knowledge, such as Computer Science (CS), Artificial Intelligence (AI), Human Machine interface (HMI), Cognitive science, Didactics, Pedagogy, and Information and Communication Technology (ICT). One possible way to define the TEL domain is the study of how we learn and teach with interactive technologies, and how to design and evaluate effective technologies for learning [50]. To address the manner of conducting such a study, we identified three complementary manners that could be used to categorize it.

The first manner concerns learning theories. In fact, a learning theory is meant to explain and help us understand how people learn [100]. As stated by Picciano [100], it also involves multiple disciplines, such as psychology, sociology, neuroscience, and education.

The following learning theories are among the most used ([55], [86]): (1) The association-ist/empiricist/behaviorist theories which commonly defines learning as an activity where knowledge is defined as a collection or sequence of skill-components and their associations. (2) The constructivist theory defines learning as achieving understanding. Learners develop their understanding by building upon their existing knowledge and previous experiences [99]. The context of learning also affects students' construction of knowledge. Being actively involved in the learning process, the learner becomes responsible of his/her learning. It is unlike previous learning theories (e.g. Behaviorism) where the responsibility rests with the instructor to teach with the learner playing a receptive role. The social constructivist theory defines learning as a social practice.

The second manner to organize TEL concerns the categorization of TEL environments as computational tools. A wide range of computer software has been developed and has emerged since 1960, with Computer Assisted Instruction systems. Among the best known and widely used categories of environments in TEL research, there are Microworlds, Intelligent Tutoring Systems (ITS), Online learning (Moodle, MOOCs, etc), and Computer Supported Collaborative Learning (CSCL). The design and implementation of a TEL environment is usually influenced by one or more of the above listed learning theories.

A third categorisation manner relies on five different perspectives (computational, cognitive, pedagogical, social and epistemological), identified in [29], that may be considered in TELE. The computational perspective establishes a link between TEL and Computer Science to answer questions such as: how can the presence of technology change the content itself to be learnt ? According to the cognitive perspective, questions should be investigated on which cognitive abilities new technologies may be required, fostered or hindered ? From a pedagogical perspective, one has to consider the design and co-evolution of new learning activities. The social and cultural perspective stresses the environmental context that influences learning. The epistemological perspective focuses on how the characteristics of a knowledge domain impact the design and use of learning technologies.

According to this categorization manners of how TEL could be seen, our proposal is not to focus on a specific manner of categorization, or a particular learning system. We rather aim

to federate all these contributions (systems, theories and perspectives) on a single environment and couple this with other connected resources, despite the challenge of data heterogeneity and pedagogical variety.

Objectives and Research Questions

After exploring and reflecting on our background elements, we propose an investigation of University 4.0 based on the retained key concepts and principles. We propose Educational Cyber Physical Systems (ECPS) as a core concept for it. ECPS are defined to be based on IoT/CPS concepts (from Industry 4.0). They implement MAPE control loops (from AC perspective) and analytics to assist administrators in taking the required actions. Furthermore, they could rely on various learning systems (from TELE) and aim to achieve educational purposes in a more flexible way.

To do so, this thesis work addresses the following main research questions :

- **RQ1** : How to define ECPS as an innovative concept, with a particular focus on the different educational processes at University ?
- **RQ2** : What is the required method to design ECPS by considering the evolution of both connected and educational aspects ?

Contributions

Based on an exploratory approach, this thesis gives, firstly, a vision of University 4.0 based on elements from the background. We introduce what we call Educational Cyber Physical Systems (ECPS) as the core concept for such smart environment. We define them as educational complex systems constituted of physical and cyber components that aim at implementing control loops to manage the educational processes according to educational stakeholders' requirements. As we focus on connected environment, we rely on IoE as an extended paradigm from IoT that defines ECPS components (Things, Services, Objects, Humans). Moreover,

we propose an ECPS generic layered architecture that considers the latter as the system infrastructure layer. The different layers aim at implementing control loops by managing data at a cyber level and finishing by taking actions in the physical level. This layered architecture could then be executed in three kinds of educational processes at university : classroom ECPS for short-term processes, course ECPS for mid-term processes, and curriculum ECPS for long-term processes.

Secondly, this thesis proposes a Model-Driven Engineering method called **ModelECPS** for designing these systems. ModelECPS is founded on a modeling process and three Domain Specific Languages (DSL). The modeling process is issued from the Object Management Group (OMG) standards that define the Y process based on three modeling pillars of the MDE method. The platform Independent Model (PIM) represents the educational meta-model, called EML4.0. The Platform Definition Model (PDM) is the connected environment meta-model, called CPSML, that extends modeling “things” into modeling “everything”. Finally, the ECPSML defines the meta-model that describes the educational connected environment and thus the Platform Specific Model (PSM). Its design is based on a concept alignment process between both PDM and PIM models and is semi-automatically generated through applying ATL rules (Model-to-Model transformation). A utilization modeling process is given in order to guide the different stakeholders in designing such systems (ECPS) regarding both connected and educational aspects.

Thesis Structure

This thesis manuscript is composed of three main parts. The first part investigates the related works. The second part presents the contributions regarding the definition and modeling of ECPS. The third part describes a case study in order to give a concrete idea of both the modeling and run-time implementation phases of ECPS.

In the first part, the first chapter presents our related works studies of the existing approaches that address the modeling of IoT and CPS architecture, both followed by comparative studies. This allowed us to draw the borders between the different domains, to get the

essence of the subjects and demystify the related concepts.

In the second chapter, we study Educational Modeling Languages (EML) and the related design perspectives. Modeling learning is viewed either as structural knowledge modeling, or through instructional process modeling. Comparative criteria are presented regarding the characteristics that we have identified for University 4.0. This first part is concluded by gathering elements from both chapters to present the main components of Educational Cyber Physical Systems on which we rely to elaborate our contributions.

In the second part, we present our contributions through, first, a generic architecture of ECPS and their definition (third chapter), and then a model-driven method, called ModelECPS, to design these systems, in chapter four. To do so, we detail the required process, the different modeling languages and the model-transformation rules that are used to generate the output model of ECPS based on two input models (one for the connected aspect and another for the domain model aspect, educational in our case).

The third part Validates our proposal in a real online environment by conducting pilot studies. We tested our design method in order to explore ModelECPS modeling process through the proposed modeling tools and illustrate different layers of the proposed generic architecture based on a case study experimentation during the covid-19 health crisis. We conclude this chapter by a discussion of our case study limitations. Finally, a general conclusion and a list of short-term/long-term perspectives are given. The list of our publications will be also provided.

Part I

RELATED WORKS

Part I Introduction

When it comes to addressing research questions in multidisciplinary complex systems, borders should be carefully drawn between the different domains to get the essence of the subjects and to demystify the related concepts. This thesis addresses connected environments on the one hand, and educational environments on the other, as we intend to benefit from the contributions of connected-ness in an educational context. Currently, billions of devices are connected everywhere and this is overwhelming various domains. Internet of Things and Cyber Physical Systems have drawn increasing attention since connectivity has become associated with various domains in our everyday lives. However, research efforts addressing IoT/CPS (and their architecting) in educational environments are still in their infancy. For this purpose, this thesis is conducted through an exploratory approach with the aim to give a vision of the university of the future in an entirely connected context. It is worth exploring the multidisciplinary aspect of the domain in order to depict the main elements of the related works regarding our proposal. Hence, we examine this with a particular interest in defining and modeling connected and educational environments.

The contributions of this first part are organized as follows: In chapter I, we review existing approaches in modeling Internet of Things and then Cyber Physical Systems. In chapter 2, a list of modeling educational languages is given. Each field is discussed through a comparative study and/or a discussion section in order to highlight the elements of the literature on which we rely on. We also present related works that cover existing approaches of IoT in education.

Chapter 1

Connected Environment Modeling

In the interest of clear-sightedness, this chapter represents our conducted review regarding the singularity aspect of the connected environment technologies. The main objective of this first part is to present the different concepts related to connected environments. We start by reporting the different definitions and existing approaches that jointly or partially address Internet of Things (IoT) and Cyber Physical Systems (CPS), their definitions and the related architecture modeling.

It is important to mention that the publications presented and analyzed in this related work chapter were extracted from several bibliographic databases, mainly: Scopus, IEEE Xplore³, ACM digital library⁴, and Springer. This extraction was done in a structured way by using appropriate keywords related to our topics of interest - in addition to several, inclusion and exclusion criteria to select relevant studies, however this study is not a complete Systematic Literature Review.

In the literature, one may find several definitions of IoT and CPS to such a degree that they could be considered as a look-alike concepts, however this is not the case. Nonetheless, they are technically highly linked. More notably, it was stated in [58] that CPS can be used as a synonym of IoT since both terms have recently been mentioned coincidentally. In this chapter, we look into both definitions and give a review of their proposed models in Software Architecture (SA).

1.1 Internet of Things (IoT)

1.1.1 Definition

Changes in technology are in constant evolution since the beginning of the current century especially with the emergence of the Internet. More affordable solutions ensuring pervasiveness are deployed through smart devices. This has resulted in the rise of Internet of Things (IoT) also called Internet of Objects [89] where billions of objects are inter-networked to the Internet through various types of wireless technologies (RFID, ZigBee, Sensor networks, etc.).

Back to 1999, the term "Internet of Things", was first used by Kevin Ashton to describe a system where the Internet is connected to the physical world via ubiquitous sensors. Since then, the research community has started exploring this concern. Various definitions of IoT can be found in [92]. For each definition, a specific point of view or perspective is considered. In fact, some research works (e.g. [68], [125]) highlight the *machine-to-machine interactions* when qualifying this paradigm. Others focus on *the real-time collaboration and interaction* [120].

More generic definitions are also given in [110] where the authors consider IoT as *a worldwide network of interconnected entities*. In [95], it is seen as *the ability to connect, communicate with, and remotely manage an incalculable number of networked and automated devices*. It is transforming our physical world into a giant information system [52].

From another view point, IoT is considered as *the internal/external communication of intelligent components via the Internet in order to improve the environment through providing smarter services* [92]. Also, a definition given in [96] characterizes IoT as *an ecosystem that interconnects physical objects with telecommunication networks, joining the real world with the cyber space and enabling the development of new kinds of services and applications*. We notice that the latter represents IoT as Cyber Physical Systems, as it will be presented in section 1.2.

Many buzzwords, after the IoT revolution, have arisen, for example, Internet of Every-

thing, Internet of Nano-Things, Web of Things, etc. In the following, we only study existing approaches for modeling IoT, as modeling interests are not developed, especially regarding Internet of Everything where we not only deal with things, but also with humans, services and data [14]. We will come back to this concept later (at the end of this section) to which we take a particular interest.

1.1.2 Existing Approaches for Modeling Internet of Things

In the literature, one may find several proposals regarding IoT application development. For instance, OpenHAB [7], a general-purpose framework based on OSCi modules, allows building smart home solutions as the most common application area of IoT. DSL-4-IoT [114], editor and designer, is based on a high visual programming language and particularly focuses on the networking heterogeneity issue of the IoT.

IoT application development is considered as a complex task as it requires a wide range of expertise [101]. Hence, its modeling seems crucial in order to reduce this complexity. In recent years, diverse studies in the field of IoT system architecture modeling were published. Each study has addressed a specific view of the IoT system, in a similar way to the IoT definition.

DiaSuite [25] provides a design language that is oriented to handle the Sense/Compute/Control (SCC) applications. PervML [80] is a Domain Specific Language that provides a set of constructs to describe system pervasiveness in a technological independent way. It also addresses the design of embedded technologies. Another attempt for a model driven software development workflow was developed in [107] where the authors suggest a code generation module based on a platform independent model which specifies how messages can be consumed by the application (message model) in order to overcome the technological gap between networked embedded objects and enterprise back-end systems.

The LAURA architecture [123], a service-oriented architecture, facilitates the development of IoT systems in order for BPM experts, domain experts, and other high-level users, to interact with the system. It provides automatic code generation from BPMN specifica-

tions. The aim is to free the developer of business models from the task of knowing details of lower level programming aspects.

In the following, we provide the main related works in the scope of IoT that are particularly aligned with our work. We then dress a comparative study between these to highlight our findings.

IoT Architecture Reference Model

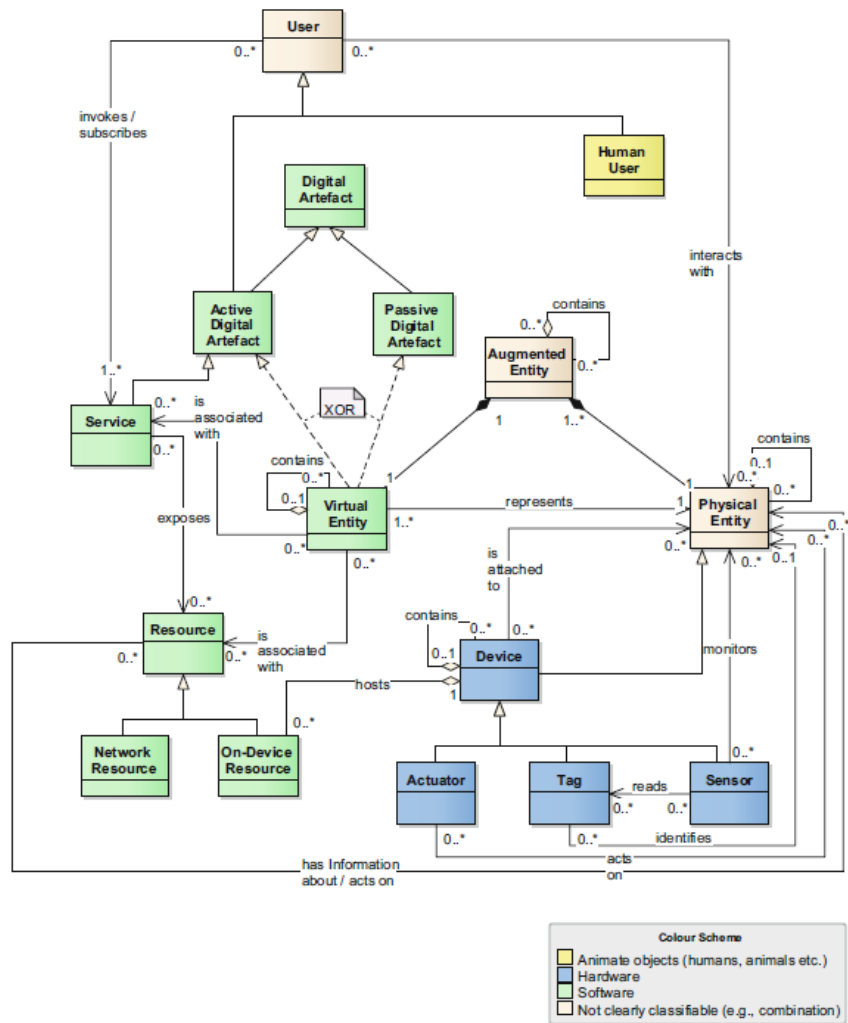


Figure 1.1: IoT ARM Domain Model from [24]

As its name implies, the IoT ARM (IoT Architecture Reference Model) is a reference

model for IoT systems [24]. It provides a domain model highlighting the main concepts and definitions for IoT architecture. IoT ARM could be considered as a mature and well-defined reference model that provides a common structure and guidelines to deal with the core aspects of developing, using and analysing IoT systems. However, the emphasis is laid on structuring services and modeling things (devices). Human and data modeling are gaining less interest here.

As shown in Figure 1.1, IoT ARM represents a connected system by a set of augmented entities. Each augmented entity is composed of a physical entity (device) and a virtual entity (service). Despite the fact that IoT ARM provides a domain-specific model for IoT as well as its acknowledgement by the scientific community, it does include very generic concepts and relationships which could be challenging to use for non-professionals. Many research works have adopted IoT ARM, such as, IoTLink [101], an IoT prototyping Toolkit based on this proposal to enable inexperienced developers to implement IoT prototypes rapidly.

Opportunistic IoT Service

The work in [52] focuses on designing IoT service models which, according to the authors, are the actual IoT drivers. The considered opportunistic properties concern the following aspects of an IoT system : dynamicity, context-awareness, co-location, and transience. In this work, the focus is set on the *Service* which is designed as a composition of a service model and a service profile, in the IoT system. The service model specifies the details of attributes and relationships describing the IoT service itself (e.g. name, description, category and parameter of the service itself).

The service profile handles the properties of the process that implements the service (e.g. id, input, output, and precondition of the process). In this work, a service is considered as a set of processes. Each process is an operation that contributes concretely to the implementation of the service. The authors define Petri nets and their components as an operational modeling tool to provide formal verification and simulation. According to the authors, such modeling allows to control the correctness of IoT services among dynamic context changes and their compliance to a given set of specifications.

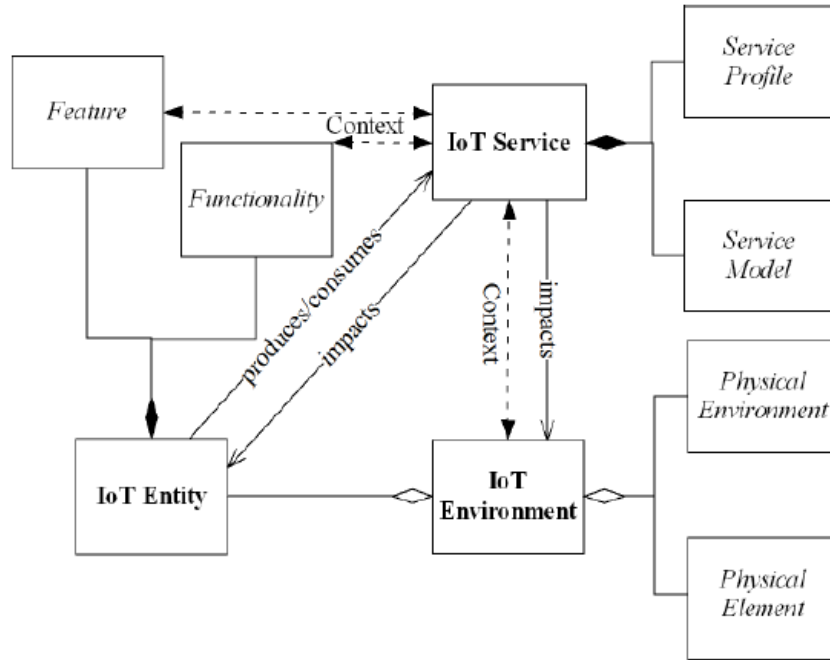


Figure 1.2: IoT Service Model from [52]

SysML4IoT

As stated above, some research works have adopted the ARM reference model in order to design and analyze IoT applications. For instance, a SysML profile, called SysML4IoT, was proposed in [42] to be used by IoT application engineers in order to model an IoT system. The framework also consists of ysML2NuSMV (a model-to-text translator) that converts the model and automatically checks its properties such as availability or performance.

Regarding modeling, SysML4IoT is based on IoT-ARM where the components of the system (hardware and software) and the system itself are considered as stereotypes and thus as specification of the "block" concept of SysML. The authors propose concepts like "system-of-interest" and "enabling system" in order to characterize the implementation of the connected system. This work is extended in [62] to propose a model driven development for adaptive IoT systems.

1.1.3 Discussion

In our work, we consider that IoT is mostly related to "machine-to-machine" interactions. Over the past few years, it gets an expanded dimension of its things. Currently, we deal with Everything. There are not only machine-to-machine interactions but also human-to-machine and human-to-human ones. Regarding IoT modeling in SA, a review of IoT architectural styles in [92] notes that the layered style dominates in the different research works. Cloud-based and service-oriented styles also have an important impact. We believe that data-oriented architecture is not sufficiently developed in the literature. Object-oriented architecture (in the sense of smart-object) could also be considered as a style in IoT architecting. We give a view of the related works that we consider useful with regards to our research questions, in Table 1.1, and dress them according to their perspective oriented modeling. As it is shown, none of the studied works consider object, human, data and service design simultaneously. Most of the research works are either focusing on service or object design. In our work, we are interested in a "four-oriented" architecture for connected environments, which we detail later.

	IoT ARM [24]	SysML4IoT [42]	Opportunistic IoT services [52]
Service-oriented Model	✓	✓	✓
Object-oriented Model	✓	✓	N/A
Data-oriented Model	N/A	N/A	N/A
Human-oriented Model	-	-	-

Table 1.1: Comparative Study Between IoT Modeling Approaches

Nevertheless, our proposal is build upon ideas from the IoT ARM regarding particularly object (device) design. Actually, we consider that actuators, tags and sensors are components of the device which is contrary to the inheritance relationship that is used in the studied model. We further discuss this point in the scope of our contribution (Part II). Moreover, interactions are designed as a relation between a Human and a physical entity. This relationship is very generic and does not highlight the different stakeholders of the connected environment.

Despite the fact that IoT ARM provides a domain-specific model for IoT as well as its acknowledgement by the scientific community, it does include very generic concepts and relationships which could be challenging when used by non-professionals. Since it is aligned with our goals, our purpose is to build upon ideas from this reference model and provide a proposal that could be interpreted by non professional users (e.g. educational stakeholders) to ensure its usability.

Various research works are in the pursuit of standardising IoT reference Architecture (e.g. [24], [58], etc). Towards an open architecture, authors in [127] pointed out some design principles (flexibility, customizability, and extensibility) that should be taken into account to control and guarantee the evolution of the system over time. The authors in [17] reviewed proposed architectural structures which show similarities between the proposals, but one of the raised issues was how to manage the high amount of generated data. Thus, modeling data is not very specified in the studied research works.

In this first section of the related works part, regarding connected environments, we examined research works that mainly focus on IoT and considered some issues regarding its architecting. In addition to the IoT, CPS represents another key element of connected environments. In the following, we explore CPS definitions and architecture modeling approaches and provide some insights. Then, based on these two key elements (IoT and CPS), we define our proposed CPS modeling language in Part II building it upon our findings from the discussed related works.

1.2 Cyber Physical Systems (CPS)

1.2.1 Definition

Cyber Physical Systems (CPS) are defined as transformative technologies for managing interconnected systems between its physical assets and computational capabilities [23]. As presented in Figure 1.3, they represent the current technological evolution since 2006. They monitor, analyse and control the physical processes and act accordingly. The purpose is to

interconnect the physical with the digital world. They are smart systems [56].

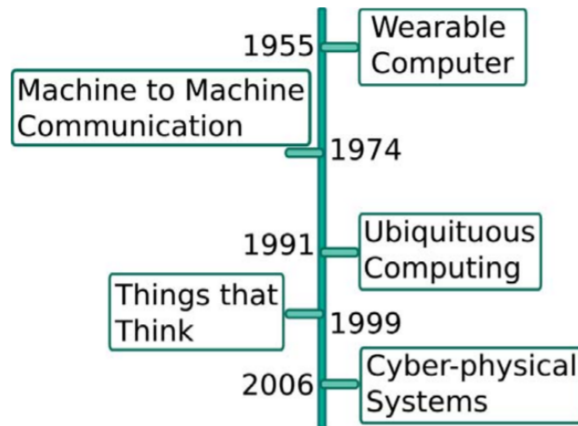


Figure 1.3: CPS Development Timeline from [90]

Moreover, CPS are among the key elements in the field of Industry 4.0. According to [61], they are systems which make a fusion between a virtual and physical world possible. They monitor and control the physical processes. A 5C architecture is proposed in [78] where the authors specify two main functional components of CPS. The first one consists in the advanced connectivity that ensures real-time data acquisition from the physical world and information feedback from the cyber space. The second functionality is intelligent data management, analytics and computational capability that constructs the cyber space.

Various features are identified in [56] that differentiate CPS, comparing them to a conventional system, such as the time-sensitive combination of the cyber and the physical, as well as their connected-ness. A consistency between Cyber Physical Systems and autonomic managers emerges through an intelligent aspect with feedback loops where physical processes affect computations and vice versa. They represent a suitable example of a SoS (System of Systems [85]).

Developing CPS software is also a major concern in the research community. As described in [108], it is a sophisticated activity as systems are logically and physically distributed. Platform diversity, real time properties and communication issues are challenging this concern. Therefore, modeling CPS represents a part of the solution facing both diversity and heterogeneity challenges.

1.2.2 Existing Approaches for Modeling Cyber Physical Systems

Many research works have been interested in connected environments, especially IoT. In spite of this, up to this date and to the best of our knowledge, CPS are barely addressed at the architectural phase of the software development life cycle. So far, architecting CPS is still in its infancy. There are presently no standardized or reference models for such systems. But, CPS and SA are currently gaining important attention of researchers. Looking through the last 5-year-proceedings of the most recognised conferences on Software Architecture and modeling (ECSA, ICSA and Models) has allowed us to select some interesting works gathering both SA and CPS modeling.

CAPS

Driven by a preliminary study on architecting CPS, Muccini et al., have proposed an architecture framework for modeling and simulation of situational-aware CPS [93]. The authors propose an architecture framework CAPS for modeling and simulation of situational-aware CPS. This work is based on the IEEE 802.15.4 standard to deal with various issues and different architectural views of CPS. In a classic way, Software Architecture, as an architectural view, mainly deals with structural and behavioral aspects. In addition to the software view, the proposed framework handles multi-architectural view points: the hardware view and the physical space. These three proposed modeling views are linked via two auxiliary views in order to create a combined software, hardware, and space view, through mapping and deployment modeling languages.

The hardware and physical spaces are modeled as a low-level specification of the concepts we may find in the IoT environment models. The structural aspect in CAPS software architecture could also be considered in this way. The Behavioral aspect is globally defined as a set of actions and events which are highly related to the domain model application. In this work, the emphasis is put on device modeling with a specified vision of the latter in a lower granularity. In addition, CAPS does not specify any generalized action type that may occur in a connected environment.

Digital Twins and CPS

In [69], the authors specify DT (Digital Twins) within CPS. DT are considered as a representation of cyber-physical systems and their related models. The proposed solution deals with the separation of concerns regarding the communication structure and the application. Modeling both CPS and DT information systems is useful especially when it comes to working with complex systems. According to the authors, this enables the generation of a required infrastructure in order to keep the CPS synchronized with the DTs. The addressed issue is how data could be represented, something that will be ensured by DT.

A Feature-Based Ontology for CPS

The authors in [124] propose a CPS meta model as well as a Feature-Based Ontology for CPS. In this research work, it is stressed that CPS are inherently heterogeneous by being multidisciplinary systems. Regarding the CPS meta model, it is important to notice that the proposal does not differ from the IoT ARM regarding the CPS component modeling. It is worth mentioning that the authors model communication network and communication protocol as components of the CPS system.

The proposed CPS ontology aims at supporting multi-paradigm for CPS. The identified features refer to constituent elements, non functional requirements, application domains, disciplines, and architecture. Each of which lists a non exhaustive set of values. The proposal seems to represent all the aspect of such systems as it deals with many features and somehow the complete mobilized ecosystem. As mentioned by the authors, it is not an approach for specific domain modeling.

CPS framework

Another attempt in representing CPS is depicted in [56] through a framework where a CPS analysis methodology and vocabulary to describe it are provided. One of the issues addressed by the authors is data exchange that is considered as a prominent dimension of a CPS operation. A layered architecture is simply presented here instead of a detailed software

architecture that could represent the components of a CPS.

In a recent study [69], further related works on modeling CPS are also presented. MontiArc [108] is a model-driven framework where CPS are modeled as Components and Connectors in order to make communication and dependencies explicit in the models. ThingML [59] is a DSL aiming at capturing the architecture of CPS and the communication required between the different components. It includes component types with ports and asynchronous messaging, event-based reactive programming, composite state machines, and action language.

1.2.3 Conclusion

Modeling CPS has taken several dimensions. Sometimes the border regarding Things (from IoT) is not clear. Some contributions are modeling IoT as part of Modeling CPS and vice-versa. Some cases could be confusing, for instance, in ThingML we are modeling CPS and not Things. Others are modeling connectors and communication issues. Presenting both domains allows us to spell out some misunderstandings. IoT is highly linked to environmental issues (hard devices, networking, etc) as a paradigm. CPS concern systems in a software and digital level (digital twins [69], computational capabilities, control, decision-making). Yet and as stated above, both subjects are thoroughly connected in order to construct a smarter world. Regarding modeling, the trend in IoT architecting is highly directed towards service modeling. However, we are facing new paradigm such as *aaT (Everything-as-a-Thing) [81] or IoE. In our work, we are interested not only in things but rather in Everything. IoE includes not only smart machines (things), but also humans, data and Services [14]. Interactions are then established between all the pillars of IoE. Thus, IoE could be defined as an infrastructure of a smart world where CPS could carry out high-performing computational capacities.

Chapter 2

Educational Modeling Languages (EML)

2.1 Introduction

In this chapter of the related works, we present some relevant elements of the literature concerning our application domain modeling which deals with educational environments. Instructional design is a major concern in the TELE (Technology Enhanced Learning Environments) domain, especially since the beginning of the Covid-19 health crisis. From then on, emergency remote teaching [16] has been widely used. Accordingly, the primary objective in these circumstances is not to re-create a robust educational ecosystem but rather to provide adapted access to instructional supports, learning materials, services and objects.

Various research works have been conducted since the early 2000s especially with the emergence of online learning via, for instance Moodle platforms. Hence, e-learning started to gain popularity and has revolutionized the educational system. Still, since the beginning of the Covid-19 crisis, researchers and practitioners as well as governors have put all their efforts and attention to explore, upgrade and rethink TELS (Technology Enhanced Learning Systems). Basically, our work aims at providing a vision for the future universities. Due to the health crisis, we have found ourselves in a perfectly-fit-situation with what we are

designing at an educational level because of the expansion of online systems' use.

In the literature, modeling education has taken several dimensions through Educational Modeling Language (EML). An EML is *a semantic information model and binding, describing the content and process within a 'unit of learning' from a pedagogical perspective in order to support reuse and interoperability* [106]. In short terms, EMLs are used to describe units of learning that in turn describe the learning process [84]. A unit of learning is defined in [73] as the smallest unit providing learning events for learners, satisfying one or more interrelated learning objectives.

In the following, we provide, first, a description of some existing EML, and a comparative study. Then, we present approaches that take into account both educational and connected environments.

2.2 Existing approaches for Modeling Educational Environments

One may find two different perspectives when it comes to defining a learning process and how it could be modeled. In fact, there are some research works that design learning through Activity Modeling Language (e.g. E2ML [30], ColeML [83]). Others design learning through Content Structuring Language (e.g. PoEML [34], (SCORM, 2003)). Some others focus on both perspectives, Activity and Content Structuring Languages, (e.g. IMS LD [74], CoUML [47]). Furthermore, there are modeling languages that rather concern evaluation modeling (e.g. IMS QTI [102]).

In fact, in the first category, the learning process is modeled as an activity where a learning activity is part of a real lifelong learning scenario in a related field. It involves the learner's interactions with a structured set of learning environment resources. Various related works design a learning process as a set of activities. One can refer to the learning theories that were mentioned earlier in the scope of the General Introduction, and more specifically to the constructive theory. The latter considers that learning is achieving understanding through

the process of connecting elementary mental or behavioral units. This process is based on sequences of activity [86]. According to [54], research on activity systems focuses on the way by which the individual components act and interact with each other, and larger contextualizing systems that provide resources and constraints for those actions and interactions.

On the other hand, a learning process is designed with regard to the knowledge-to-be-taught (the second category). According to Wilensky [128], what is concrete for an individual is not for another one. Indeed, knowledge needs to be adapted regarding its structure and the different stakeholders (researchers, students, teachers, community) who intervene before and during its transmission in both teaching and learning processes. Didactic is the science which studies such phenomena. In fact, the theory of the Didactic Transposition of Knowledge studies the transformations of several types of knowledge, as proposed by Chevallard [38]. From the organization of *the scholarly knowledge*, as it is done by scholars and researchers, to the definition of *the knowledge-to-be-taught* in curricula and manuals, a so-called external didactic transposition happens. Then, the selected knowledge-to-be-taught is tailored by the teacher for a particular context in the classroom (*taught knowledge*). This taught knowledge is learned by students (*learned knowledge*). Didactic issues are not the purpose of our work, however, this category is constituted of languages that allow the designers to create the learning units regarding the content structure, the learner's needs and performance. In this thesis, the "knowledge-to-be-taught" refers to its content structure.

Regarding EML, the authors in [72] identify some characteristics that should be validated. For example, an EML needs to be formally defined and machine-readable, so the script created with the EML can be interpreted by a computer application. An EML must be pedagogically neutral so different teaching trends can be applied in a unit of learning using the same EML. An EML needs to allow designers to create complete units of study that include the activities to be executed by the learner (what to do), the people involved in the activity (with whom), and the environment where the activities will be performed (which learning materials are needed, which software tools, etc.). The units of learning created using an EML should be resilient to technical changes, evolution, and platforms, since their purpose is to facilitate reusability between systems and tools.

In the following, a scope is given to the main EML related works according to the presented categorisation. First, some languages presenting learning as an activity are given : CoUML, E2ML, COleML, and IMS LD. Then, some languages presenting learning as a content are provided through PoEML, SCORM, LOM and IMS LD. Finally, IMS QTI will be given as an example of language considering learning as an assessment.

2.2.1 CoUML

The Cooperative Unified Modeling Language (CoUML) [47] extends UML with a modeling profile specifically designed to enable the modeling of complex and cooperative learning environments. It is to model structural and dynamic aspects of blended courses. As a primary artifact, one may find the Course Activity Model (CAM) that comprises a number of activity diagrams showing the course's activities at different levels of detail. When it comes to modeling complex courses, another primary artifact is created and maintained with the CAM in order to model a course structure through the Course Structure Model (CSM). It is also possible to model structural elements as secondary artifacts (i.e., goals, roles, and documents). The used notations are adopted from UML and then adapted to extend the required concepts for CoUML.

2.2.2 E2ML

Educational Environment Modeling Language (E2ML) [30] is designed through an informal visual language composed of chronological (the activity flow) and structural diagrams (the dependencies diagram). The "actions" represent the main element to be modeled in the educational environment. According to the authors, the activity flow is a sequence of actions represented as boxes that can be articulated with split and join devices. The dependencies diagram is a network of action, represented as boxes that can be connected by two types of connections to represent prerequisites and products. EML does not offer means to the development or implementation tool.

2.2.3 ColeML

The Collaborative Learning Modeling Language (ColeML) [32] is for making learning designs active, that is, readily usable in actual teaching and learning based on a collaboration between students and teachers. ColeML adopts modeling principles from the workflow area and business process. In this context, the main modeled concept is *activity* which, added to its name and description, can have inputs, outputs, actors, and resources (objects, tools, and instructional material, etc.). Parallel activities also have their own graphical representation for modeling collaborative learning. ColeML seems suitable for complex teaching methods like inquiry/project based learning.

2.2.4 LDL

The Learning Design Language (LDL) allows the representation of various situations, particularly collaborative ones, with few concepts [83]. The proposed meta-model puts "interactions" in the center of learning modeling. Their aim is to specify the exchanges between participants during the learning activity regarding their "role". An interaction is an action from one participant to another in a specific place. This interaction reflects the activity and thus the cooperation between the participants. Situations happen in "Arena" which is the means of situating the activity. An arena can be either a service (a search engine, a chat room, etc.) or a content (a course, an exercise, etc.).

2.2.5 PoEML

The Perspective-oriented EML (PoEML) [34] is conceived with a separation-of-concerns principle. It is based on the activity theory which is a meta-theory about the essence of activities as well as their components. This theory's expanded mediation model states that any activity involves a subject playing a role acting on an object to achieve a certain goal. This connection is influenced by the environment and the community where the activity is performed. The model has guided the identification of PoEML perspectives which are mainly about Structural, Functional, Environment, Participants, Data, Tools, Organizations, Order, Temporal,

Authorization, Awareness, Interaction, etc. PoEML provides a graphical representation for all the perspectives.

2.2.6 SCORM

The Shareable Content Object Resource Reference Model (SCORM) [13] is a set of technical standards developed by the ADL initiative (Advanced Distributed Learning) to provide a communication method and data models that allow e-Learning content and LMSs to work together. SCORM addresses interoperability and reusability challenges in e-learning through three sub-specifications. The Content Packaging part specifies how content should be packaged and described with XML. The Run-Time part specifies the launch and communication of the described content within Learning Management Systems using ECMAScript. Finally, the Sequencing part, described in XML, specifies how the learner can navigate between parts of the course (SCOs).

2.2.7 LOM

Learning Object Meta-data (LOM) [21] is a data model used to describe a learning object and similar digital resources to support learning. It is described in XML in order to foster reusability and interoperability within Learning Management Systems. By definition, an object is a collection of content items, practice items, and assessment items that are combined based on a single learning objective, according to CISCO [3]. IEEE defines it as any entity, digital or non-digital, that may be used for learning, education or training [3]. The type of object, author, owner, format, technical information, life cycle, and pedagogical attributes are described as meta-data of the learning object. A learning object is an independent structure of the knowledge-to-be-taught which is highly related to the learning activities as well as the educational context.

2.2.8 IMS-LD

IMS-LD [74] is the most referred design learning language as it represents a standard in the educational technology community. IMS-LD stands for Instructional Management System - Learning Design. It provides a conceptual model that defines the basic concepts and relations in a learning design. The specification is divided into 3 levels (A, B and C). All the conceptual elements of the different levels are presented in Figure 2.1. Each level is based on the model and semantics defined in the previous level. Level A contains the core model components of the design language. It is known as the static part which is composed basically of activities, roles (type of participants), and the environment (learning objects and services). This level also specifies prerequisites and learning objectives. Level B and C are known as the dynamic parts of IMS-LD. Level B defines properties and conditions in order to personalize the units of learning to the learner based on if-then-else rules. Level C introduces notification behaviours based on events that occur in a run-time environment. Most of the concepts of Figure 2.1 are detailed in the contribution section as we adopt this language for a model extension.

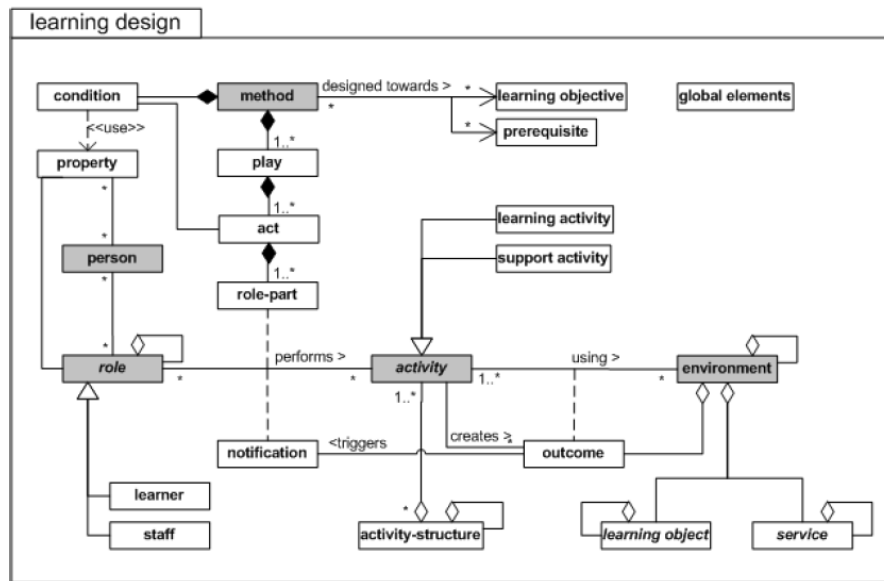


Figure 2.1: IMS LD from [63]

2.2.9 IMS-QTI

The IMS Question and Test Interoperability (IMS-QTI) [102] is a specification that describes a data model for the representation of question (assessment item) and test (assessment) data and the corresponding reports of the results. This model is supported by an XML representation to foster interoperability and data exchange between the different educational artefacts (e.g. learning systems, authoring tools, etc.). However, it is described abstractly, using the Unified Modeling Language (UML).

2.3 Discussion

All the listed EML seem to have a common goal about ensuring and fostering interoperability and reusability in complex learning systems. The studied concepts either focus on activities (called actions in some models) or units of learning (called also learning objects). As shown in Table 2.1, some criteria were chosen in order to position the studied related works. These criteria are highly related to the consideration of the main elements of the next revolutionary step of educational technologies at university that we specify : the consideration of inter-connectivity based on IoT or IoE, learning analytics, and present mode. Obviously, none of the studied related works considers connected environments or learning analytics. The latters have been recent challenges, after EML was established. This leads us to rethink EML and the way they represent learning according to the technological evolution and consider both physical and cyber learning systems. The integration of synchronous and asynchronous activities, including interaction patterns in face-to-face, traditional online and poly-synchronous learning environments is stressed in [44].

As described previously in the listed EML related works, they are mostly related to electronic learning and Learning Management Systems. Physical interactions in the real world are not really considered. EMLs enable instructors and educators to build and exchange courses by designing the whole learning process and the learning units. They serve as a communication language between pedagogical and technical staff. Attempts for gathering

	Present Mode	IoT/E Consideration	LA Consideration	Learning modeling though
CoUML [47]	blended	No	No	Activity
E2ML [31]	N/A	No	No	Activity
ColeML [32]	online	No	No	Activity
LDL [51]	online	No	No	Activity
poEML [34]	blended	No	No	Content structure
SCORM [13]	online	No	No	Content structure
LOM [109]	online	No	No	Content structure
IMS-LD [63]	online	No	No	Activity and Content structure
IMS-QTI [64]	online	No	No	Assessment

Table 2.1: Positioning of EML Regarding University 4.0 characteristics

different EML in the same management system are conducted in [116] through a model driven approach where the system programmatically generates a final course in different platform-specific target models based on a pre-defined platform-independent course model. However, this does not consider University 4.0 characteristics.

In this first part, we studied modeling in the related works regarding first connected environments and then educational ones. In the following, we give some highlights regarding the conducted efforts in connected educational environments (at the same time).

2.4 Existing approaches for Educational and Connected Environments

Our learning society has been through different formal and informal approaches from traditional learning to virtual classrooms. A list of concepts, which includes e-learning, ubiquitous learning, m-learning, etc., shows a real evolution in the educational domain. The difference between these concepts is sometimes difficult to distinguish, for non practitioners. In [115] and [98] review studies, the related areas are clarified. In fact, it is stated that Distance learning (D-learning) is an approach of remote learning where the learners and instructors are not physically present in a traditional classroom. The principle is to eliminate regular

face-to-face contact. Virtual learning (V-learning) relies in particular on the Internet and software applications to provide virtual classrooms regardless of user location. Electronic learning (e-learning) simply refers to the use of electronic devices that provide learning resources in the learning process, whatever its environmental disposition. In this approach, as shown in Figure 2.2, Mobile learning (M-learning) is considered when a used electronic device is mobile, such as smartphones, tablets, etc. In Ubiquitous learning (U-learning), the emphasis is put on embedded devices and technologies in order to support the permanency, accessibility, immediacy, interactivity and adaptability [79] of learning everywhere and at any time.

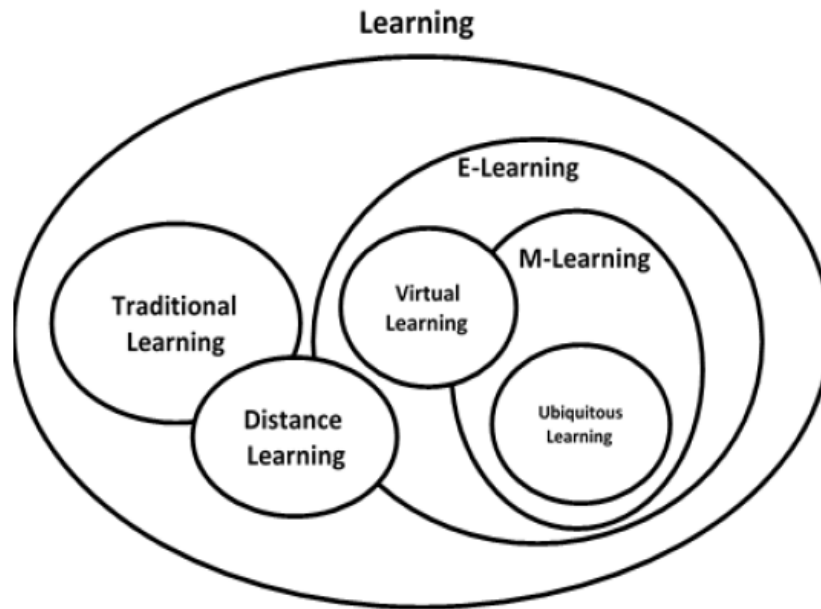


Figure 2.2: Learning Approaches from [115]

This section provides a crossed manner of highlighting related works that are positioned in both educational and connected environments. It is a sort of merging related work fundamentals in order to give another viewpoint of the state of the art. As the main characteristic of U-learning is the use of technological and embedded devices, the use of IoT/IoE in education could be associated to this approach. However, to the best of our knowledge, CPS are not that present in the research papers with regards to such issues. Consequently, only IoT in education is covered.

As far as connected environments are concerned, many research works have had the interest of the added values of IoT in education. Several value propositions [22] are empowering, directly or not, students' achievements. One of them is related to the real-time personalization of learning experiences that could engage more students. A review in [104] has recently been published about IoT applications in education. Interesting efforts have been conducted to improve the educational ecosystem based on this paradigm. We classify them according to the way IoT is used. We opt for the following classification :

- Learning/Teaching **of** Internet of Things : this classification represents the teaching and learning of IoT as a learning subject. The aim is to teach/learn the core knowledge of the subject. Research works such as in [113], [103], and [33] are examples of such research works. Education 4.0 could be classified in this type of IoT use. It focuses on teaching/learning IoT to prepare future professionals to acquire required competences and skills of the subject to enable them to work on IoT equipped environments such as Industry 4.0.
- Learning/Teaching **by** Internet of Things : this classification represents the use of Internet of Things as an artifact to acquire other knowledge. [129], [91] are examples of research works which focus on such aspect. Experiment based Learning is one of the conducted pedagogies that could be viewed as a way to serve knowledge through the bias of smart object manipulation.
- Learning/Teaching **based on** Internet of Things : This perspective is addressed in some research works regarding the monitoring of students' healthcare or in classroom access control, like in [122], [97]. Learning analytic techniques are adopted as a means to provide feedback to the different stakeholders.

Another category could be drawn according to further use of IoT in the educational context, which are not directly linked to learning and teaching. It focuses on energy management, enhancing safety, improving comfort, etc. [22].

2.5 Conclusion

Regarding EML, in our work, we build upon ideas from both Activity and Content Structuring Languages, in order to ensure both structural and behavioral aspects of educational modeling in a learning process. For this purpose, in our work, we adopt IMS-LD, as it is a learning model standard, in order to extend it and consider the listed comparison criteria. Even if it is meant to mainly describe units of learning, as described in the literature, we consider that IMS-LD provides means to describe the activities as well as the learning environment which fits a connected environment modeling. Hence, it is worth mentioning that some languages could be situated in both categories (learning as an activity and learning as a knowledge-to-be-taught) although they consider some elements as black-boxes such as learning object in IMS-LD.

Regarding educational connected environments, when talking about general issues related to IoT, the referred domains discount Education as a solid area where we can explore different IoT perspectives. Nonetheless, IoT has clearly affected the educational business model by providing several ways to do so. We consider that there is still a great deal of work to do in order to directly improve educational processes **based on** IoT. We thus adopt the positioning of our work on the third categorisation of IoT application (Learning/teaching **based on** IoT). This does not exclude the possible association with the other purposes of IoT applications.

Regarding modeling connected educational environments, to the best of our knowledge, there are very few related works in this area. According to our review, one may find an ontology combining IMS-LD and IoT that is developed in [121] to propose ontological models that allow intelligent interpretation of contextual data. Ontological models enable a formal and specific description of the meanings of things, according to the authors. Also, a Service-oriented architecture is proposed in [45] to integrate IoT systems into educational environments, considered as a System of System (SoS) and implemented by Multi Agent System (MAS) to handle the availability of requested resources between Constituent System (CS).

Many research works have addressed IoT in education by an empirical approach but

very few are those that deal with its modeling in a connected environment. In this thesis, the emphasis is put on modeling perspectives based on elements from both educational and connected views. It is worth mentioning that many researchers believe that security and privacy are among the main challenges of IoT in education [18]. These issues are important, however, they are not in the scope of this thesis.

Part I Conclusion

In this first part, we addressed related works in different areas of interest to this thesis work. First, we analyzed software architecture modeling in connected environments with regards both to IoT and CPS. Regarding IoT, it is concluded that none of the studied works considers object, human, data and service design at the same time. Most of the research works are either focusing on service or object design. Then, we presented CPS related works which allowed us to spell out some misunderstanding between these two sub-domains of connected environments. Both subjects are thoroughly connected in order to construct a smarter world.

Secondly, Educational Modeling Languages are studied to investigate the essence of their models which mainly focus either on activities or knowledge-to-be-taught (also called learning object, unit of learning) modelings. We conducted a comparative study based on a set of criteria. These criteria are highly related to the main elements of the next revolutionary step of educational technologies at the university. They are related to the consideration of inter-connectivity based on IoT or IoE, learning analytics, and blended present mode (to ensure more flexibility). Obviously, none of the related studied work considers connected environment or learning analytics.

Thirdly, we combined both previously studied domains (i.e. IoT/CPS and educational modeling) in order to study how IoT is explored in educational systems. According to our elaborated classification, we consider that there is still a great deal of work to do in order to directly improve educational processes **based on** IoT. We thus adopt the positioning of our work on the third categorisation of IoT applications (Learning/teaching **based on** IoT). The aim is to take advantage of this high connected-ness of smart objects (and not only) for collecting data and analysing educational behaviors and interactions and act accordingly.

Finally, these three main studied domains somehow reflect the different contribution directions for the rest of this thesis. First, a more complete model of connected environment will be presented based on the main components of CPS. Actually, we are interested in a "four-oriented" architecture for connected environments. Then, an adapted educational model is presented which will later lead to an educational connected system modeling.

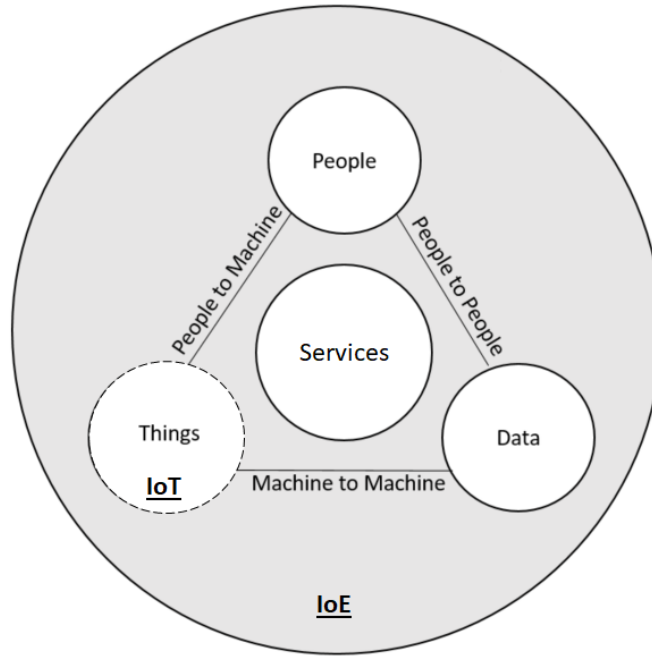


Figure 2.3: The Main Components of Cyber Physical Systems (revised from [45] and [14])

In fact, Figure 2.3 shows the borders among this plethora of concepts in the composition of CPS. The examined related works model either CPS or IoT. However, we deal with the same concepts and sometimes the same architecture. CPS are defined as System of Systems and so surround Everything. Thereby, as illustrated in the figure, the IoT is presented as part of the Internet of Everything (IoE). In the educational context, we consider that the main components of CPS, referring to [45] and [14], are :

- ”*Human*” refers to students, teachers, administrators or other stakeholders who are involved in the learning/teaching processes. This knowledge about user profile is essential to create personal and professional linkages by fostering collaboration and cooperation. As it is stated, people themselves will become nodes on the Internet, with both static information and a constantly emitting activity system.
- ”*Object*” refers not only to physical devices that can establish connection with the Internet and utilize sensors to capture environment information (as it is presented in [45]) or actuators to act on the environment, but also to smart learning resources that

can establish connection with the Internet (e.g. computer, telepresence-robot, etc.).

- ”*Service*” refers to how people and objects must interact to generate data that can be transformed into usable knowledge through service invocation. According to [52], each IoT Service is designed as a composition of the Service Model (details about attributes and relationships describing the IoT Service itself) and the Service Profile (details about the process implementing the service). These processes allow information to be addressed to the right people at the right time [45]. Consequently, well-designed processes can make the educational process more efficient and motivating.
- ”*Data*” refers to all the different data flows coming from the different elements of the connected system. Cloud computing and learning analytics are examples of technologies for data management and its transformation into information. Data could be structured, semi-structured, or non-structured in a connected environment.

In our work, we are interested in exploring IoE and so model all the types of components within the different interactions (people-to-people, people-to-machine, and machine-to-machine). As its name implies, IoT mainly deals with machine-to-machine interactions.

Part II

TOWARDS UNIVERSITY 4.0 : Educational Cyber Physical Systems (ECPS)

Part II Introduction

As stated above, our exploratory research methodology is conducted based on elements of our research context which are related to Industry 4.0, Autonomic Computing and Technology Enhanced Learning Environments. Obviously, there is a big gap between industrial processes and human learning processes. However, we are trying to see what is behind the "4.0" version. Industry 4.0 principles (IoT and CPS) are studied in the first part of this thesis where a review of the related works is given in order to position our contribution that we detail in this part. Moreover, conducted researches in TELS have made contributions to enhance education, as it was presented earlier. But, they are conducted through an ad-hoc development manner. Pedagogical models are still linear and do not consider the massive data that could be generated regarding the potential use of connected environments everywhere.

Since the emergence of smart and ubiquitous learning environments, many research studies have been interested in integrating Internet of Things in the educational context. We classified them into three categories (learning/teaching of IoT, learning/teaching by IoT, learning/teaching based on IoT) in the previous part. Existing related works have mostly dealt with IoT as a body of knowledge-to-be-taught or as learning artifact. However, there is a limited interest in using IoT in order to monitor the learning experience itself.

In our work, we believe in the power of connected-ness of everything. For the remaining parts of this thesis, the IoE concept will not only be limited to Internet of Things but also Internet of Services, Internet of Data and Internet of Human [14]. In fact, the proposal, based on this concept, will be presented through two phases. The first one gives a holistic architecture of what we call University 4.0 via its core element (ECPS), with a particular attention to the educational purpose, as at university, one may find several branches (research, administration, logistics, etc.). Then, the second phase is dedicated to the ECPS modeling method based on some elements from the literature.

Chapter 3

Educational Cyber Physical Systems Architecture

Fostered by the accelerated digitalization of the modern society, research works have been interested in the future of education in order to adjust it according to the potential of technology. As stated in [82], the main idea of an innovative smart society will be the development of human potential. Various emerging concepts (smart education [82], education 4.0 [40], smart university [126]) have been proposed regarding this purpose. For instance, Education 4.0 focuses more on innovative educational content from a didactical perspective. Its aim is the training of future professionals who will be working in an Industry 4.0 context.

Furthermore, a conceptual definition of smart university is given in [126]. It is defined as an n-tuple of n elements that can be constituted of : smart features (adaption, sensing, inferring, self learning, anticipation, self-optimising or re-structuring), smart stakeholders, smart curriculum, smart pedagogy, smart classrooms, smart hardware and software, and smart technology and resources. This definition of the "smartness" in "smart university" is limited to the organisation and deployment of modern technological devices within connected features. More importantly, we believe that it should be implemented to support both learning and teaching activities more efficiently to generate the suitable pedagogical scenario regarding learning objectives and students' performances.

As IoE is being more and more used in our daily lives, we consider that, from an educational perspective, University 4.0 is an intelligent integration of the learning and teaching processes, devices with networked sensors and software, educational stakeholders and related data used to monitor, analyse, plan, control and react for better education and societal outcomes according to the educational context. This intelligent integration is ensured by Educational Cyber Physical Systems (ECPS). In addition to educational features, smart administration, smart research and smart resources are also parts of University 4.0. We consider that ECPS are the key components behind the smartness of each part of University 4.0. In our work, we only focus on the educational part. In this chapter, we specify this core element by proposing a definition and a holistic software architecture view of this innovative concept. This represents our first contribution which allows us to deal with our first Research Question (*How to define ECPS as an innovative concept, with a particular focus on the different educational processes at University ?*).

3.1 Software Architecture

Defining, modeling or implementing a complex system first needs an awareness of its Software Architecture (SA). According to Sommerville [119], SA is a description of how a software system is organized. Properties of the system are influenced by the used architecture. The author identified two levels of abstraction when designing SA. *Architecture in the small* focuses on the architecture of individual programs and their decomposition into components. *Architecture in the large* focuses on the architecture of complex enterprise systems that include other systems, programs and program components. SA is important as it affects the robustness, distribuability, and maintainability of a system [28] as well as other properties. In our work, we design a SA in the large and give a holistic vision of ECPS. Moreover, in SA, we deal with architectural design decisions according to different properties (performance, security, safety, availability, maintainability, etc.), architectural view (logical, process, physical, development, etc.) and architectural pattern (layered, MVC, etc.). We contextualize these elements regarding our work :

- Architectural design decisions : Our main objective is to ensure the supervision of the ECPS at first so that the system could maintain its performance regarding collecting and analyzing data.
- Architectural view : We are eager to design the structural and behavioral view of the system. Further models regarding the design of the CPS physical level do already exist (e.g. [93]).
- Architectural pattern : As we have presented so far, we rely on a layered generic architecture of ECPS. However, our aim is to define the main components of our system at a generic level through a MDE method.

3.2 ECPS Definition

Many research works have been interested in Cyber Physical Systems and their extended applications in various domains (Industry, Healthcare, Energy, etc.). But there are barely any studies which have dealt with their potential use in Education. To the best of our knowledge, Educational Cyber Physical Systems are not yet emerging in the scientific community as a well defined research concept. In [35], the authors define an ECPS as a localised system involving software (cyber) and physical (experimental) resources or a more advanced distributed collaborative system. This definition is very reductive to the vision we have of ECPS.

Domains and applications of CPS were identified in the previous studied works (robotics domain, electric vehicles, supervisory systems, embedded systems, sensors and actuators networks, and smart grids). We, thus, consider that its educational application could also be viewed as a supervisory system aiming at gathering real-time data from connected educational components in order to constantly monitor and make decisions about not only the adaptation of pedagogical scenarios but also other educational oriented decisions (course design, curriculum process).

ECPS are educational complex systems constituted of physical and cyber components that aim at implementing control loops to manage the educational processes according to educational stakeholders' requirements. These components refer to IoE.

ECPS are based on a network of cyber-physical components within an IoE educational ecosystem, where the aim is to improve not only the learning and teaching processes, but also the whole educational ecosystem. At different operational levels, ECPS implement MAPE (Monitor, Analyse, Plan and Execute) control loops to supervise and possibly self-adapt the whole curriculum, the courses' structure, and pedagogical scenarios depending on students' skills, capacities and the run-time evolution, etc. They are Cyber Physical Systems applied to the world of Education to make the fusion between the physical and cyber worlds. They are set of objects, services, data and humans, that collaborate to carry out teaching and learning scenarios in a short/medium/long term context (Classroom, Course, and Curriculum).

Generating the required processes should be implemented based on a particular data processing which highly depends on the educational context. Data collection and analysis are in a cyber level in order to monitor and supervise the physical world and adapt the processes when it is needed. This monitoring is based on executing control loops, adopted from the Autonomic Computing perspective, presented in the general introduction of this thesis. The aim is not to provide an instructional system related to specific requirements but rather gathering different services, different educational software and any other component that could contribute directly or indirectly to data generation in an open architecture based on the autonomic computing workflow.

It is worth mentioning that human decision making is very crucial in our ECPS proposal. The latter could be considered as a CPSS (Cyber Physical Social System) [49]. The purpose of CPSS is to carefully consider the impact of individual human beings that are an integral part of the systems, both as a user as well as a source of disruption, according to the authors.

3.3 ECPS Generic Architecture

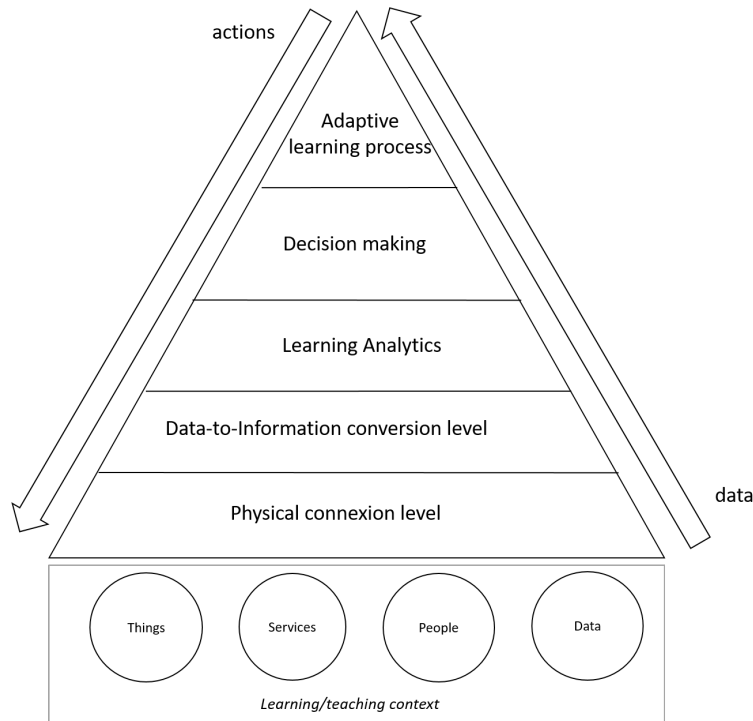


Figure 3.1: The Generic Layered Architecture of Educational Cyber Physical Systems

Regarding the educational purpose, we architect *in the large* a generic layered model of ECPS that could be implemented at different levels of educational processes at the university. Different parameters are involved to decide about a required improvement or an educational need based on an important data flow management. The MAPE control loop principle is the core process applied to provide contextual decisions. In fact, we identify three different levels of ECPS execution (classroom, course, curriculum). In order to specify the latter, we first present the ECPS generic architecture based on IoE through the different layers :

- Physical level : This level represents the interconnected components of ECPS which are mainly about IoE. It is worth highlighting them once again here :
 - *Human* refers to students, teachers, administrators or other stakeholders who are involved in the learning/teaching processes. This knowledge about user profile is

- essential for creating personal and professional bindings by fostering collaboration and cooperation. As it is stated, people themselves will become nodes on the Internet, with static and dynamic information. Dynamic aspect is ensured through the constantly emitting activity system.
- *Object* refers not only to physical devices that can establish connection with the Internet and utilize sensors to capture environment information (as it is presented in [45]) or actuators to act on the environment, but also to smart learning resources that can establish connection with the Internet (e.g. computer, telepresence-robot, etc.).
 - *Service* refers to how people and objects interact to generate data that can be transformed into usable knowledge through service invocation. According to [52], each IoT Service is designed as a composition of the Service Model (details about attributes and relationships describing the IoT Service itself) and the Service Profile (details about the process implementing the service). These processes allow information to be addressed to the right people at the right time [45]. Consequently, well-designed processes can make the educative process more efficient and motivating.
 - *Data* refers to all the different data flows coming from the different elements of the connected system. Cloud computing and learning analytics are examples of technologies for data management and its transformation into information. Data could be structured, semi-structured, or non-structured in a connected environment.
- Physical connection level : This layer consists of the data collection that may be collected through sensors (e.g. biometric sensors) or obtained from systems that are involved in the educational ecosystem. Particularly, SIS (Student Information Systems) that are used in an administrative level could provide a significant amount of data. TELS are also becoming more and more present especially since the Covid-19 health crisis. Moreover, data could be tracked regarding human-to-human interactions (student-student / student-teacher) through, for instance, the use of particular services.

- Data-to-information conversion level : Before proceeding to data analysis and knowledge extraction, a data-to-information conversion process is needed. It consists of data filtering where one may focus on a part of the data set and use that subset for viewing or analysing. Due to various involved systems in an open connected environment, different data formats would not support interoperability in the analysis phase (next layer). Unified forms of information allow flexibility and interoperability. Hence, this is considered as the second challenge for this layer in order to prepare significant data and so information for the next one.
- Learning Analytics (LA) : According to a review in [36], there are many goals of using LA, such as, monitoring and analysis, assessment and feedback, adaptation, prediction and intervention, personalization and recommendation, tutoring and monitoring, etc. Basically, our aim in this generic architecture is to provide an open system that could support any of these goals. LA uses techniques of artificial intelligence such as data mining, statistics, information visualization, etc. in order to process for the analysis step. Analysis results are highly related to the predefined goal. For instance, when the goal is to monitor and analyse, the analysis result could simply be about communication based on information visualisation as a technique. When the aim is to adapt a learning process, the analysis result is expected to make decisions and act accordingly.
- Decision making : The analysis result (communication and/or actions) is then communicated to another cognitive level where decision making could be done automatically, semi-automatically, or manually with a human intervention, based on analyzed information. In our case, both learners and teachers or even educational staff could contribute to the decision making process. For example, the learner could choose between different recommendation tips offered by the system or the teacher. The teacher could simply be aware of the analyzed result at a given point during a run-time session.
- Adapted learning process : This phase is the output of the ECPS. By learning process, we mean the educational process and ecosystem in general. Whatever the taken

decision, it is about adapting the process by acting according to a scenario or its components. At this level, the educational ecosystem components are expected to implement decisions by taking an appropriate action based on users' requirements.

- *Data and Action* : As we deal with Everything, it is worth mentioning that data is the key element of such educational complex systems. In this generic architecture, the aim is to have a continuous feedback based on collected data from the physical space, and a set of actions that are meant to be executed in the physical space as well. The cyber space is involved in order to bridge the gap between raw data and required decisions. The bidirectional data/action flow represents control loop flows around the layered architecture.

To conclude this generic architecture description, the physical environment represents the main components of the educational ecosystem (IoE) that differ from one situation to another regarding both temporal constraints and educational objectives. In fact, we adopt and define three types of learning processes : (i) Classroom process for short-term situations (ii) Course process for mid-term situations (iii) Curriculum process for long-term situations. We believe that these three types of process refer to different learning outcomes throughout the whole learning journey (knowledge (classroom process), competence (course process), expertise (curriculum process) [60]).

In the following, we present the implementation of each level with a contextualization of the learning and teaching context especially regarding the physical connection layer. Each level stresses a different use of the generated data.

3.4 Different Levels of ECPS

3.4.1 Classroom ECPS

A classroom process refers to a combination of a set of activities occurring in a physical and/or virtual classroom. The learning outcomes of such process are the learnt-knowledge by

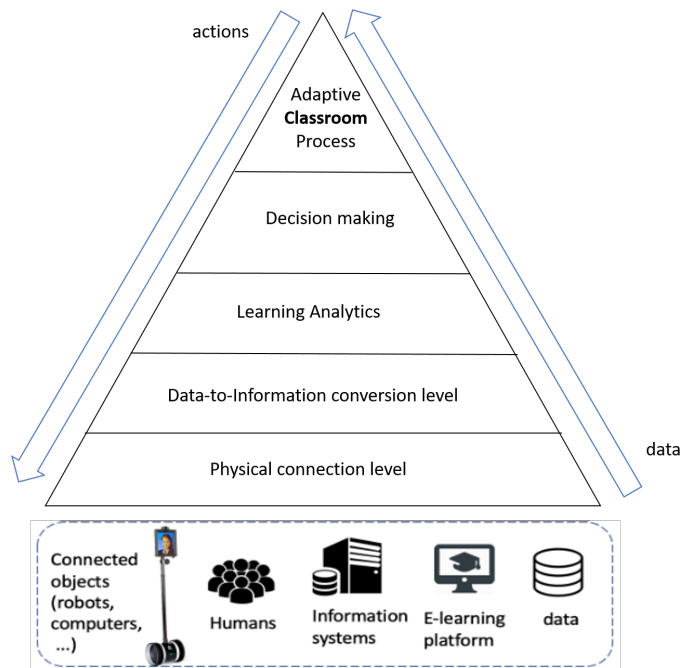


Figure 3.2: Educational Cyber Physical Systems in a Classroom Process

students. As presented in [60], learning is highly related to knowledge management and must address its acquisition, creation, storage, redistribution and application through activities. Learners and teachers are the main actors of this management process which highly depends on the choice of pedagogy.

Figure 3.2 shows the use of the generic layered architecture in a classroom level implementation. The basic layer consists of a set of both physical and virtual resources. This mixture of different natures of managed resources strengthens, for instance, blended learning. The aim is to bring together a sort of everything (Human resources, connected objects, information systems, TEL Systems, data, etc.) to ensure not only the capacity to collect a maximum amount of data for a better pedagogical knowledge management, but also to ensure learning flexibility and adaptation.

Data is collected during the progress of the activity from the classroom ECPS components. It must be stored in a common format in a data warehouse, and analysed to deduce knowledge/know-how validation, but also the level of engagement of the present and/or distant students, or any kind of analytics the educational stakeholders would like to implement. The interactions between teachers and/or students in the classroom could be considered as

an indicator for the efficiency of the learning process. Decision making in this level involves only actions related to a pedagogical scenario (short-term).

3.4.2 Course ECPS

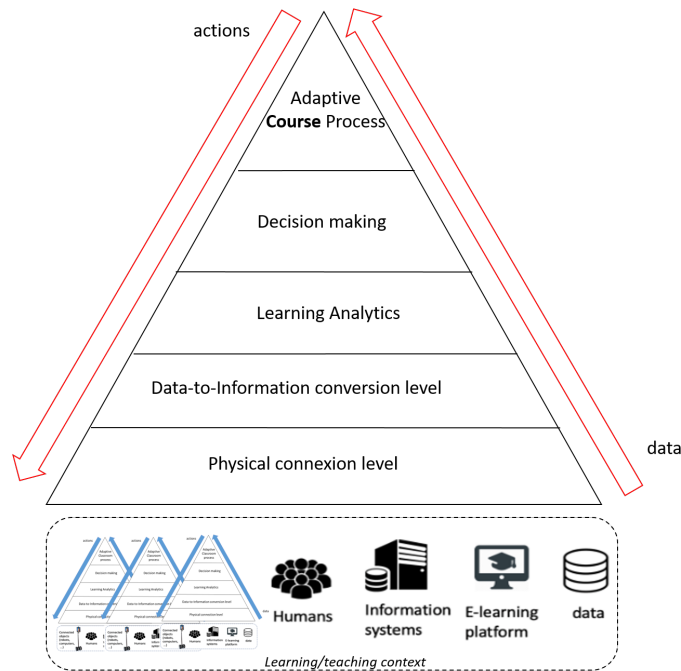


Figure 3.3: Educational Cyber Physical Systems in a Course Process

A course process covers the structure of the knowledge-to-be-taught for a mid-term period. Competences are the learning outcomes of this kind of process. They imply the ability to demonstrate a consistent level of performance. Despite the fact that this process has a didactic perspective, the monitoring of a course process seems important for students in order to adapt learning sessions according to their needs by adding, updating, or deleting (un)necessary sessions. Monitoring this kind of learning process ensures the achievement of learning objectives and thus the required competences within a particular course.

A course ECPS (Figure 3.3) is based on collected data from classroom ECPS presented by different instances as well as further data coming from other ECPS components like SIS, TELS, etc. The classroom ECPS instances designed in the physical level are presented with

filled blue arrows to specify their earlier implementation. Hence, we rely on the correspondent analyzed information to couple them with further data in the context of a mid-term educational process.

The aim is also to boost and enhance the efficiency of a course design and its related sessions as well. Pedagogical staff are the main actors in this level. In the case of a human decision maker, the teacher (in the classroom process) may not be involved in the Course ECPS if he/she is not the responsible of the whole course management.

3.4.3 Curriculum ECPS

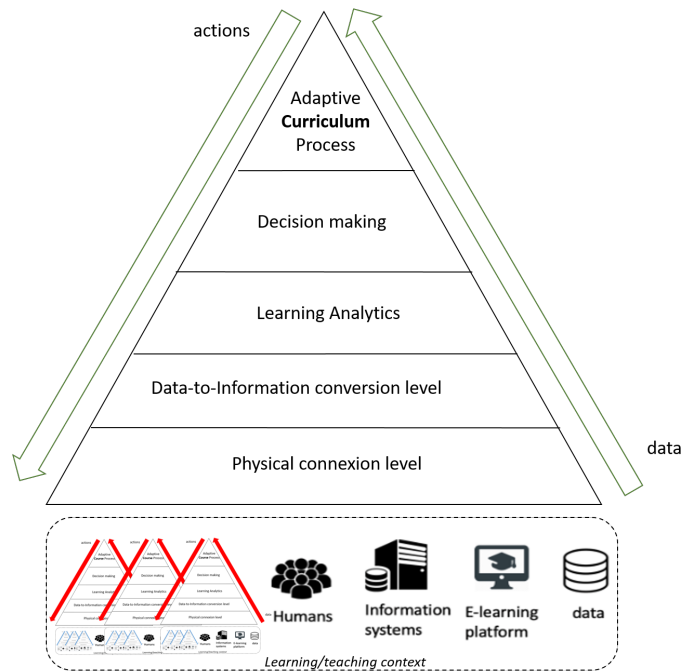


Figure 3.4: Educational Cyber Physical Systems in a Curriculum Process

A curriculum process covers the strategy of learning and teaching that highly depends on the learners' background, performances and needs in a long-term perspective. This will lead to expertise acquisition which comprises related value judgements, knowledge and skill sets, lived experiences, and problem solving abilities.

Once different instances of ECPS courses occur in different contexts (different courses in different classroom processes), a curriculum ECPS, as illustrated in Figure 3.4, executes

almost the same stages as in Figure 3.3 but from another viewpoint. In fact, the IoE components in a high level of such learning process management will be focused not only on physical resources (connected objects, humans, learning systems, data, etc.) but also and particularly on meta-data traced back to different instances of Course ECPS processes.

The network of ECPS courses coupled to further data (e.g. Competence portfolio) allows a decision making about the whole curriculum process. Based on that, we could be able to determine if, for example, a student has upgraded his/her knowledge in a particular subject. This improvement could be ensured by attending a MOOC to deepen students' knowledge or, in the worst case, attending another course. Hence the concept of such ECPS is supposed to automatically adapt the generic curriculum to each student based on the acquired competences of a long-term expertise acquisition. The main actors may (or may not) be the same actor in the previous listed processes. Generally, they are the administrative pedagogical staff.

3.5 ECPS Coordination

To conclude what is stated above, Figure 3.1 shows the coordination between the different learning processes. Data management is the core element of all the processes. This management allows to retrieve information from data, and then knowledge from information. Data sources are different from one level to another. It depends on the type of process as well as the used components of the ECPS.

In the rest of this thesis, we focus on only the classroom process level as it represents the basis for data collection and further processes as well. The learner is also the primary actor of the process contrary to the course and curriculum processes where the educational staff are the most involved actors. In addition, machine-to-machine, machine-human and human-human interactions are highly active at this level. Hence, the provided definition and modeling tools are highly oriented to answer classroom process requirements within a particular learning and teaching context. In the next chapter, we detail our proposed method to design ECPS in classroom processes.

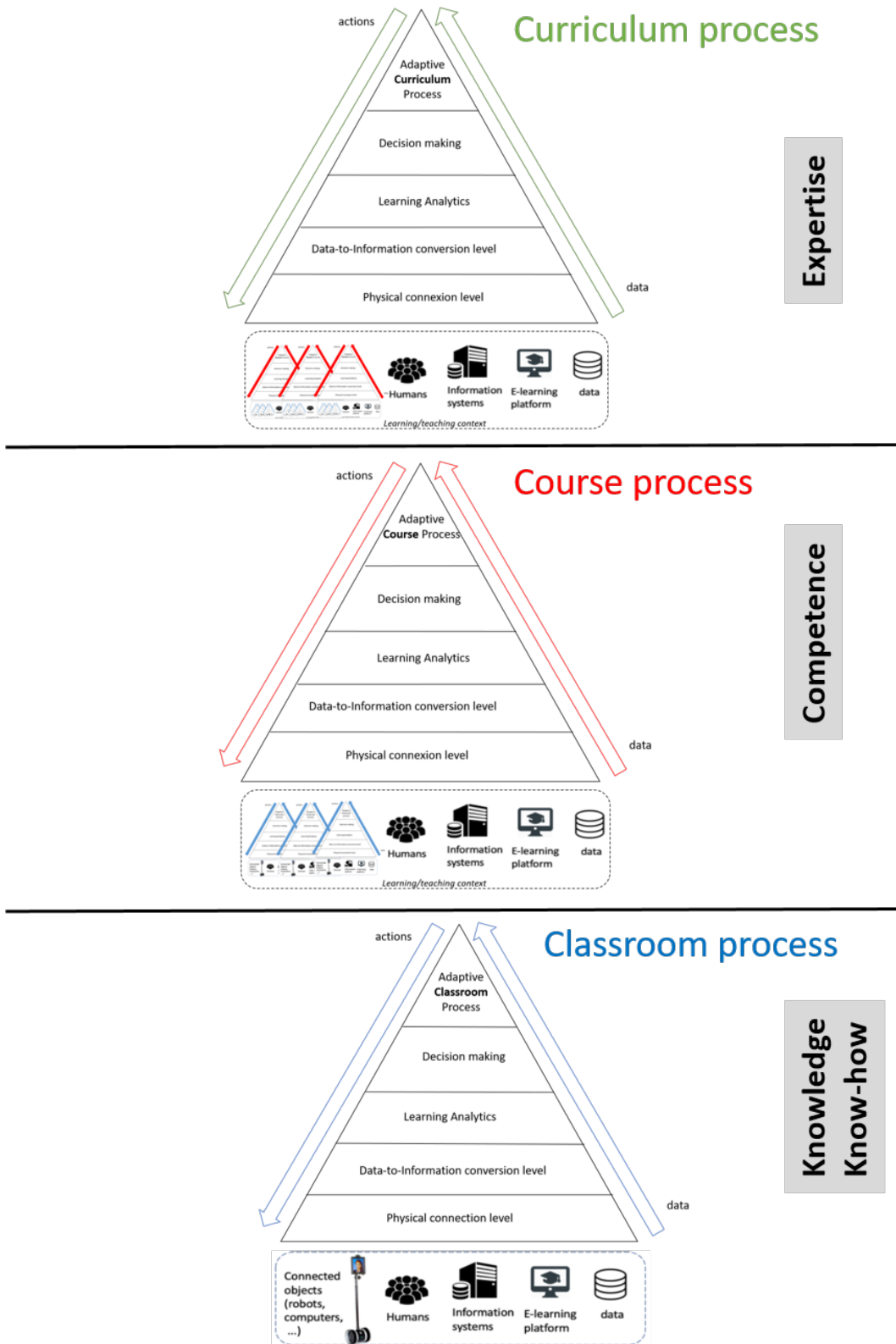


Figure 3.5: Educational Cyber Physical Systems Coordination Between the Different Learning Processes

Chapter 4

ECPS Modeling

It is worth reminding that our methodology is mainly exploratory in defining and modeling ECPS. This concept is related to both technological and educational evolutions and so to the TELE domain. In the latter, any research contribution is conducted either with technological or educational requirements, or both [70]. Due to technological evolution, educational aspects need to take profit from this dimension in order to provide a better life-long learning experience for learners. Open educational requirements should also get the required service in learning systems. In fact, our contribution proposes evolution in both perspectives and designs ECPS accordingly regarding a model driven engineering method called *ModelECPS*. In this chapter, we present the method process and modeling languages that we developed for this purpose. This allowed us to ask our second Research Question (*What is the required method to design ECPS by considering the evolution of both connected and educational aspects?*).

4.1 Model Driven Engineering (MDE)

MDE is a software development method that uses its core models not only as inputs but also as outputs in order to reduce the gap between problem domain and solution domain. It is more general than the set of standards and practices recommended by the OMG's (Object Management Group) Model Driven Architecture (MDA) proposal [75]. MDA is a standard

proposed by the OMG to deal with the separation of platform dependent and independent aspects in information systems. According to [75] as well, MDE handles separation and combination of various kinds of concerns in software or data engineering. It allows the exploitation of models to simulate, estimate, understand, communicate and produce code [39]. The different concerns are mainly about :

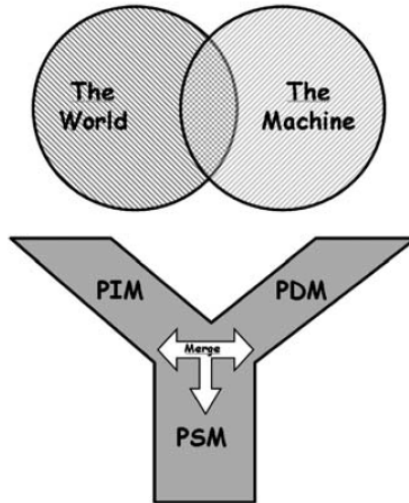


Figure 4.1: The World and The Machine from [26]

- PDM (Platform Definition Model) which is highly linked to the kind of platforms and their related definition. According to [26], this could be very difficult to build. It could represent Operating Systems, Data Bases, Virtual Machines, platform infrastructure, etc. In the literature, PDM are not really explored.
- PIM (Platform Independent Model) which outlines the description of the domain model itself that may contain different specifications of the enterprise business model, but not specifications about the system platform. PIM are supposed to survive the constant changes in software technologies that are transformed into new PSMs.
- PSM (Platform Specific Model) which is produced by the transformation of the Platform Independent Model according to its platform definition (PDM). It specifies how

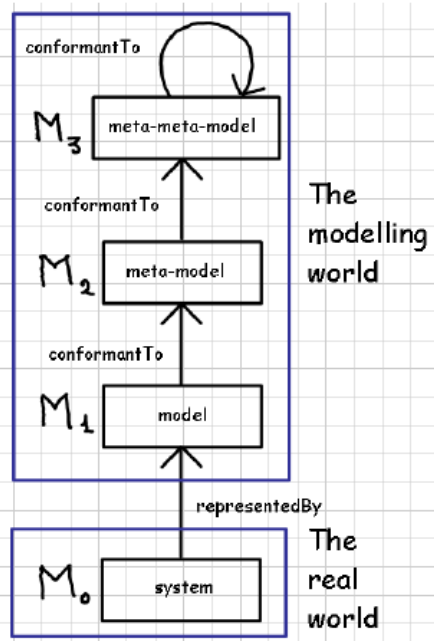


Figure 4.2: Modeling and (Meta)MetaModeling from [26]

the system uses the chosen platform. A PSM corresponds to a generated model (e.g. a textual program, a graphical model).

Figure 4.1 represents the different models regarding the real world and the machine. Each of these three models may be presented through several models from different views. According to [119], it is required to model a system from different views as we cannot represent all relevant information about its architecture in a single one. For instance, a graphical model can only show one view or perspective of the system. In fact, modeling, in its broadest sense, is the cost-effective use of something representing something else for some cognitive purpose. It allows us to use something that is simpler, safer or cheaper instead of reality for some purpose [111]. Hence, a model is an abstraction of reality in the sense that it cannot represent all aspects of reality. This allows us to deal with the world in a simplified manner, avoiding the complexity, danger and irreversibility of reality. A Meta-Model is a model which describes a specific domain model through a modeling language, also called DSL (Domain Specific Language). Thus, a terminal model conforms to a Meta-Model, which means that we use only pre-defined elements of the meta-model to describe a model. A Meta-Model,

in its turn, confirms to a Meta-Meta-Model. Furthermore, a Meta-Meta-Model is a model which conforms to itself. The MOF is the unique Meta-Meta-Model in MDA. It represents the extensible and platform independent model driven framework for modeling and Model Driven Architecture. Figure 4.2 presents both the real and modeling worlds at different levels.

4.1.1 Domain Specific Languages (DSL)

In view of our Research Question on Modeling, we consider MDE as an efficient method to adopt into our research method. In fact, Domain Specific Languages (DSLs) are the complementary part of MDE which is considered as the main means to define models. DSLs are languages tailored to a specific application domain, in our case modeling educational connected environment. In contrast to general-purpose languages, they offer substantial gains in expressiveness and ease of use [88]. DSLs are defined through different parts (abstract syntax, concrete syntax, and semantics) and allow model generation features in MDE. Referring to the literature, these parts are specified in [39] as following :

- Abstract Syntax : it expresses the notions that are specific to the domain of the language. In MDE, the domain is modeled through the use of a Meta-Model. Therefore, the Meta-Model describing the domain represents the abstract syntax of the modeling language. Meta-Models play the same role for modeling languages as grammar does for programming languages
- Concrete Syntax : the syntax domain defines the symbols used to represent the concepts in the language. A modeling language may have different concrete syntaxes (e.g. a textual program, a graphical model).
- Semantics : the semantic domain is the meaning of the concepts in the language. It may be defined through the semantic mapping towards the precise semantics of an existing programming language.

4.1.2 Model Transformations

In MDE, one may deal with model transformations when it comes to deal with models as artefacts. A transformation is the generation of a target model form a (or many) source model(s). This generation is ensured through transformation definition. The latter is a set of transformation rules that describes how one or more constructs in the source language can be transformed into one or more constructs in the target language [71].

There are different types of model transformations. The basic concepts are positioned in Figure 4.3. Endogenous transformations are transformations between models expressed in the same language. Exogenous transformations are transformations between models expressed using different languages [27]. Similarly, one may find horizontal and vertical transformations. The latter is related to transformations where the source and target models reside at the same abstraction level. The former is related to transformations where the source and target models reside at different abstraction levels [87].

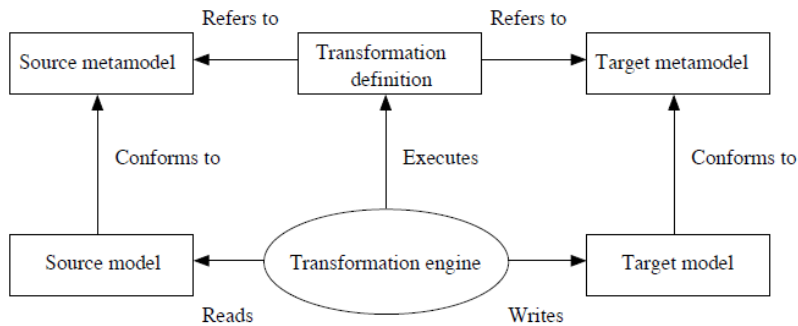


Figure 4.3: Basic Concepts of Model Transformation from [43]

The main objective of the MDA is the separation of concerns through platform dependent and independent modeling. This initiative is motivated by the constant change in implementation technologies and the recurring need to handle software from one technology to another [66]. For this purpose, MDA comprises a set of modeling standards. It is about UML(profiles) and MOF (Meta Object Facility) which are specified to define models and modeling languages. UML stands for Unified Modeling Languages. It enables the analysis, design and implementation of software based systems for different stakeholders (system ar-

chitects, software engineers, and software developers). XMI (XML Meta-data Interchange) is put forward to represent and exchange meta-data and thus models. It can also be used for serialization of models. OCL (Object Constraints Language) is a declarative language that provides constraints and rules which cannot be expressed by diagrammatic notation. The designed system is constrained by these expressions through constant conditions.

OCL is considered as a key component of the QVT (Queries, Views, Transformations) specification, an OMG standard too. The latter provides a description of model transformations through hybrid declarative and imperative languages. A layered architecture, presented in [19], shows the declarative languages (Relations, Core) and the imperative languages (Operational Mappings) that compose QVT. The Relations language specifies transformations as a set of relations among models where the semantics are handled by the Core language. The Operational Mappings language handles the extension of these languages to address the issue of incomplete declarative solutions for a given transformation problem. ATL (Atlas Transformation Language) is a transformation language that is highly inspired by QVT. We rely on ATL in our model-to-model transformations. Our technological context is oriented towards a Model Driven Engineering method. Modeling and models are in the heart of our proposed framework to handle the separation and combination of various kinds of concerns in software and data engineering. In the following, we present our modeling method.

4.2 ECPS Design Method (ModelECPS)

We rely on MDE principles to explore the power of modeling. As described by [111], modeling is everything. Naturally, in turn, everything could be modeled. However, as modeling represents an abstraction of a particular system from a particular viewpoint, one could not model everything at the same time. A model view represents a (real) system from specific system concerns. The latters represent interests of a system that are relevant to one or many of its stakeholders [65]. Hence, our proposal aims at modeling Everything (in the sense of IoE) in an educational system but not Everything (in an exhaustive way). This inscribes our approach in modeling ECPS architecture in the large [119].

In the following, we provide a detailed description of our adopted method that we implement through MDE standards. In [42], the authors identified some design challenges related to IoT systems. We believe that they remain the same regarding IoE systems. They are about (i) the heterogeneity of hardware devices and software components; (ii) the lack of mechanisms to address multiple stakeholders' concerns; and, (iii) the lack of a method to design IoT applications. In our work, we provide a method for the ECPS design. As stated in [105] and according to the OMG, a method, in software engineering, consists of : (1) a modeling language : a set of modeling conventions (syntax and semantics); and (2) a process which:

- provides guidance as to the order of the activities,
- specifies what artifacts should be developed using the modeling language,
- directs the tasks of individual developers and the team as a whole, and
- offers criteria for monitoring and measuring a project which includes products and activities.

As previously mentioned, we consider MDE as an efficient method to be adopted in this thesis, as it handles separation and combination of various kinds of concerns (such as platform models and code generation) in software engineering in order to bridge the gap between the design and implementation phases. Our method, called *ModelECPS*, adopts the classical “Y development cycle” from MDA as the method process, as illustrated in Figure 4.4. The objective of this process, according to [26], is to enable the generation of Platform Specific Models (PSMs) from Platform Independent Models (PIMs) and Platform Description Models (PDMs).

For each platform model, we propose a DSL. Firstly, the CPSML meta-models the platform description domain related to connected environments based on the IoE paradigm. This contribution is independent from the application domain. Secondly, the EML4.0 represents the application domain related to educational purposes. This is independently meta-modeled from the implementation details. Thirdly, we define ECPSML which is the third DSL aiming

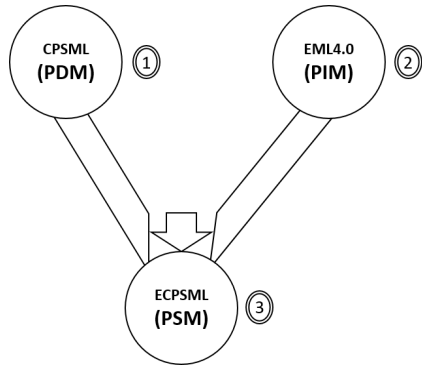


Figure 4.4: Modeling Process for the ModelECPS method

at combining both connected and educational aspects. More concretely, when it comes to design ECPS models confirming to these DSLs, the purpose is to elaborate the CPS and EML4.0 models and then the ECPSML model is supposed to be generated automatically based on a set of transformation rules that we identify through a concept alignment process.

Our method defines three modeling languages that together define the whole ECPS knowledge. Due to the multidisciplinary of the latter, our modeling languages aim at helping ECPS designers in representing both educational and connected aspects of such system. Hence, the CPS expert(s) addresses the connected-ness of the system through CPSML, after becoming aware of the educational context and instructions from the educational / pedagogical expert(s). Moreover, the pedagogical expert relies on EML4.0 in order to model the pedagogical scenario and related concepts of the learning session(s). Then, both of them collaborate together to couple their models using ECPSML and the concept alignment process.

Aiming a higher flexibility, the goal is also to produce software assets that are resilient to changes in technologies. MDE stresses the importance of PIMs since they are supposed to survive the constant changes in software technologies that are transformed into new PSM [75]. By contrast, PDM are highly linked to the kind of platforms (connected context in our case). In an educational purpose, we can deploy the same pedagogical scenario in different platforms (something we have noticed during lock-downs). In this case, the Y development cycle is a real advantage, allowing us to redefine only the PDM model, with no change in the educational one (PIM). Furthermore, several pedagogical situations can be modeled in

the same environment, by changing only the PIM model. In addition to this, our method could be adopted in order to model other domains. To do so, the generated meta-model, the application domain meta-model and part of the transformation rules have to be adapted. The CPSML remains the same as well as the whole processing mechanism regarding the Y development cycle.

Our three modeling languages (DSL) are implemented using the Eclipse Modeling Framework that we present in the next section.

4.3 Eclipse Modeling Framework (EMF)

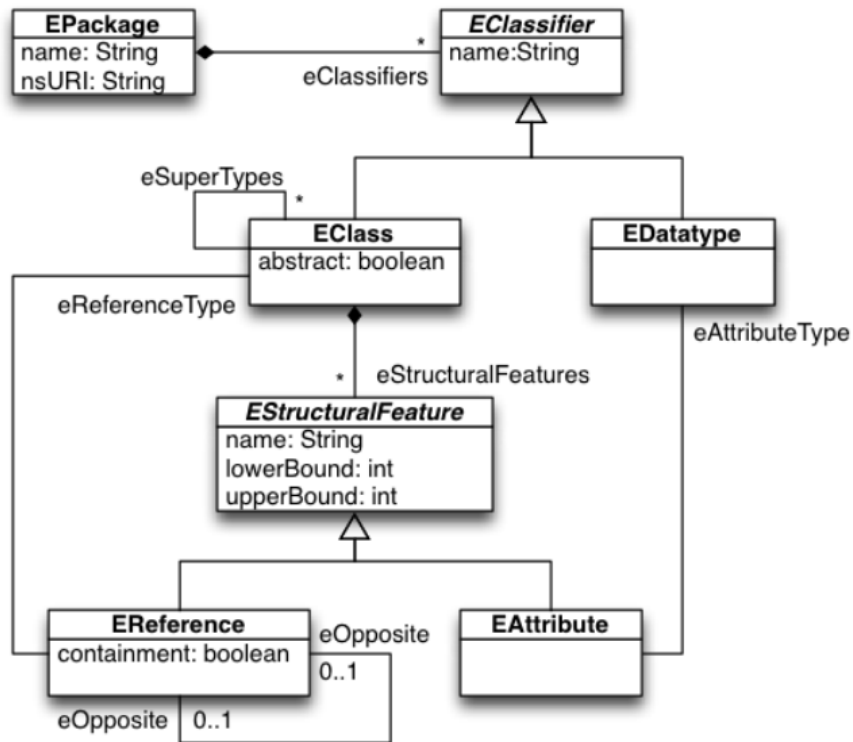


Figure 4.5: Ecore Model from [10]

In our work, we develop our modeling languages using the Eclipse Modeling Framework (EMF). It is a modeling framework that provides code generation facilities and allows to build tools and other applications based on a unified set of modeling tools and standard

implementations [10]. It handles the evolution and promotion of model-based development technologies within the Eclipse community. For instance, XMI is used to describe the model specification that could be transformed to Java classes through EMF tools and runtime support. This framework consists of three components : the core EMF which includes the Ecore metamodel to describe models and additional runtime utilities, EMF-Edit to build editors of described models, and EMF.codegen to generate options needed to build a complete editor for an EMF model through a GUI facilities. Regarding model description, Ecore is defined as the EMF metamodel (model) that allows the definition of domain specific languages using general metaconcepts like EClass, EAttribute, EDataType, etc., as presented in Figure 4.5.

4.4 Profile Extension Mechanism

There are different ways to implement a modeling tool. It could be created from scratch which could bring many challenges especially regarding the cost of its development. However, refining an existing modeling language by enriching, extending or specializing some of its general concepts, provides advantages regarding the reusability of existing language structures and tools [118]. In this thesis, we extend the System Modeling Language (SysML) to allow the modeling of the IoE educational ecosystem components as well as the different kinds of activities related to AC perspective through modeling the MAPE control loop.

Accordingly, creating a *profile*, usually associated with UML, represents a lightweight mechanism in system modeling. It consists in extending a reference model with a specific domain concepts that is related to a particular domain through *stereotypes* and *constraints*, as shown in Figure 4.6. According to the OMG, a stereotype defines a new modeling construct by generalizing a generalizable element or specifying another existing construct with new properties and constraints in order to gather the domain specificities. In the following, we describe our DSL (CPSML, EML4.0, ECPSML), specifying in each case the related abstract and concrete syntax. It is worth mentioning that all the proposed metamodels together represent the whole ECPS knowledge.

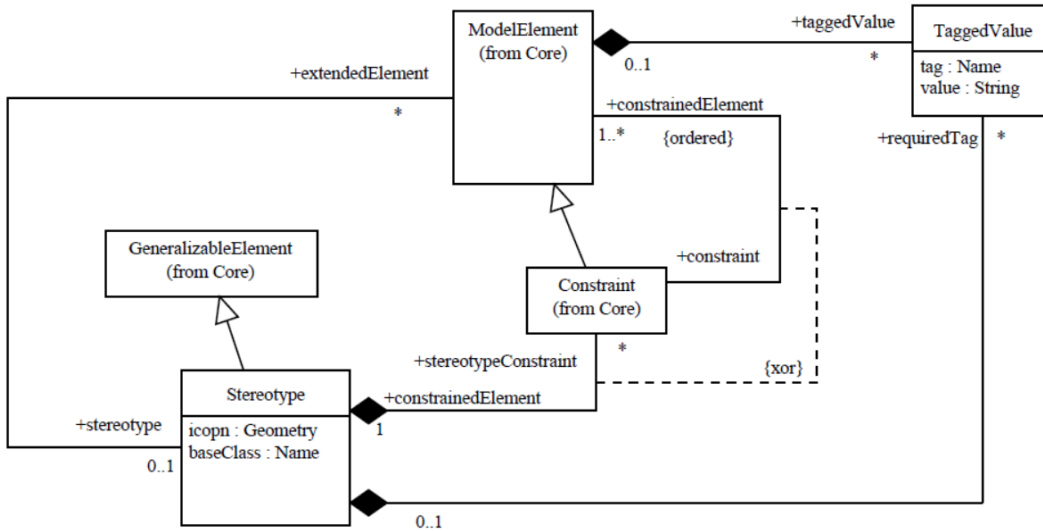


Figure 4.6: Profile Definition Mechanisms from [11]

4.5 Cyber Physical Systems Modeling Language (CPSML)

In view of our review of connected environment modeling in the first part of this thesis, we propose a specific language for Cyber Physical Systems that represents the platform definition model (PDM) at a generic level. The latter could later be coupled to any kind of domain specific language that represents the platform independent model (PIM). Our aim is to provide a highly reusable methodology in modeling a connected environment that could be responsive for most of the application domain. To do so, we reuse SysML (System Modeling Language) and take advantage of its expressiveness through stereotypes' definition.

4.5.1 SysML

According to the OMG [11], SysML is a general-purpose modeling language for system engineering and software-intensive applications. It is considered as a UML profile where the former (e.g. Block diagram) is an extended customization of the latter (Class diagram). SysML is a highly referenced modeling language among the scientific community. In the literature, SysML is a highly used modeling language, as it has proved its efficiency and reusability. Actually, it provides a solid basis for representing both behavioural and structural

aspects in the modeling of specific system features through nine diagrams. Nevertheless, it lacks specifications regarding connected environments' components. So, as it represents a key approach offering language and tool extension possibilities, it seems to be suited for us to adopt and extend for our modeling languages.

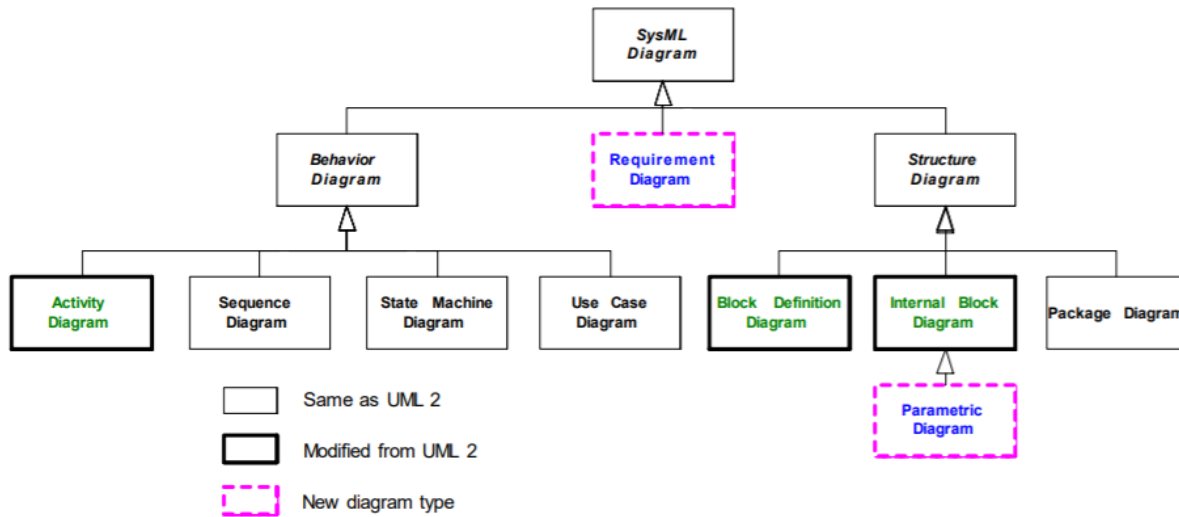


Figure 4.7: SysML Taxonomy from [9]

Sometimes, SysML reuses some major diagrams of UML in an integral way, like the use case, sequence, and state machine diagrams. However, some UML diagrams are adapted so that they fit with SysML extensions. For example, both block and activity diagrams are adapted to support SysML extensions and restrictions. In Figure 4.7, it is shown that the activity, the Block Definition Diagram (BDD) and the Internal Block Diagram (IBD) are the main ones that are modified from UML.

In SysML, the Block Definition Diagram defines the blocks, their features and their relationships (associations, generalizations, and dependencies). Another structural diagram, called the Internal Block Diagram, is meant to capture the internal structure of the defined blocks (from BDD) through (1) properties that can be values, parts, or references, (2) connectors that are considered as ports (special class of property) used to specify types of interactions between blocks, and (3) constraints that are a special class of property used to constrain other defined properties of blocks. These extensions are not supported in UML.

Regarding behavioral aspects, it is supported by a set of diagrams in order to illustrate the dynamic constructs of a system. We have a particular interest in the activity diagram that aims at describing the sequence of actions connected through inputs and outputs. Compared to UML, controls are not meant to enable actions. They rather support disabling actions that are already in the action of being executed. They could be based on complex control operations to construct control values (data) to execute actions.

4.5.2 CPSML Meta-Model

CPSML corresponds to the PDM perspective as mentioned in Figure 4.4 (1). We propose CPSML (Cyber Physical Systems Modeling Language), a specific modeling language, in order to design a connected environment regardless the domain model. CPSML is a SysML profile where we extend its reference MetaModel with IoE system specificities. Hence, our proposal handles both structural and behavioral perspectives in the modeling of specific system features. Moreover, CPS are considered as systems of systems where the components within a system interact between each other or with other systems. So it is crucial that our proposal takes into account the system components' modeling first, and then their behaviors.

Accordingly, we extend the block diagram by specifying the *block* concept through different stereotypes, as specified in Figure 4.8. According to the SysML specification [9], *blocks are modular units of system description. Each block defines a collection of features to describe a system or other element of interest.* The idea is to allow the designer to specify the nature of the used blocks (object, service, human, data) which refers to the main components of CPS illustrated in Figure 2.3. In this context, we use "stereotype" as a specification of the predefined metaclass (Block). Our defined stereotypes the followings ones :

- *Human* refers to people who are involved in a connected environment as actors. The user profile is the core element when providing services or answering a requested feature of the system. It allows to ensure the customization of the system with both static information and a constantly emitting activity system through collaboration and cooperation.

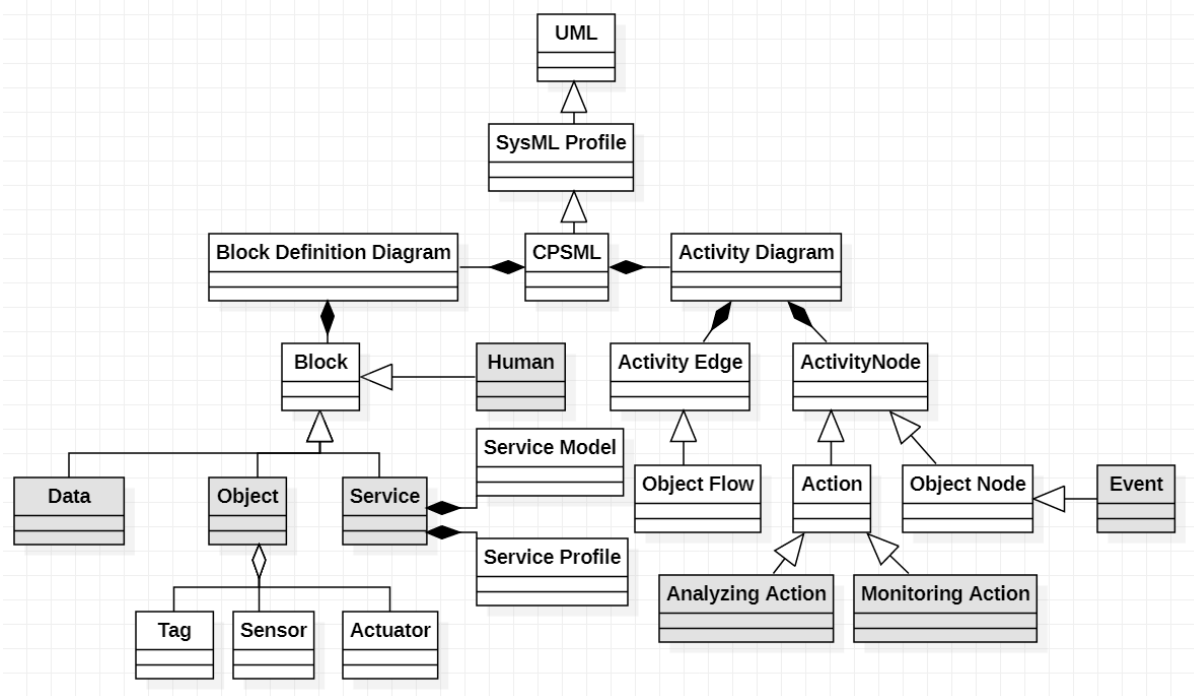


Figure 4.8: CPS Modeling Language

- *Object* refers to physical devices that can establish connection with the Internet and utilize sensors to capture environment information (as it is presented in [45]) or actuators to act on the environment. It could also refer to any resources that could provide valuable data for system processing, for instance, through tags. It is worth mentioning that our Object modeling differs from the IoT ARM proposal which considers that a sensor, a tag, and an actuator "is-a" device. We rather model it as an aggregation (white diamond) of these elements.
- *Service* refers to how people and objects must interact to generate data that can be transformed into usable knowledge through service invocation. We model a Service according to [52] where each Service is designed as a composition of the Service Model (details about attributes and relationships describing the IoT Service itself) and the Service Profile (details about the process implementing the service). These processes allow information to be addressed to the right people at the right time [45].
- *Data* refers to all the different data flows coming from the different elements of the

connected system. Cloud computing and learning analytic services are examples of technologies for data management and its transformation into information. Data could be structured, semi-structured, or non-structured in a connected environment.

It is worth pointing out on how objects and services are modeled. In Figure 4.8, it aims particularly at relying on some related works from the literature. As specified in the related works part, connected systems are often modeled through specifying either objects or services. Our aim is to give a global model of CPS gathering all the components of the IoE ecosystem. We do not particularly focus on a low level modeling, for instance the internal architecture of an object or a service are provided in the literature. Moreover, one may rely on the internal block diagram of SysML in order to model the required internal architecture of our defined stereotypes (which is not the scope of this thesis).

Likewise, we extend the activity diagram to add the Autonomic Computing (AC) [41] features and specify action types regarding the CPS control, analysis and computational capabilities to ensure the behavioral modeling a connected system. AC covers different aspects of self-management that implement control loops MAPE (Monitor, Analyse, Plan and Execute) which collect details from the system and act accordingly. More precisely, we extend the *Action* concept (which represents the behavior of the system) in SysML by defining both *Analyzing Action* and *Monitoring Action* stereotypes. We believe that they play an important role throughout the design process of the connected environment such as the explicit way to represent the autonomic properties of the CPS. Other types of actions like 'decision-making' could be simply designed as *Action* with the SysML action concept. In such systems, we also deal with events. For this purpose, it is also important to explicitly model them. In CPSML, Event is modeled as a stereotype of the *Object Node* concept. Events refers to indicators that one may model to check their states in a run-time system and acts accordingly.

By introducing these structural and behavioral concepts that define the abstract syntax (MetaModel) of the CPSML, we represent the main components of the IoE as well as a way to describe more specifically the different activities. It is worth mentioning that our model

uses modeling "architecture in the large" (as stated earlier). Again, we do not deal with modeling sensors or actuators at a low level of granularity. We consider that by definition the Internal Block Diagram in SysML ensures this requirement as it captures the internal structure of a block in terms of properties and connectors between them [9].

4.5.3 CPSML Concrete Syntax

As stated before, a DSL is basically defined through an abstract syntax and one (or many) concrete syntaxes. The latter corresponds to the human-readable notation that translates the diagram elements into a set of textual or visual constructs. For instance, a programming language such as Java uses textual concrete syntaxes, where a set of structured keywords to express the language are used. Whereas, in our case, we rely on a visual concrete syntax where a set of graphics and forms are used to handle the expressiveness of the language. As CPSML is a SysML profile. We reuse the concrete syntax of the latter.

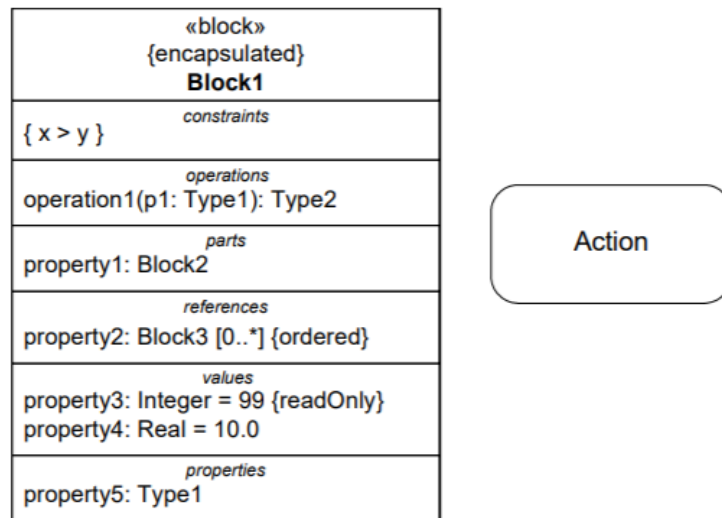


Figure 4.9: Part of the SysML Concrete Syntax

Some examples are given in Figure 4.9, particularly regarding the visual graphic representing the Block concept (used in the Block diagram) and the Action concept (used in the Activity diagram). Moreover, whenever a stereotype is defined, its name appears in the related visual notation (see models in Part III) to profile the used concept. The other notations

are presented in the specification of SysML ([9]).

4.5.4 CPSML Modeling Tool

In our ModelECPS method, we use the EMF to implement our modeling tool as it offers a set of automated mechanisms and utilities in generating either graphical editor or source code. EMF represents a promising framework as it allows open contributions to be reusable and standardized. Consequently, we implemented our CPSML profile, in the Papyrus modeling tool that we integrated as an Eclipse modeling component. Papyrus is an eclipse open source Model-Based Engineering tool. It could be integrated in the EMF to facilitate its customization and extension which considerably reduces the development time and effort. It has notably successfully been used in industrial projects and is the base platform for several industrial modeling tools [12].

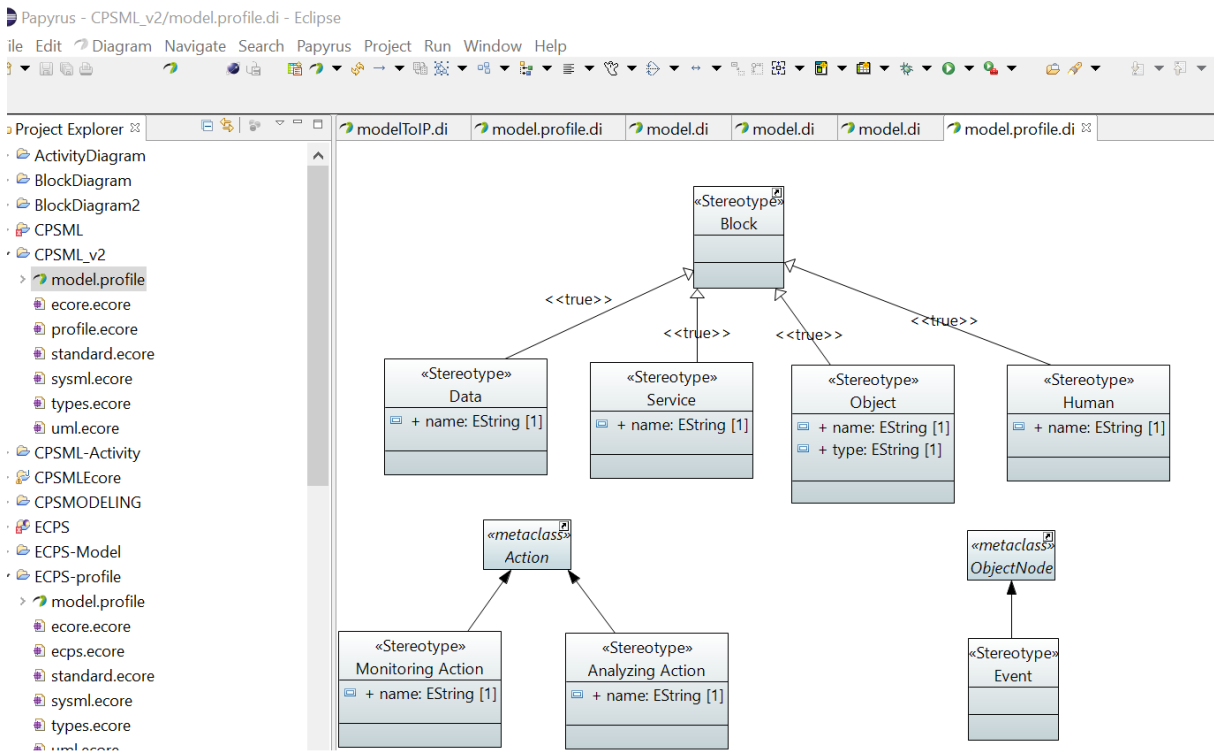


Figure 4.10: CPS Modeling Language Implemented in Papyrus as a SysML Profile

As shown in Figure 4.10, we implemented the CPSML profile using the Papyrus version

1.0.2. The main implemented stereotypes specialize SysML meta-classes with the inheritance association. Papyrus automatically integrates the new concepts to its graphical editor in order to allow the designers to use them in their own models (Figure 4.11). Papyrus allows the update of the graphical tool whenever any kind of edit happens regarding the evolution process of the metamodeling. As a result of this customization process, CPSML gains the same advantage of SysML in the Papyrus tool, from its usability to its reusability.

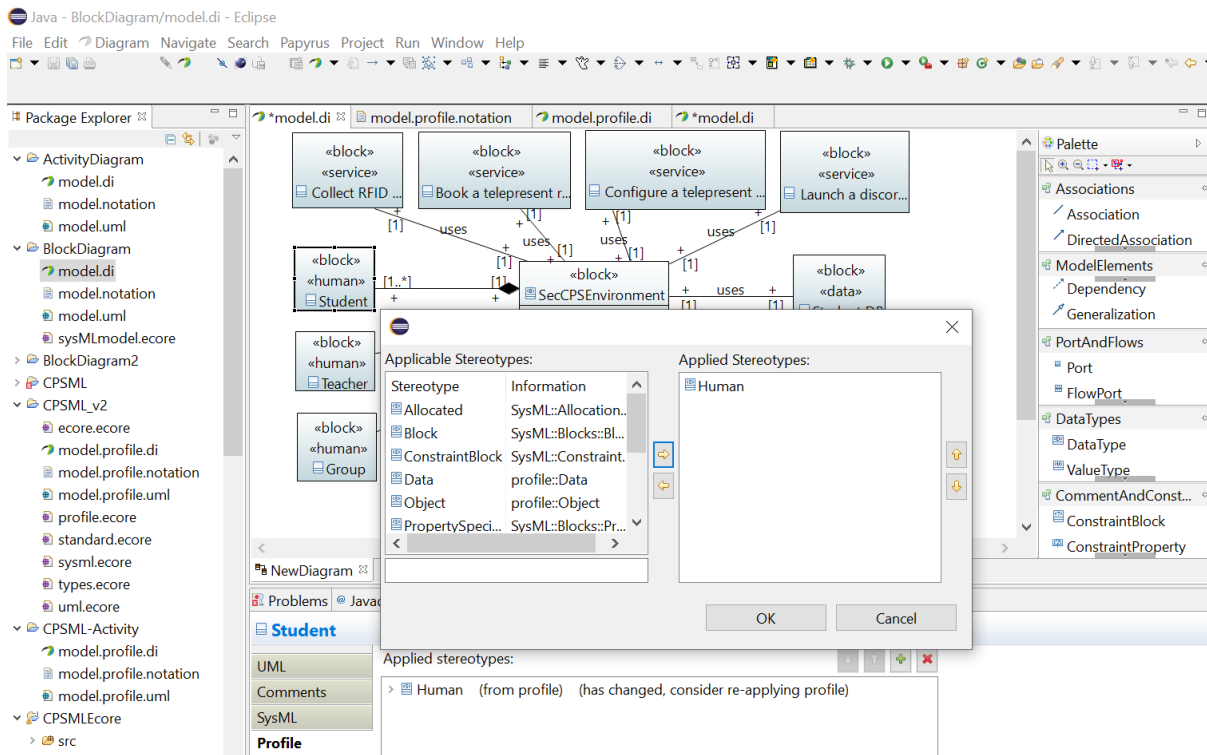


Figure 4.11: CPSML Stereotypes Integration in Papyrus

4.6 Educational Modeling Language (EML4.0)

Regarding our ModelECPS method defined below, we here describe the PIM part of the OMG Y process where we metamodel our application domain. As our interest goes to educational environments, our aim is to focus on the process where the student becomes the most involved entity. This is highly related to the classroom process which represents the basis for data management and other educational processes (course and curriculum), as

explained in chapter 3. In addition, machine-to-machine, machine-human and human-human interactions lead to a dynamic data generation in this level. Hence, the provided modeling is highly oriented to answer classroom process requirements within a particular learning and teaching context. A set of educational modeling languages were given in the related works part and a critical view point was given. In our work, we adopt IMS-LD as it represents a standard in the educational technology community and is the most referred design learning language.

4.6.1 IMS-LD

Modeling education has taken several dimensions. Various research works have been conducted since the early 2000s especially with the emergence of online learning via, for instance, moodle platforms. One may consider an educational modeling language according to a particular perspective. In our work, we build upon ideas from Activity and Content Structuring Languages, in order to ensure both structural and behavioral aspects of educational modeling in a learning process.

Our adopted EML (IMS-LD) provides a conceptual model that defines the basic concepts and relations in a learning design. The specification is divided into 3 levels (A, B and C). Level A contains the core model components of the design language. It is known as the static part. Level B and C are known as the dynamic parts of IMS-LD. Level B defines properties and conditions in order to personalize the units of learning to the learner based on if-then-else rules. Level C introduces notification behaviours based on events that occur on a run-time environment. Regarding the different levels of IMS-LD, we particularly focus on level A as the dynamic aspect of the model is ensured by the connected part of our method based on control loops.

4.6.2 EML4.0 Meta-Model

EML4.0 is our proposed Educational Modeling Language that is based on IMS-LD and other elements from the literature, designed to fit with our university 4.0 vision. It is worth

mentioning that our proposal considers both structural and behavioral elements. Regarding IMS-LD, we consider that dynamic parts refer to the setting up of the AC perspective with its MAPE control loops, our focus goes to the static part of IMS-LD and thus part of its level A. In the following, we describe the core concepts designed as structured constructs of the classroom process (Figure 4.12), inspired by IMS-LD and [112]. We describe our added concepts as well.

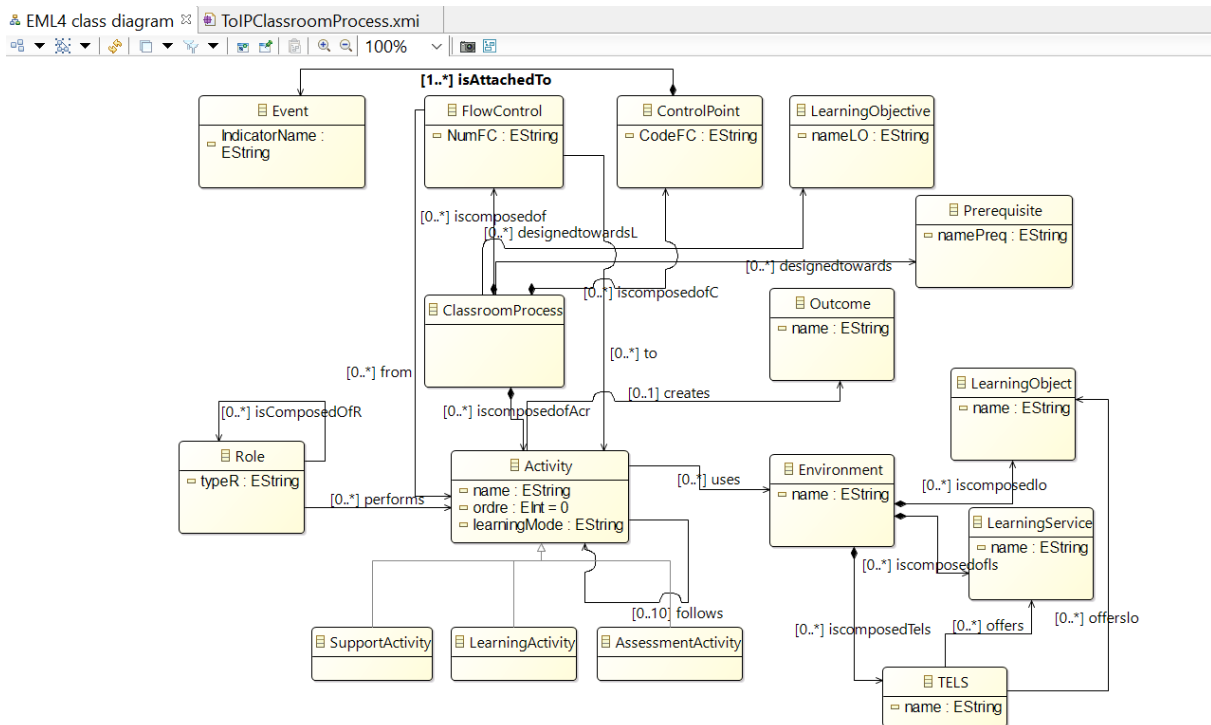


Figure 4.12: EML 4.0 MetaModel Implemented in EMF based on its Ecore MetaMetaModel

According to the adopted specification, the aim is to perform activities that are considered as scripts and description of what to do, according to a defined pedagogy or method. An *Activity* may be a *Learning Activity*, a *Support Activity* or an *Assessment Activity* (added concept regarding the IMS-LD specification). All these types of activities allow to meet *Learning Objectives*. The latter are specifications of learning *Outcomes* for the learners. Obviously, any activity requires a set of *Prerequisites* essential for the accomplishment of a task. The performer may have different *Roles* (Student, Group, Class, Staff) regarding an activity. The performer could also be performing an activity in a particular learning Mode.

Covid-19 has driven us to consider all kind of possibilities (It is also proposed in [112]). It could be a face-to-face, a distant, or a blinded learning mode. The performance happens in an *Environment* containing resources needed to carry out the designed activities. The resources could be *Learning Services* or *Learning Objects*. According to [4], a learning object is any entity, digital or non-digital, that can be used, during learning and/or teaching. It could be about multimedia contents, instructional contents, instructional software and software tools. A learning service refers to the set-up of any service that is needed during learning such as communication services and collaboration services.

The classroom process designer, often the teacher, would control the whole process referring to activities that he/she may judge critical or crucial for the understanding of the learners. For this purpose, assigning a control point is possible to allow the monitoring of the students' progress according to a flow control and events that may be interpreted during the learning and teaching processes on run-time sessions. The control flow specifies the order flow between the different activities. These designed behavioral elements are inspired by [112]. It is worth pointing out that at this level, the instructional designer only focuses on the pedagogical perspective. The "how could be these events detected, or how the control point could be implemented" questions represent the whole interest of coupling this educational viewpoint to CPS capabilities.

Finally, we extend this set of elements with a very important concept **TELS** that we consider fundamental in representing all kinds of technology enhanced learning systems, as part of the learning environment. A teacher could prepare his/her pedagogical scenario founded on a set of heterogeneous learning systems. For instance, he/she could rely on a MOOC to reinforce some basic competences that are essentials for acquiring the news ones. Also, Moodle could be provided to share learning resources and quizzes for the learners. Accordingly, a TELS is a learning system that could provide learning objects and learning services.

4.6.3 EML4.0 Modeling Tool

Regarding our ModelECPS method, EML4.0 corresponds to (2) in Figure 4.4. IMS-LD is an implemented specification based on an XML Schema. To the best of our knowledge, there is no standardised tool for it. Many attempts have been conducted in order to provide a modeling tool to design learning according to the IMS specification, however, they are not open-source and cannot support extensions as they are often at a prototyping stage or a restricted proof of concept. Due to the fact that we are in a model driven approach implemented with EMF, we realized an EML4.0 modeling tool from scratch in this framework using the Ecore metaModel [1].

Ecore is an essential part of EMF. It allows the definition of Domain Specific Languages. EMF also provides a code generator that produces Java classes from the Ecore model through an EMF generator model. The different nodes defined through Ecore EClasses, as shown in Figure 4.5 which represents the core concepts of Ecore. The EClass concept allows to define classes. The EAttribute refers to the definition of an attribute of a defined class. The EReference allows the definition of an association between two defined classes. The common super type of EDataType and EClass is EClassifier. EStructuralFeature is the superclass of EAttributes and EReferences and allows to define a name and a cardinality by setting lowerBound and upperBound. Figure 4.12 represents the implementation of the EML4.0 Ecore metaModel. Basically, we defined EClasses, EAttributes and EReferences. For the time being, implementing EML4.0 models refers simply to creating dynamic instances using EMF instead of a graphical tool.

4.7 Educational Cyber Physical Systems Modeling Language (ECPSML)

According to our ModelECPS method, our aim is to define an ECPS meta-model based on both CPS and EML4.0 meta-models. It is the result of a process fusion between the latters as presented in the Y process of our ModelECPS method presented in Figure 4.4. This

fusion is supported by an alignment concept process that aims at approaching the different concepts from both input meta-models. Obviously, not all the concepts will be considered in this fusion as our method defines three modeling languages that together define the whole ECPS knowledge. To do so, we primarily consider the concepts that are related to the IoE educational ecosystem as well as the MAPE control loops used to monitor the students' progress within learning activities.

4.7.1 ECPSML Meta-Model

The ECPSML integrates structural and behavioral concepts derived from both CPS and EML4.0 meta-models. We define ECPSML as a SysML profile too, as shown in Figure 4.13. Regarding the provided CPSML description in Section 4.5.2, the main extended stereotypes and relations are represented by a dark grey color in Figure 4.13. They are about :

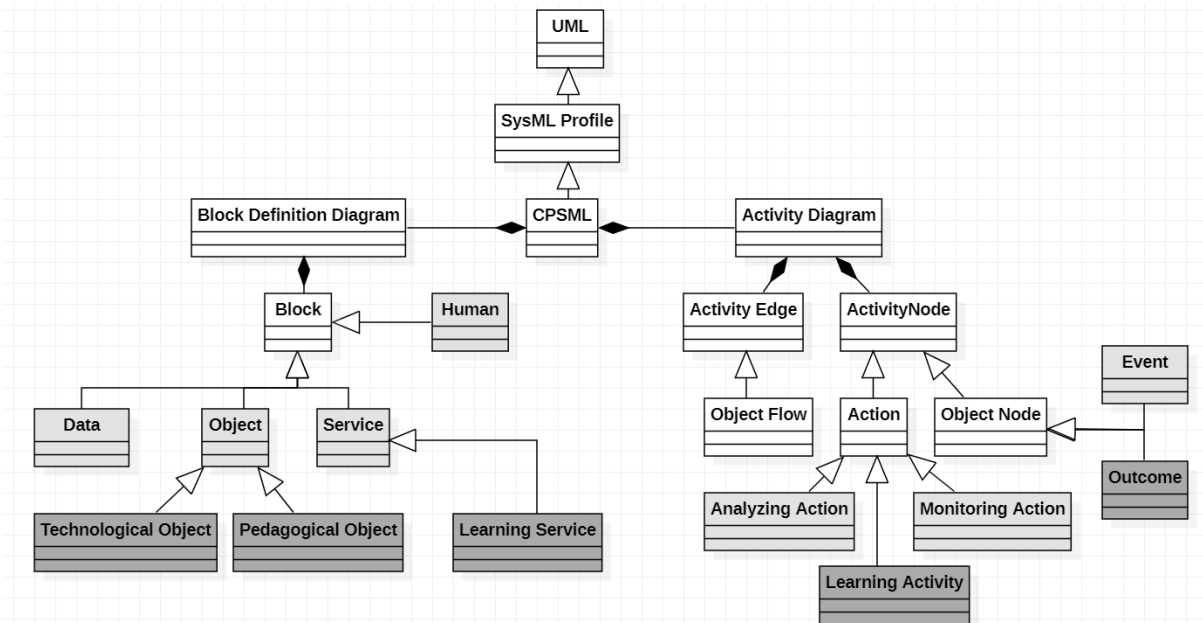


Figure 4.13: ECPS Modeling Language Designed as a SysML Profile

- Technological Object and Pedagogical Object are stereotypes that aim at specializing the available objects in the educational connected environment. For instance, a technological object may contribute to the monitoring of the general behavioral of a

student. A pedagogical object provides pedagogical resources important in the process of learning. It could also contribute to the monitoring activities if it generates data logs.

- Learning Service stereotype is-a service and so specializes this super class to support the definition of the related learning system facilities.
- Particularly, the TELS concept in EML could be represented by a SysML block in ECPSML, as we take advantage of the reusability of SysML. It could be linked to other defined stereotypes (learning service, pedagogical object).
- The previous listed extensions are about structural elements. Learning activity design has, as final goal, interactive practices. Then, it is designed as a behavioral concept specializing the action super-class in an activity diagram, in addition to the monitoring and analyzing action stereotypes (from CPSML).

As the ECPSML is a sort of combination between the pedagogical aspect from EML4.0 and CPSML, we determine a concept alignment process in order to generate the ECPSML models. Table 4.1 mostly presents the main structural and behavioral concepts and the different cases that designers may find. This list is far from being exhaustive, it is open for further extension based on empirical research.

We identified the emergence of various cases that could associate concepts from input models to an output model concept. Regarding the structural perspective, learning objects (from EML4.0) become pedagogical object (in the ECPSML model), whatever the corresponding concept on the counter-side of CPSML model (Object, Service, -). We estimate that it is interesting to qualify objects as pedagogical when they are used for learning. Obviously, a pedagogical object, regardless the fact that it could be in a non-digital format, could allow data generation when it is also technological. For this reason, a type property is specified in the corresponding stereotype (Pedagogical Object) to state if it is digital or not. Objects, Services, and Data from the CPSML model become respectively Technological Object, Service, and Data. The TELS concept is mapped to a SysML Block concept as

EML4.0	CPSML	ECPSML
Learning Object	Object	Pedagogical Object
Learning Object	Service	Pedagogical Object
Learning Object	-	Pedagogical Object
Learning Service	Service	Learning Service
Learning Service	-	Learning Service
-	Object	Technological Object
-	Service	Service
-	Data	Data
TELS	-	Block (from SysML)
TELS	Service	Learning Service
Role	Human	Human (role)
Role	-	Human (role)
-	Human	Human (role)
Control Point	-	Analyzing Activity
Outcome	-	Outcome
Event	-	Event
-	Event	Event
Learning Activity	-	Learning Activity
-	Monitoring Activity	Monitoring Activity
-	Analyzing Activity	Analyzing Activity
Control Flow	-	Object Flow
-	Data	Monitoring Activity

Table 4.1: Concept Alignment Between the Different DSL (EML4.0, CPSML, ECPSML)

TELS could host learning services or pedagogical objects. The concept "Role" from EML4.0 completes the Human "component" of the ECPSML model, and so, it becomes a property of the latter to specify if it is a student, a teacher or a staff member.

Regarding the behavioral elements (Event, Monitoring Activity and Analyzing Activity), coming from the CPSML model, they remain the same in the ECPSML output model. However, a control point coming from the EML4.0 model is mapped to an Analyzing Activity based on the related Events (defined in the EML model). Control flow is mapped to an object Flow from SysML. Data modeled in the CPSML are expected to get Monitoring Actions in order to later translate them into software components that are required to collect and gather data for further analysis.

In addition, associations between the different concepts are supposed to remain the same in both Block and Activity diagrams (CPSML, ECPSML). In this fusion, the ECPSML block diagram will be somehow upgraded by concepts coming from EML4.0 (e.g. learning services and pedagogical objects). Hence, associations in the CPSML block diagram remain the same unless some are removed because of their “inutility” in the ECPS. For instance, a designed data in the CPSML block diagram could be not retained in the ECPSML model as it is not useful in the validation of any of the modelled learning indicators. This is expected to be approved by both experts when it comes to validate the output model during the collaboration phase. Furthermore, both experts are expected to validate the output model when things seem incomplete regarding the concept alignment in Table 4.1. For instance, they design the required associations related to concepts from EML4.0 as they will be associated to other ones in the ECPSML model.

```

1  -- @path CPSMM=/CPSML_v2/model.profile.di
2  -- @path EMLMM=/ECPS-v2/MetaMordels/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 4.14: Part of the ATL Transformation Rules

The generation of our ECPSML model is supposed to be automatically handled by the ATL module. As described earlier, ATL is a model transformation language. It is integrated

in the EMF which, in addition to its modeling framework features, provides facility model generation for building tools and other applications based on the different XMI models. Rules are the heart of ATL transformations. They describe how output elements (ECPS) are produced from input elements based on the input model(s). Input models could be multiple which is the case of our method (CPSML, EML4.0). Each rule expresses a mapping between a (or multiple) input element(s) and an output element. Figure 4.14 presents part of our ATL rules (A Learning Object in the EML4.0 model becomes a Pedagogical Object and an Object in the CPSML model becomes a Technological Object). According to the different types of model transformation, listed below, our contribution confirms an exogenous transformation as transformations are carried out between expressed models using different languages and a vertical transformation as the source and target models reside at different abstraction levels. Figure 4.15 presents the Model transformation process based on the defined metamodels.

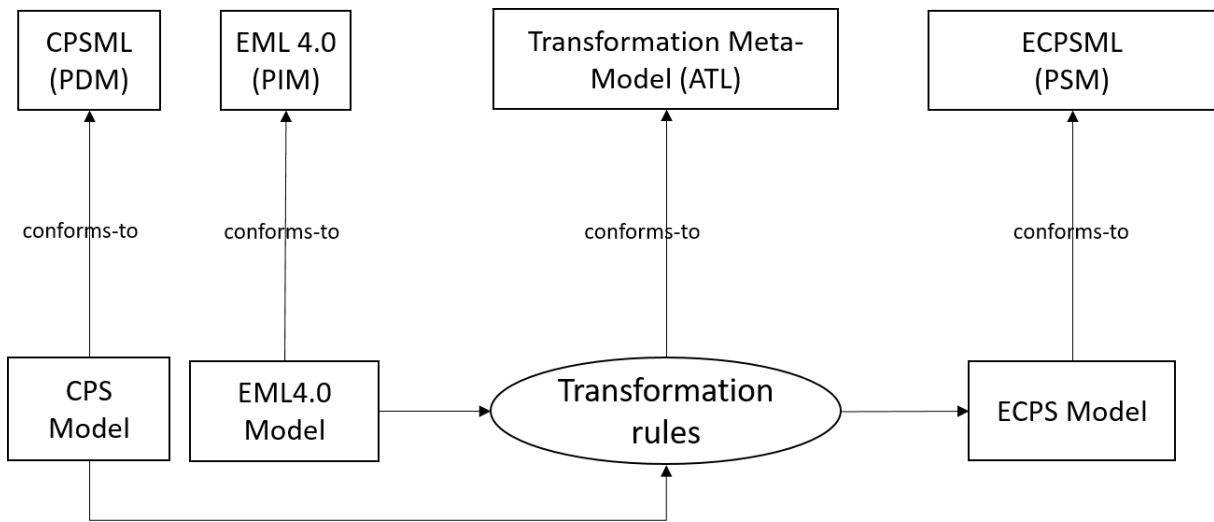


Figure 4.15: ModelECPS Transformation Process

Both CPSML and ECPSML are SysML profiles. Likewise, we reuse the concrete syntax of the latter in order to take advantage of the reusability and expressiveness of its DSL to model ECPSML.

4.7.2 ECPSML Modeling Tool

We reuse Papyrus to also implement the modeling tool of the ECPSML based on the definition of Stereotypes, as shown in Figure 4.16. As mentioned before, this tool proved its worth as it allows the update of the graphical tool whenever any type of edit occurs regarding the evolution process of the metamodeling, in addition to its usability and reusability.

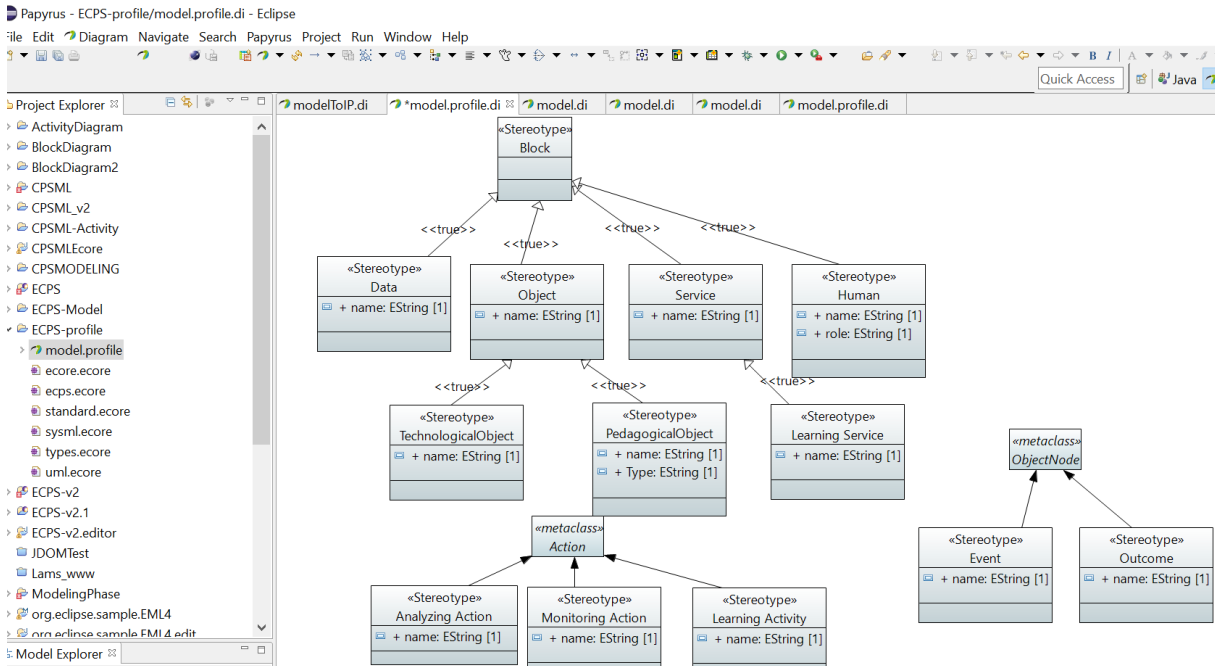


Figure 4.16: ECPS Modeling Language Implemented in Papyrus as a SysML Profile

It must be emphasised that the objective is not only to provide a pedagogical modeling tool for educational and connected environments, but it is also meant to retrieve the different blocs of elements where MAPE control loops could be applied. A control point designed at a pedagogical level, coupled with the abilities and capacities of deployed cyber systems will result in a merged, consolidated, monitored and analyzed environment. After all, ECPS consists of a big data management system, in which the components of the IoE educational system are treated as data.

For implementation reasons, we generated Ecore MetaModels for both CPSML and ECPSML after their extensions using Papyrus. The aim is to unify the DSLs as EML4.0 is implemented from scratch using Ecore. This facilitates the implementation of ATL rules. For

this purpose, EMF provides generator tools to support the required transformation process.

4.7.3 ModelECPS - Modeling Process

After presenting the different DSL proposals, we provide, in the following, a process that situates the designers regarding the use of the defined modeling languages. This process is based on a sort of a requirements' specification where the educational context and instructions are given and discussed between both designers (CPS and pedagogical experts). In fact, the pedagogical expert models the educational environment by instantiating the EML 4.0 concepts. The core elements are mainly about learning activities on which the pedagogical expert rely to identify the required environment (learning object, learning services, roles, etc.). Learning activities have to be designed according to their order flows. After this, he/she defines the used technological learning systems (TELS), as well as the learning objects and services. Then, the expert stresses specific activities that he/she judges critical for students during the learning process. Based on that, a set of control points could be identified/defined to keep track of the students' progress. Control points are translated by checking the predefined events (learning indicators).

From a connected system viewpoint, the CPS expert designs the provided components of the IoE educational ecosystem. At this level, the aim is to focus exclusively on connected components and not handling pedagogical issues. At first, the expert models the provided smart objects, services related to, for instance, communication / collaboration system, humans and data. The expert assesses the whole setting in order to retrieve some elements. Drawing interactions between the modelled components ensures a better awareness of the ubiquitous environment. Then, the expert models indicators through events which could give feedback about, for instance, the students' presence or a particular physical state. Based on this, the expert defines the required monitoring and analyzing activities.

Based on these two models which confirm to correspondingly EML4.0 and CPSML, both experts collaborate in order to couple and integrate their modelings. The CPS expert supports the pedagogical expert in defining the required monitoring and analyzing activities

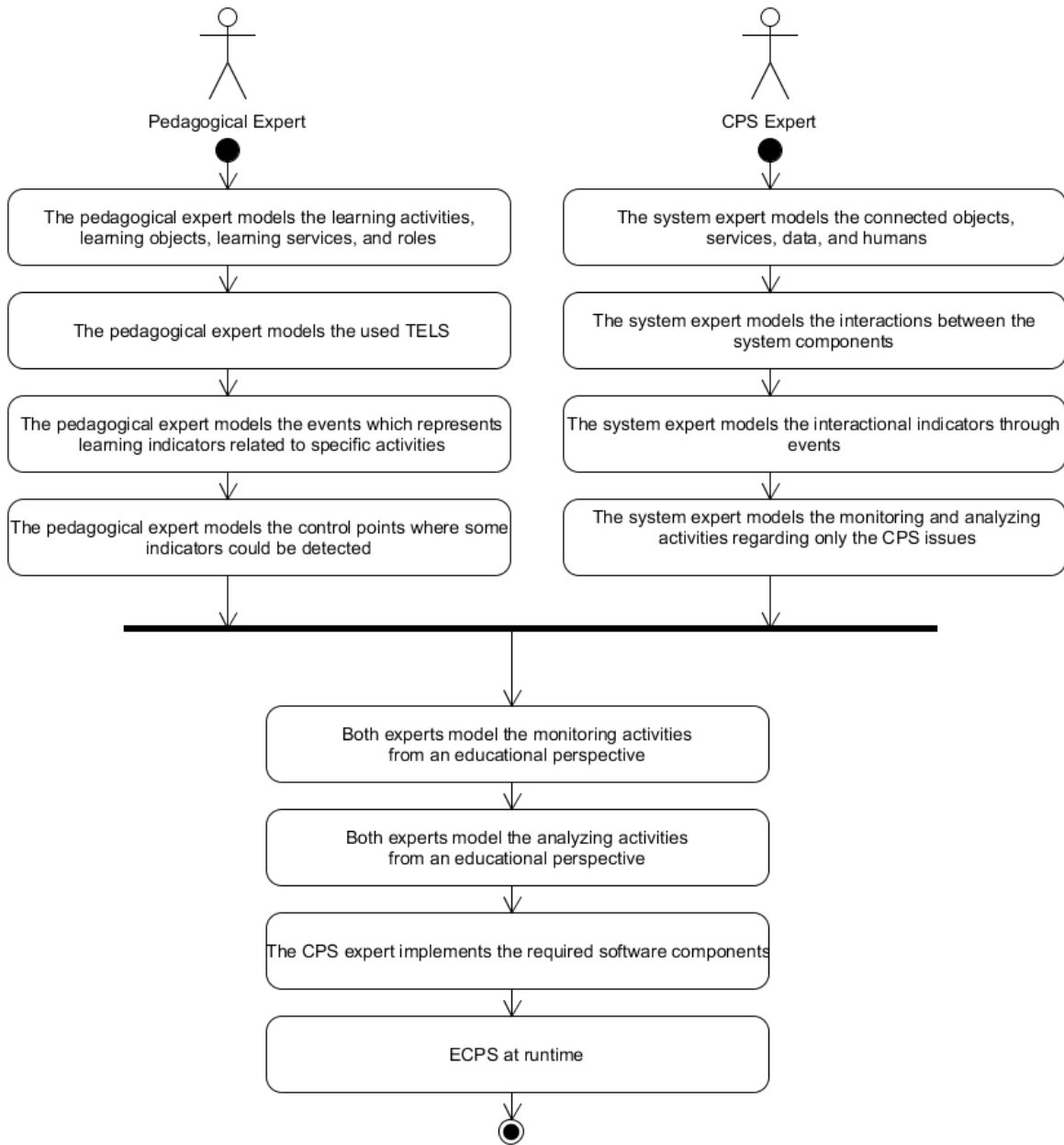


Figure 4.17: ModelECPS Modeling Process

that aim at carrying out the predefined control points at an educational level. The feasibility of these control points would depend on the available designed components of the CPS, especially regarding the provided data logs. This merge is undoubtedly the purpose of the ECPSML. The CPS expert implements the necessary services and software components in

order to accomplish the required MAPE activities.

Part II Conclusion

In this part of the thesis, we give our vision of the university of the future. First, a generic architecture of the educational connected environment is given through the definition of ECPS which are educational complex systems constituted of physical and cyber components that aim at implementing control loops to manage the educational processes according to the educational stakeholders' requirements. As pointed out previously, the connected components are not only about devices (IoT), but, they refer to all the components of the IoE ecosystem (objects, services, human and data). Then, three levels of ECPS are given based on elements of the literature and how educational processes are managed at the university. The classroom process consists in the short-term implementation of a pedagogical scenario during real classroom sessions. Course ECPS and Curriculum ECPS are implemented to adapt either the course process (mid-term) or the whole curriculum process (long-term).

Then, we focused on the classroom level and provide more in-depth its design through the ModelECPS method. Actually, we adopted the classical “Y development cycle” from MDA as the process of our method. The objective of this process is to enable the generation of Platform Specific Models (PSM, ECPS in our case) from Platform Independent Models (PIM, EML4.0 in our case) and Platform Description Models (PDMs, CPSML in our case). We proposed modeling tools for both EML4.0 and CPSML MetaModels. Regarding ECPSML, our aim is to automatically generate its model based on transformation rules that we implemented using the ATL transformation language. All the rules were not implemented, due to technical issues. It is worth reminding that our models are built upon ideas from the literature to foster usability and reusability of a reference architecture and a set of standardized tools. A utilisation process of the modeling languages is given after providing their description. As meta-modeling aims at creating some sort of abstraction level. The next part of this thesis brings forth a concrete case study experimentation in order to enlighten this and design concrete models confirming to the proposed meta-models.

Part III

VALIDATION OF OUR APPROACH

- Towards University 4.0

Chapter 5

Case study - Covid-19 Educational Context

Part III Introduction

In this part, we will provide a case study implementation in order to illustrate and put into practice our proposed modeling languages. Basically, our work aims at providing a vision for what would be a highly connected university. Due to the health crisis, we have found ourselves in a perfectly-fit-situation with what we are designing at an educational level. This crisis has particularly amplified the intensity of online services' request and humans' engagement. During the three lock-downs in France, the teaching staffs put in place solutions to ensure educational continuity. In the first one, a new term appeared to describe the switch of teaching and learning from a fully face-to-face to a fully online mode. It is about emergency-based teaching [16]. The purpose was not about developing learning tools from scratch, but rather about adapting the access and use of different existing online platforms. Thus, the teachers were derived to handle not only pedagogical challenges but also technological ones.

During the first lock down, learning and teaching processes were ensured fully online, whereas during the second and third, flexibility and convenience of real classrooms were made available under strict constraints to keep the academic activities going. This was due

to challenges a high number of students were facing, for instance connectivity and psychological issues [76]. Thus, presence mode was blended with online mode at various universities. We conducted several observations and experimentations during the third lock down (January to June 2021). We particularly focused on practical work sessions of the ToIP (Telephony over Internet Protocol) course that was ensured by two teachers for more than fifty students (organized into four practical work groups) in a blended context in the Networks and Telecommunications department (Réseaux and Télécommunications - RT in French) of our institute. The educational context is presented in Table 5.1.

Institution	IUT - Mont de Marsan, France
Department	Network and Telecommunication (RT)
Period	January - June 2021
Course	Telephony over IP (ToIP)
Presence mode	Blended
Number of students	54
Number of teachers	2

Table 5.1: General Context of the Case Study

ToIP is a technology that is based on establishing communication between hardphones or softphones in a local or large-scale organization. This technology is based on VoIP (Voice over IP). It ensures the setup of a communication session regarding services like signalling, transfer, on-hold and voice message. The main objective of this course is to address the different technologies in order to allow the students to mainly set up a private telephony network (phones + server + network devices), configure different services (e.g. messaging service), and establish a connection through a public network (internet, ISDN).

Figure 5.1 shows the infrastructure used during the ToIP practical works. These sessions are organized according to a collaborative and blended process. In fact, peer working is achieved through a student in present mode and another in distant mode. So, only half of the students are present in real classrooms. Actually, the purpose is to enable the present student to interact with physical phones by performing the required configurations and connections. The remote student will interact with his own softphone (a software that emulates

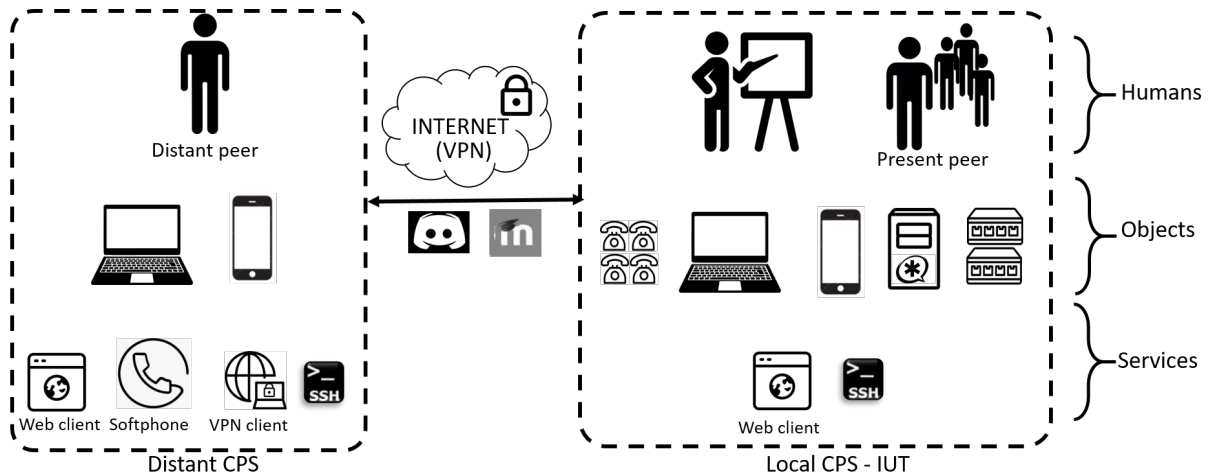


Figure 5.1: Infrastructure of the ToIP Course

a physical phone on a computer). Then, both of them will interact remotely with an Asterisk server to which all the phones are connected. It is expected that the student peers interact with each other to set up the configuration of the server based on given instructions. The communication channel is ensured via a Discord server. Moreover, specific guidelines are given to the present or remote students when needed, as shown in Figure 5.2.

Actually, there were 4 working sessions with 4 student groups. It is worth pointing out that the same CPS setting was designed for all the monitored sessions. However, it is not the case, in general, at a pedagogical level as learning objectives evolve from one session to another. And so, the pedagogical model is different in each classroom process. As far as the ECPS classroom process is concerned, the implementation of the different layers is conducted through generated data from different sources. The latter are either static or dynamic when it comes to a real time monitoring of the educational ecosystem components.

This presentation of the general context gives a glance of the ECPS through a part of its infrastructure (CPS) and how activities are organized in a pedagogical way. In the rest of this part of the manuscript, more details will be given regarding both perspectives through their models (a CPSML model for the CPS infrastructure, an EML4.0 model for pedagogical concerns, and a generated ECPSML model). The modeling will obviously be based on our proposed meta-models and tools in order to illustrate their expressiveness. We present both

[étudiant présentiel]

Câblez vos trois hardphones sur les prises terminales de la goulotte reliées à la baie de brassage. Faites le brassage dans la baie de brassage pour connecter ces téléphones au switch Cisco. Configurez leurs paramètres IP (et le serveur TFTP pour les aastra)

Créez les profils SIP « patron » et « technicien » sur Asterisk.

Créez les fichiers de configuration des postes aastra pour qu'ils s'enregistrent avec les bons profils sur votre asterisk : patron sur poste 57i, technicien sur poste 55i

Créez le plan de numérotation suivant : patron (101), technicien (102).

Créez une boîte vocale pour les postes patron et technicien. Faites en sorte que si ces postes ne répondent pas au bout de 10s, alors l'appel sera dirigé vers la boîte vocale.

[étudiant à distance]

Créez les profils SIP « standard » et « commercial » associés respectivement au GrandStream et au Linphone. Ces deux profils seront en enregistrement automatique. Vous devez attendre que l'étudiant en présentiel ait configuré l'adresse IP du GrandStream pour pouvoir le configurer par interface Web.

Créez (ou complétez) le plan de numérotation interne suivant dans Asterisk : standard (100), commercial (103)

Créez une boîte vocale pour les postes standard et commercial. Faites en sorte que si ces postes ne répondent pas au bout de 10s, alors l'appel sera dirigé vers la boîte vocale.

Créez l'extension permettant d'aller lire les messages dans les boîtes vocales, et faire en sorte qu'apparaisse une notification sur les postes quand un message a été déposé.

Figure 5.2: Example of Task Sharing Between the Remote and Present Students (in French)

structural and behavioral aspects of the implemented ECPS classroom processes. Then, we give an insightful view of the Learning Analytics (LA) layer where an implementation of control points are given to enable the monitoring and analyzing activities at run-time, as well as the generated dashboards that give teachers feedback about their students' progress.

5.1 Connected Aspect Modeling (CPSML Model)

Regarding the provided modeling process in Section 4.7.3 and referring to the CPSML, the purpose here is to model the main components of the IoE educational environment and the different potential actions (monitoring, analyzing, etc). This is done by the CPS experts who define the required models following the pedagogical expert requirements. In fact, the given block diagram in Figure 5.3 somehow represents the structural infrastructure modeled

through the different components that are used during teaching and learning activities in all the practical work sessions (as shown in Figure 5.1). It shows the use of our modeling language in modeling the structural connected aspect in this case study through a block diagram. The extension of Papyrus allows us to first apply the CPSML profile to the whole project, and then to apply the defined stereotypes for the required IoE components in order to specialize them : Object (in yellow), Service (in green), Data (in grey) or Human (in red). Each time a component is modeled, our modeling tool shows it graphically under Papyrus. The latter are physically situated in different physical spaces : the real classroom, the distant student space, and over the Internet, as described in Figure 5.1.

As we extend SysML, we take advantage of its expressiveness to model associations and compositions as well as SysML blocks (in white). As students are dispatched into two environments, the engaged components are qualified differently under two SysML blocks ("Distant CPS" and "Physical CPS"). The "Connexion Support" block is designed as a composition of two main services (VPN service and Discord Service) that ensure the interconnexion between both distant and physical CPS as well as the human-to-human interactions between distant and present students over the discord service. In this diagram, services are often hosted by objects, for instance the distant computer hosts the distant web client, the distant SSH client, the softphone and the VPN client. In our case, different logs "Data" could be generated either by services or objects.

The dynamic aspect of the system could be designed through the monitoring of the generated data. The latter are mainly tracked through service invocation using the provided objects. Figure 5.4 shows an activity diagram, and so a behavioral aspect, to implement a MAPE control loop over the whole system while permanently collecting the different logs throughout a working session. The "Service logs" output from this activity diagram contains various types of logs related to the interactions of the students with the Asterisk server. These logs are specified later in Section 5.4. Here, the CPS expert designs all the possible data logs that could be generated, their formatting and saving. Their concrete exploration will depend on the final application (EML4.0 model in our case). In this activity diagram, we specialize the type of action based on the loaded stereotypes from the profile extension

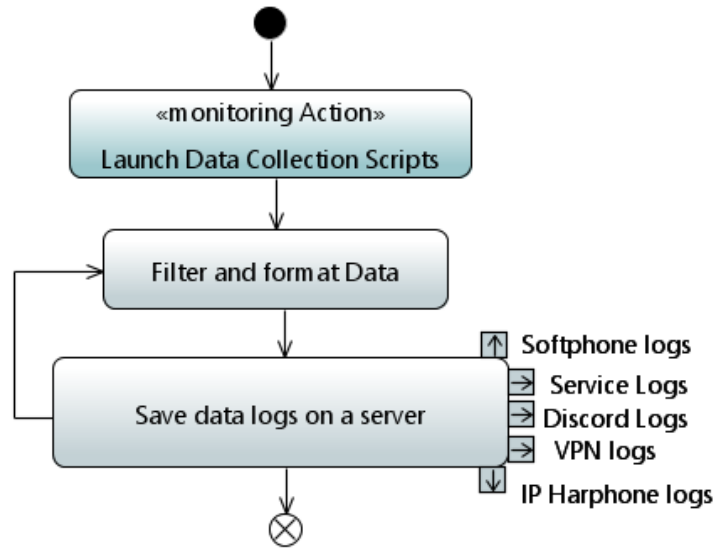


Figure 5.4: A CPSML Activity Diagram for a MAPE Control Loop

mechanism, for instance the monitoring action is mentioned in a grey color. This MAPE control loop is designed independently from pedagogical concerns so it could be reused for each practical work session in the ToIP course.

5.2 Pedagogical Aspect Modeling (EML4.0 Model)

Our modeling method deals with both PIM and PDM perspectives. In this part of the case study, we focus on our application domain model (PIM) which is about classroom processes during the ToIP teaching as previously detailed in the case study context. We will focus more particularly on designing learning activities as well as some other related concepts to the learning environment. At this stage of modeling, it is for the pedagogical expert(s), in our case two teachers, to identify and model the different practical work sessions based on the EML 4.0 modeling language. They do not need to know more about technological details when modeling the pedagogical elements. We present both structural, through basically the components of the learning environment (learning objects, learning services and used TELS), and behavioral aspects, through learning activities and control points, of the studied scenario.



Figure 5.5: Manual Draft of the Learning Scenario

Several sessions (four) were designed during this course. As stated above, the main objective of the ToIP course is to address the different technologies in order to allow the students to mainly set up a private telephone network (session 1), to configure different services (session 2), and to establish a connection to a public network (session 3). Each session is represented through a separate EML4.0 model, however, they all are happening in the same CPS infrastructure. The different ECPS classroom models are generated by merging each practical work session model with the unique CPSML model (Section 5.1). We detail in particular the first practical work session model as our primary aim is to illustrate the use of our modeling proposals. Figure 5.6 shows the pedagogical model scenario of the learning session which is represented as an implementation of dynamic instances of the

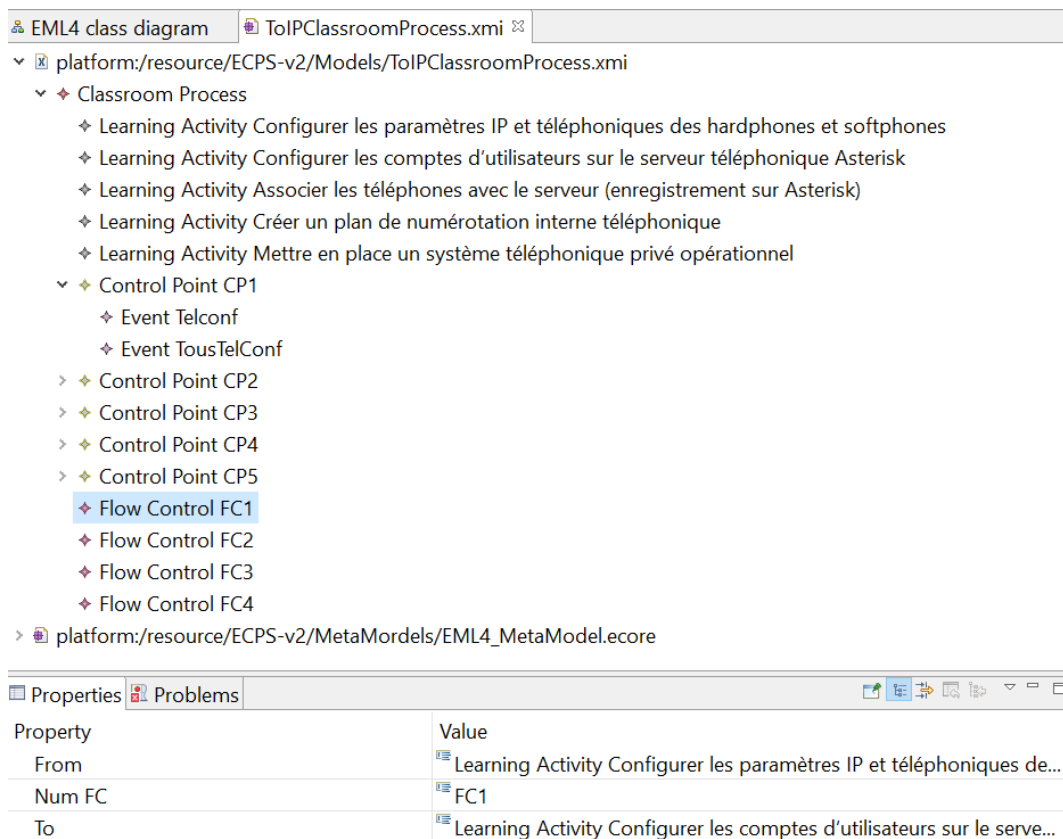


Figure 5.6: Modeled Learning Activities, Control Points, Control Flows and Events (confirms to EML4.0)

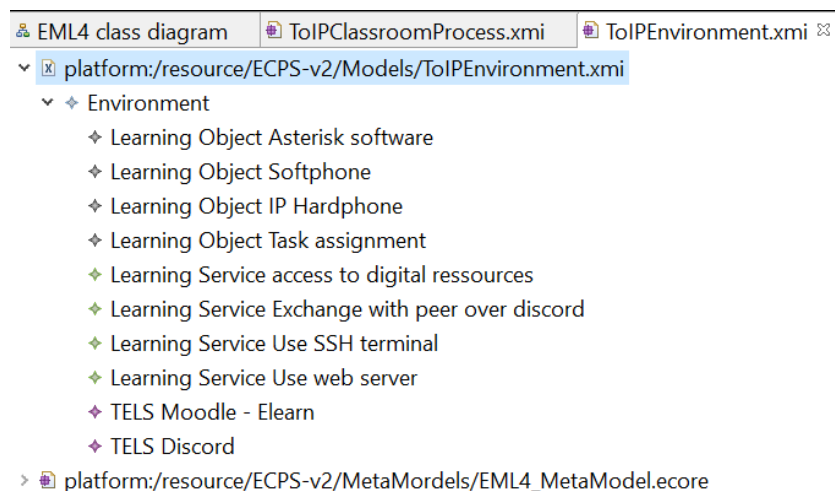



Figure 5.7: The Modeled Learning Environment Through Learning Objects, Learning Services and TELS (confirms to EML4.0)

EML 4.0 modeling language in EMF. It is a model transcription of the pedagogical expert manual draft that is shown in Figure 5.5: learning activities are represented as rectangles, control points are between brackets and control flows are between learning activities (arrows). In the same way, the learning environment is described through learning objects, learning services and TELS and is implemented through the creation of dynamic instances in EMF, as presented in Figure 5.7.

The designed classroom process in Figure 5.6 shows the different learning activities and specifies the order flow through the use of control flows. In our case, the learning activities are meant to be sequentially ordered. But it is not always the case, for example if two learning activities can be accomplished independently (in parallel from a modeling view), the "from" and "to" properties (see the bottom of the Figure 5.6) of a control flow could be attached to one or many learning activities. Also, several control points are designed in order to check the students' progress during the working session. A control point represents the set of learning indicators (Events) that are meant to be checked through the implementation of MAPE control loops. In this example, each control point is related to a specific learning activity, which may be different in another case when we do not have control over some learning activities. For instance, the CP1 control point in Figure 5.6 refers to the learning indicators (Events : TelConf and TousTelConf) that the pedagogical expert would like to implement in order to monitor their validation by the students who are supposed to properly configure all the phones.

5.3 ECPS Model

Regarding our ModelECPS method, the ECPSML model refers to our Platform Specific Model (PSM). As we mentioned before, our case is about a model-to-model transformation. In fact, Figure 5.8 shows part of the generated model based on both the CPSML model (Figure 5.3) and the pedagogical session model (Figure 5.7) through the implementation of two ATL rules (A Learning Object in the EML4.0 model becomes a Pedagogical Object and an Object in the CPSML model becomes a Technological Object). These transformation rules are implemented according to the concept alignment in Table 4.1. In fact, this process of model merge aims at representing both educational and technological aspects of the ECPS through a tight coordination between both CPS and pedagogical experts. At first, the pedagogical expert only cares about what to learn and how to achieve by students based on pedagogical learning objects and services. From the other side, the CPS expert examines the provided technology in order to identify the useful components, Objects in this example. Then, in the ECPSML model, each Object is qualified by the required properties (pedagogical or technological). Due to implementation reasons, the complete concept alignment process was not implemented using ATL. We are currently investigating the issue.



```
ECPSGeneratedModel.xml ❏
<?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="Distannt 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 5.8: Automatic Generation of a Part of the ECPSML Model Through the Transformation Rules given in Figure 4.14

Figure 5.9 shows the structural aspect of the ECPS through the definition of a graphical block diagram directly generated from Papyrus using our extended stereotypes. This is supposed to be generated automatically by the ATL module as in Figure 5.8. Regarding the CPS block diagram model, presented in Figure 5.3, the pedagogical fusion could be seen through

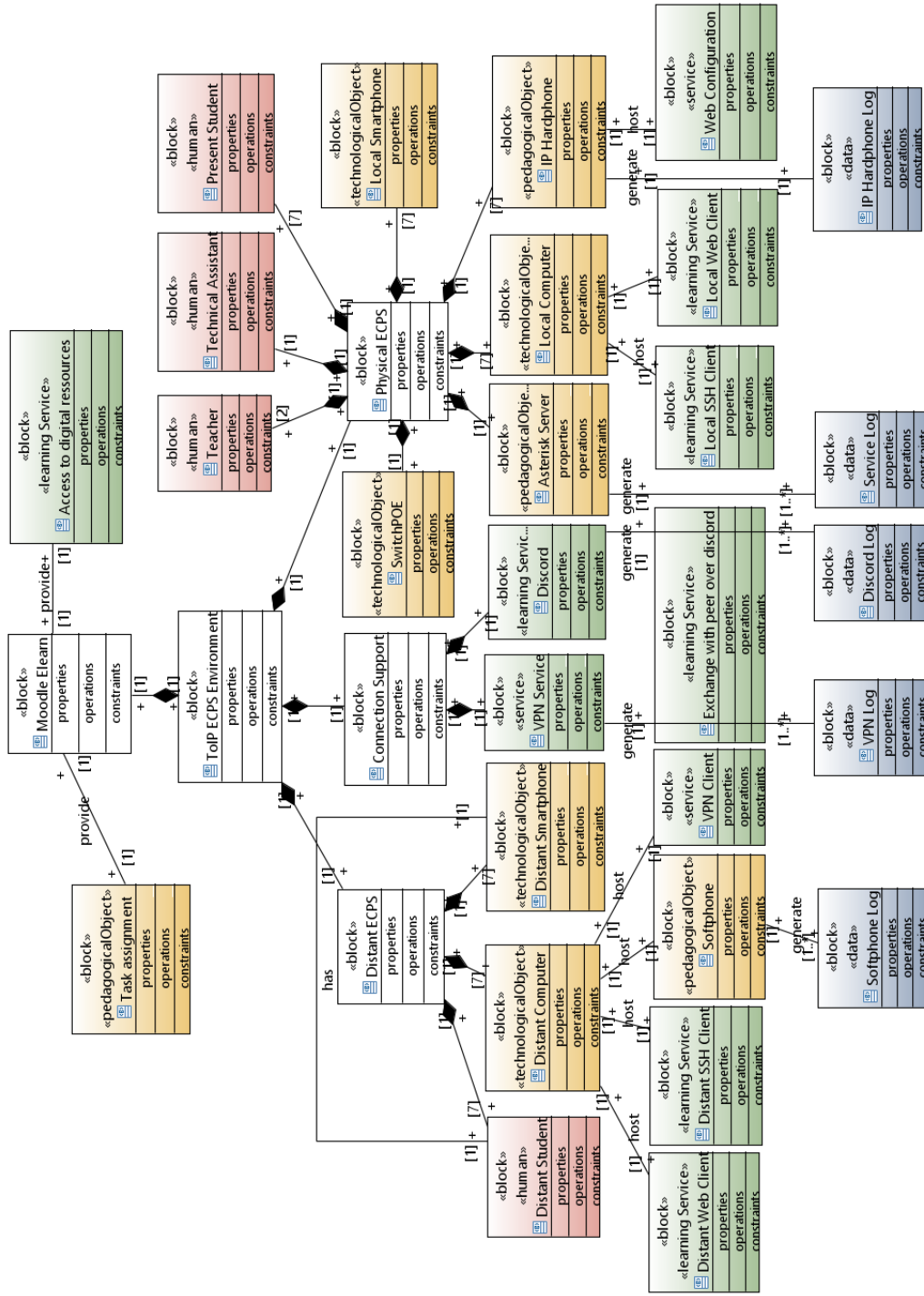


Figure 5.9: The ECPS Block Diagram Generated Directly From Papyrus Corresponding to the Defined Concept Alignment

basically the identification of human roles, the used TELS, the learning services, as well as the pedagogical and technological objects. The technological and pedagogical objects are presented in yellow, services and learning services are highlighted in green, humans are in red, data are in grey and SysML blocks are in white. Regarding the defined concept alignment, the learning objects (Asterisk server, Softphone, and Task assignments) become pedagogical objects even if the Softphone is considered as a service in the CPSML. The pedagogical aspect dominates the output transformation. Also, the defined TELS, for instance Moodle - Elearn becomes a sysML Block that provides a learning service (Access to digital resources) and a pedagogical object (Task assignment). Data elements remain the same regarding the CPSML block diagram.

Regarding the behavioral aspect of the ECPS, it is modeled through an activity diagram implemented on Papyrus after the application of our defined SysML profile, as shown in Figure 5.10. And so, we can see the specialization of the "action" concept to specify in each case if it is a learning (in green), a monitoring (in grey) or an analyzing activity (in yellow). The object nodes could be represented through output pins in Papyrus as it is not supported as an output node from the activity in this tool. The designed MAPE control loops in the CPSML model are kept identically in the ECPSML model. Data logs are aligned with monitoring action in the ECPS. This is for example the case of the "Data log collection" monitoring action, which deals with collecting, filtering, and formatting data coming from objects and services modeled in the CPS (Figure 5.4). Learning activities and flow controls come from the EML4.0 model, and are also kept identical. Control points are transformed into analyzing actions, according to our concept alignment in Table 4.1.

The modeled events represent the pedagogical indicators which are directly linked to the designed control points as output pins. Each analyzing action takes as input the required data logs (designed as input pins - coming from the "data logs collection" monitoring action) to check the related learning indicators. The monitoring activity to generate a dashboard also runs permanently as a listener for the validation or not of the learning indicators in order to display their correspondent state. At a run time level, these events will be implemented into software components that will allow the generation of the required dashboards regarding

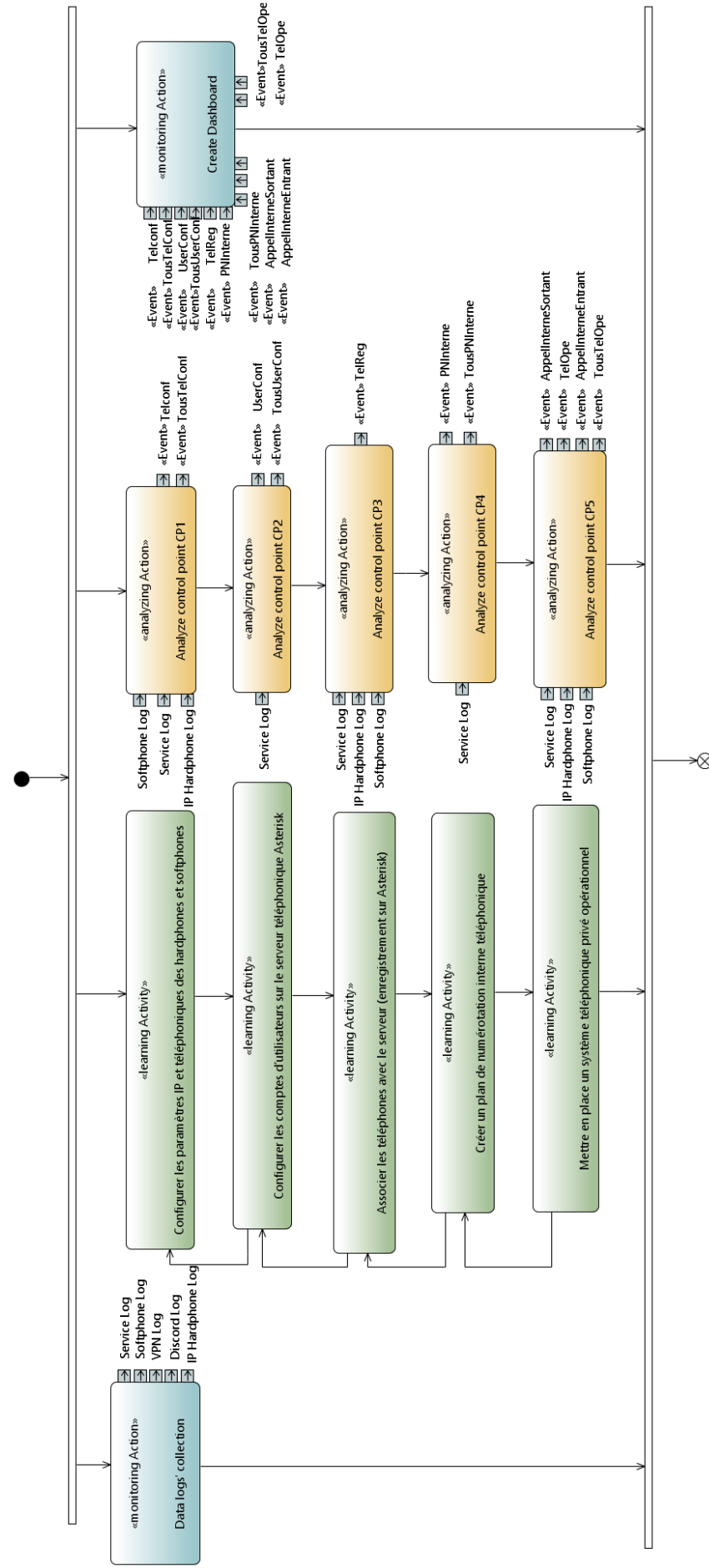


Figure 5.10: The ECPS Activity Diagram Generated Directly From Papyrus Corresponding to the Defined Concept Alignment

the student's progress.

5.4 ECPS on Run-Time

Our designed models are used to specify the structural and behavioral aspects of the ECPS. Software components are needed to implement (in a low level) our models and somehow translate the defined learning indicators (events) in order to see a concrete result of students' progress monitoring in our real study case. In fact, incorporating a learning activity into MAPE control loops seems very promising as it gives a concrete feedback regarding the achievement of learning objectives by the students. The CPS expert translates the pedagogical metrics into a concrete data management process and verifies its feasibility regarding the provided data. Hence, our modeling phase is powered by further software components in order to concretely implement the required MAPE control loops.

To do so, the implementation of the ECPS should be supported by a set of pre-analysis and modeling of the IoE components, in particular regarding data logs. Various services and objects are connected to allow the teachers to teach and the learners to learn, and so many interactions to generate data happen. For this purpose, the data collection process is somehow centralized over a particular server that orchestrates the used infrastructure to gather logs from "everywhere". This phase of data collection and analysis was assisted by three interns (Valentin Lubon, Mathieu Fouine and Matteo Guyon). In the following, we present how an example of a learning service (interactions over Discord) and another for the whole learning scenario (session 3) are specified regarding data management and learning metrics implementation.

5.4.1 Learning Service Implementation

In our Classroom ECPS, we are seeking to finally monitor the state of the described components and especially the interactions between each other. Actually, there are various types of interactions (human-to-human, human-to-service, human-to-object, service-to-service, object-

to-object, etc.). As the course and activities are scheduled in a blended way, peer working is ensured through a communication service via a discord server where the students (distant and present) would be able to exchange between each other on the created channels. Thus, we model monitoring and analyzing actions over this communication server. To do so, during the first lock-down, we established some observations and implementations to track some students' behaviors through the collection of Discord traces. Then, we implemented a bot in Discord in order to collect students' data in real time. This was done only after obtaining the concerned student's consent. An example of such a collected log event is given in Figure 5.11.

```
1 {"event_type":"session_start","event_id":"AQECzAMveFsNtuLWNeZ28prGGABvP1A=","event_source":"api","user_id":"690279991996580011","domain":"Reporting","freight_hostname":"api-prd-main-xbvq","ip":"37.173.51.0","day":"4","chosen_locale":"en-US","detected_locale":"en-US","user_is_authenticated":true,"browser_user_agent":"Discord-Android/1064","browser":"Discord Android","device":"RNE-L21, RNE-L21","os":"Android","os_version":"8.0.0","os_sdk_version":"26","client_build_number":"1064","client_version":"13.2","country_code":"FR","time_zone":"Europe/Paris","isp":"Free Mobile SAS","session":"9706496e4518684db4e141f32e0da1c1","client_send_timestamp":"null","client_track_timestamp":"null","timestamp":"\2020-03-24T18:15:01.068Z\""}}
```

Figure 5.11: Example of a Discord Log (Start Session Log)

A very significant amount of logs is detected, however, we are only interested in those that could be explored to detect some issues or informational behaviors of the users. The "eventtype" (in discord logs) is the main attribute which guided us during the log filter stage. In fact, there are different events that we classified into various categories. We identified them accordingly into : behavioural events, message events, channel events, account events, private conversation events, research events, and server events. In a first stage, we decided to focus only on some events that we estimated valuable for distant learning, such as, start speaking, voice connection success, voice disconnect, send message, join voice channel, leave

voice channel, ack message, channel opened, video stream started, and video stream ended.

It is worth mentioning that selecting these events goes through different stages. First of all, we developed a python script in order to filter logs corresponding to the selected server that is used for learning in our case. Then, all the logs are merged on one file. After this, we apply another filter by date to extract only events that are related to particular time slots (date and time of the pedagogical session). Finally, the logs are correlated and sorted. Therefore, some metrics are given in Figure 5.12 in which we have noticed some interpretations regarding the students' behaviors. The "event" concept used in describing discord logs (Figure 5.11) is different from the "event" concept we use in our modeling languages. The former describes elementary events happening when using the communication channels over Discord. The latter is about the defined learning metrics/indicators that allow to deduce some meaningful interpretations.

Actually, as the lockdown affected pedagogical practices, it was important to evaluate students' status and behaviors regarding the general context. The given example of metrics shows some interpretations during a run-time learning session. For instance, it is possible to analyze if a student participates actively in a conversation, if he/she is streaming something, etc. Basically, these indicators were initially expected to monitor human-to-human interactions in the designed CPS, yet, we have been able to identify other characteristics of the CPS like the internet connection quality of the distant students, for instance if the student joins and leaves a channel several times in a short time, he/she may face a slow internet connection. To resume this communication service over discord, first; the discord bot is implemented to collect data, a set of scripts are then executed to filter, sort, and merge valuable information. Some interpretations are established based on metrics in order to characterize the student's behaviors.

This characterization could be interpreted similarly in another context with other users. The data management example could be seen as fully independent from the learning and teaching activities, however, coupling them with other results from an educational context could enrich our observations and analysis which we consider one of our objectives. The implementation of this MAPE control loop is expected to be executed on run-time so the

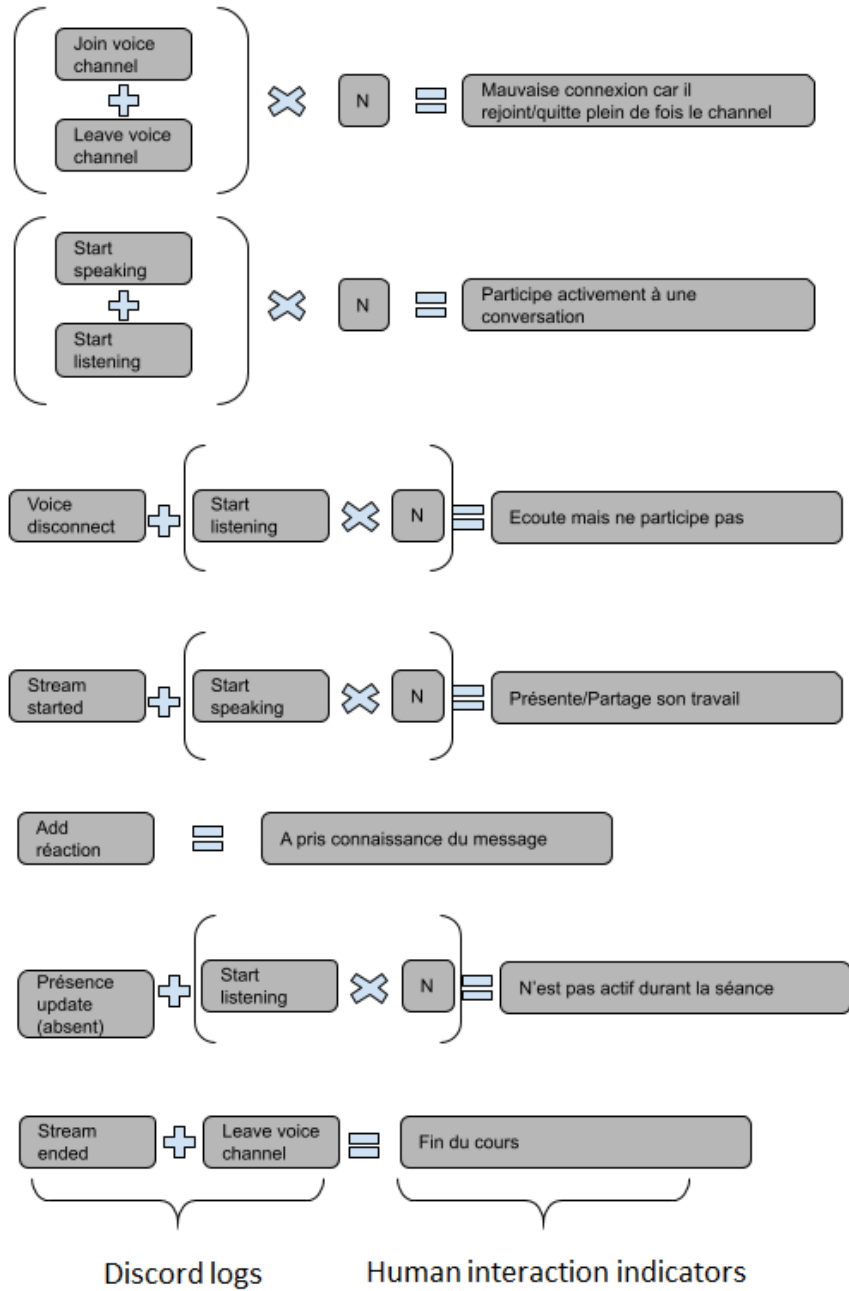


Figure 5.12: Discord Metrics

teacher could get significant result and act accordingly. Therefore, this MAPE control loop cannot be considered to be really part of the ECPS modeling. It is rather a preliminary work that was carried out by our CPS experts, to create the "discord service" used in the CPSML model and to generate the "discord logs" of this service. These logs allow the identification of some human interaction indicators, as shown in Figure 5.12.

5.4.2 Learning Scenario Implementation

TP3	Bashsyslog	Chan SIP log	Dahdi	Fulllog	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 5.13: All Collected Educational Logs from the Asterisk Server

Figure 5.13 shows all the collected data from the Asterisk servers. It represents the logs of the third practical session (TP3) of 4 student groups. Each group is composed of 14 students and thus 7 peer groups (represented by 7 Asterisk servers from number 1 to 8, number 2 does not exist). Actually, a set of scripts were prepared, in advance, by our CPS experts and coordinated with the teachers to allow the record of students' activities during the practical

sessions that will be saved and extracted through commands on/from the telephony servers. Executing these scripts allows to extract the list of connected phones to the server which will be stored in the "registrar" file (column 9). The internal and public numbering plans are stored in the "Plandenum" file (column 8). The state of the card allowing access to the telephone network is stored in the "Dahdi" file (column 4). In addition to these files, we also recorded the full logs of each student server, in the "fullog" file (column 5), which is then filtered to have everything related to the SIP protocol as well as the log of the iPBX (columns 3, 6 and 7). Other logs are also saved in the "Bashsyslog" file to keep the issued commands by the students (column 2). Unfortunately, we were not able to benefit from some logs (e.g. phone logs state), due to technological reasons, however their interactions with the IPBX were tracked over the allocated server.

In the presented logs, the black lines refer to servers that have not been used. Red boxes refer to an absence/unusability of the corresponding logs due to technical issues (e.g. activation of the server, SSH client, etc.).

bashsyslog	timestamp	serveur	utilisateur	IP Étudiant	Commande				
fullog	timestamp	niveau	Composant	informations					
dahdi	timestamp	Chan	Extension	Context	Language	MOH Interpret	Blocked	State	
plan de num	timestamp	context	extension	Priority	Action				
registrar	timestamp	name/username	host	Dyn	port	status			
sauveftp	timestamp	hostname	PID	Action	Destination				
mailbox	timestamp	contexte	username	num BV	notif activée ?				

Figure 5.14: Generated Students' Logs Format

Converting all the generated data to a unique format was also anticipated by our CPS experts. Figure 5.14 presents the different logs recorded in the correspondent files (first column on the left). As the different file may contain redundant information, as shown by the different colors (e.g. "serveur" in the bashsyslog file is "hostname" in sauveftp file - yellow color), they import all the useful data in a unique format to make it accessible for the next analysis process, with respect to a particular timestamp. They also abandon all the uninteresting data for a better data processing.

The analysis results, based on the defined metrics, are coupled to the use of LA techniques. To do so, we review a set of LA tools. Some of them are reported in [37]. Basically,

RapidMiner Studio [8], Jupyter [5], Kibana [6], and Grafana [2] are the tools which are mainly tested and evaluated to basically measure their ability to perform real-time processing, the presence or not of a graphical interface, integrated languages for the software, license price, and Open source-ness. In this case study, our primary objective is to first generate dashboards so to give the teacher an idea about the progress of the students according to the defined metrics. Thus, we evaluated the different tools with regards to this perspective.

Grafana was retained as it could consider different data sources as well as scripts on runtime. Actually, various scripts were implemented to validate the different metrics in the tool and display the designed dashboards accordingly. For each learning activity, a set of indicators has been assigned that we validate for each server (peer group). To do so, each step of the whole process is checked by a Python function separately. Figure 5.15 shows an example of code to validate one of the learning activities in the studied learning scenario regarding its correspondent learning indicators.

```

1 import sys
2
3 def etape12 (traces, serveur) :
4     indicateurEtape12 = [
5         ['PMModif', False],
6         ['AppelExt', False],
7         ['PMModifbis', False]]
8
9     for i in range (len(traces)): #On regarde parmi toutes les traces (-1 pour pas sortir du tableau)
10        if "0." in traces[i][-3] : #Recherche de "0." dans le champ "Extension du csv"
11            indicateurEtape12[0][1] = True #Si "0." est présent, l'indicateur PMModif devient vrai et on ajoute la date ou ça s'est vérifié
12            indicateurEtape12[0].append(traces[i][0])
13
14        elif "Called g" in traces[i][-1] : #Cette fois on cherche "Called g" pour Called g0/9558000040 par exemple dans le champ "Infos"
15            indicateurEtape12[1][1] = True #Si on le trouve, on pass l'indicateur AppelExt en true et on ajoute la date
16            indicateurEtape12[1].append(traces[i][0])
17
18        elif "0XXXXXXXX" in traces[i][-3] : #Recherche de "0XXXXXXXX" dans le champ "Extension du csv"
19            indicateurEtape12[2][1] = True #Si "0XXXXXXXX" est présent, l'indicateur PMModifbis devient vrai et on ajoute la date ou ça s'est vérifié
20            indicateurEtape12[2].append(traces[i][0])
21
22    for i in range(len(indicateurEtape12)): #On affiche tout l'indicateur
23        print(indicateurEtape12[i])
24
25    with open('traces10.2.17.20'+sys.argv[1][-1]+''.csv') as csvTraces:
26        traces = csvTraces.read().split('\n')
27        for i in range(len(traces)):
28            if traces[i] == '':
29                traces.pop(i)
30            else:
31                traces[i] = traces[i].split(',', maxsplit=9)
32
33    etape12(traces, sys.argv[1]) #On peut lancer le programme avec "./Etape12.py AsteriskX" ou X est le numéro du serveur asterisk. Le fichier traces doit s'appeller "traces10.2.17.20X"
34

```

Figure 5.15: Example of a Python Code to Check the Validation of a Learning Activity

In order to display the progress on Grafana, we choose to define two dashboards. Each one presents a particular viewpoint regarding the students' progress. The first dashboard will show the validation of the learning steps regarding the corresponding servers (Figure 5.16). Figure 5.17 shows the details of learning indicators validation regarding each server. The second shows the saved information regarding the students' progress, after the analysis

phase, by assigning two notes to each peer group (Figure 5.18). The first note represents the number of the validated learning steps. The second note shows the number of the successive validated learning steps (this represents the true note on which the teachers rely to assess the students). This second dashboard can be used by both teachers and students.

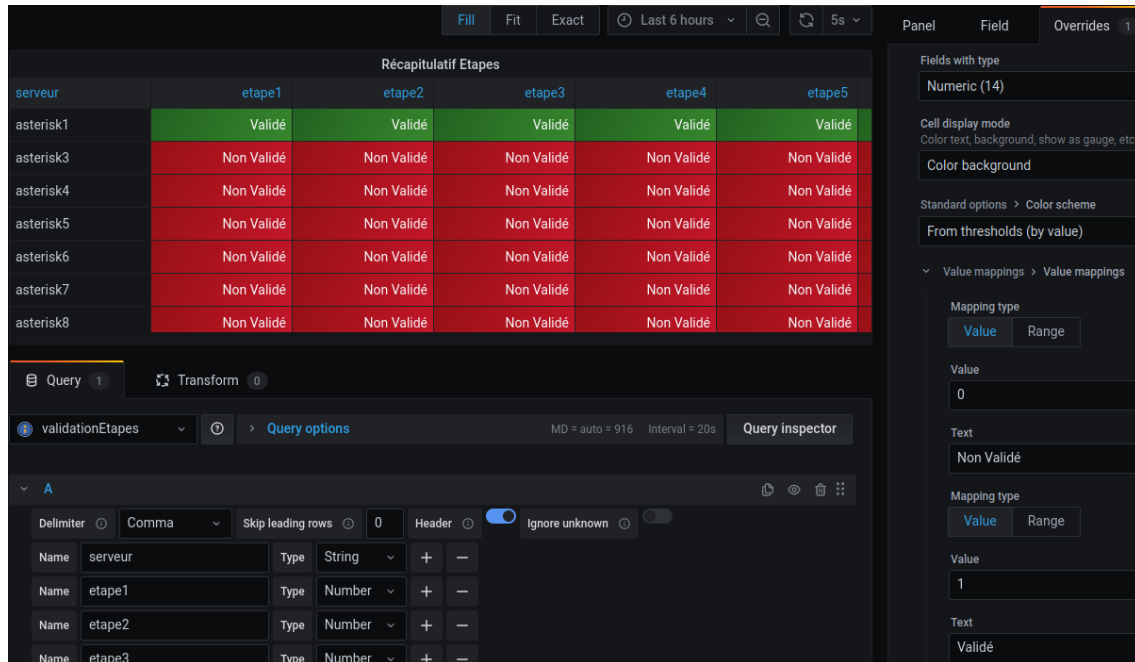


Figure 5.16: Activities Validation and Their Corresponding Servers

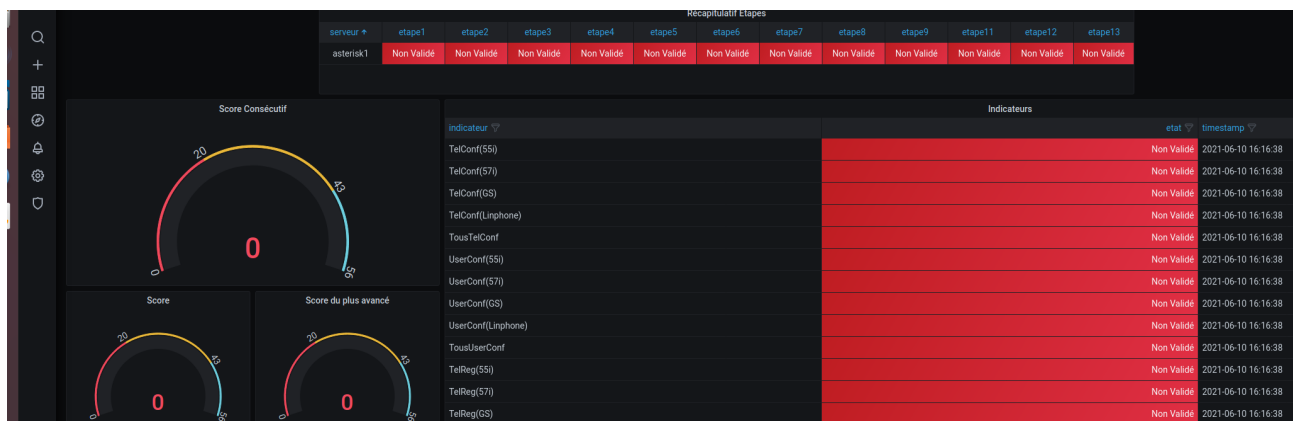


Figure 5.17: Indicators' Validation and Their Corresponding Servers

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 5.18: Number of the Validated Indicators

5.5 Case Study Limitations

Our modelECPS method is an innovative method as, to the best of our knowledge, there are no research studies that address the separation of concerns regarding the complexity of the ECPS domain. By applying our method through defining first a CPS model and then a pedagogical model, we illustrated the followed modeling process to provide the generation of the ECPS model. However, regarding the complexity of the domain, our case study has certain limitations related to its CPS infrastructure, model generation and analysis issues.

Unfortunately, we did not have the availability to use other connected objects / sensors (e.g. RFID readers, smart cameras, bio-metric sensors, etc.) that could be separated from the educational purpose. Yet, we estimate that the highly connected services in the ToIP configuration mechanism are sufficiently miscellaneous that the provided data was particularly valuable for the analysis phase. Coupling our finding results with other control loops over further "independent" components undoubtedly promotes a very enriching analysis and feedback.

Secondly, we do not model everything on a low level. LA, for instance, is only modeled

through "actions". Its implementation differs from one learning domain to another. Analysis actions are developed separately through software components to fill this gap. So, when educational data model change, how could we manage this flexibility in a model generation level and fill this gap automatically / semi-automatically in order to link between the generated model and the complementary python scripts ? Finally, the running environment was a simulation of real classroom processes. The latter impacts somehow the capacity of the teachers to make real-time feedback for the students. It gave them, though, a delayed analysis.

Part III Conclusion

In this part, we presented the use of our modeling method (ModelECPS) in order to design ToIP classroom ECPS, during the lockdown in our institute due to the global health crisis. At first, we described the CPSML model. Although there was limited use of smart objects and sensors, we modeled some features of the CPS that are independent from the domain learning model. Then, we showed the modeling of the classroom educational process through instantiating learning activities, control points as well as the learning environment.

Generating the ECPSML model showed us the importance of implementing further components in order to make its usefulness concrete regarding the monitoring and analysis of students' actions. For this purpose, we developed the required components and explored them using a visual tool (Grafana) to allow their visualization by teachers and students as well. A refined process was conducted by our CPS expert in order to collect, filter and unify data format.

Regarding our ECPS generic architecture, we have been able to first model the whole environment through our modeling languages to specify the different components of our IoE educational environment and then to implement the first three layers through a physical connection level, data-to-information conversion level and learning analytics to explore some visualization tool capacities and give a first insight of the monitoring and analyzing processes.

To conclude, the health crisis shows us the amount of connected-ness and that adapting learning could take place whenever, wherever and however educational stakeholders are willing to do it. A particular CPS environment could handle different knowledge domains and vice-versa. This points out how particular Education is open to any circumstances.

GENERAL CONCLUSION

Educational Cyber Physical Systems (ECPS) are first about Education which is a major concern in our society, especially during the Covid-19 health crisis. They are about complex systems of systems where not only a particular system generates data, but multiple ones do by invoking services and tracking various types of interactions (human-to-human, human-to-machine, machine-to-machine). In this thesis, we have derived to explore technological advancement in the domain of industry 4.0. Obviously, this domain is oriented towards industrialization and smart factories, however, we are more interested in demystifying and profiling the motivation behind the "4.0" vision. After inquiring, we pointed out that a set of advanced technologies (IoT, CPS, AI, etc.) are basically promoting the digitalisation and automation of manufacturing processes. Afterwards, we mostly investigated IoT and CPS definition and modeling as well as Educational Modeling Languages.

Studying both domains (IoT and CPS) allows us to spell out some misunderstandings. Actually, IoT is highly linked to environmental issues (hard devices, networking, etc) as a paradigm. CPS concern systems on a software and digital level, computational capabilities, control, and decision-making. In our thesis work, we are interested not only in Things but rather in Everything. IoE includes not only smart machines (things), but also humans, data and Services [14]. Interactions are then established between all these pillars of the IoE. Thus, IoE could be defined as an infrastructure of a smart world where CPS could carry out high-performing computational capacities. We consider IoE components as the main constituent elements of a CPS.

Fulfilling the Objectives and Contributions

In this thesis, we address connected complex system in educational applications in order to give a vision of the future university (University 4.0). It is based on the Educational Cyber Physical Systems (ECPS) concept. To the best of our knowledge, ECPS are not addressed in the literature. Hence, we have conducted an *exploratory* approach in order to **define, model** and then **implement** these environments.

First, we made an overview of the literature of IoT and CPS modeling. Then, we coupled

these findings with retrieved insights from the EML (Educational Modeling Languages) literature as it is our application domain. None of the studied research works addresses neither the modeling of connected aspect of educational environments nor the implementation of the MAPE (Monitor, Analyze, Plan and Execute) control loops regarding the instructional design. These conclusions lead us to rethink the modeling approach in an educational environment with regards to the expansion of connected-ness. Hence, we stated the following definition of ECPS : *ECPS are educational complex systems constituted of physical and cyber components that aim at implementing control loops to manage the educational processes according to educational stakeholders requirements. These components refer to IoE.*

Based on this definition, we proposed a generic architecture of ECPS which could be implemented in three levels at the university (classroom ECPS for short-term process, course ECPS for mid-term processes, and curriculum process for long-term processes). Regarding Software Architecture principles, our architecture follows a level of abstraction in the large. Different layers are meant to construct the whole system, from the physical connection, data-to-information, learning analytics, decision making, to actions. Data management between the physical and cyber levels is the heart of this whole process. Our generic layered architecture is then refined through a model driven engineering method to explicitly show the main structural and behavioral aspects.

We propose a modeling method (ModelECPS) as it is one of the main design challenges stated in the literature regarding connected environments, in addition to the lack of mechanisms to address multiple stakeholders' concerns and the heterogeneity of hardware devices and software components. Our aim is to support educators in developing IoE applications to fully achieve the benefits of this new paradigm in teaching and learning. The approach comprises a process and three Domain Specific Languages that offer substantial gains in expressiveness, ease of use and benefit from model generation features. More specifically, two kinds of experts are involved in this modeling process (the pedagogical expert(s) and the CPS expert(s)).

To do so, we adopt the classical “Y development cycle” from MDA as the process of our method. The objective of this process is to enable the generation of Platform Specific

Models (PSMs) from Platform Independent Models (PIMs) (specific to the business model) and Platform Description Models (PDMs). Accordingly CPSML is the proposed DSL to describe the connected aspect of the environment (PDM). EML4.0 is the proposed DSL to describe the domain model which corresponds to educational applications on a classroom level (short-term process). PSM refers to the final ECPS system (ECPSML models) where the output model is generated through ATL transformation based on both input models from CPSML and EML4.0.

The pedagogical expert models the instructional scenario principally based on the learning environment and activities by specifying the control points that he/she would like to explore to validate some metrics regarding the task accomplished by students. The CPS expert models the connected environment and designs from his/her side monitoring and analyzing activities that he/she considers valuable to consider regardless the instructional design. His/her perspectives in this part could be applied in any other application domain. The process of modeling ends with a collaboration between both experts to merge their modeling and implement (by the CPS expert) the required components for the defined control points (by the pedagogical expert) to make them concrete in a run-time environment. The CPS expert translates each control point and the related metrics. This translation means implementing the required features for both monitoring and analyzing actions. We presented in Part III an experimentation for a case study in order to illustrate and validate the implementation of our proposed models.

In this thesis, our contributions are based on MDE principles as they handle separation and combination of various kinds of concern (such as platform models and code generation) in software engineering. One of the objectives of modeling is to bridge the gap between the design and implementation phases and produce software assets that are resilient to changes in technologies. For instance, MDE stresses the importance of models since they are supposed to survive the constant change in software technologies.

ModelECPS is an innovative method regarding Educational Cyber Physical System concept definition and modeling. This concept, not yet emerging in the scientific community, is addressed through its multidisciplinary aspects (educational and connected in our case).

Throughout our thesis work, our proposals come up with strong points to answer the defined research questions. However, like any research work, it does have some limitations regarding mainly model generation and analysis issues. Actually, all the ATL model-to-model transformations regarding the concept alignment process are not all automatically implemented in EMF as we identified some technical issues. Model-to-text transformations are not addressed in this thesis, which would facilitate for CPS experts the implementation of further components through generated code. Furthermore, our vision of University 4.0 is particularly addressed through Classroom ECPS modeling. It is worth improving our approach to incorporate Course and Curriculum processes' specificities. These limitations and further improvements are addressed as perspectives of this thesis that could be improved in future works.

Perspectives

In this section, we present some perspectives and a number of possible future directions that would improve our research. The short-term perspectives address limitations and challenges that we identified thanks to our case study which makes us aware of the complexity of the ECPS systems during experimentation. The long-term perspectives are defined with the intention to explore our methodology in order to design other application domains and address related challenges regarding their connected-ness.

Short-term Perspectives

First, regarding the CPS infrastructure, it is interesting to gather data from other smart objects and/or services (RFID readers, smart cameras, bio-metric sensors, etc.) that are independent from the instructional design model and study their correlation with educational analysis. Further data about students' health-state, motion or presence would foster learning indicator definitions. This should be improved regarding the CPSML model. Coupling CPS analysis results with other educational indicators undoubtedly promotes a very enriching analysis and feedback.

MDE allows not only model generation but also code generation. Hence, we are looking forward to first improve the ATL M2M transformations in order to implement all the concept alignment process that we identified. Then, a code generation module could be considered to provide a complete framework where both model-to-model and model-to-text transformations are supported. Actually, the generated code would help in implementing further components in order to fill the gap regarding the final use of ECPS in run-time learning sessions. For our case study, we implement software components from scratch.

Finally, the running environment was a simulation of real classrooms processes. This limitation somehow impacts the capacity of the teachers to give a real-time feedback for the students. It gave them, though, a delayed analysis. Conducting experimentations in real time would be efficient to evaluate our methodology based on more improved dashboards for better students monitoring.

Long-term Perspectives

Our first perspective is to test the genericity of our ModelECPS method by changing the educational model and dealing with the same CPSML model and vice-versa. And so, we could validate the fact of acting on either PIM or PDM in order to generate the required educational environment. In addition, it is worth exploring this model driven method to design other connected environments (smart grid, smart building, medical environment, etc.). The transformation rules and PIM model should be adapted accordingly.

At university, we do have various specialization domains. Sometimes, we use the same infrastructure, and thus CPSML model, in different courses and experimentations. It is the case of our Network and Telecommunications department. This leads us to think about the idea of developing design patterns that could address the ECPS implementation when needed in particular domains for educational purpose.

In this thesis, a global vision of the future of the university is given, however, with regards to the decision making layer in the generic layered architecture that we have designed, for the time being, we do not consider developing AI components that would assist the teacher in his/her decision making. As perspective of this work, AI techniques could be adopted in

order to implement such features. Nevertheless, the required components have to be adapted to each application domain model. So, how could the common modeling and coding features be designed so that they fit with our model-driven method ?

Last but not least, our modeling tools could be regularly improved with regards to the mentioned perspectives (short-term and long-term). At this stage, only Papyrus and ecore basic tools are used. To do so, Eclipse Graphical Modeling Framework (GMF) would be the required tool to generate the graphical editors. This framework provides the fundamental infrastructure and components to develop visual models and modeling surfaces in Eclipse.

List of publications

- Bachir, S., Gallon, L., Abénia, A., Aniorte, P. (2021, September). Educational Cyber Physical Systems : Data and Metrics. In DMCESE 2021 Workshop. *(Selected for a detailed version submission to be published in the Rank **Q1** Open Access journal “Sensors”)*.
- Bachir, S., Gallon, L., Aniorte, P., Abénia, A. (2021, July). A Model Driven Method to Design Educational Cyber Physical Systems. In 16th International Conference on Software Technologies (ICSOFT 2021). **(Rank B in CORE, Acceptance rate = 15%)**
- Bachir, S., Gallon, L., Abénia, A., Aniorte, P., Lubon, V. (2021, June). Enseignement à Distance en Situation d’Urgence: Retour d’Experiences du Premier Confinement. In Congrès national de la recherche des IUT (CNRIUT 2021).
- Bachir, S., Abenia, A. (2019, September). Internet of Everything and educational cyber physical systems for University 4.0. In International Conference on Computational Collective Intelligence (pp. 581-591). Springer, Cham. **(Rank C in CORE)**
- Bachir, S., Gallon, L., Abenia, A., Aniorté, P., Exposito, E. (2019, August). Towards autonomic educational cyber physical systems. In 2019 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCCom/IOP/SCI) (pp. 1198-1204). IEEE. **(Rank B in CORE)**

Bibliography

- [1] Ecore. https://www.eclipse.org/Xtext/documentation/308_emf_integration.html. Accessed : 2021 – 09 – 01.
- [2] Grafana. <https://grafana.com/>. Accessed: 2021-05-16.
- [3] Ieee. <https://www.ieeeltsc.org/working-groups/wg12LOM/lomDescription/>. Accessed: 2021-07-23.
- [4] Igi global. <https://www.igi-global.com/dictionary/designing-blended-learning-strategies-for-rich-content/16905>. Accessed: 2021-09-01.
- [5] Jupyter. <https://jupyter.org/>. Accessed: 2021-05-16.
- [6] Kibana. <https://www.elastic.co/fr/kibana/>. Accessed: 2021-05-16.
- [7] Openhab : Open source project website. www.openhab.org. Accessed: 2021-07-01.
- [8] Rapidminer studio. <https://rapidminer.com/>. Accessed: 2021-05-16.
- [9] Sysml. <https://sysml.org/.res/docs/specs/OMGSysML-v1.4-15-06-03.pdf>. Accessed: 2021-07-15.
- [10] Emf : Eclipse modeling framework. <https://www.eclipse.org/modeling/emf/>, 2000. Accessed: 2021-02-05.
- [11] Omg : Object management group. <https://www.omg.org/>, 2000. Accessed: 2021-02-05.

- [12] Papyrus : Eclipse papyrus modeling environment. <https://www.eclipse.org/papyrus/>, 2000. Accessed: 2021-02-05.
- [13] Scorm : Shareable content object resource reference. <https://adlnet.gov/projects/scorm/>, 2003. Accessed: 2021-02-05.
- [14] Cisco : Internet of everything. https://www.cisco.com/c/dam/en_us/about/business_insights/docs/ioe_value_index_faq.pdf, 2013. Accessed : 2021 - 02 - 09.
- [15] Educause, <https://er.educause.edu/articles/2016/6/the-internet-of-things-riding-the-wave-in-higher-education>, 2016 (accessed May 27, 2019).
- [16] Educause : The difference between emergency remote teaching and online learning. <https://er.educause.edu/articles/2020/3/the-difference-between-emergency-remote-teaching-and-online-learning>, 2020. Accessed: 2021-02-09.
- [17] Dogukan Aksu and M Ali Aydin. A survey of iot architectural reference models. In *2019 16th International Multi-Conference on Systems, Signals & Devices (SSD)*. IEEE, 2019.
- [18] Mostafa Al-Emran, Sohail Iqbal Malik, and Mohammed N Al-Kabi. A survey of internet of things (iot) in education: opportunities and challenges. *Toward social internet of things (SIoT): Enabling technologies, architectures and applications*, pages 197–209, 2020.
- [19] Francisco Pérez Andrés, Juan De Lara, and Esther Guerra. Domain specific languages with graphical and textual views. In *International Symposium on Applications of Graph Transformations with Industrial Relevance*, pages 82–97. Springer, 2007.
- [20] Francisco Araque, Concepción Roldán, and Alberto Salguero. Factors influencing university drop out rates. *Computers & Education*, 53(3):563–574, 2009.
- [21] Paris Avgeriou, Anastasios Koutoumanos, Symeon Retalis, and Nikolaos Papaspyrou. *Learning object metadata in a Web-based learning environment*. University of Groningen, Johann Bernoulli Institute for Mathematics, 2000.

- [22] Maryam Bagheri and Siavosh H Movahed. The effect of the internet of things (iot) on education business model. In *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 435–441. IEEE, 2016.
- [23] R Baheti and H Gill. The impact of control technology: Cyber-physical systems. *IEEE Control Systems Society*, 2011.
- [24] Martin Bauer, Nicola Bui, Jourik De Loof, Carsten Magerkurth, Andreas Nettsträter, Julinda Stefa, and Joachim W Walewski. Iot reference model. In *Enabling Things to Talk*. Springer, Berlin, Heidelberg, 2013.
- [25] Benjamin Bertran, Julien Bruneau, Damien Cassou, Nicolas Lorient, Emilie Balland, and Charles Consel. Diasuite: A tool suite to develop sense/compute/control applications. *Science of Computer Programming*, 79:39–51, 2014.
- [26] Jean Bézivin. On the unification power of models. *Software & Systems Modeling*, 4(2), 2005.
- [27] Jean Bézivin and Olivier Gerbé. Towards a precise definition of the omg/mda framework. In *Proceedings 16th Annual International Conference on Automated Software Engineering (ASE 2001)*, pages 273–280. IEEE, 2001.
- [28] Jan Bosch. *Design and use of software architectures: adopting and evolving a product-line approach*. Pearson Education, 2000.
- [29] Rosa Maria Bottino. Which perspectives can drive the analysis of technology-enhanced learning environments? In *Educational Technology Use and Design for Improved Learning Opportunities*, pages 172–184. IGI Global, 2014.
- [30] Luca Botturi. E2ml-educational environment modeling language. In *EdMedia+ Innovate Learning*. Association for the Advancement of Computing in Education (AACE), 2003.
- [31] Luca Botturi. E2ml: A tool for sketching instructional designs. In *Handbook of visual languages for instructional design: Theories and practices*, pages 111–131. IGI Global, 2008.

- [32] Jeppe Bundsgaard. Introducing the collaborative learning modeling language (coleml). In *European Conference on e-Learning*. Academic Conferences International Limited, 2014.
- [33] Barry Burd, Lecia Barker, Monica Divitini, Felix Armando Fermin Perez, Ingrid Russell, Bill Siever, and Liviana Tudor. Courses, content, and tools for internet of things in computer science education. In *Proceedings of the 2017 ITiCSE Conference on Working Group Reports*, pages 125–139. ACM, 2018.
- [34] Manuel Caeiro-Rodríguez. poeml: A separation of concerns proposal to instructional design. In *Handbook of visual languages for instructional design: theories and practices*. IGI Global, 2008.
- [35] Joe Cecil and Damon Chandler. Cyber physical systems and technologies for next generation e-learning activities. *Cyber Physical Systems and Technologies for Next Generation e-Learning Activities*, (10):1–11, 2014.
- [36] Mohamed Amine Chatti, Anna Lea Dyckhoff, Ulrik Schroeder, and Hendrik Thüs. A reference model for learning analytics. *International Journal of Technology Enhanced Learning*, 4(5-6):318–331, 2012.
- [37] Florence Cherigny, Hassina El Kechai, Sébastien Iksal, Marie Lefevre, Hugues Labarthe, and Vanda Luengo. *L’analytique des apprentissages avec le numérique Groupes thématiques de la Direction du numérique pour l’Éducation (DNE-TN2)*. PhD thesis, Direction du numérique pour l’éducation, 2020.
- [38] Yves Chevallard. On didactic transposition theory: Some introductory notes. In *International Symposium on Research and Development in Mathematics, Bratislava, Czechoslovakia*, 1988.
- [39] Vanea Chiprianov. Collaborative construction of telecommunications services. an enterprise architecture and model driven engineering method, 20 july 2021.
- [40] Monica Ciolacu, Paul Mugur Svasta, Waldemar Berg, and Heribert Popp. Education 4.0 for tall thin engineer in a data driven society. In *Design and Technology in Electronic Packaging (SIITME), 2017 IEEE 23rd International Symposium for*, pages 432–437. IEEE, 2017.

- [41] Autonomic Computing. An architectural blueprint for autonomic computing. *IBM White Paper*, 31:1–6, 2006.
- [42] Bruno Costa, Paulo F Pires, Flávia C Delicato, Wei Li, and Albert Y Zomaya. Design and analysis of iot applications: a model-driven approach. In *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, (DASC/PiCom/DataCom/CyberSciTech)*. IEEE, 2016.
- [43] Krzysztof Czarnecki and Simon Helsen. Feature-based survey of model transformation approaches. *IBM systems journal*, 45(3):621–645, 2006.
- [44] Barney Dalgarno. Polysynchronous learning: A model for student interaction and engagement. *Rhetoric and reality: Critical perspectives on educational technology. Proceedings Ascilite Dunedin 2014*, pages 673–677, 2014.
- [45] Rafael de Amorim Silva and Rosana TV Braga. An acknowledged system of systems for educational internet of everything ecosystems. In *Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings*, page 25. ACM, 2018.
- [46] Peter de Lange, Boris Bähre, Christiane Finetti-Imhof, Ralf Klamma, Andreas Koch, and Leif Oppermann. Socio-technical challenges in the digital gap between building information modeling and industry 4.0. In *Proceedings of the 3rd International Workshop on Socio-Technical Perspective in IS development (STPIS'17) co-located with (CAiSE 2017), Essen, Germany, June 13, 2017.*, pages 33–46, 2017.
- [47] Michael Derntl and Renate Motschnig-Pitrik. couml: A visual language for modeling cooperative environments. In *Handbook of visual languages for instructional design: Theories and practices*. IGI Global, 2008.
- [48] Rainer Drath and Alexander Horch. Industrie 4.0: Hit or hype?[industry forum]. *IEEE industrial electronics magazine*, 8(2):56–58, 2014.

- [49] Falko Dressler. Cyber physical social systems: Towards deeply integrated hybridized systems. In *2018 International Conference on Computing, Networking and Communications (ICNC)*, pages 420–424. IEEE, 2018.
- [50] Erik Duval, Mike Sharples, and Rosamund Sutherland. Research themes in technology enhanced learning. In *Technology Enhanced Learning*, pages 1–10. Springer, 2017.
- [51] Christine Ferraris, Christian Martel, and Laurence Vignollet. Ldl for collaborative activities. In *Handbook of visual languages for instructional design: Theories and practices*, pages 224–251. IGI Global, 2008.
- [52] Giancarlo Fortino, Wilma Russo, Claudio Savaglio, Mirko Viroli, and MengChu Zhou. Modeling opportunistic iot services in open iot ecosystems. In *WOA*, 2017.
- [53] Bror Giesenbauer and Georg Müller-Christ. University 4.0: Promoting the transformation of higher education institutions toward sustainable development. *Sustainability*, 12(8):3371, 2020.
- [54] James G Greeno and Yrjö Engeström. *Learning in activity*. na, 2006.
- [55] Resnick L Greeno JG, Collins AM. *Cognition and Learning*. In the Handbook of Educational Psychology, 1996.
- [56] Cyber Physical Systems Public Working Group. Framework for cyber-physical systems, release 1.0. *National Institute for Standards and Technology*, pages 1–266, 2016.
- [57] Mamadou Gueye and Ernesto Exposito. University 4.0: The industry 4.0 paradigm applied to education. In *IX Congreso Nacional de Tecnologías en la Educación*, 2020.
- [58] Jasmin Guth, Uwe Breitenbücher, Michael Falkenthal, Frank Leymann, and Lukas Reinfurt. Comparison of iot platform architectures: A field study based on a reference architecture. In *2016 Cloudification of the Internet of Things (CIoT)*. IEEE, 2016.

- [59] Nicolas Harrand, Franck Fleurey, Brice Morin, and Knut Eilif Husa. 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*, 2016.
- [60] Richard W Herling and Joanne Provo. Knowledge, competence, and expertise in organizations. *Advances in Developing Human Resources*, 2(1):1–7, 2000.
- [61] Mario Hermann, Tobias Pentek, and Boris Otto. Design principles for industrie 4.0 scenarios. In *System Sciences (HICSS), 2016 49th Hawaii International Conference on*, pages 3928–3937. IEEE, 2016.
- [62] Mahmoud Hussein, Shuai Li, and Ansgar Radermacher. Model-driven development of adaptive iot systems. In *MODELS (Satellite Events)*, 2017.
- [63] IMS. *IMS LD*, <http://www.msglobal.org/learningdesign/index.html>.
- [64] IMS. *IMS QTI*, <https://www.msglobal.org/question/index.html>.
- [65] IEC ISO. Ieee, “international standard iso/iec/ieee 42010: 2011 systems and software engineering—architecture description,” iso/iec, 2011.
- [66] Frédéric Jouault, Freddy Allilaire, Jean Bézivin, Ivan Kurtev, and Patrick Valduriez. Atl: a qvt-like transformation language. In *Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications*, pages 719–720, 2006.
- [67] Henning Kagermann, Wolfgang Wahlster, and Johannes Helbig. Final report of the industrie 4.0 working group. *Federal Ministry of Education and Research*, 82, 2013.
- [68] Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, and Shahid Khan. Future internet: the internet of things architecture, possible applications and key challenges. In *2012 10th international conference on frontiers of information technology*. IEEE, 2012.
- [69] Jörg Christian Kirchof, Judith Michael, Bernhard Rumpe, Simon Varga, and Andreas Wortmann. Model-driven digital twin construction: synthesizing the integration of cyber-physical

- systems with their information systems. In *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*, 2020.
- [70] Roland Klemke. Introduction to phd in technology-enhanced learning, Monday 03 May 2021.
- [71] AnnekeG Kleppe. andwarmerjos, andbast, wim. mdaexplained: The model driven architecture: Practice and promise. addison-wesley longman publishing co. *Inc., Boston, MA, USA*, 2003.
- [72] Rob Koper. From change to renewal: Educational technology foundations of electronic learning environments. <http://eml.ou.nl/introduction/docs/koper-inaugural-address.pdf>, 2000.
- [73] Rob Koper. Modeling units of study from a pedagogical perspective—the pedagogical meta-model behind eml. <http://eml.ou.nl/introduction/docs/ped-metamodel.pdf>, 2001.
- [74] Rob Koper and Bill Olivier. Representing the learning design of units of learning. *Journal of Educational Technology & Society*, 7(3), 2004.
- [75] Ivan Kurtev, Jean Bézivin, Frédéric Jouault, and Patrick Valduriez. Model-based dsl frameworks. In *Companion to the 21st ACM SIGPLAN Symposium on Object-oriented Programming Systems, Languages, and Applications*, OOPSLA '06, pages 602–616, New York, NY, USA, 2006. ACM.
- [76] Sarah Le Vigouroux, Aurélie Goncalves, and Elodie Charbonnier. The psychological vulnerability of french university students to the covid-19 confinement. *Health Education & Behavior*, 48(2):123–131, 2021.
- [77] Edward A Lee. Cyber physical systems: Design challenges. In *11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369. IEEE, 2008.
- [78] Jay Lee, Behrad Bagheri, and Hung-An Kao. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing letters*, 3, 2015.

- [79] T-Y Liu. A context-aware ubiquitous learning environment for language listening and speaking. *Journal of Computer Assisted Learning*, 25(6):515–527, 2009.
- [80] Tomás Sánchez López, Damith C Ranasinghe, Mark Harrison, and Duncan McFarlane. Adding sense to the internet of things. *Personal and Ubiquitous Computing*, 16(3):291–308, 2012.
- [81] Zakaria Maamar, Noura Faci, Mohamed Sellami, Emir Ugljanin, and Ejub Kajan. Everything-as-a-thing for abstracting the internet-of-things. In *ICSOFT*, 2018.
- [82] Irina Makarova, Ksenia Shubenkova, Polina Buyvol, Vadim Mavrin, and Eduard Mukhametdinov. Interaction between education and business in digital era. In *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, pages 503–508. IEEE, 2018.
- [83] Christian Martel, Laurence Vignollet, Christine Ferraris, and Guillaume Durand. Ldl: a language to model collaborative learning activities. In *Proc. of the 2006 World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA '2006)*, 2006.
- [84] I Martinez-Ortiz, Pablo Moreno-Ger, Jose Luis Sierra, and Baltasar Fernandez-Manjon. Educational modeling languages. In *Computers and Education*, pages 27–40. Springer, 2007.
- [85] Mark W Mayer. Architecting principles for system of systems. *Syst. Eng*, 1(4):267–274, 1998.
- [86] Terry Mayes and Sara De Freitas. Review of e-learning theories, frameworks and models. *JISC e-learning models desk study*, (1), 2004.
- [87] Tom Mens and Pieter Van Gorp. A taxonomy of model transformation. *Electronic notes in theoretical computer science*, 152:125–142, 2006.
- [88] Marjan Mernik, Jan Heering, and Anthony M. Sloane. When and how to develop domain-specific languages. *ACM Comput. Surv.*, 37(4):316–344, December 2005.
- [89] Mahdi H Miraz, Maaruf Ali, Peter S Excell, and Rich Picking. A review on internet of things (iot), internet of everything (ioe) and internet of nano things (iont). In *2015 Internet Technologies and Applications (ITA)*, pages 219–224. IEEE, 2015.

- [90] Daniel Moldovan, Georgiana Copil, and Schahram Dustdar. Elastic systems: Towards cyber-physical ecosystems of people, processes, and things. *Computer Standards & Interfaces*, 57:76–82, 2018.
- [91] Filipe T Moreira, Andreia Magalhães, Fernando Ramos, and Mário Vairinhos. The power of the internet of things in education: an overview of current status and potential. In *Conference on Smart Learning Ecosystems and Regional Development*, pages 51–63. Springer, 2017.
- [92] Henry Muccini and Mahyar Tourchi Moghaddam. Iot architectural styles. In *European Conference on Software Architecture*. Springer, 2018.
- [93] Henry Muccini and Mohammad Sharaf. Caps: Architecture description of situational aware cyber physical systems. In *2017 IEEE International Conference on Software Architecture (ICSA)*. IEEE, 2017.
- [94] Mohammad Reza Nami and Koen Bertels. A survey of autonomic computing systems. In *Autonomic and Autonomous Systems, 2007. ICAS07. Third International Conference on*, pages 26–26. IEEE, 2007.
- [95] Deepika Navani, Sanjeev Jain, and Maninder Singh Nehra. The internet of things (iot): A study of architectural elements. In *2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE, 2017.
- [96] Luiz Henrique Nunes, Julio Cezar Estrella, Charith Perera, Stephan Reiff-Marganiec, and Alexandre Claudio Botazzo Delbem. Multi-criteria iot resource discovery: a comparative analysis. *Software: Practice and Experience*, 47(10), 2017.
- [97] Daniel Palma, Juan Agudo, Héctor Sánchez, and Miguel Macías. An internet of things example: Classrooms access control over near field communication. *Sensors*, 14(4):6998–7012, 2014.
- [98] Alejandro Peña-Ayala and Leonor Cárdenas. A revision of the literature concerned with mobile, ubiquitous, and pervasive learning: A survey. In *Mobile, ubiquitous, and pervasive learning*, pages 55–100. Springer, 2016.

- [99] Inhelder B Piaget J. *The Psychology of the Child*. Basic Books, Inc., 1990.
- [100] Anthony G Picciano. Theories and frameworks for online education: Seeking an integrated model. *Online Learning*, 21(3), 2017.
- [101] Ferry Pramudianto, Carlos Alberto Kamienski, Eduardo Souto, Fabrizio Borelli, Lucas L Gomes, Djamel Sadok, and Matthias Jarke. Iot link: An internet of things prototyping toolkit. In *2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops*, pages 1–9. IEEE, 2014.
- [102] IMS QTI. Ims global learning consortium: Ims question & test interoperability: Ims question & test interoperability specification, 2005.
- [103] Pablo-Alejandro Quezada-Sarmiento, Liliana Enciso, Hironori Washizaki, and Wilmar Hernandez. Body of knowledge on iot education. In *WEBIST*, pages 449–453, 2018.
- [104] Dosheela Devi Ramlowat and Binod Kumar Pattanayak. Exploring the internet of things (iot) in education: A review. In *Information Systems Design and Intelligent Applications*, pages 245–255. Springer, 2019.
- [105] Raman Ramsin and Richard F Paige. Process-centered review of object oriented software development methodologies. *ACM Computing Surveys (CSUR)*, 40(1), 2008.
- [106] Adrian Rawlings, Peter Van Rosmalen, Rob Koper, Miguel Rodriguez-Artacho, and Paul Lefrere. Survey of educational modelling languages (emls). 2002.
- [107] Till Riedel, Dimitar Yordanov, Nicolaie Fantana, Markus Scholz, and Christian Decker. A model driven internet of things. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)*. IEEE, 2010.

- [108] Jan Oliver Ringert, Bernhard Rumpe, and Andreas Wortmann. From software architecture structure and behavior modeling to implementations of cyber-physical systems. *arXiv preprint arXiv:1408.5690*, 2014.
- [109] USEATYOW RISK. Draft standard for learning object metadata. *IEEE standard*, 1484(1), 2002.
- [110] Rodrigo Roman, Jianying Zhou, and Javier Lopez. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, 57(10), 2013.
- [111] Jeff Rothenberg, Lawrence E Widman, Kenneth A Loparo, and Norman R Nielsen. The nature of modeling. in *Artificial Intelligence, Simulation and Modeling*, 1989.
- [112] Alexandra Ruiz, José Ignacio Panach, Oscar Pastor, Fábio D Giraldo, José L Arciniegas, and William J Giraldo. Designing the didactic strategy modeling language (dsml) from pon: An activity oriented eml proposal. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 13(4), 2018.
- [113] Samuel Mensah Sackey, Andre Bester, and Dennit Adams. Industry 4.0 learning factory didactic design parameters for industrial engineering education in south africa. *South African Journal of Industrial Engineering*, 28(1):114–124, 2017.
- [114] Adnan Salihbegovic, Teo Eterovic, Enio Kaljic, and Samir Ribic. Design of a domain specific language and ide for internet of things applications. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 996–1001. IEEE, 2015.
- [115] Mohamed Sarrab. Distance, virtual, electronic, mobile and ubiquitous learning environments: taxonomy study. In *Proceedings of the 2019 7th international conference on information and education technology*, pages 36–43, 2019.
- [116] Goran Savić, Milan Segedinac, Dušica Milenković, Tamara Hrin, and Mirjana Segedinac. A model-driven approach to e-course management. *Australasian Journal of Educational Technology*, 34(1), 2018.

- [117] G Schuh, T Potente, C Wesch-Potente, and A Hauptvogel. 10.6 sustainable increase of overhead productivity due to cyber-physical-systems. 2013.
- [118] Bran Selic. The theory and practice of modeling language design for model-based software engineering—a personal perspective. In *International Summer School on Generative and Transformational Techniques in Software Engineering*, pages 290–321. Springer, 2009.
- [119] Ian Sommerville. *Software Engineering*. Pearson Education, 2015.
- [120] Madiha H Syed, Eduardo B Fernandez, and Mohammad Ilyas. A pattern for fog computing. In *Proceedings of the 10th Travelling Conference on Pattern Languages of Programs*, 2016.
- [121] Aroua Taamallah and Maha Khemaja. Designing and experiencing smart objects based learning scenarios: an approach combining ims ld, xapi and iot. In *Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality*, pages 373–379, 2014.
- [122] Temitope Takpor and Aderemi A Atayero. Integrating internet of things and ehealth solutions for students’ healthcare. In *Proceedings of the World Congress on Engineering*, volume 1. World Congress on Engineering, London, UK, 2015.
- [123] Sergio Teixeira, Bruno Alves Agrizzi, José Gonçalves Pereira Filho, Silvana Rossetto, Isaac Simões Araújo Pereira, Patrícia Dockhorn Costa, Adriano Francisco Branco, and Ruan Rocha Martinelli. Laura architecture: Towards a simpler way of building situation-aware and business-aware iot applications. *Journal of Systems and Software*, 161, 2020.
- [124] Bedir Tekinerdogan, Mittal Rakshit, Rima Al Ali, Iacono Mauro, Navarro Eva, Vanherpen Ken, and Ankica Barisic. A feature-based ontology for cyber-physical systems, 2021.
- [125] Nupur Tyagi. A reference architecture for iot. *International Journal of Computer Engineering and Applications*, 10(I), 2016.

- [126] Vladimir L Uskov, Jeffrey P Bakken, Srinivas Karri, Alexander V Uskov, Colleen Heinemann, and Rama Rachakonda. Smart university: Conceptual modeling and systems' design. In *International Conference on Smart Education and Smart E-Learning*, pages 49–86, 2017.
- [127] Bahtijar Vogel and Dimitrios Gkouskos. An open architecture approach: Towards common design principles for an iot architecture. In *Proceedings of the 11th European Conference on Software Architecture: Companion Proceedings*, 2017.
- [128] Uri Wilensky. *Abstract meditations on the concrete and concrete implications for mathematics education*. Epistemology and Learning Group, MIT Media Laboratory, 1991.
- [129] Yang Yuqiao and Yu Kanhua. Construction of distance education classroom in architecture specialty based on internet of things technology. *International Journal of Emerging Technologies in Learning (iJET)*, 11(05):56–61, 2016.