UNIVERSITY OF CRETE
DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF SCIENCES AND ENGINEERING

CognitOS: A Unified Student-centric Working Environment of Adaptive Educational Applications for the Attention-aware Intelligent Classroom

by

Anastasia Ntagianta

MSc dissertation submitted in partial fulfillment for the

Master of Science degree in Computer Science

Heraklion, April 2019

This work has been performed at the University of Crete, School of Sciences and Engineering, Computer Science Department.
The work has been supported by the Foundation for Research and Technology – Hellas (FORTH), Institute of Computer Science (ICS).

UNIVERSITY OF CRETE
DEPARTMENT OF COMPUTER SCIENCE

CognitOS: A Unified Student-centric Working Environment of Adaptive Educational Applications for the Attention-aware Intelligent Classroom

by **Anastasia Ntagianta**

in partial fulfillment of the requirements for the
Master of Science degree in Computer Science

APPROVED BY:

____

Author: Anastasia Ntagianta

____

Supervisor: Constantine Stephanidis, Professor

____

Committee Member: George Papagiannakis, Associate Professor

____

Committee Member: Margherita Antona, Principal Researcher

____

Director of Graduate Studies: Antonis Argyros, Professor

Heraklion, April 2019

To my parents

Abstract

The emergence of Intelligent Classrooms, and in particular classrooms equipped with facilities for identifying the students' attention level, has raised the need for appropriate student-friendly tools that not only facilitate application hosting, but also act as the means to re-engage inattentive students in the educational process. The diversity of the potential classroom artifacts introduces additional interaction challenges, stemming from the cross-platform migration of the available applications, and outlines the need for a unified working environment for students.

This thesis introduces CognitOS, a web-based working environment hosting a variety of educational applications (e.g. exercises, multimedia viewer, digital book) that instantiates a common Look 'n' Feel across all available artifacts, thus transforming the classroom into a unified environment, rather than a group of isolated units. Additionally, utilizing the ability of Attention-aware Intelligent Classrooms to identify problematic situations inside the educational setting (e.g. one or more distracted students, tired or unmotivated students) through the LECTOR framework, CognitOS is able to act as a host for delivering interventions that aim to re-engage the students in the educational process.

Since each classroom environment possesses diverse characteristics (e.g. students' numbers or abilities, classroom setting), CognitOS supports extensive configuration of its functionality by design. Its modular architecture allows new applications to be easily integrated, while its functionality can be extended/modified as needed. To that end, CognitOS provides an expandable API that can be further utilized by the educators to customize the system's behavior and define rules for the different use cases, in order to properly address the educational needs of the target classroom.

In comparison to related work, CognitOS is an innovative, attention-aware, student-centric working environment that not only hosts educational applications, but also constitutes a configurable medium through which personalized interventions can be efficiently delivered to the students and the educator.

Keywords: E-Learning, Educational Applications, Intelligent Classroom, Migrating UIs, Ambient Intelligence, Attention Monitoring

Περίληψη

Η ανάπτυξη της Έξυπνης Τάξης, και πιο συγκεκριμένα μιας τάξης εφοδιασμένης με εγκαταστάσεις για την ανίχνευση των επιπέδων προσοχής των μαθητών, έχει εγείρει την ανάγκη για φιλικά προς τους μαθητές εργαλεία τα οποία δεν παρέχουν απλά εφαρμογές, αλλά έχουν επίσης τη δυνατότητα να χρησιμοποιηθούν ως μέσα για την επαναφορά της προσοχής των μαθητών στην εκπαιδευτική διαδικασία. Η ποικιλομορφία των πιθανών αντικειμένων (artifacts) μέσα στην τάξη εισάγει πρόσθετες προκλήσεις αλληλεπίδρασης, που απορρέουν από τη ‘μετανάστευση’ των διαθέσιμων εφαρμογών μεταξύ των συσκευών, και περιγράφει την ανάγκη για ένα ενοποιημένο εργασιακό περιβάλλον για τους μαθητές.

Η παρούσα εργασία παρουσιάζει το CognitOS, ένα διαδικτυακό εργασιακό περιβάλλον που παρέχει μία ποικιλία από εκπαιδευτικές εφαρμογές (π.χ. συλλογή ασκήσεων, πρόγραμμα προβολής πολυμέσων, ψηφιακά βιβλία) και δημιουργεί μια κοινή εμφάνιση και αίσθηση προς το χρήστη σε όλα τα διαθέσιμα αντικείμενα, μεταμορφώνοντας έτσι την τάξη σε ένα ενοποιημένο περιβάλλον εργασίας, αντί για ένα σύνολο από απομονωμένες μονάδες. Επιπροσθέτως, αξιοποιώντας την ικανότητα μιας Έξυπνης Τάξης, που έχει επίγνωση του επιπέδου προσοχής των μαθητών, να εντοπίζει προβληματικές καταστάσεις μέσα στο εκπαιδευτικό περιβάλλον (π.χ. έναν ή περισσότερους κουρασμένους μαθητές ή μαθητές που έχουν χάσει την προσοχή ή το ενδιαφέρον τους για το μάθημα) μέσω του συστήματος LECTOR, το CognitOS μπορεί να λειτουργήσει ως κανάλι παρουσίασης και εφαρμογής παρεμβάσεων, οι οποίες έχουν σκοπό να επαναφέρουν τους μαθητές στην εκπαιδευτική διαδικασία.

Δεδομένου ότι κάθε σχολικό περιβάλλον εμφανίζει διαφορετικά χαρακτηριστικά, που αφορούν τόσο τους μαθητές όσο και τη διαρρύθμιση της τάξης, το CognitOS υποστηρίζει εκτεταμένη διαμόρφωση της λειτουργικότητάς του. Η αρθρωτή αρχιτεκτονική του επιτρέπει την εύκολη ενσωμάτωση νέων εφαρμογών, ενώ η λειτουργικότητά του μπορεί να επεκταθεί/τροποποιηθεί κατά βούληση. Για το σκοπό αυτό, το CognitOS παρέχει ένα επεκτάσιμο API το οποίο μπορεί να χρησιμοποιηθεί περαιτέρω από τους εκπαιδευτικούς για την

προσαρμογή της συμπεριφοράς του συστήματος και τον καθορισμό κανόνων για τις διάφορες περιπτώσεις χρήσης, προκειμένου να αντιμετωπιστούν σωστά οι ανάγκες του εκάστοτε εκπαιδευτικού περιβάλλοντος.

Συγκριτικά με τις υπάρχουσες συναφείς εργασίες, το CognitOS είναι ένα καινοτόμο εργασιακό περιβάλλον, που εστιάζει στους μαθητές και παρακολουθεί την προσοχή τους στο μάθημα. Το σύστημα αυτό, εκτός από το να παρέχει εκπαιδευτικές εφαρμογές, αποτελεί και ένα διαμορφώσιμο περιβάλλον, μέσω του οποίου εξατομικευμένες παρεμβάσεις μπορούν να επιτελεστούν με αποτελεσματικότητα στους μαθητές και τον εκπαιδευτικό.

Λέξεις κλειδιά: Ηλεκτρονική Μάθηση, Εκπαιδευτικές Εφαρμογές, Έξυπνη Τάξη, Μεταφερόμενες Διεπαφές, Διάχυτη Νοημοσύνη, Παρακολούθηση Προσοχής

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the recent past, there has been a growing interest in how technology can improve the efficiency of education at all levels. When used appropriately, Information and Communication Technologies (ICTs) can expand access to information, strengthen the relevance of education to the modern digital workplace and raise educational quality by transforming learning into an engaging, active process connected to real life [1]. Until recently, these technologies were mainly used to empower concepts like distant learning [2], intelligent tutoring and e-learning applications [3], and educational games [4]. Starting the past decade, the notion of Intelligent Classrooms has prevailed, envisioning classrooms equipped with technologically enhanced artifacts that aim to replace the traditional furniture (i.e., student and educator desks, classroom board) [5]. This vision was incited by the emergence of Internet of Things (IoT) and the increased availability of Ambient Intelligence (AmI) environments, which have subsequently led to an abundance of internet-enabled physical devices and online services that can communicate and collaborate [6] in an intelligent manner, forming environments able to anticipate user needs and improve the quality of life [7]. Hence, aiming to take advantage of the opportunities offered by such intelligent environments, the domain of education has made several steps towards the creation of student-oriented Intelligent Classrooms that promote and advance the learning process [8], [9] which, for example, may support: video and audio capturing of the lecture [10], adaptation of the environment according to the context of use (e.g., lowering the lights for a presentation) [11], enhancement of lecture's video streaming with annotations, delivery of personalized content [12] and information sharing between students.

Nevertheless, these approaches have only scratched the surface of how AmI can enhance the educational process. Recently, researchers have been exploring more sophisticated ways for engaging students. The concept of "student engagement" is predicated on the belief that learning improves when students are inquisitive, interested, or inspired, and that learning tends to suffer when students are bored, dispassionate, disaffected, or otherwise "disengaged." [13]. An indicator of whether students are motivated to participate to the course's activities is their degree of attention. Hence, it became a necessity for Intelligent Classrooms to incorporate mechanisms that enable the identification of students that exhibit inattentive behaviors and require assistance to re-engage with the activity at hand. To that end, the concept of the Attention-aware Intelligent Classroom [14] has emerged.

1.1 Overview

Currently, such environments consist of a wide range of technologically augmented artifacts able to present intelligent, multimodal and context-aware educational applications, such as dictionaries and translators, multimedia viewers, digital versions of exercises, electronic books, etc. The most commonly encountered artifacts are the students' personal computers and tablets, the smart whiteboards, as well as custom-made student desks with integrated PCs [15]. In order to successfully enhance the educational process and achieve a better learning experience, these artifacts need to not only work independently, but collaborate as well. However, their diverse characteristics outline on the one hand the need for a unified working environment to assist both students and educators, and on the other hand the interaction challenges that stem from the cross-device migration of the available educational applications.

To address these issues, this work introduces CognitOS, a web-based window manager that hosts educational applications and instantiates a common Look 'n' Feel across the various classroom artifacts, thus transforming the classroom into a unified environment rather than a group of isolated units.

Apart from being a simple educational application host, CognitOS also aims to support the mission of Attention-aware Intelligent Classrooms as reflected by its Latin-based name that means "motivated". Therefore, CognitOS can – when it is deemed necessary by the LECTOR framework [14] – act as a host of interventions aiming to (i) draw the educator's attention on problematic situations and (ii) re-engage distracted or unmotivated students in the educational process.

In addition, CognitOS implementation supports the notion of software design reusability – a major concern of the programming community. CognitOS is a customizable framework, offering tools to create new applications. Also, its structure can be altered on demand, by adding or removing existing applications, as well as configuring their functionality. Finally, CognitOS provides an API containing functions for customizing its behavior. Inside a classroom environment, there are several different cases of use for CognitOS, as for example during a lecture or an examination period. Through the provided API, various rules can be defined that alter CognitOS's behavior, according to each use case.

1.2 Thesis structure

The rest of this master thesis is divided in eight (8) main chapters as indicated in the table of contents. The structure is as follows:

- Chapter 2. The second chapter presents an extensive literature review on the introduction of ambient intelligence in classroom environments and how it affects the educational process. Also, the correlation of attention with learning is analyzed and the best practices to re-engage students are explored further. Finally, the importance of utilizing unified environments is discussed and how this laid the groundwork for CognitOS.

- Chapter 3. The third chapter explores the state of the art in the direction of the Attention-aware Intelligent Classroom before this work and how this system can further enhance this concept, introducing elaborate features in a versatile multimodal application.

- Chapter 4. The forth section explores the design process followed in order to construct the front end of the system. In this section, the requirement elicitation process is presented, as well as the iterative prototype evaluation, before finalizing the front end design. In addition, the requirements extracted during this process are also presented.

- Chapter 5. The fifth chapter is dedicated to an elaborate analysis of the CognitOS service. In more details, the system architecture is explained, followed by an analysis of the system's three (3) main facets. A full presentation of CognitOS's front end on all artifacts that are supported is also given, as well as a description of the system's technical aspects.

- Chapter 6. The sixth section explores the aspects that extend CognitOS beyond a service.

- Chapter 7. The seventh chapter illustrates the cases of use of the system, as well as the different modes it alternates between based on contextual parameters. In addition, examples of the system's behavior for each mode are also presented.

- Chapter 8. The eighth section contains an analysis of the evaluation process followed, the scenarios and tasks that were created to achieve it and a report on the findings that were extracted.

- Chapter 9. The final chapter presents a summary of this work, as well as a discussion on future projects that stemmed from CognitOS.

Chapter 2

Background Theory and Related Work

2.1 Background Theory

2.1.1 Learning Management Systems (LMS)

Learning Management Systems were introduced as the framework that can successfully handle all aspects of the learning process [16]. An LMS is the infrastructure behind managing and delivering instructional content, able to identify and assess individual learning, track learning progress, and collect and present data for supervising the learning process [17]. An LMS's primary goal is to deliver content, but also handle course registration and administration, skills gap analysis, tracking and reporting [18].

Other concepts, such as distance learning and virtual learning, are often confused for LMS [19] [20]. However, these concepts represent modern advancements in the education process, which in most cases involve the utilization of ICT tools and technologies [21]. LMS is also perceived as a software application that uses the internet as a medium to support education and the learning process [22].

Also referred to as "learning platforms", "distributed learning systems", "course management systems", "content management systems", "portals", and "instructional management systems", LMS combine a range pedagogical tools, as well as course and subject management, in order to construct online learning environments by providing a means to designing, building and delivering such infrastructure. LMS have the potential to affect the core business of teaching and learning in unanticipated ways [23].

Pedagogic scenarios, otherwise inaccessible to the traditional learning process, can be successfully achieved through these systems, which employ a broad range of Internet technologies, such as personalization, simulation, and mobility [24] [25].

LMS feature a variety of mechanisms to assist educators in managing and delivering online courses [26]. However, these systems do not typically take into consideration the different characteristics of individual learners, hence they treat them all equally, regardless of their personal needs and learning characteristics. Different prior knowledge, cognitive abilities, learning styles and motivation are some examples of individual differences that affect the learning process and are the reason why some learners are having difficulties on a specific course, whereas others can learn it with ease [27].

2.1.2 Ambient Intelligence in the Classroom

AmI environments are designed based on a user-centered approach, aiming to anticipate user needs and support sensibly and proactively their everyday life [7], [28]. Transforming the traditional classroom into an AmI environment could potentially have significant positive impact on learning, collaboration, communication and performance of the students, but also assist educators during their demanding mission of disseminating knowledge [29]. Inside an AmI classroom, activities are enhanced and augmented through the use of pervasive and mobile computing, sensor networks, artificial intelligence, robotics, multimedia computing, middleware and agent-based software [30]. Additionally, commercial artifacts such as interactive whiteboards, touch screens and tablet PCs, have gained popularity and have become a major tool in the educational process. However, the proliferation of ICTs raises the need for making human-computer interactions natural and transparent to the users, thus helping them to focus on learning, or lecturing, rather than on the technology [29]. That way, the educational process is improved by facilitating the communication between educators and students, as well as classroom-wide collaborations [8].

Regarding the necessity of maintaining students’ engagement throughout a course, attention-aware classrooms [14] take advantage of the ambient facilities of the AmI environment to identify when students require assistance and intervene to support them. Such classrooms have much to contribute to educational research and practice. Particularly, they can influence the delivery of instructional materials, the acquisition of such materials from presentations (as a function of focused attention), the evaluation of student performance, and the assessment of learning methodologies (e.g., traditional teaching, active learning techniques) [31].

2.1.3 Attention and Education

Psychology defines attention as the cognitive process of selectively concentrating on one aspect of the environment while ignoring other things. An indicative example of an attentive behavior is when listening carefully to what someone is saying, while ignoring external sounds of the surrounding environment, such as noise or other conversations. In other words, attention means focusing the consciousness and preferentially responding on a stimulus or a range of stimuli. According to Packard [32], "classroom attention" is defined as a set of stimulus-response relationships. These relationships involve both educator's instructions and students' behavior (i.e., looking, listening, being quiet).

People generally find difficulties in maintaining a constant level of attention over extended periods of time, while passively listening. This mental state often results in mind wandering and attention lapses, which are recognized to be common phenomena both within the traditional classroom environment (e.g., [33]–[35]) as well as online education (e.g., [36], [37]). In addition, according to [38], [39] activities such as lecturing or studying require higher attentional demands than others (e.g., cooking, commuting, etc.).

Research has shown that learning is tightly coupled with paying attention during a lecture [31], [39]. Obtaining and maintaining the students' attention is very often considered as a fundamental prerequisite of learning, as it ensures motivation, engagement, and overall successful academic performance [40], [41]. However, inattentive behaviors are inevitable during educational activities and produce negative consequences, which have long been a major concern of educators [42], [43], who put a lot of effort trying to maximize students' engagement.

McKeachie [44], in his book on tips for lecturers, suggests that student attention will inevitably drift during passive lectures, unless interactive strategies, that can maintain student attention, are introduced. Additionally, experimental studies [33], [34] reveal that students do not pay attention continuously for more than twenty (20) minutes during a lecture. Instead, their attention alternates between being engaged and non-engaged in ever-shortening cycles throughout a class period.

Inside the classroom environment, student behaviors can be classified as either "appropriate" or "inappropriate" [45]. The first term – appropriate – includes paying attention to the educator during a lecture, actively participating by raising hand and patiently waiting for response, following along while reading texts and working on assignments while seated. The latter term – inappropriate – is used to describe inattentive characteristics, such as rattling papers, tapping feet or even getting out of seat, loud talking or carrying conversations with other students, laughing, singing or pointing at other classmates. While, to many educational activities, the aforementioned behaviors would in fact be disruptive, students should not be forced to be quiet, docile and obedient “young adults” [46]. On the contrary, in order to ensure effective learning, students' curiosity should be enhanced, while their desire to think and act for themselves remains intact. In order to get the most out of the knowledge that both the educator and the ICTs have to offer and refrain from creating an obsessively still and quiet environment, a classroom environment that aims to motivate students to be engaged in the learning activities, should support the aforementioned requirements.

Attention aware systems have much to contribute to education, since they can influence the delivery of instructional materials, the evaluation of student performance, and the assessment of learning methodologies, traditional or not [31]. However, existing approaches [47]–[51], concentrate mainly on computer-driven educational activities and lie far from employing attention monitoring in a real classroom setting.

2.1.4 Re-engaging students to the Educational Process

Literature suggests several strategies to regain student attention and increase the level of engagement in learning activities; among them, Active Learning [52]–[54] was acknowledged as the most effective instructional method in terms of resetting the students' concentration and decreasing attention lapses during lectures. Unlike the traditional way of lecturing where students passively receive information from the instructor, it introduces active student participation in the course. When designing active learning activities, the main goal should be to deliver important learning outcomes, while promoting thoughtful and meaningful student engagement. In addition to Active Learning, encouragement is another technique which, when supported throughout the educational process, has proven to have positive effects on student behavior. According to [55], most students thrive in encouraging environments where they receive specific feedback and have the opportunity to evaluate their own behavior and work. Thus, through encouragement, the classroom can be transformed into an environment free from the fear of failure, where students look forward to continuous evaluation and learn from their mistakes, while they do not have to meet certain standards of excellence. Moreover, literature [34] states that changing pedagogical activities within a class period, could help in engaging students in subsequent learning formats, apart from also being seen as means to present concepts in alternate formats.

Literature indicates various applications that have been conceptualized and developed for resolving issues of inattention and disengagement inside the educational settings. Morgan [56] researched the effects of an interactive whiteboard used in conjunction with a computer and found that it has positive effects to student attention, as opposed to a traditional setting of a plain board and an educator’s personal computer. Szafir et al. [57] and Brown et al. [58] reviewed the effects of a robotic educational agent. Both studies focused on identifying inattentive behavior using either neural signals [57] or idle time during a human-tablet interaction [58] and used behaviors comparable to that of a human educator, such as verbal (praise, motivation, etc.) and non-verbal (gestures), to deliver instructions. The results were positive, mentioning that students didn’t want to “disappoint” the agent, and thus were more focused on their tasks. FishBuddy [59] is a self-paced learning environment that introduces an Apple Watch application that measures students' heart rate variations in order to identify whether they feel anxious. As an anxiety-reducing intervention, it creates visualizations of a fish that swims freely in a rhythm inversely related to the student's heart rate.

2.2 Related Work Overview

People are increasingly using an abundance of devices daily and, in many cases, these devices need to work together in a seamless manner towards achieving the same goal, a term known as cross-device interaction [60]. Regarding systems that have the ability to migrate amongst various devices, Rowland [61] wrote that it is not enough to design individual User Interfaces (UIs) for each device in isolation. On the contrary, the top priority should be to create a coherent understanding of the system, as well as a solid intercommunication between devices. Wäljas et al. [62] define three key concepts for cross-platform service User Experience (UX), which together ensure a coherent experience:

Composition refers to the way the functionality of a service – especially the user-facing functionality – is distributed across devices. Good composition distributes functionality between devices to make the most of the capabilities of each device. Designers should consider the context in which each device will be used, and what users expect each to do.

Consistency works to create a sense of coherence of the overall system. It is important to make the devices look, feel and sound like a family so that users form a clear mental model of the system and its capabilities.

Continuity refers to the flow of data and interactions in a coherent sequence across devices. The user should feel as if they are interacting with the service through the devices, not with a bunch of separate devices.

Since nowadays Intelligent Classrooms contain a wide range of technologically augmented artifacts, a mechanism that transforms the classroom into a unified environment rather than a group of isolated units, respecting the concepts of Composition, Consistency and Continuity is of outmost importance and CognitOS aims to deliver such an experience.

Over the years, several commercial web-based applications have been developed that aim to facilitate the educational needs inside a classroom environment. The vast majority of those are game-based cross-platform systems that host either a collection of quizzes, sets of educational activities or offer dashboards to facilitate learning. In more detail, the various systems can be divided in three categories, regarding their general approach, i.e., educator-oriented, student-oriented activity hosts and dashboard-related approaches.

2.2.1 Student-oriented Activity Hosts

There is a variety of commercial systems offering various educational activities, whose main aspiration is to enhance learning and engage students in alternate learning formats. This collection of systems can be divided in two categories; those that offer configurable learning content and those with predefined activities.

Acadly [63] offers a mechanism to keep student attendance, on the press of a button, while engaging them to a variety of polls and multiple choice quizzes, customizable by the educator (Figure 1). Acadly also supports messaging between the students and the educator to support real-time communication and assistance.

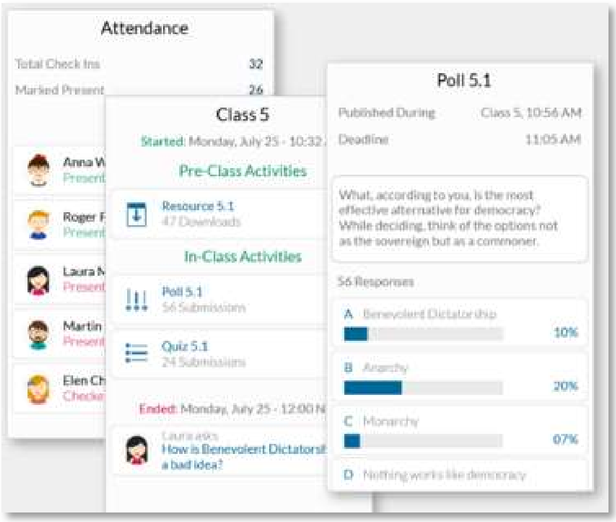


Figure 1. Acadly keeps track of student attendance and offers polls and multiple-choice quizzes to ensure student engagement

Courselle [64] is an activity host, which facilitates the creation of interactive training courses (Figure 2), enriched with quizzes, tests, expert and peer-graded assessments. Students can engage in the aforementioned activities and review their progress with report cards and statistical analysis.

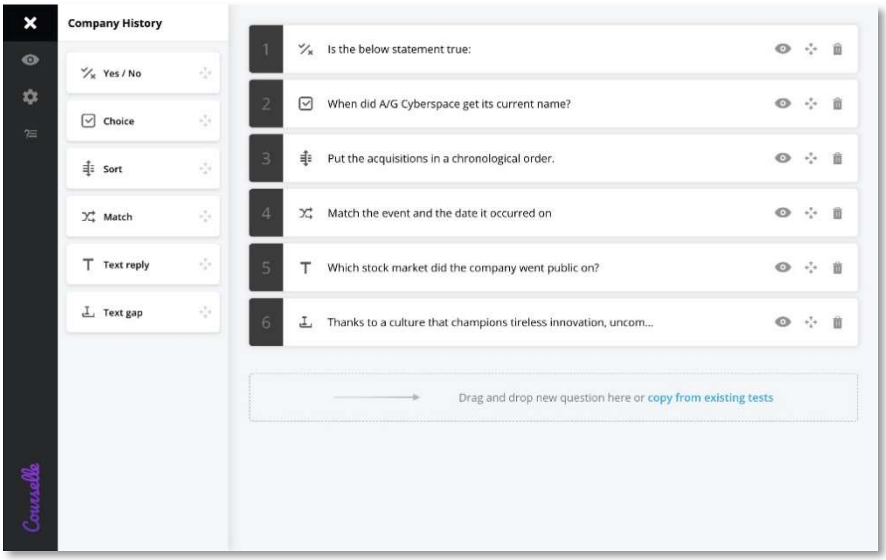


Figure 2. Creation of interactive training course through Courselle's editor

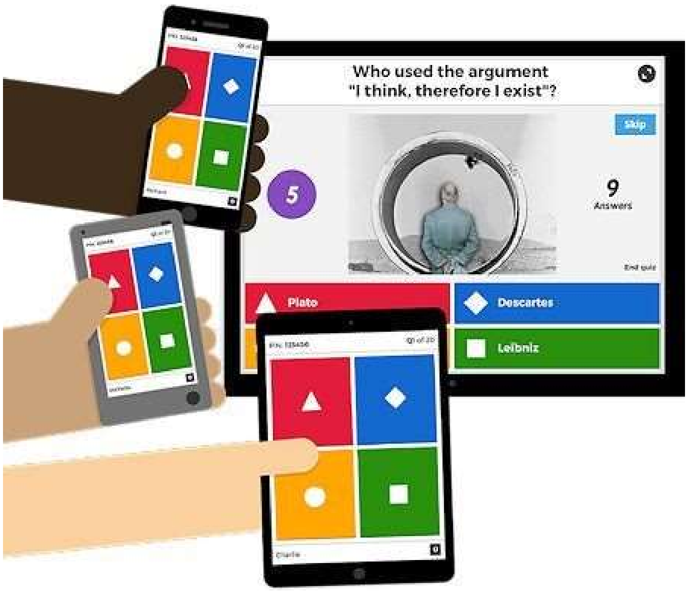


Figure 3. Kahoot! displays a question on a public artifact, while students answer using personal devices

Kahoot! [65] is a game-based system hosting a variety of learning games and trivia quizzes, in the form of multiple-choice questions. The content of these quizzes (or 'kahoots') can be defined by the educator and enhanced with videos, images or diagrams to amplify engagement. These kahoots are presented in a public artifact, while each student uses their own device to answer (). Kahoots can be used as assignments and can help review student progress.

Quizalize [66] is a game-based system, which aims to turn boring tests into fun classroom team games. It offers a collection of customizable educational activities and quizzes presented on the whiteboard. Students team up to collaboratively solve the quizzes and get on-the-fly assistance by the educator. Quizalize provides educators with instant student feedback to direct their focus on students who need assistance (Figure 4). After each quiz, students are provided with follow-up material depending on the level of their understanding of each topic.

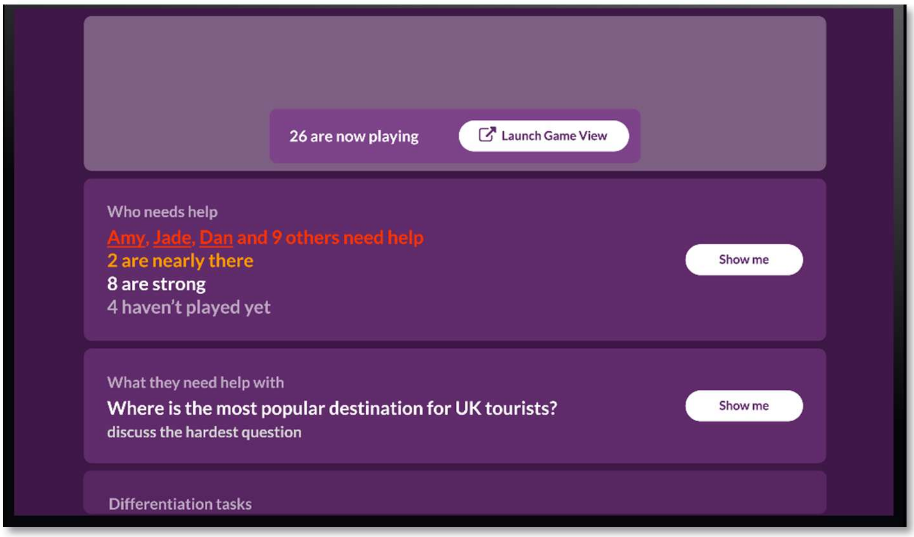


Figure 4. Quizalize offers real-time feedback on students who need the educators assistance

Classflow [67] is an activity host for laptops, tablets and smartphones, where students can (individually or collaboratively) solve a variety of ready-made quizzes, polls and engage with various educational activities (e.g., multiple choice, true/false, creative responses, etc.) (Figure 5). Classflow incorporates a badging system to reward outstanding performance.

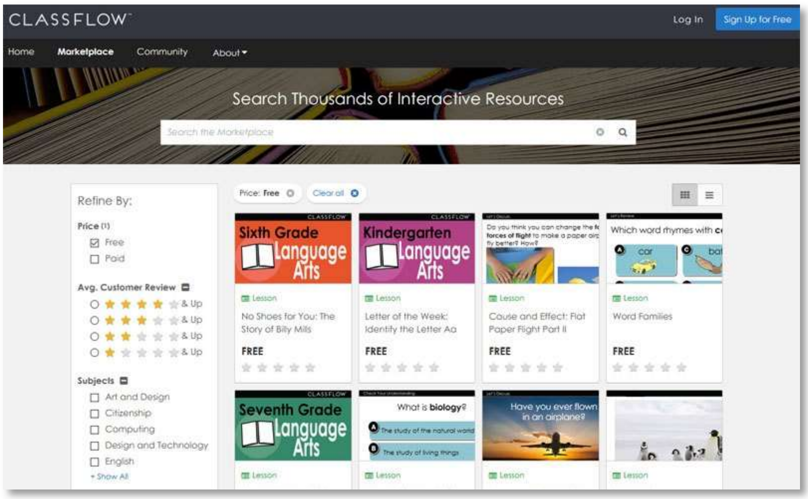


Figure 5. Classflow offers ready-made educational activities

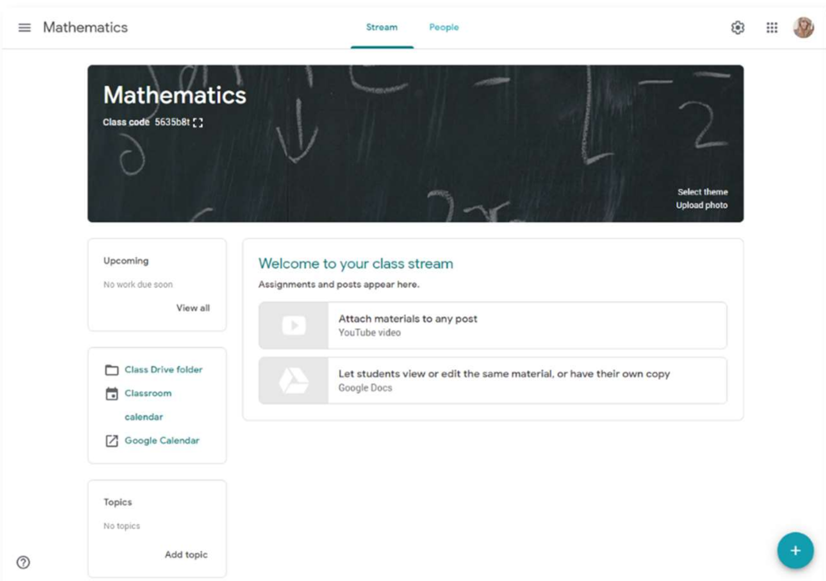


Figure 6. Google Classroom contains courses and lessons, created by the educators

Google Classroom [68] is the virtual counterpart of a physical classroom, containing various courses and lessons. Educators can create courses, lessons and assign homework to their students. Students can review, solve and submit their assignments, as well as get an overall course overview (Figure 6).

Moodle [69] is a learning platform, which provides a single robust, secure and integrated system to create personalized learning environments. Moodle's functionality can be expanded by applying various plugins offered by the system. These plugins (e.g., Reengagement, Mootivated, Engagement analytics) offer tools that can motivate and reengage students to the lecture (e.g. a reward system for completing small challenges, such as completion of courses or collaborative assignments) (Figure 7). Moodle also offers report cards, where students can review their progress per course.

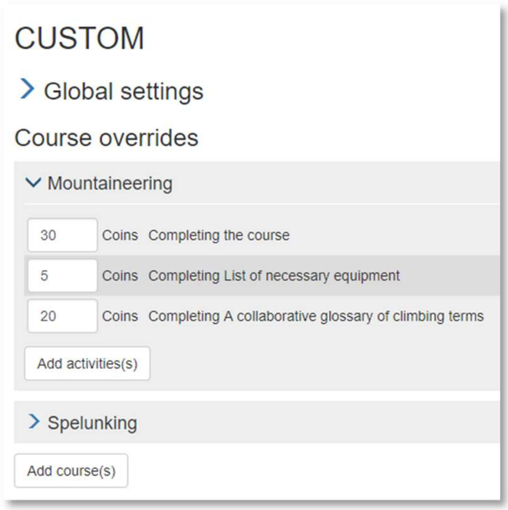


Figure 7. Moodle's 'Mootivated' plugin customization and reward system

Quizlet [70] is a game-based system, hosting a variety of study sets. These include predefined educational activities, such as flashcards, matching games (Figure 8), spelling quizzes, multiple choice questions and true/false quizzes. The study sets are divided by course and incorporate various learning styles ('Study Modes'), like 'Flashcards', 'Write', 'Spell', etc., to fit better the educational needs.

2.2 Related Work Overview

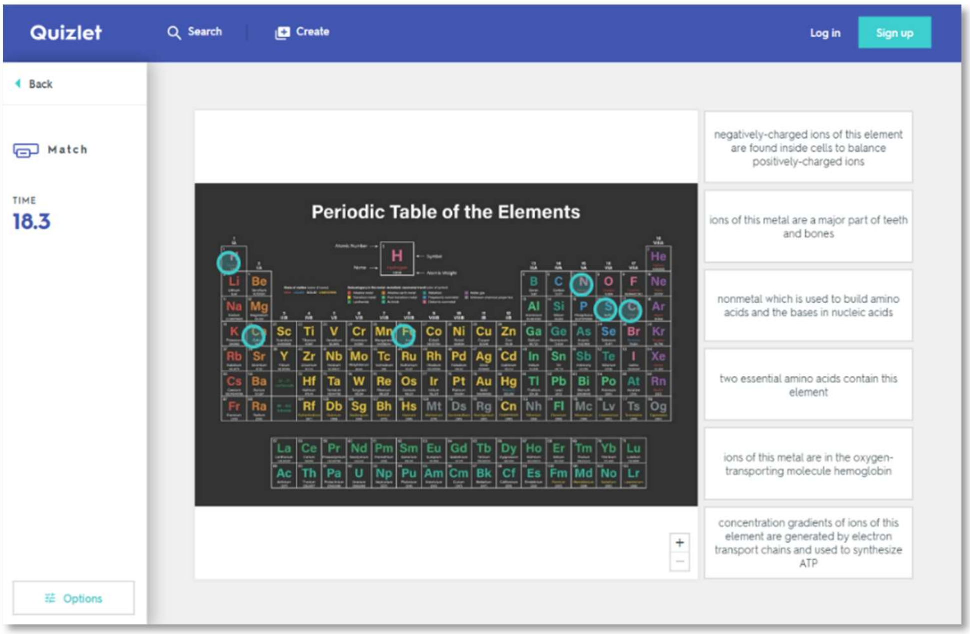


Figure 8. Quizlet's matching game on Chemistry

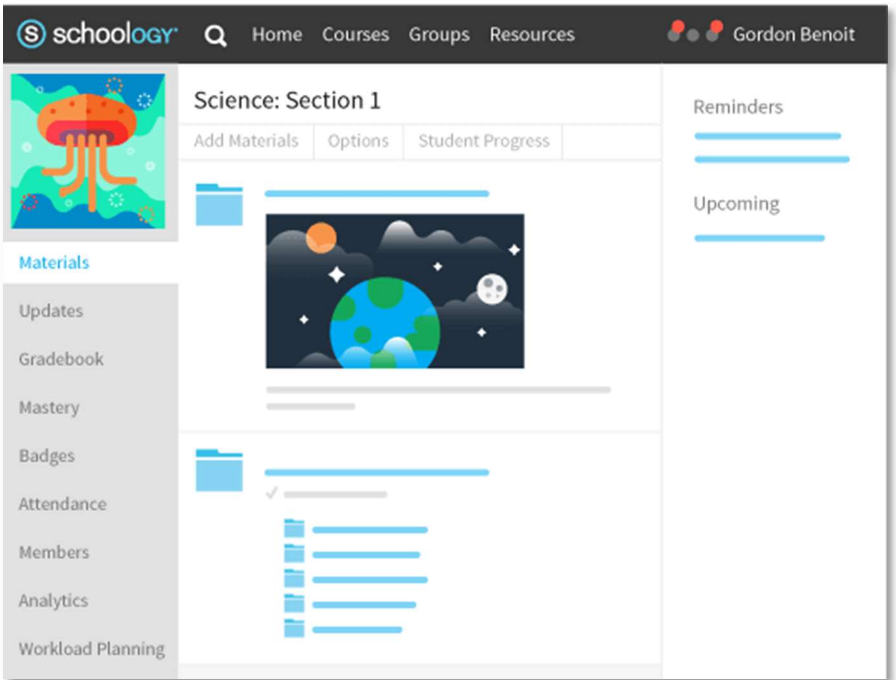


Figure 9. Schoology's course overview, containing tools such as 'Attendance' and 'Analytics'

Schoology [71] is a learning management system containing a variety of educational tools. These tools aim to improve student performance and engagement, such as course analytics and student mastery reporting, foster collaboration and personalize learning (Figure 9). Students can review analytics on their performance as well as their grades per course.

Seesaw [72] contains a library of activities created and shared by educators (Figure 10). The activities include image annotating, drawing and multiple choice questions, which aim to encourage student engagement, while it offers tools to keep track of student progress such as insights to understand students' strengths.

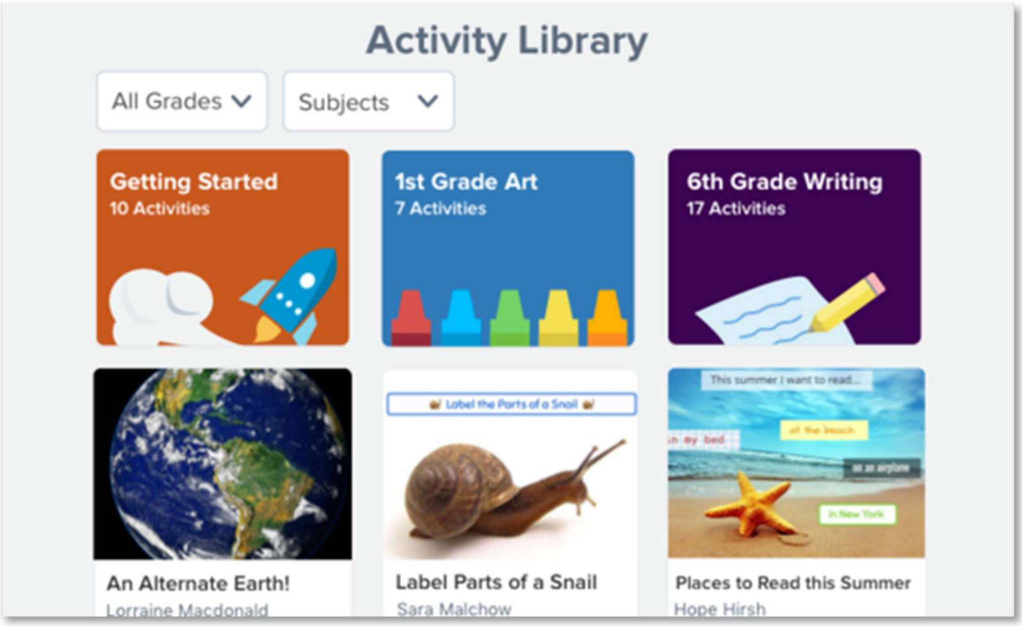


Figure 10. Seesaw offers a library of activities, which include drawing, writing and more

2.2.2 Educator-oriented Systems

This category features systems that support educational activities from the perspective of the educators.

BookWidgets [73] is an educator-oriented application for iPads, tablets or laptops, where educators can create their own interactive exercises or use any of the predefined educational apps that get instantly graded. Its library offers a collection of various activity types, such as flash cards, crosswords, multiple choice and “fill the blanks” quizzes (Figure 11).

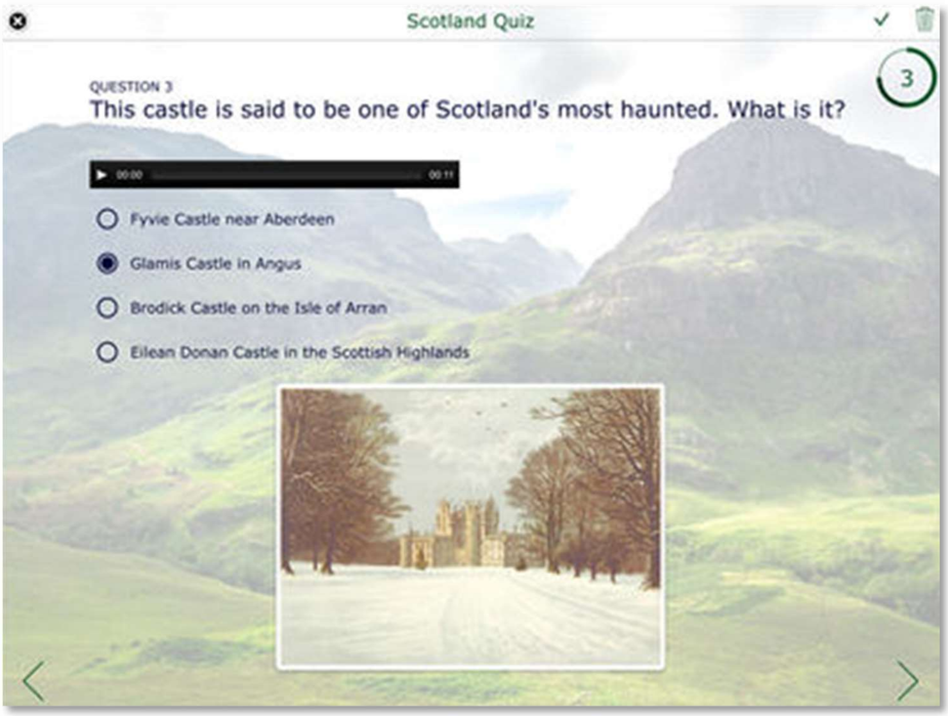


Figure 11. BookWidgets offers multiple choice quizzes

Classtime [74] is an educator-oriented system, which compliments in-class teaching with immediate feedback on students’ level of understanding (Figure 12). Classtime offers libraries with collections of questions from various domains (e.g. Mathematics, Science, Chemistry), that enable educators to create their own quizzes. Also, it supports collaboration by letting students engage with various collaborative challenges (e.g., fighting city pollution, engineering a roller coaster, etc.).

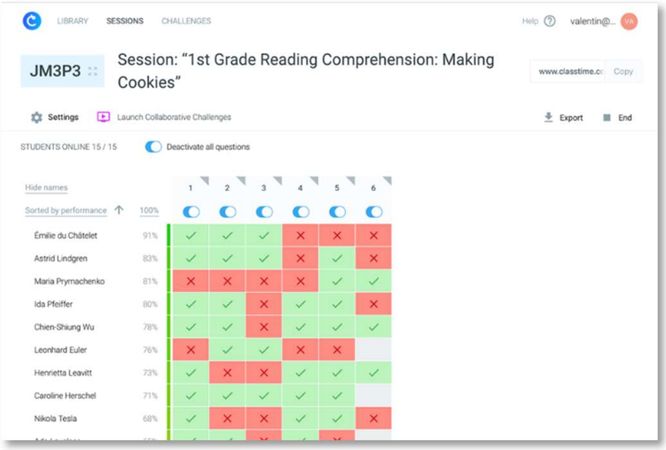


Figure 12. Classtime offers immediate student assessment

Edmodo [75] is an educator-oriented system hosting a variety of educational activities, such as assignments and quizzes (e.g., multiple choice, "fill the blanks", true/false questions). Educators can review submitted assignments and provide students with comments, highlights and annotations (Figure 13). The grades for each course are stored in a universal gradebook, which is accessible only by the educator.

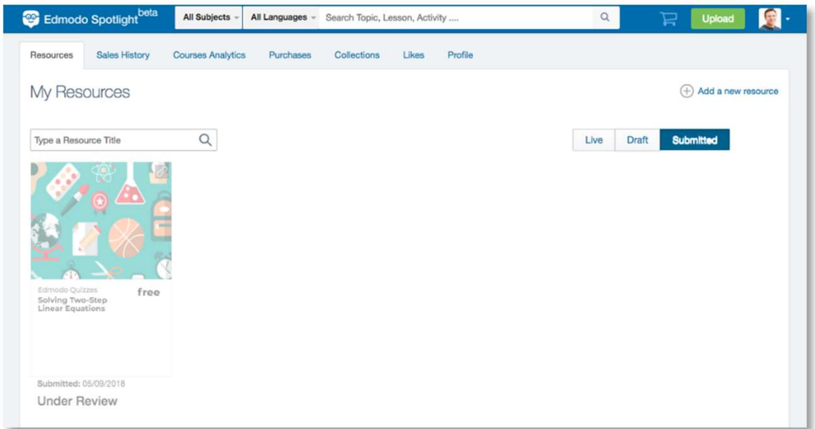


Figure 13. Edmodo contains a collection of all pending assignments for reviewing

Figure 14. Socrative supports quiz configuration

Socrative [76] is an educator-oriented application for computers, laptops, tablets and smartphones, which offers configurable quizzes (Figure 14). These quizzes support: (i) multiple choice, (ii) true/false questions, or (iii) questions which require short answers. These quizzes can either be solved individually or collaboratively. Socrative automatically generates report cards to assist educators with reviewing student progress.

2.2.3 Dashboard-related Approaches

This category features two distinctive applications developed to facilitate learning in a different manner than the aforementioned systems.

Padlet [77] is a virtual student pad, which uses dashboards to support learning. Students can choose from premade dashboard templates or create their own. By collaborating with their peers, they can create shared dashboards, where they can add content, comment and make edits. These dashboards can include photos, documents, web links, videos or plain text (Figure 15).

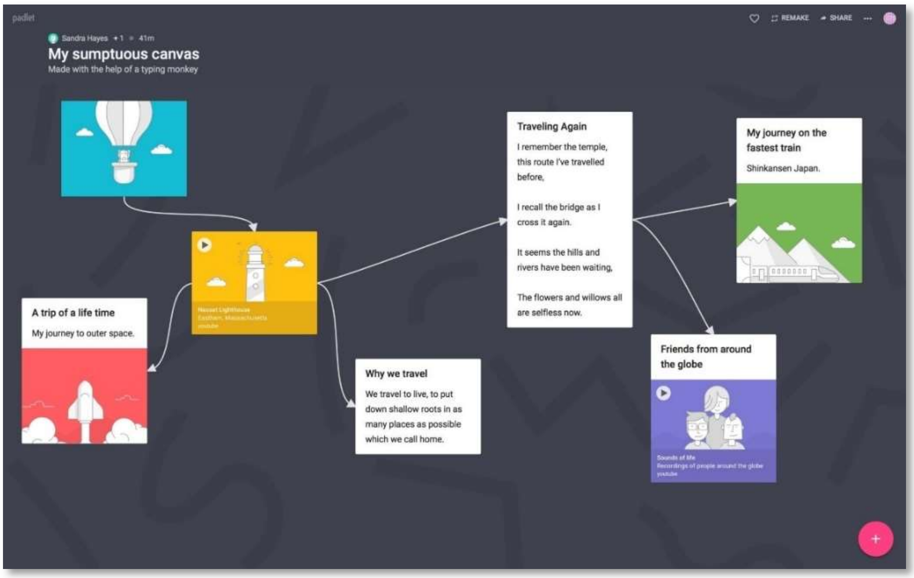


Figure 15. Padlet uses individual or collaborative dashboards, which contain photos, documents or text to support learning

Popplet [78] is a tool for desktops and tablets to facilitate capturing and organizing ideas. It acts as a mind map (Figure 16), which supports visual learning through mapping images with facts about them.

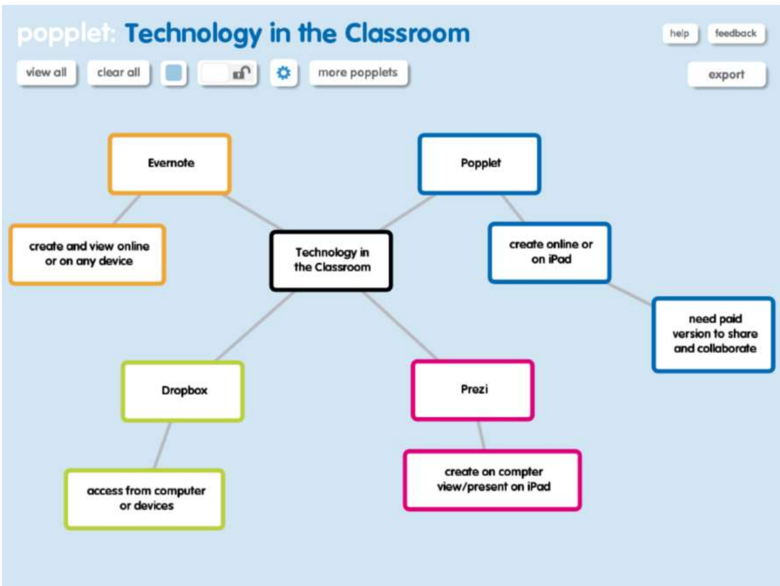


Figure 16. Popplet is a virtual mind map to assist students in mapping facts

2.2.4 Discussion

The most important aspect that differentiates CognitOS from conceptually similar systems is its ability to communicate and collaborate with the classroom environment, in order to adapt to the various parameters inside an educational setting and facilitate learning (Table 1). Apart from being context-aware, CognitOS can also present interventions, when an inappropriate behavior inside the classroom is identified, in order to re-engage students to the educational process. Finally, CognitOS enables educators to configure its overall functionality, by choosing appropriate applications to launch at each specified time during a lecture, with specific attributes and content.

Table 1. Related work compared to CognitOS

		Applications			Interventions		Service	
		Hosts Activities	Student Assessment	Uses Dashboards	Attention-aware	Supports Interventions	Configurable Learning Content	Configurable Behavior
Activity Hosts	Classflow [67]	✓	✓					
	Google Classroom [68]	✓	✓	✓				
	Moodle [69]	✓	✓	✓				
	Quizlet [70]	✓						
	Schoology [71]	✓	✓					
	Seesaw [72]	✓	✓					
Configurable Content	Acadly [63]	✓					✓	
	Courselle [64]	✓		✓			✓	
	Kahoot [65]	✓					✓	
	Quizalize [66]	✓	✓				✓	
Educator-oriented	BookWidgets[73]	✓						
	Classtime [74]		✓				✓	
	Edmodo [75]	✓	✓	✓				
	Socrative [76]	✓	✓				✓	
Other Ideas	Padlet [77]			✓				
	Popplet [78]			✓				
Attention and Engagement	Brown [58]				✓	✓		
	FishBuddy [59]				✓	✓		
	Morgan [56]				✓			
	Szafir [57]				✓	✓		
	CognitOS	✓	✓	✓	✓	✓	✓	✓

2.2 Related Work Overview

Chapter 3

The Attention-aware Intelligent Classroom

3.1 Intelligent Classroom B.C. (Before CognitOS)

During the past decade, an initial approach has been made towards integrating AmI services in the classroom environment inside the FORTH-ICS AmI Facility Building. An in-vitro simulation space of an Intelligent Classroom has been set up, containing technologically augmented traditional classroom artifacts, as well as custom made pieces. The most notable example of a custom-made artifact is the technologically augmented student desk [15], [79]. It features a 27-inch multitouch-enabled All-in-One PC, integrates various sensors and cooperates with the students' personal devices (e.g., tablet).

The architecture of the Intelligent Classroom [12] is a stack-based model, where the first layer, namely the middleware infrastructure, serves the interoperability needs of the classroom. The next two layers, namely the ClassMATE [80] and the PUPIL [81] frameworks, expose the core libraries and finally the remaining layer contains the educational applications.

The ClassMATE framework is an integrated architecture for pervasive computing environments, that monitors the ambient environment and makes context-aware decisions, in order to assist the student in conducting learning activities and the teacher with administrative issues. The PUPIL system delivers interaction workspaces customized to the needs of each "intelligent" artifact, thus forming an educational-ecosystem across the entire class. The PUPIL framework offers a library of ready-to-use, adaptive, educational-oriented graphical components designed to optimally interact with the various educational applications of the ambient classroom on different devices.

Inside the Intelligent Classroom, every educational application can be launched, manipulated and migrated in any intelligent artifact. For that to be achieved a collection of adaptation-aware Classroom Window Managers (CWMs) was developed [82] to host the various educational applications and automatically adapt them according to the needs of the targeted classroom artifact, thus ensuring their portability.

The aforementioned window managers are employed in the technologically enhanced ambient classroom, where every artifact incorporates situation-aware functionality offered by the ClassMATE platform. The cWMs were developed following a user centered design process (UCD) and focus on the students' needs gathered through a requirements elicitation process. Furthermore, each cWM has unique characteristics, as it is purposed for a specific artifact (i.e., student's desk, classroom board) trying to create an ideal workspace environment.

Through a combination of technology, which is fascinating by its nature, and augmented interaction through physical objects, AmI technologies have the potential to enhance the classroom learning experience ([12], [80]), as learners seem to be more willing to spend time on modern, technology-enabled, applications rather than traditional approaches. In the context of the above, two simple, yet indicative, educational tabletop mini-games were built [83] that combine learning, entertainment and ambient intelligence, as a proof of concept that the transparent integration of AmI technologies in learning activities is feasible and cost-friendly. The developed games are: (i) a multiple choice quiz game and (ii) a geography-related game. Both games use physical cards as the primary interaction source.

Finally, AmI-RIA [84] monitors and analyzes students' activities in real-time so as to identify potential difficulties, either at an individual or at a classroom level, and notify the teacher accordingly (through the teacher's front-end application). The teacher can therefore concentrate on the lecture and rely on the system to monitor the classroom and prompt for intervention only when necessary (e.g., a student is out of task or performed poorly in a quiz). In addition to real-time monitoring, AmI-RIA offers a performance analysis tool that provides extensive metrics of students' progress and performance (based on previously collected data) that the teacher can use to either identify topics that require further elaboration or adapt the teaching methodology. Finally, AmI-RIA integrates tools that automate common classroom procedures, like attendance record keeping, quiz assessment and preparation of lesson's curriculum.

3.2 Intelligent Classroom A.C. (After CognitOS)

Since the development of CognitOS, the Intelligent Classroom has transformed drastically, incorporating newer and more robust software. The AmI Desk remains one of the augmented classroom artifacts, accompanied by commercial devices, such as a touch sensitive interactive whiteboard, while the students are expected to utilize their personal tablets, smartphones and smart watches as auxiliary output devices. Additionally, an abundance of internet enabled devices is installed in appropriate locations inside the classroom, in order to support monitoring the overall environment and the learners' actions (e.g., microphones, user-tracking devices, speakers).

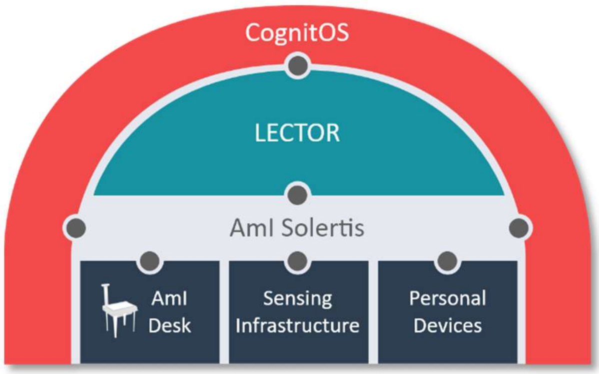


Figure 17. The Intelligent Classroom software infrastructure

Figure 17 presents the software infrastructure that enables the integration, interoperation and inter-communication of CognitOS with the environment. The ClassMATE and PUPIL frameworks have been substituted by the AmI-Solertis middleware. In more details, the AmI-Solertis middleware [85] – the core component of the Intelligent Classroom – is responsible for (i) the collection, analysis and storage of the metadata regarding the environment's artifacts, (ii) their deployment, execution and monitoring in the AmI-Solertis-enabled systems to formulate a ubiquitous ecosystem. On top of AmI-Solertis resides the LECTOR framework [14], which has contributed towards the transformation of the Intelligent Classroom into an attention-aware environment. In particular, it offers a sophisticated mechanism for identifying student behaviors that require remedial actions (e.g., unmotivated student) and a mechanism for intervening and offering help or support. LECTOR also features an authoring tool, named LECTORstudio [86], which supports both developers and educators in creating simple “if-then” rules that dictate the behavior of the classroom (e.g., “if students are chatting during a lecture, then launch a multimedia presentation (based on the lecture’s topic) on every student’s desk, dim the lights to 50% and half-close the blinds”).

CWMs and AmI-RIA have been replaced by CognitOS, which has the ability to incorporate both of their core functionalities. AmI-Solertis monitors the classroom environment, by collecting, analyzing and storing artifact metadata, which afterwards constitute the input feed for LECTOR. Utilizing this input, LECTOR can decide whether problematic situations have been identified. When a remedial action is needed, LECTOR dictates appropriate intervention, for one or more artifacts and gives a command to CognitOS. When a command from LECTOR is received, CognitOS immediately acts on it, by displaying the appropriate intervention dictated by LECTOR. When no problematic situations are present, CognitOS functions as an application host, residing on every classroom artifact, transforming the classroom ecosystem into a unified environment.

The introduction of CognitOS in the Intelligent Classroom enables the fluidity and adaptability of the structure to fit better the wide range of needs inside an educational setting. Since CognitOS is a configurable service, its functionality can be altered according to the various parameters and conditions of the environment. At any time, applications can be added, edited or removed from CognitOS, while the behavior of existing ones can change on demand. Through the available expandable API of CognitOS, these adjustments have not only been made possible – a concept otherwise impossible to the previous classroom setting –, but also easy to achieve.

Chapter 4

Requirement Elicitation

4.1 Design Process

CognitOS philosophy revolves around students and their benefits, hence it was of foremost importance to focus on facilitating their wants and needs, as well as address their limitations. In order to efficiently address this requirement, the User-Centered Design (UCD) [87] approach was selected.

In order to analyze users' characteristics and their tasks, apart from literature review, a team of experts including members of various backgrounds – such as teachers, psychologists, designers, developers and usability experts – was formed to conduct an iterative process of brainstorming sessions. CognitOS focuses on students who have reached the formal operational stage [88]. Literature states that at the aforementioned stage children tend to display similar thinking process to that of adults, while maintaining different interests. As a result, students have the ability to apprehend abstract concepts supported by computer systems, some of which they may be already familiar with. This comes as no surprise following the exposure of today's children to a variety of technological systems – including electronic toys, multimedia and communication devices – from a young age.

According to [89], teenagers experience more difficulty in analyzing and making sense of complex information than adults, as a consequence of having much less research experience than them. Additionally, teenagers often opt for quick answers and refrain from complicated interactions, since they tend to have little patience. Finally, [90] states that a clean and modest design, with visually meaningful icons and age-appropriate instructions easy to comprehend and remember, would be preferred from teenagers over a more childish one, since they are sensitive with their age.

Druin et al. [91] have found that children want to have expressive tools at their disposal and be in control of technologies, which they can use with their peers. To them, it is important that an application is “cool” and does not require much effort to learn. In addition, when an application provides any kind of multimedia, it automatically becomes more appealing. Regarding the latter, Said [92] studied the engagement of children at the ages of nine (9) to fourteen (14) with multimedia and revealed that they become more engaged when immediate feedback is provided, as well as when there is an environment in which they can be in control and set their own goals.

Furthermore, children at this age are searching for activities that promote socialization, which does not only include talking to their classmates, but also share resources they find interesting and team up to collaboratively solve an exercise. Kaplan et al. [93] revealed that even though children (10-14 years old) may be co-present in the same space, they would like to be supported with tools to share their experiences with their friends and teachers.

Since the early 1960, several studies have taken place [94]–[96] researching student goals and motives as drives for learning, which incited a list of tasks that the students should be able to perform to be devised. These tasks include: (i) solving a variety of exercises (e.g., multiple choice, fill the blanks), (ii) getting assistance on exercises, (iii) submitted completed assignments for grading, (iv) retrieve additional resources about something interesting or about an assignment dictated by the teacher, (v) have access to assistive applications (e.g., calculator, dictionary, etc.), (vi) have access to multimedia, (vii) maintain a personal area with access to history of homework, and (viii) collaborate with classmates to complete a task.

This research revealed a set of preliminary system requirements, used to visualize plausible solutions. Through an iterative process of design and evaluation, a set of prototypes were created and evaluated during a formative evaluation experiment. The results extracted from this process were then used to develop the basic structure of CognitOS, focusing on better meeting the users’ needs, in this context the students’ needs. Once the development phase had advanced and interactive prototypes had been created, a heuristic evaluation was conducted, aiming to eliminate serious usability problems before proceeding to user testing. CognitOS was then improved according to the heuristic evaluation results. Finally, after the full scale development of CognitOS, a second iteration of the heuristic evaluation experiment has been conducted and the results are presented later in this work.

4.2 CognitOS Requirements

4.2.1 Application Hosting

Literature review raised the need for a student-centric unified environment that instantiates a common Look ‘n’ Feel across the different classroom artifacts (e.g., the desktop stations, students’ personal tablets, classroom public board). This environment should host educational applications, offering a student-friendly working environment. Across the various artifacts, these applications need to communicate and collaborate in a seamless manner, leading the users into thinking they are interacting with an ecosystem, rather than a group of isolated units. Each application should be able to launch on demand on every artifact it applies, while also being capable of transitioning between them without losing its state, to ensure continuity – one of the three aspirations to ensure a coherent mental model of the system.

4.2.1.1 Desktop Requirements

All desktop stations should offer a personal, as well as private, workspace for each student, making their needs a priority. Furthermore, during a class period, students may interact with their physical books, either reading or solving exercises. Following this knowledge, the need for an application hosting digital counterparts of the physical books has risen. Also, since the book offers multiple exercises, the system should also include an application containing a collection of all available exercises. Moreover, since student attention is highly required inside an educational setting and changing pedagogies – which includes alternating between lecturing format – is a way of ensuring it, a multimedia player, where educators can choose to display educational content, such as tutorial videos, should also be included to the system. Finally, as a personal workstation, each desktop should include a student profile, where they can review their grades on the various courses.

4.2.1.2 Tablet Requirements

In addition to the desktop, students can choose to utilize their personal tablets. The tablet should be able to dock next to the desktop, acting as an auxiliary device that supports student work on the desktop. In more details, a docked tablet should offer shortcuts to helpful mini applications for each course. However, since tablets are mobile devices, they should be used as a replacement for the desktop when they are not docked. At any point in time during a classroom period, where the main working environment is the desktop, the students should be able to undock them and continue working on their tablets from the point the left off on their desktops and vice versa.

4.2.1.3 Board Requirements

Moreover, as described in the previous section, students need to be able to collaborate and work together. Interactive whiteboards have proven to be able to make significant contributions to children's collaborative communication and thinking [97], [98], thus a public artifact (i.e., an interactive whiteboard) should be utilized accordingly. An Intelligent Classroom could include an interactive white board, which can be used for the aforementioned aim. Hence, the board should display exercises publicly for students to work together in solving them. Also, for the same reasons as for the desktop, an educator should be able to display multimedia on demand, when changing pedagogies or deciding to include active learning in the lecture. Finally, since the primary use of a board is writing notes during a lecture, when idle the interactive board should act as a canvas to enable the educator to freely write.

4.2.2 Intervention Hosting

Recent studies, according to section 2.1.4, researched means of re-engaging tired, distracted or unmotivated students to the educational process, since it is a prerequisite of learning inside a classroom environment. The Attention-aware Intelligent Classroom employs sophisticated mechanisms that are able to identify and act on problematic situations. Hence, a host that acts as a means of presenting the interventions dictated by the classroom mechanisms was of high importance. CognitOS should act as the host of these interventions, which should be able to utilize its applications to display appropriate content.

4.2.3 CognitOS-as-a-Service Requirements

Currently, Intelligent Classroom environments include solid systems, including a variety of preinstalled applications and components, utilizing existing infrastructure, without the ability to, at least, provide means for configuration and, furthermore, customization. A new era of Intelligent Classrooms would benefit from a system that offers ways of configuring its functionality, since it could best accommodate the diverse characteristics of both the educational setting, as well as the students and the educators. CognitOS should provide a way to easily configure its behavior according to contextual parameters. Also, designing a system that offers reusable components can facilitate the future expansion of its architecture.

4.2 CognitOS Requirements

Chapter 5

The CognitOS service

5.1 System Architecture

In order to identify active classroom artifacts and display corresponding UIs for each one, CognitOS employs two different node servers, the CognitOS UI Server and the CognitOS Core Server. The first is responsible for communicating with all different artifacts. In more details, the UI Server is responsible for initiating instances of CognitOS with corresponding UIs to each artifact, with appropriate content, as dictated by the Core Server. The second represents the brain of CognitOS, the backbone of its architecture. The CognitOS Core collaborates with the AmI-Solertis system in order to get metadata of the classroom environment, which include class schedule, current activity and running artifacts. Combining this data with each application's logic and predefined behavioral rules, the Core manages which applications to launch, with the appropriate content. For example, if an educator chooses to initiate an examination from her controller, the command is sent to the UI Server, which forwards it to the Core Server. Inside the Core Server reside predefined parameters by the educator regarding the system's behavior on the occasion of an examination. Taking into consideration those parameters, as well as currently active artifacts, the Core's decision mechanisms dictate the UI Server to launch appropriate applications with specified content. The UI Server's main function is to execute commands received by the Core Server. *Figure 18* presents a graphic representation of the chain of command inside CognitOS.

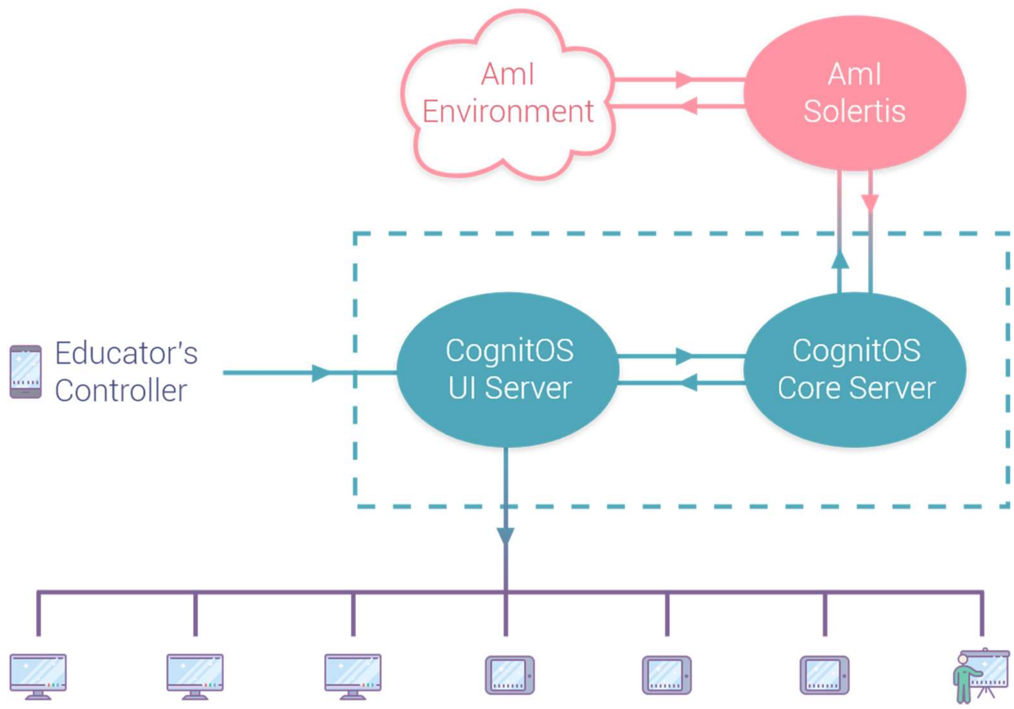


Figure 18. Example in chain of command of CognitOS architecture

Nowadays there is an abundance of front-end frameworks and libraries that support front-end development. One of them, Angular, is designed to work with data directly in the front end while using HTML as a template language. Using the Angular CLI, the command line interface of Angular, working applications can be easily created.

CognitOS's front end is developed using the Angular framework. In more detail, CognitOS is divided into different modules and components. Modules represent the different devices and applications, constructed by a number of components. The core modules of CognitOS are the different devices, i.e., the Desktop, Tablet, Board and Mobile. Each educational application (as well as any other application that can be further added to CognitOS) is also a module. Each module is responsible for routing and navigation between its components. *Figure 19* presents the module-component relationship inside CognitOS.

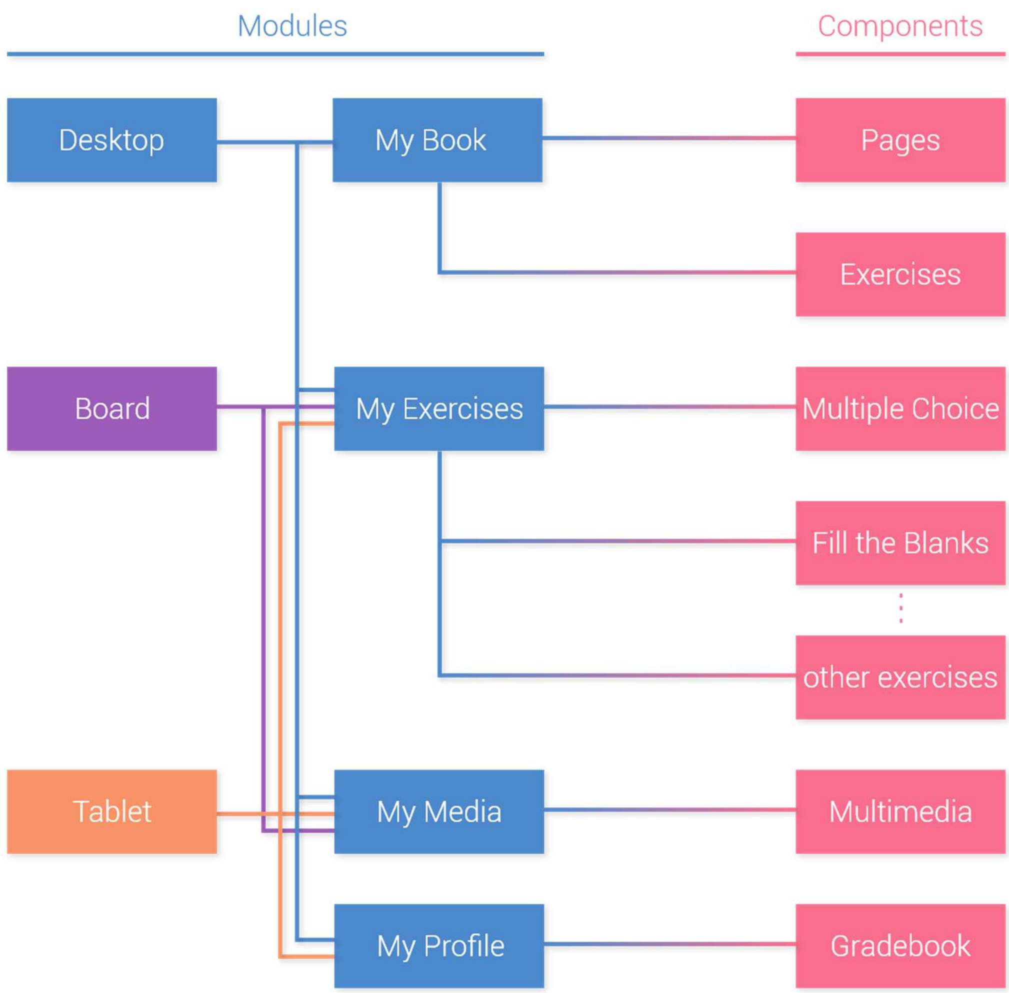


Figure 19. Relationship between modules and components of CognitOS

5.2 CognitOS as an Application Host

CognitOS is deployed currently on four different artifacts of the Intelligent Classroom, including students' and educators' personal devices – desktop PC, tablet, smartphone and smart board. For the educational suite of CognitOS, a number of different applications have been implemented, which utilize the capabilities of each different artifact to achieve an intuitive interaction. Appropriate dashboards have been designed for each of the available artifacts, in order to facilitate educational activities, by creating a practical workspace for students. The term dashboard in this context is used to describe a UI that organizes and presents information in an easy to read and interact with manner, allowing access to all kinds of applications.

One of the three aspirations of CognitOS as a cross-platform system is consistency, in order to achieve a coherent mental model. To achieve a consistent User Interface (UI) across all different artifacts, CognitOS employs a set of CSS rules to be followed in the creation of each component. This includes, among others, a specific color palette, typography rules – containing guidelines for headings, body text, titles and subtitles – and specific UI elements, such as buttons and input fields. Figure 20 represents a detailed version of CognitOS's style guide. All major components of CognitOS were designed and developed utilizing the rules available through the aforementioned style guide. Producing a guide for designing components can facilitate the creation of new applications, which in the future could expand CognitOS's structure and overall functionality.

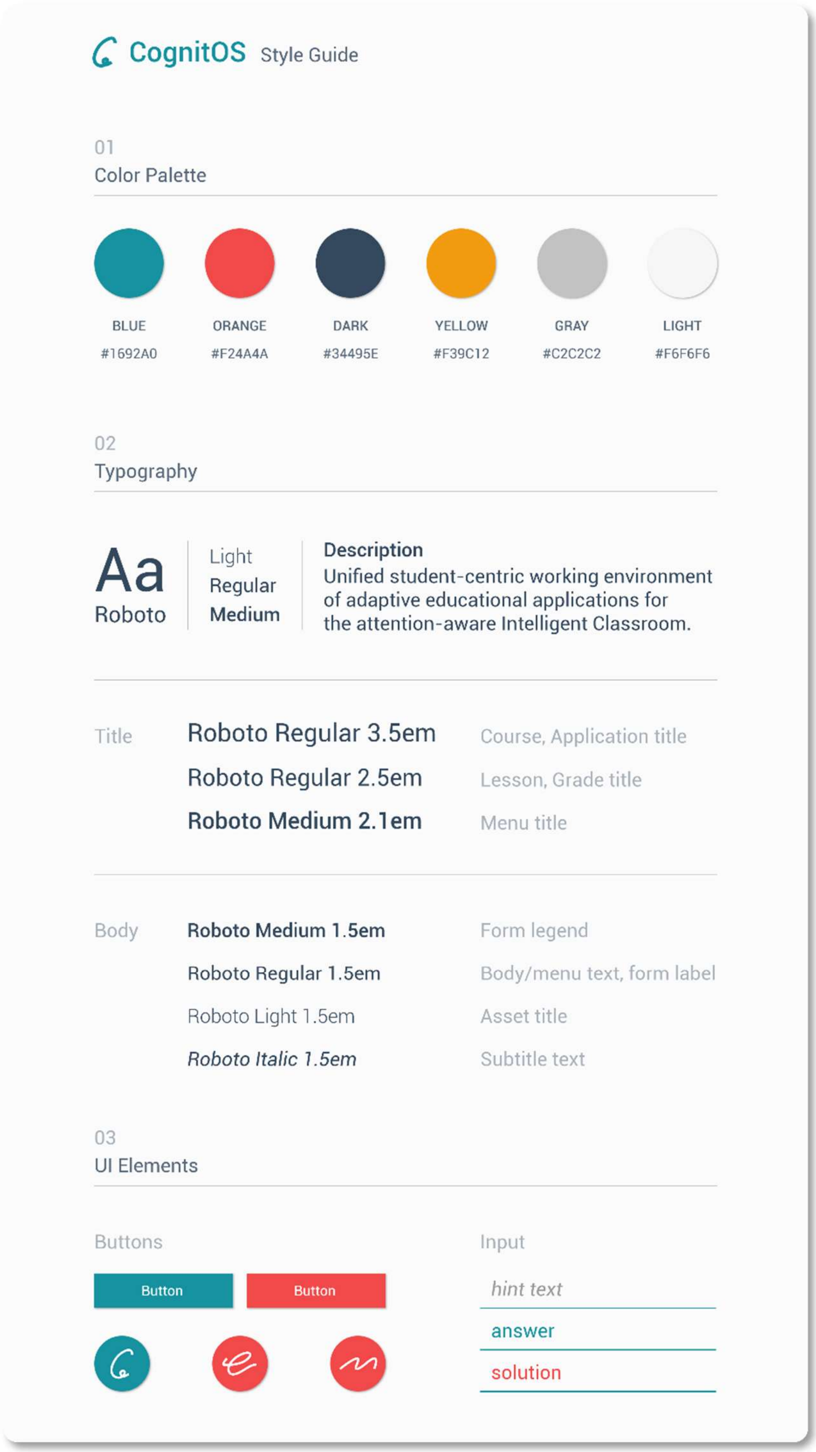


Figure 20. CognitOS's detailed style guide

5.2.1 Desktop

The Desktop (Figure 21) is the main working environment of students. It follows the metaphor of an actual student desk containing virtual items (e.g., books, notes) that can be used as launchers to respective applications. It features four (4) main applications namely ‘My Book’, ‘My Exercises’, ‘My Media’ and ‘My Profile’. In more detail, the dashboard of desktop contains: (i) a pile of books, offering a shortcut to ‘My Book’ application, (ii) a set of notebook pages, launching a set of assignments for each course, (iii) a computer monitor, hosting ‘My Media’ application and (iv) an ID card that displays the student’s name providing access to ‘My Profile’. With the exception of the profile, all applications offer a categorization of content per course and lesson. Each lesson contains specific content for the book, media and exercises applications (Figure 22).

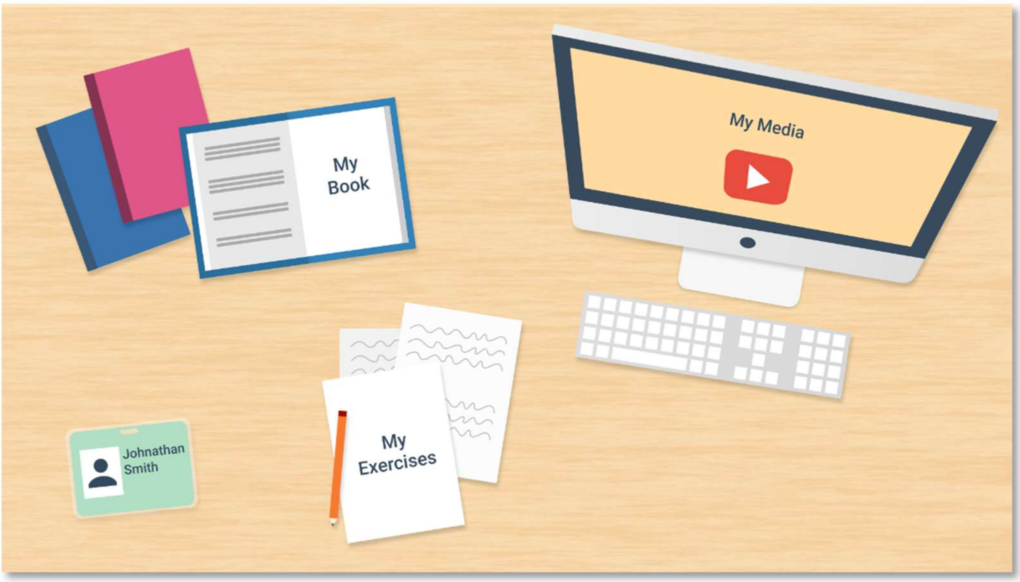


Figure 21. The Desktop environment of CognitOS

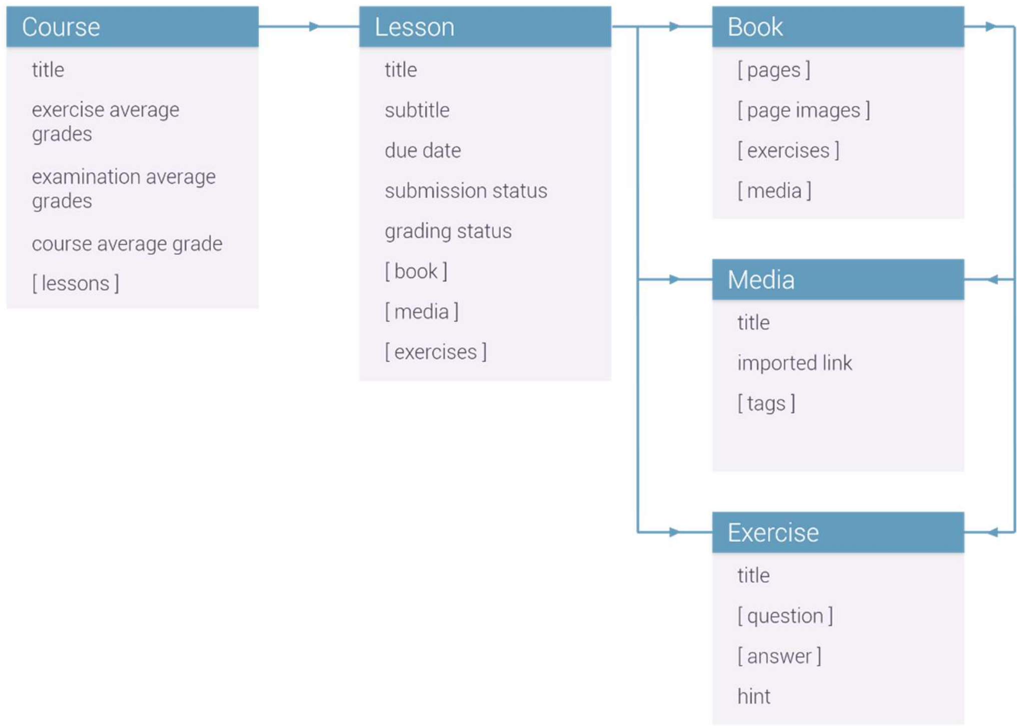


Figure 22. CognitOS data modeling

5.2.1.1 My Book

A collection of all the books of student's current school grade is available through the 'My Book' application (Figure 23), categorized by course and lesson. Upon launching the application, a menu offering shortcuts to all courses is available. Each course item contains a list of the available lessons, displaying the title, the number of pages covered in each one, as well as the date of teaching. Each book is the digital counterpart of the physical course book, where each physical page corresponds to an electronically augmented page enriched with additional material alongside the statically printed educational content. In more details, a button is provided on the side of each exercise, acting as launcher to the respective content, enabling students to solve their assignments directly on the book. Additionally, a color coded indicator is provided representing each exercise's level of difficulty. Three different colors are used for the levels, ranging from easy to hard. Light blue is used for 'Easy', yellow for 'Medium' and 'Red' for Hard, respectively (Figure 24).

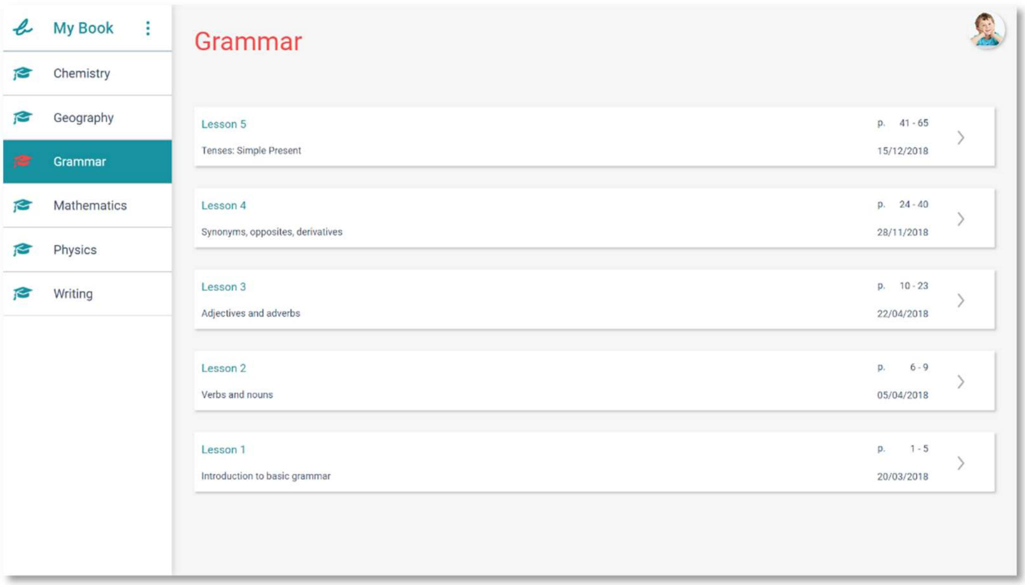


Figure 23. 'My Book' course categorization

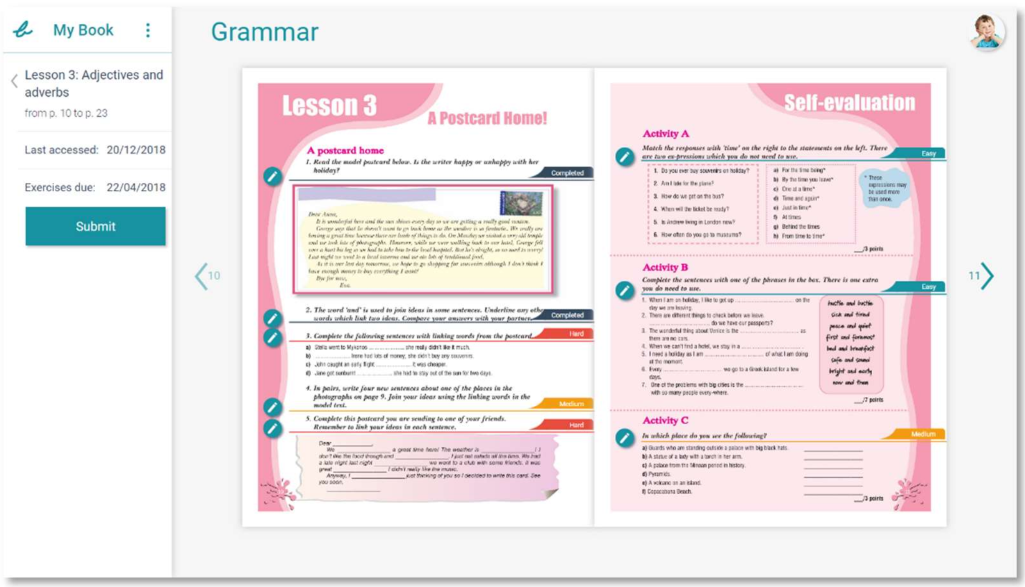


Figure 24. Snapshot of the open book

There are two alternative ways of solving an exercise. Each student can choose to display the exercise directly on the book page. In this scenario, when pressing the edit option, a modal-like pop up opens displaying the respective content (e.g., a multiple choice exercise) (Figure 25). Upon completing the exercise or closing the modal, students are able to continue browsing the book from where they left off. Finally, since all exercises that appear in the book are included – among others – in ‘My Exercises’, students can choose to launch the ‘My Exercises’ application to browse and solve their assignments, while benefitting from the additional features of the application.

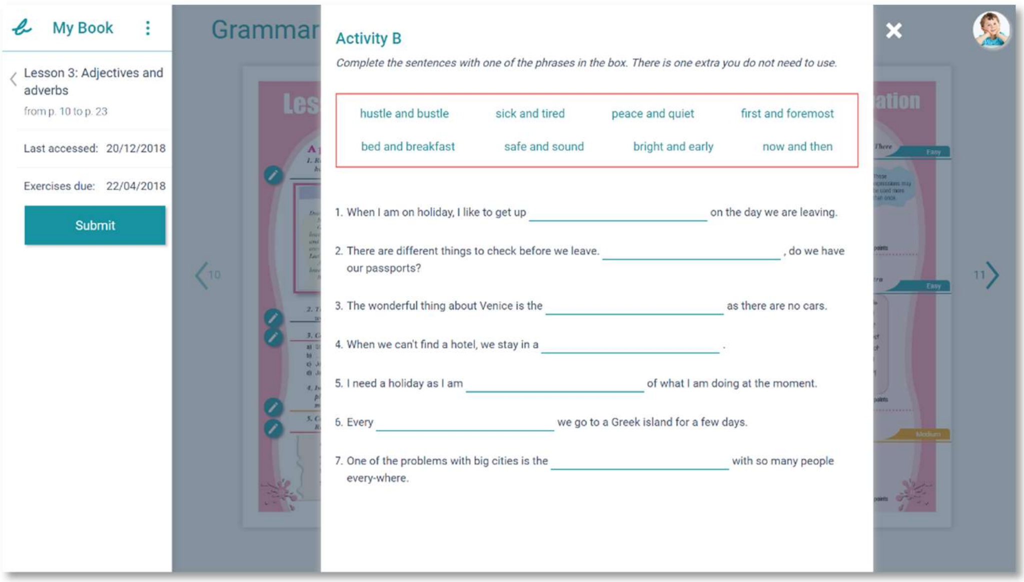


Figure 25. Solving an exercise directly in 'My Book'

5.2.1.2 My Exercises

Through the ‘My Exercises’ application, students have access to the full collection of their assignments for each course. A variety of exercise types is supported (e.g., multiple choice, fill in the blanks, matching, etc.). When launching the application during a lecture, exercises related to the current course are displayed first. While studying, a student can browse all courses and lessons for pending assignments, through the in-app menu. Since seven (7) or less has proven to be the appropriate range of items for accurately processing information for the human brain [99], [100], the last four (4) viewed assignments are displayed on the home screen of ‘My Exercises’ for quickly launching homework (Figure 26). By choosing a course from the menu, a list of all its lessons is presented, containing the due dates, as well as the status of each assignment (i.e., marked, submitted, pending and upcoming).

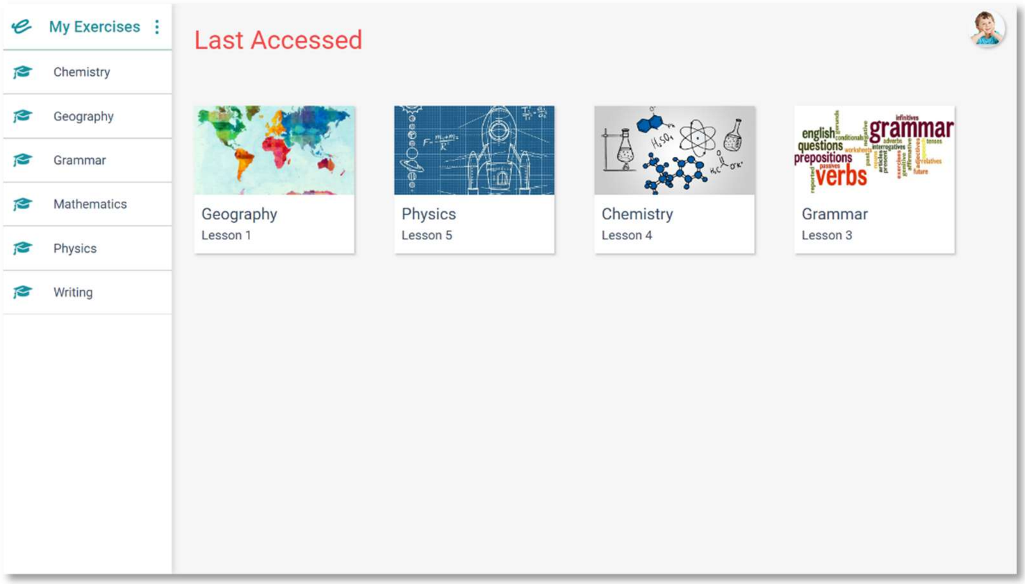


Figure 26. The 'My Exercises' home page with four last accessed items

Upon selecting a set of assignments – categorized per lesson – a full view of the exercises is available (Figure 27), accompanied by a side-menu and a submit button. On the side-menu a number of information (e.g., course and lesson title) is displayed, as well as the due date for the submission of the homework. Below, there is a list of the individual exercises of the assignment, acting as a menu for browsing each of them, alongside a completion indicator in the form of a small progress circle. In the center area, occupied by the exercise, students can engage in solving their assignments with various interaction methods, such as mouse clicking, touching, drag and drop, etc. In order to assist students during this process, CognitOS features a mechanism offering hints and solutions when explicitly asked. This complex mechanism supports three kinds of hints, gradually presented to the student, in order to assist the development of critical thinking skills. Each student can select to view a second or third hint, if the previous one was insufficient to solve the exercise. For example, in the case of solving a multiple choice exercise, the first hint would rephrase the question, the second hint provides definitions for each of the available answers, while the third eliminates half of them. If, in any case, the available hints fail to assist the student with solving the exercise, they can choose to reveal the solution. Hints and solutions used are taken into consideration in the formulation of the final grade of each assignment, so they are stored and submitted to the educator along with each exercise.

While an assignment is active, students can browse, alter and submit their exercises multiple times, until the deadline expires. The given answers are stored every time students engage with an exercise and are retrieved upon initiating again the application. Submitted answers are stored accordingly, as well as hints and solutions. On each assignment's due date, stored data are submitted to the educator for grading. Students can thereafter view educator feedback through the application or access it via their personal profiles. The 'My Exercises' application can also be accessed from students' personal tablets, where they can also solve and submit their homework to the educator.

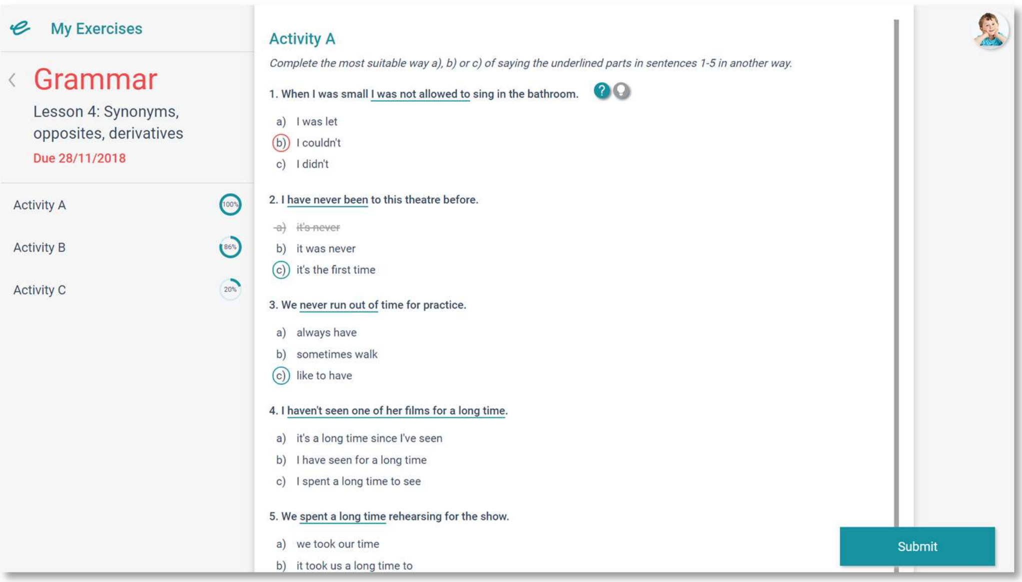


Figure 27. Full-view of an exercise with Hints and Solutions

Apart from full-view, the 'My Exercises' application is also available in mini-view. In this representation, only the main content area is displayed. Mini-view exercises mostly appear automatically on demand, following commands dictated by LECTOR or the educator. In the first case, when LECTOR identifies a troubling situation (e.g., a class-wide distraction of student attention), it can intervene by starting an interesting quiz. In the latter case, the educator –

through an external application – is able to order CognitOS to start a quick test at any point during the lecture time.

5.2.1.3 My Media

CognitOS features a multimedia application, ‘My Media’, which acts as a unified library containing a collection of complementary educational material for every course, such as videos, photos and sound files. This collection can be enriched in two ways. First and foremost, the educator has the ability to import and categorize appropriate and useful media for each course. In order to assist the educator in this task, an external service offering a content mining mechanism searches the web for educational multimedia and identifies the appropriate course topics they are associated with to carry out a categorization.

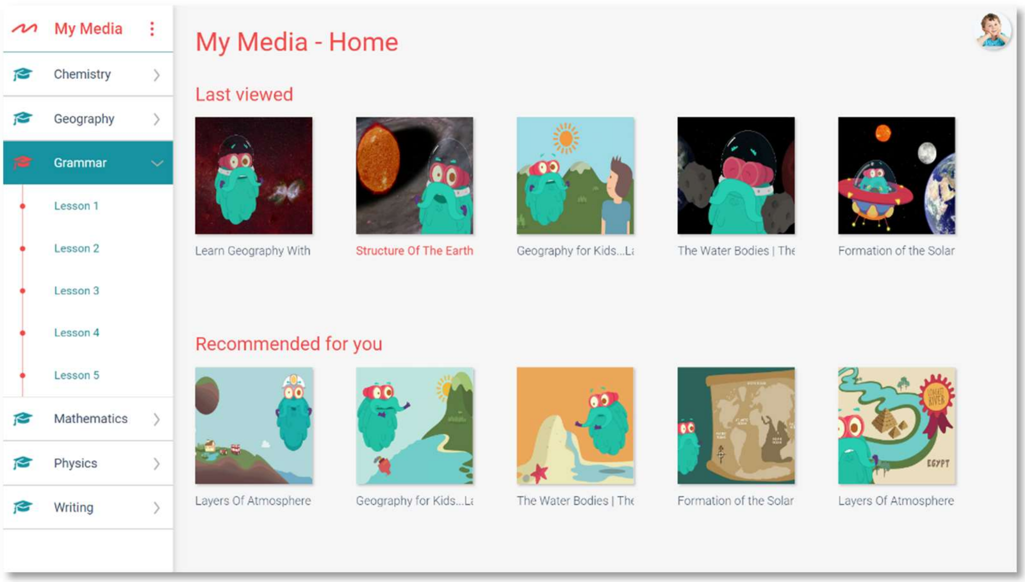


Figure 28. 'My Media' home page

On the home screen of the application, students have fast access to most recently viewed videos, that they were not able to finish during the previous lecture, as well as recommended material (Figure 28). The latter contains a set of multimedia selected via an external service utilizing an algorithm, which takes into consideration a number of factors, in order to provide the most accurate results. These factors contain (a) current lecture topic, (b) recently viewed material, (c) recently added material, (d) most popular material by student and (e) student difficulty on a specific subject. Students' difficulty is identified through a set of tags appended to every course and lesson, that is acquired throughout a student's academic course, based on strength and weaknesses on courses, lessons or specific topics. For example, during a particularly challenging topic for a specific student, the algorithm provides a set of the most popular tutorials in the current context, provided that the student has not viewed them before.

When not in a lecture or examination, students can browse through the multimedia contents of all courses they are enrolled to.

5.2.1.4 My Profile

In addition to the aforementioned applications, CognitOS also features a 'My Profile' application. In their personal profiles students can view – along with their personal information (e.g., name, age) – details regarding their academic performance. In more detail, 'My Profile' features a gradebook containing the current Overall Average Grade, as well as a list of individual grades (e.g., exercises and examinations average, course average) per course (Figure 29). Each course item in this list includes a color coded indicator (green, yellow, red), signifying the student's stronger and weaker courses. The overall grade is calculated by taking into consideration all courses' average grades, as well as the units of significance of each course.

Each course item offers a shortcut to more elaborate details on course assessment. There are graphic representations for the latest five (5) exercises and the latest three (3) examination grades, as well as a general weekly assessment of student's progress throughout the semester (Figure 30). The weekly assessment score is calculated according to weekly average grades on exercises and examinations, as well as student attendance and active participation in class. The 'My Profile' application can be accessed both via the desktop and the tablet either at home or in the classroom, provided that the student is not attending an examination at the time.

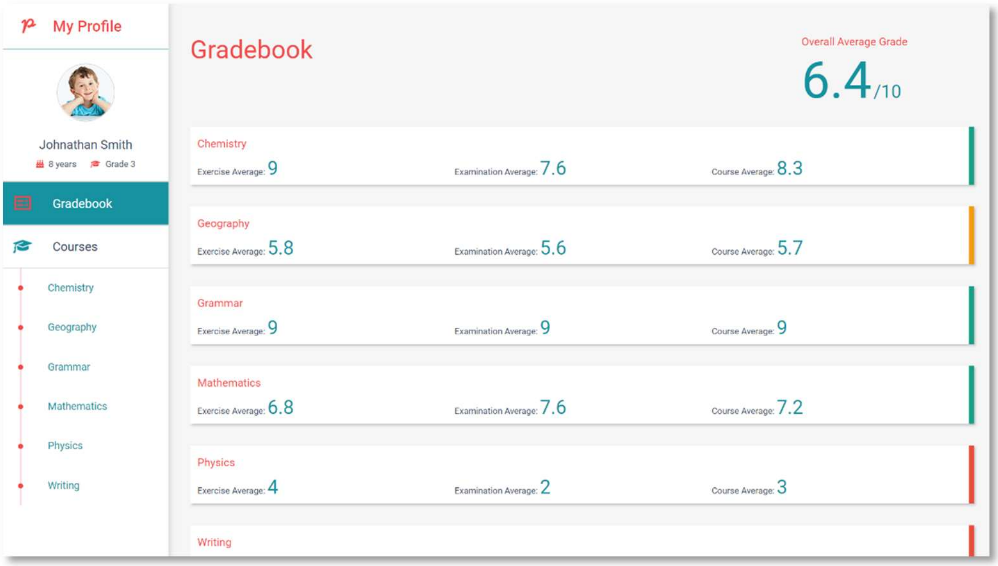


Figure 29. 'My Profile' Gradebook displaying average grades per course

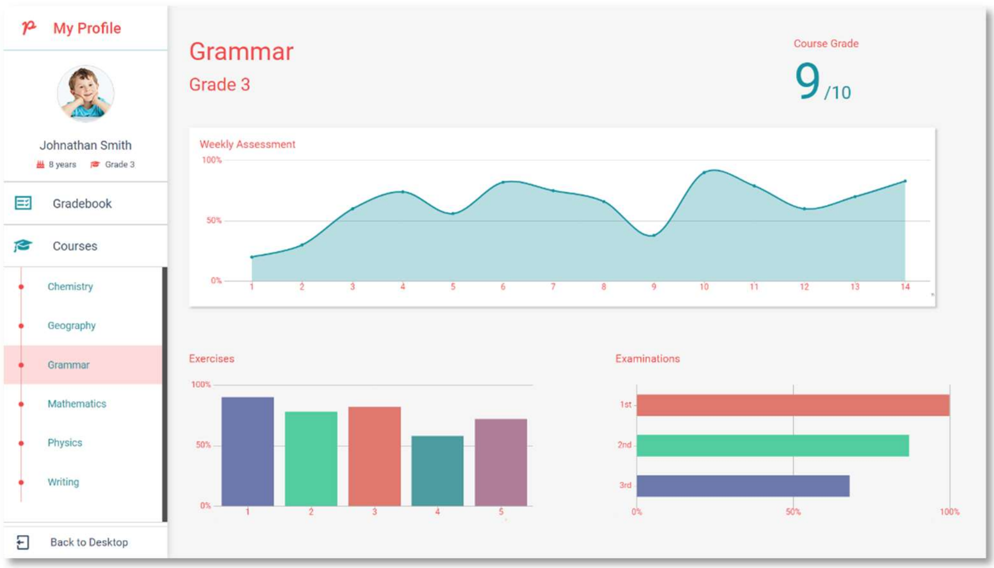


Figure 30. Grades per course in the 'My Profile' application

5.2.2 Tablet

The dashboard design for tablet follows the Springboard navigation pattern, also known as Launchpad. A Springboard consists of several screens containing options, mostly in the form of large icons, that act as launching points for applications. A paging indicator (e.g., little dots at the bottom of the screen), informs the users about the existence of other screens.

The tablet has been designed to complement the desktop environment, by offering a set of useful utilities for each course – defined each time a new course is added to CognitOS –, while docked beside the desktop, but also has the ability to work as a standalone device, when there is not a desktop PC present.

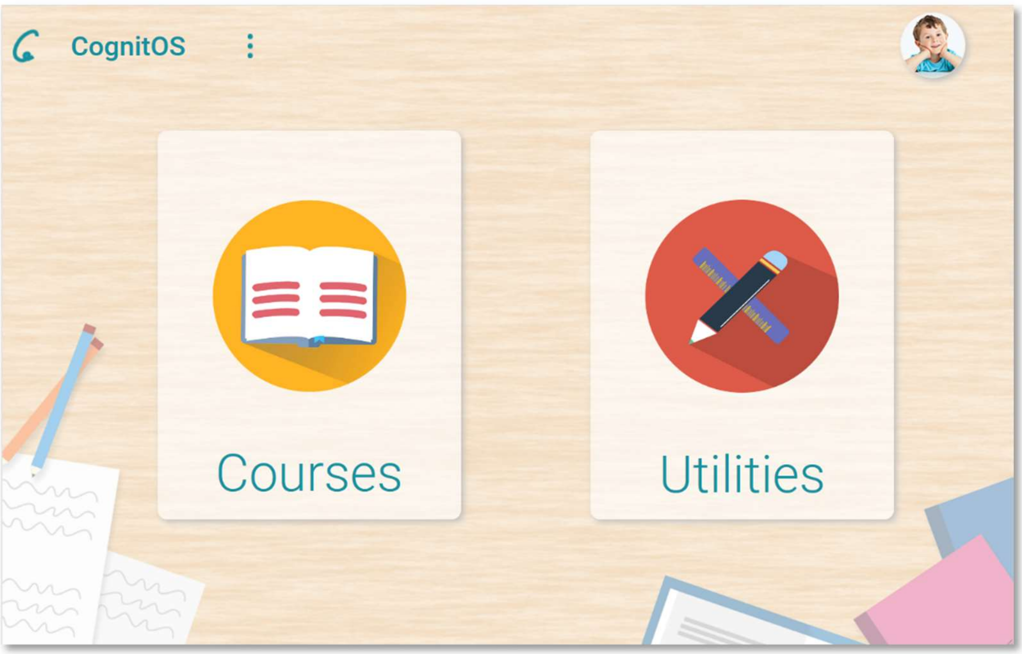


Figure 31. Tablet home page, with 'Courses' and 'Utilities' options

On the home screen, two options are available, 'Courses' and 'Utilities' (Figure 31). When the tablet is docked, the view is locked on 'Utilities', where only a number of them is available according to the current course. As soon as the tablet is undocked during a lecture period, the desktop automatically locks and the tablet becomes a standalone device. For example, during a Geography lesson, students may be solving an exercise on urban terrain. The educator at that point can choose to alternate her pedagogy by inviting the students to head outside, to the school's backyard, to view firsthand an example of an urban terrain to help with exercise solving. In that case, students interrupt their activity on desktop and undock their tablets to go outside. Through their tablets, students can proceed with solving their exercises from the point they left off. This is one informative example of how CognitOS can achieve continuity among the different artifacts, both inside and outside a classroom environment.

As a standalone device, the tablet takes the place of the desktop, hosting all courses' material. Both options ('Courses' and 'Utilities') are available. Students can browse through courses to solve exercises or view multimedia, in the same manner as they use the desktop PC during a lecture or examination period, or even when doing their homework.

5.2.2.1 Courses

Courses are only available when the tablet is not docked beside the desktop, acting as a complementary device. Whether during a studying period or when at home, courses offer a collection of all exercises and multimedia associated with each course, categorized as such. Each course offers a list of all lessons, same as the desktop version. A dropdown drawer for each lesson displays shortcuts to assignments and multimedia (Figure 32). Each assignment shortcut contains the title of the activity, an estimated time required to solve the exercise, as well as a level of completion for each of them. By clicking on a shortcut, students can navigate immediately on the respective activity, without the need to open all and browse until they find the right one.

Upon selecting a lesson, a full-view of associated assignments is displayed. On the top side, a header with information about course, lesson and due date is present, hosting also a submission button. Below, students can find the set of assignments displayed one at a time in a carousel style (Figure 33). Students can navigate between exercises by 'next' and 'previous' arrows placed on either side of each exercise or by navigation dots – one for each exercise – placed just below the header. The same interaction styles as in the desktop PC apply to the tablet version as well. Hints and solutions are also available, for each exercise respectively. Students can engage with their homework and submit it for the educator to grade. They can also, edit their answers multiple times until the submission date. After receiving grades, students can view their results through the tablet for each exercise.

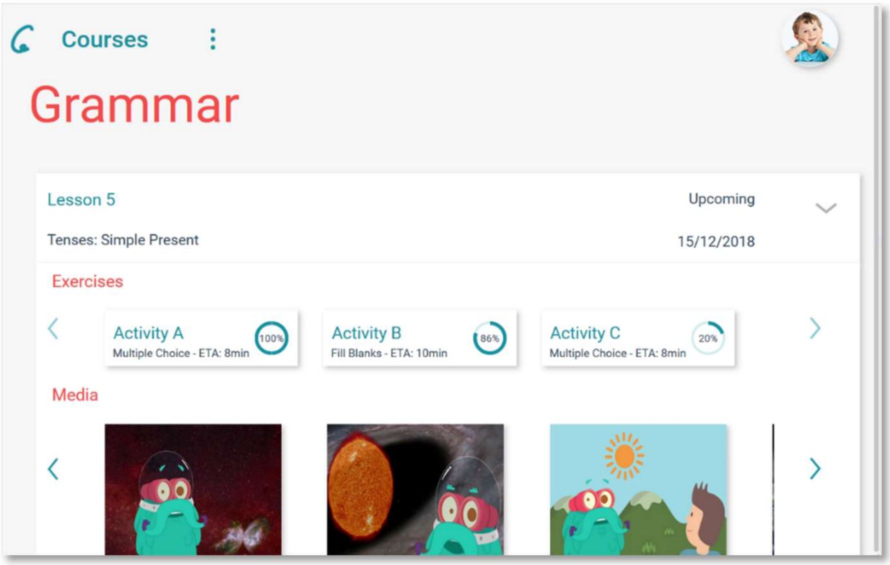


Figure 32. Lesson drawer with associated exercises and multimedia

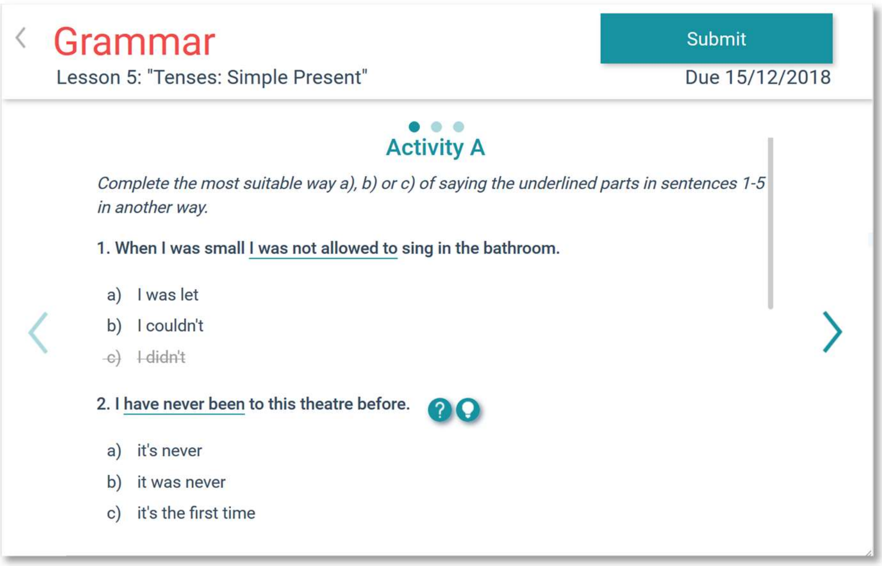


Figure 33. Solving an exercise on the tablet

5.2.2.2 Utilities

CognitOS offers a set of handy utilities through the tablet version (Figure 34). These have been designed in order to assist students with the various courses they have enrolled to. These utilities include, but are not limited to, a calculator, a calendar, an email client, a file explorer, maps, media, notes, a search engine, a thesaurus and a translator. Since the ‘Utilities’ application is course-sensitive, each mini application is available for one or more courses. One of the mostly used mini-apps is Media (Figure 35), since it is similar to the respective ‘My Media’ desktop application, containing multimedia for all courses. Media displays content corresponding to the active course each time the app is launched.

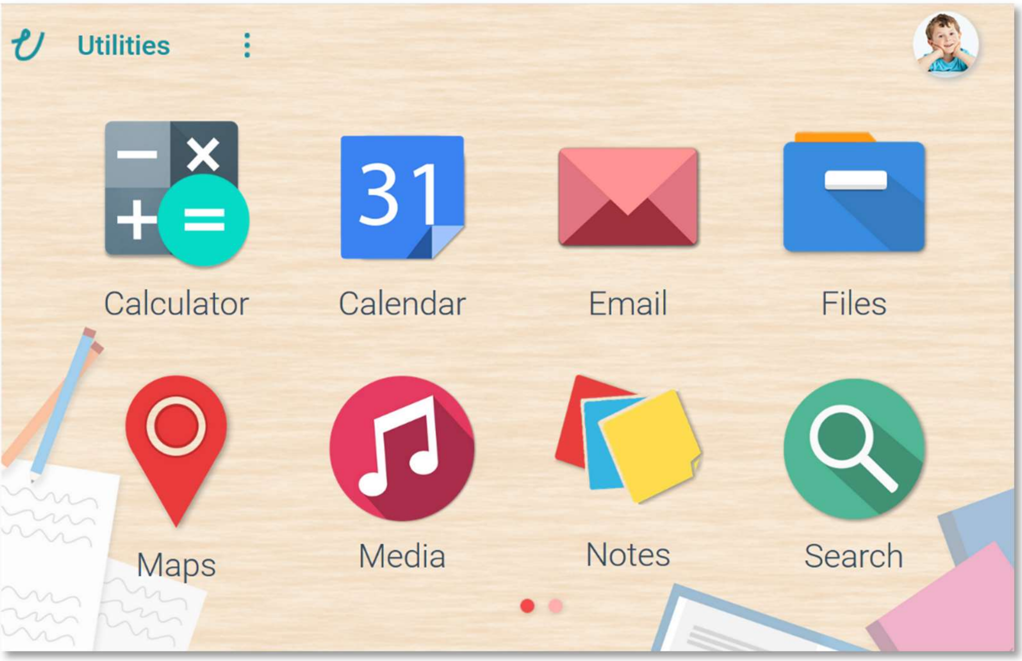


Figure 34. 'Utilities' of tablet

Utilities can be accessed during all the classroom times or at home, with a set of restrictions regarding the current course. When the tablet is docked besides the desktop during a lecture or an examination period, the UI is locked to the

'Utilities' view and does not allow students browsing through the other tablet applications. CognitOS takes into account the rules created by the educator regarding the current classroom activity – where the application behavior is defined as well –, in order to adjust which applications should be available on the tablet at any point. For example, during a lecture or examination, where the active course is Mathematics, all tablet utilities are disabled except from the calculator, the files and the videos. Were the active course Grammar, the calculator would be disabled and the Thesaurus enabled.

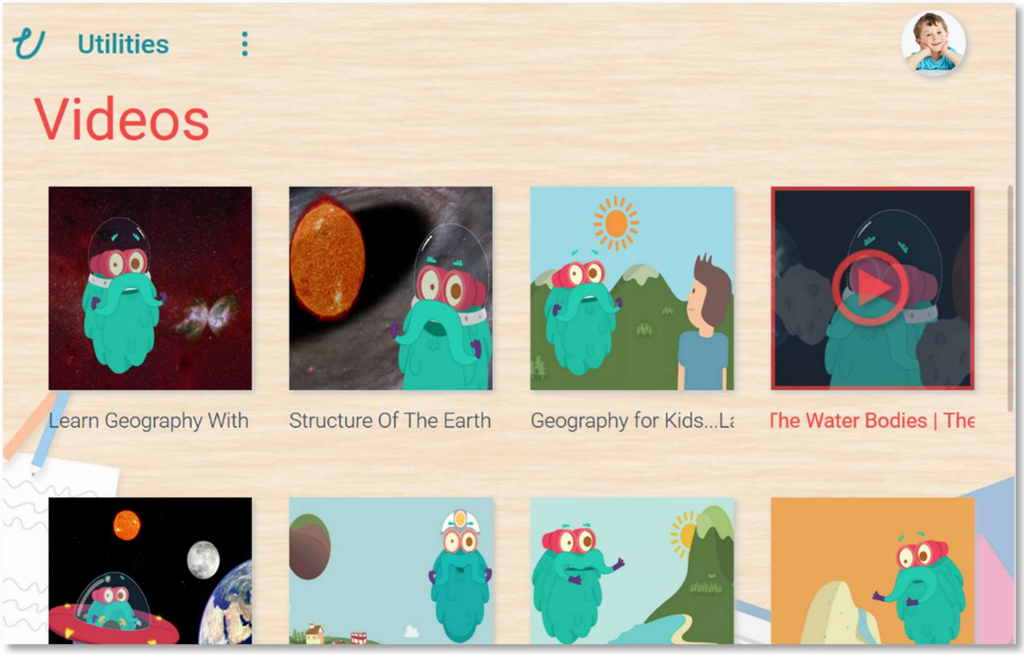


Figure 35. 'Media' mini-application of tablet's 'Utilities'

5.2.3 Board

The classroom's interactive smart white board is enhanced with course-sensitive content. Since the Intelligent Classroom has the ability to know beforehand the content of each lecture, either defined by the official school program or set explicitly by the educator, the board permanently displays the active course and lesson. In the same manner, the board also displays the current classroom activity – studying, lecture, examination – and adapts immediately to changes in activity based on the behavioral rules set by the educator.

On the bottom left, the board features the CognitOS menu, strategically placed in that position to facilitate interaction by both educator and smaller students. The main area of the board is occupied by a canvas application for educators to write notes in plain text for the current lecture or add annotations to any content (e.g., images, videos, presentations) displayed on the board. Educators can either use a touch pen to write or simply by using their fingers. These notes are collected by CognitOS and are afterwards saved into the “Notes” mini-app in the tablet Utilities, under the corresponding course and lesson.

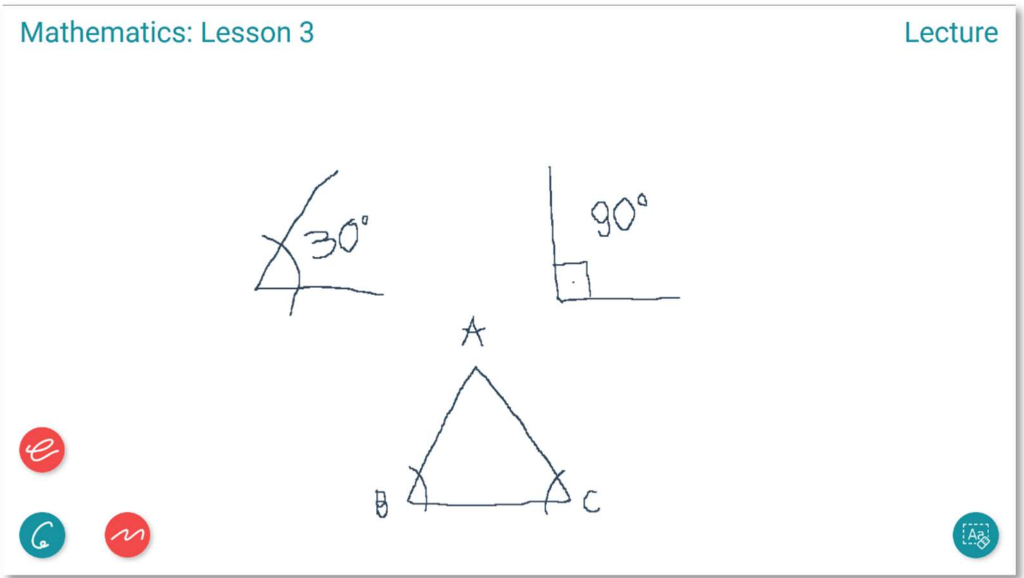


Figure 36. Smart classroom board with CognitOS menu, open to display the two options

Inside the menu, two applications can be found (Figure 36). The first one – Exercises – contains a number of course related exercises for the whole classroom to solve. In contrast with the desktop and tablet version of exercises, on the board questions are displayed one by one, with the answers right below, in order to be easily accessible by both educators and students. Educators can navigate between the different questions using ‘next’ and ‘previous’ labeled arrows on each side of a question (Figure 37).

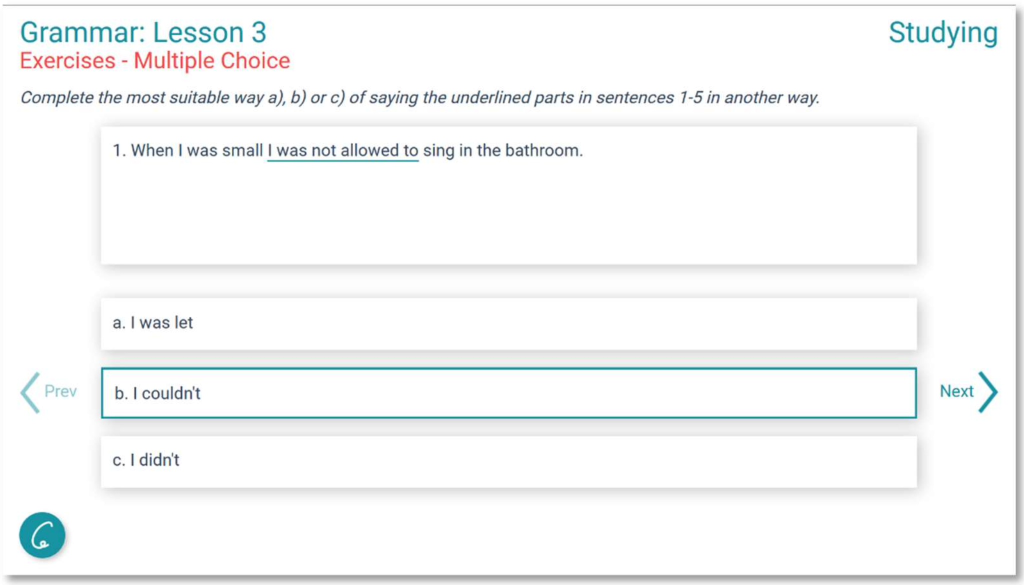


Figure 37. An exercise displayed on board.

The second application, namely Media, displays all course related multimedia. This content is presented in a carousel-like tray on the bottom of the board (Figure 38), for ease of access, even by younger students. The main area of the board is used to present selected material, e.g. play a video, with video player controls on the bottom side (Figure 39).

Apart from course sensitive, the board – along with all the other artifacts – is also classroom activity related. During an examination, the CognitOS menu is locked, disabling navigation, and the main area of the board transforms in a countdown timer. In addition, since, both during an examination and when they are solving exercises during class, students are required to submit their answers to the educator, the board displays an indication of who has submitted at any point.

Figure 38. Media application containing a multimedia carousel-like drawer.

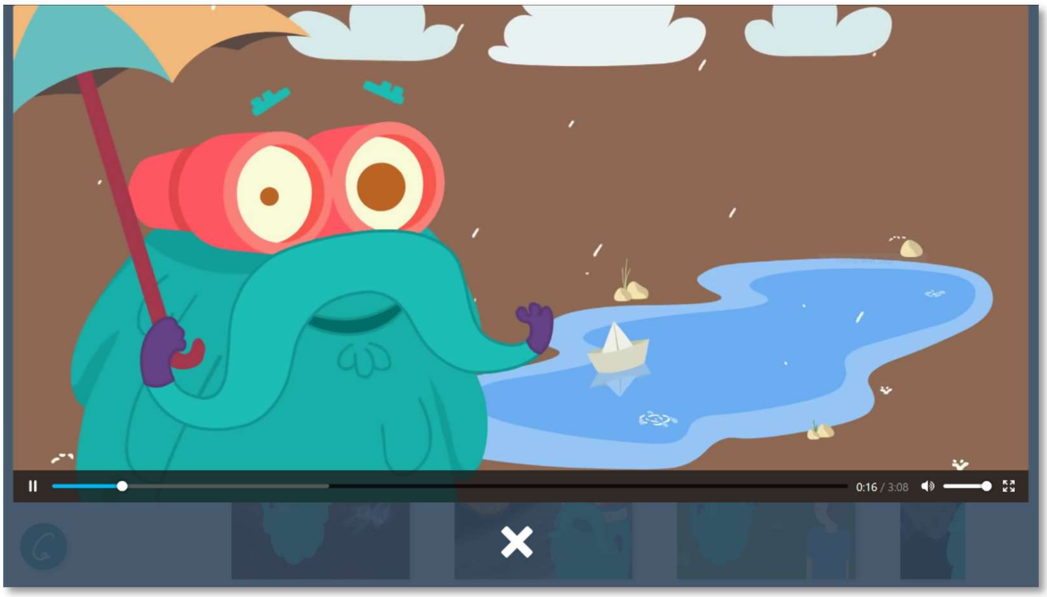


Figure 39. Video player with controls, presented on board

5.3 CognitOS as an Intervention Host

Apart from being a basic working environment acting as an application host, CognitOS supports the mission of the Attention-aware Intelligent Classroom by being the medium through which LECTOR delivers interventions. In a broader sense, an intervention is defined as *"involvement in a difficult situation in order to improve it or prevent it from getting worse"* [101]. In this context the term intervention is used to describe the actions, guided by the system, that aim to subtly interrupt the flow of a lecture in order to achieve two main goals. The first step is to inform the educator about a problematic situation, drawing her attention to the source of the problem, when an alarm is triggered. The second step targets the students, specifically the ones that triggered an alarm, and try to re-engage them somehow in the educational process.

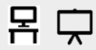
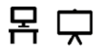
From the aforementioned statement it arises that there are two intervention recipients – the educator and the students. Depending on the recipient, the host of the intervention is defined. For the educator, the smartphone acts as the host, whereas students can be addressed either via their individual desks or the public classroom smart board. Individual student desks are used as an intervention medium when one or very few students need to be addressed. When the entire classroom requires a remedial action, the public board can be used.

Furthermore, CognitOS is equipped with a notification mechanism able to provide motivational messages. Taking into account that students tend to thrive in encouraging environments, their private artifacts are able to display specific feedback when deemed appropriate through said mechanism. In the same manner, when an intervention is targeting the educator, the same mechanism employed on their smartphone can suggest subtle changes in the lecture format. In more detail, if a lot of students are identified to be tired, LECTOR may suggest recapitulation of the lecture topics, initiation of a discussion relevant to the current course, repetition of specific material and continue lecturing at a slower pace.

5.3.1 Intervention Techniques

There are several reasons an alarm might be triggered and for each of them, a different technique can be used to intervene. The literature review reports several intervention techniques that can prove to be beneficial in tackling the various situations occurring in an educational setting (as reported in Chapter 2). When students feel tired, Active Learning has proven to be most effective, whereas Changing Pedagogies can produce best results when students are distracted from the task at hand. CognitOS, with the help of LECTOR, can apply three (3) different techniques, namely: (i) Active Learning, (ii) Changing Pedagogies and (iii) Encouragement. Table 2 summarizes the aforementioned techniques.

Table 2. List of all available educational interventions

Intervention	**Technique**	**Recipient**	**Hosts**	**Range**
Quiz	Active Learning	Students		Individual, Class-wide
Multimedia Presentation	Active Learning	Students		Individual, Class-wide
Motivational Messages	Encouragement	Students		Individual
Initiate Discussion	Changing Pedagogies	Teacher		Class-wide
Recapitulation	Changing Pedagogies	Teacher		Class-wide
Repeat Course Material	Changing Pedagogies	Teacher		Class-wide
Lower Pace	Changing Pedagogies	Teacher		Class-wide

: Student Desk : Classroom Board  : Smartphone

Active Learning and Encouragement techniques utilize the classroom artifacts (i.e., student desks and classroom board) to present appropriate content dictated by LECTOR through CognitOS's applications. On the contrary, changing pedagogies target the educator through her smartphone. When this type of technique is chosen by LECTOR, the educator receives, through

CognitOS, a notification on her smartphone suggesting the appropriate change in the lecturing format.

5.3.2 Intervention Types

As soon as LECTOR decides that a specific technique must be employed, it sends a command to CognitOS via AmI-Solertis, which includes details regarding the type and content of the intervention to initiate. The types of interventions that CognitOS can deliver are divided into four (4) categories as described below:

Notices. CognitOS is equipped with a notification mechanism which is able to present toast notifications on students' private artifacts (i.e., tablet, smartphone, smart desk). The term toast notification is used to describe small pop up message boxes that appear at a certain location of the screen and disappear automatically after a few seconds. This intervention type is utilized to deliver motivational messages (technique: Encouragement) to the students who seem unmotivated, troubled or disengaged from the task at hand. Figure 40 gives an example of a motivational message delivered through CognitOS.

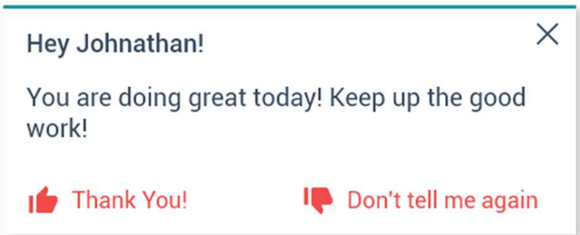


Figure 40. A toast notification displaying a motivational message

Augmentations & Alterations. Each of the educational applications CognitOS hosts – described in the previous section, can be launched on demand with specific content optimized for each dictated intervention. All of the above available applications have been designed featuring two views on the students' desks, namely mini- and full-view. Augmentations are represented as mini-view applications launched alongside with other material (Figure 41), in order to present auxiliary content, whereas alterations refer to full-view applications that aim to monopolize and redirect students' attention.

Restrictions. CognitOS features an API to create rules to deny access to application content, when either the educator or LECTOR deems it irrelevant to the current course or activity. For example, as depicted in Figure 42, during an examination period, CognitOS disables students from browsing applications other than the task at hand (i.e., the current exercises of the examination).

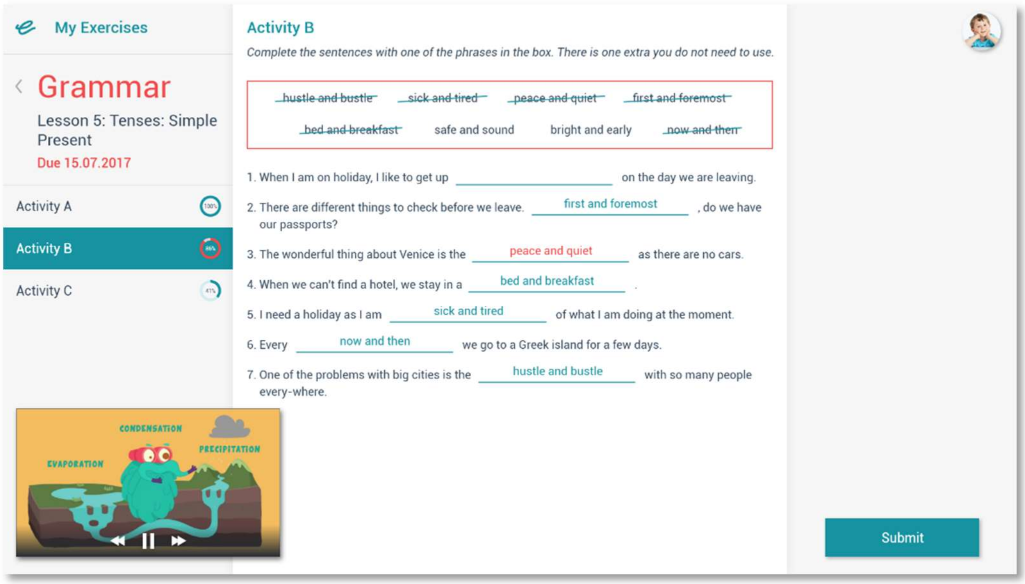


Figure 41. An example of augmentation. A tutorial is displayed while solving an exercise

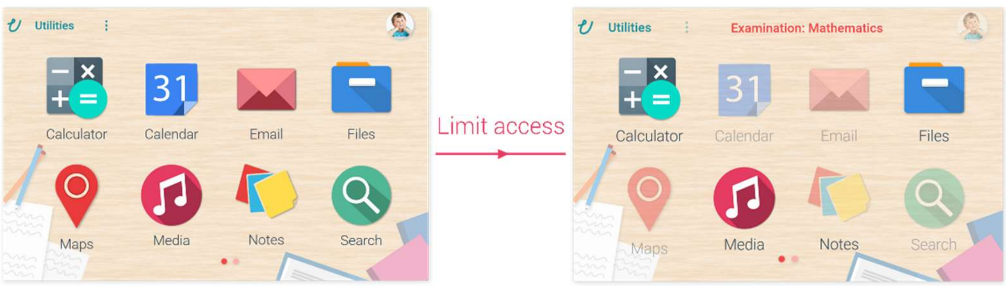


Figure 42. An example of a restriction on tablet

5.4 CognitOS as a Service

Software design reusability has been a major concern in the programming community, when designing program components and developing large-scale applications. A set of widely used techniques for this approach exist among the object-oriented community, while only a few of them are documented [102]. One of the most important kinds of reuse is reuse of designs, where a collection of abstract classes can be used to express an abstract design. The design of a program is usually described in terms of the program's components and the way they interact.

A framework is a reusable, “semi-complete” application that can be specialized to produce custom applications [103], with code that is resistant to more conventional reuse attempts. When components are designed independent from specific applications, code reusing is rather easy. Frameworks are not to be confused with skeleton programs – the conventional way of code reusing – since, unlike them, frameworks can ensure consistency between components under changing requirements [102]. Frameworks have the ability to provide stable interfaces that can enhance reusability through generic components that can be reapplied to create new applications. This kind of reusability stems from the domain knowledge and effort of developers to avoid recreating common solutions for the same type of application requirements and design challenges.

Understanding and building an application using a framework should take less effort and time than building an equivalent program without a framework [104]. Reuse of framework components can yield substantial improvements in programmer productivity, as well as enhancing the quality, performance, reliability and interoperability of software. When building an application with rapidly evolving requirements, the need of flexibility in the pattern of interactions between components of a framework is high [105]. Exploiting framework extensibility is essential in order to ensure the timely development of new application services and features.

Towards this approach, CognitOS major components have been designed and developed for reuse (e.g., each exercise type). Each component is constructed with a basic backbone customizable structure, which, given different types of data, can be used in various cases, even outside an Intelligent Classroom environment. In addition, CognitOS supports application configuration. The system offers an API containing a number of functions (Table 3), in order to customize the behavior of all different applications. Through these API functions, any application can launch and terminate on demand, on one or more artifacts, containing specific data and with a certain status. An external source can exploit the given API and control on demand the general behavior of CognitOS. The API itself can be enhanced with more functions, since the system is ever-growing.

Table 3. CognitOS API functions

Server	Category	Functions
UI Server functions	Actives	GetOpenInstances
		GetRunningApps
	Time	GetAppRunningTime
		GetInstanceIdleTime
		GetAppIdleTime
	Statistics	GetExerciseStatus
		GetLastAccessedLessons
Core Server functions	Status	GetAppStatus
		GetClassroomStatus
		SetAppStatus
		SetClassroomStatus
	Behavior	SetAppContent
		SetBehaviorRule
	Actions	LaunchApp
		TerminateApp
		InitiateActivity
		TerminateActivity

5.4.1 API Functions

5.4.1.1 UI Server Functions

5.4.1.1.1 GetOpenInstances

URL

`POST https://cognitos.ics.forth.gr/api/v1/GetOpenInstances`

Returns the names of all open instances of CognitOS

Information

Response format	**JSON**

Parameters

No parameters.

Example Response

```
{
      “getOpenInstances”:
      {
            “instances”:
            [
                  “Desk1”,
                  “Desk2”,
                  “Tablet2”,
                  “Board”
            ]
      }
}
```

5.4.1.1.2 GetRunningApps

URL

`POST https://cognitos.ics.forth.gr/api/v1/GetRunningApps`

Returns the names of all running applications on a specific artifact

Information

Request format	JSON
Response format	JSON

Parameters

Name	Required	Description	Example
host_id	required	Contains the requested applications' artifact name	"Desk1"

Example Request

```
{
	"getRunningApps":
	{
		"host_id": "Desk1"
	}
}
```

Example Response

```
{
	"getRunningApps":
	{
		"host_id": "Desk1",
		"apps":
		[
			"My Book",
			"My Media"
		]
	}
}
```

5.4.1.1.3 GetAppRunningTime

URL

```
POST https://cognitos.ics.forth.gr/api/v1/GetAppRunningTime
```

Returns the time a specific application has been running on a specific artifact

Information

Request format	JSON
Response format	JSON

Parameters

Name	Required	Description	Example
app	required	Contains the name of the application whose running time is requested	`“My Book”`
host_id	required	Contains the artifact name the requested application is running on	`“Desk1”`

Example Request

```
{
      “getAppRunningTime”:
      {
            “app”: “My Book”,
            “host_id”: “Desk1”
      }
}
```

Example Response

```
{
      “getAppRunningTime”:
      {
            “app”: “My Book”,
            “host_id”: “Desk1”,
            “time”: “00:28:43”
      }
}
```

5.4.1.1.4 GetInstanceIdleTime

URL

```
POST https://cognitos.ics.forth.gr/api/v1/GetInstanceIdleTime
```

Returns the time a specific instance has been idle for

Information

Request format	JSON
Response format	JSON

Parameters

Name	Required	Description	Example
host_id	required	Contains the artifact name of a requested idle instance	“Desk1”

Example Request

```
{
	“getInstanceIdleTime”:
	{
		“host_id”: “Desk1”
	}
}
```

Example Response

```
{
	“getInstanceIdleTime”:
	{
		“host_id”: “Desk1”,
		“time”: “00:28:43”
	}
}
```

5.4.1.1.5 GetAppIdleTime

URL

`POST https://cognitos.ics.forth.gr/api/v1/GetAppIdleTime`

Returns the time a specific application has been idle for

Information

Request format	JSON
Response format	JSON

Parameters

Name	Required	Description	Example
app	required	Contains the name of the application whose idle time is requested	"My Book"
host_id	required	Contains the artifact name the requested application is idle on	"Desk1"

Example Request

```
{
      "getAppIdleTime":
      {
            "app": "My Book",
            "host_id": "Desk1"
      }
}
```

Example Response

```
{
      "getAppIdleTime":
      {
            "app": "My Book",
```

```
            "host_id": "Desk1",
            "time": "00:28:43"
      }
}
```

5.4.1.1.6 GetExerciseStatus

URL

```
POST https://cognitos.ics.forth.gr/api/v1/GetExerciseStatus
```

Returns a specific exercise status (pending, ongoing, completed, submitted, marked)

Information

Request format	JSON
Response format	JSON

Parameters

Name	Required	Description	Example
course	required	Contains the name of the requested course	"Grammar"
lesson	required	Contains the name of the requested lesson	"Lesson 3"
host_id	required	Contains the artifact name the requested exercise is on	"Desk1"

Example Request

```
{
      "getExerciseStatus":
      {
            "course": "Grammar",
            "lesson": "Lesson 3",
            "host_id": "Desk1"
```

```
    }
}
```

Example Response


```
{
    “getExerciseStatus”:
    {
        “course”: “Grammar”,
        “lesson”: “Lesson 3”,
        “host_id”: “Desk1”,
        “status”: “Pending”
    }
}
```


5.4.1.1.7 GetLastAccessedLessons

URL

```
POST
https://cognitos.ics.forth.gr/api/v1/GetLastAccessedLessons
```

Returns the last accessed lessons and their respective course for a specific application

Information

Request format	JSON
Response format	JSON

Parameters

Name	Required	Description	Example
app	required	Contains the name of the application whose last accessed lessons are requested	`“My Book”`

host_id	required	Contains the artifact name the requested application is on	“Desk1”

Example Request

```
{
	“getLastAccessedLessons”:
	{
		“app”: “My Book”,
		“host_id”: “Desk1”
	}
}
```

Example Response

```
{
	“getLastAccessedLessons”:
	{
		“app”: “My Book”,
		“host_id”: “Desk1”,
		“lessons”:
		[
			{
				“course”: “Grammar”,
				“lesson”: “3”
			},
			{
				“course”: “Physics”,
				“lesson”: “2”
			}
		]
	}
}
```

5.4.1.2 Core Server Functions

5.4.1.2.1 GetAppStatus

URL

```
POST https://cognitos.ics.forth.gr/api/v1/GetAppStatus
```

Returns a specific application status (locked or unlocked)

Information

Request format	JSON
Response format	JSON

Parameters

Name	Required	Description	Example
app	required	Contains the name of the application whose status is requested	“My Book”
host_id	required	Contains the artifact name the requested application is on	“Desk1”

Example Request

```
{
	“getAppStatus”:
	{
		“app”: “My Book”,
		“host_id”: “Desk1”
	}
}
```

Example Response

```
{
	“getAppStatus”:
	{
```

```
            “app”: “My Book”,
            “host_id”: “Desk1”,
            “status”: “Locked”
      }
}
```

5.4.1.2.2 GetClassroomStatus

URL

`POST https://cognitos.ics.forth.gr/api/v1/GetClassroomStatus`

Returns classroom status (examination, lecture, studying)

Information

Response format	**JSON**

Parameters

No parameters.

Example Response

```
{
      “getClassroomStatus”:
      {
            “status”: “Examination”
      }
}
```

5.4.1.2.3 SetAppStatus

URL

`POST https://cognitos.ics.forth.gr/api/v1/SetAppStatus`

Sets the status of a specific application on a specific artifact (locked or unlocked)

Information

Request format	JSON

Parameters

Name	Required	Description	Example
app	required	Contains the name of the application whose status to be set	`"My Book"`
host_id	required	Contains the artifact name the requested application is on	`"Desk1"`
status	required	Contains the status to be set	`"Locked"`

Example Request

```
{
      "setAppStatus":
      {
            "app": "My Book",
            "host_id": "Desk1",
            "status": "Locked"
      }
}
```

5.4.1.2.4 SetClassroomStatus

URL

`POST https://cognitos.ics.forth.gr/api/v1/SetClassroomStatus`

Sets the general classroom status (examination, lecture, studying)

Information

Request format	JSON

Parameters

Name	Required	Description	Example
status	required	Contains the status to be set	“Examination”

Example Request

```
{
      “setClassroomStatus”:
      {
            “status”: “Examination”
      }
}
```

5.4.1.2.5 SetAppContent

URL

`POST https://cognitos.ics.forth.gr/api/v1/SetAppContent`

Sets the content of a specific application

Information

Request format	JSON

Parameters

Name	Required	Description	Example
app	required	Contains the name of the application whose content to be set	“My Book”
host_id	optional	Contains the artifact name the requested application is on If all artifacts should display the same application content, host_id is omitted	“Desk1”

content	required	Contains the content to be set	`{ data }`

Example Request

```
{
	“setAppContent”:
	{
		“app”: “My Book”,
		“content”:
		[
			{
				“pages”: [pages],
				“pageImages”: [page images],
				“exercises”: [exercises],
				“media”: [media]
			}
		]
	}
}
```

5.4.1.2.6 SetBehaviorRule

URL

`POST https://cognitos.ics.forth.gr/api/v1/SetBehaviorRule`

Sets a new behavior rule, or alters an already existing one, with specific parameters regarding the behavior of each application on each artifact according to a classroom activity, as well as specify the appropriate content when launching.

Information

Request format	JSON

Parameters

Name	Required	Description	Example
name	required	Contains the name of the rule to be created or edited	`“Examination”`
course	optional	Contains the name of the course the rule refers to If omitted, the rule applied to all courses	`“Grammar”`
parameters	required	Contains the parameters of the rule It contains the status of specific applications on specific artifacts to be set It contains the appropriate content to launch an application	`{ data }`

Example Request

```
{
      “setBehaviorRule”:
      {
            “name”: “Examination”,
            “course”: “Grammar”,
            “parameters”:
            [
                  {
                       “host_id”: “Desk1”,
                       “appStatus”:
                       [
                             {
                                   “app”: “My Book”,
                                   “status”: “Locked”
                             },
                             {
```

```
                        “app”: “My Media”,
                        “status”: “Locked”
                    },
                    {
                        “app”: “My Exercises”,
                        “status”: “Unlocked”,
                        “content”:
                        [
                            {
                                “question”: “”,
                                “answers”:
                                [
                                    “...”,
                                    “...”,
                                    “...”
                                ],
                                “hint”: “”,
                                “solution”: “”
                            }
                        ]
                    }
                ]
            },
            {
                “host_id”: “Board”,
                “appStatus”:
                [
                    {
                        “app”: “My Media”,
                        “status”: “Locked”
                    }
                ]
            }
        ]
    }
}
```

5.4.1.2.7 LaunchApp

URL

`POST https://cognitos.ics.forth.gr/api/v1/LaunchApp`

Launches a specific application on one or more artifacts, with specific content

Information

Request format	JSON

Parameters

Name	Required	Description	Example
app	required	Contains the app to be launched	“My Book”
host_id	required	Contains the host(s) the app should be launched at	“Desk1”
content	required	Defines the appropriate content to launch the application Could vary among artifacts	{ *data* }

Example Request

```
{
      “launchApp”:
      {
            “app”: “My Book”,
            “hosts”:
            [
                  {
                        “host_id”: “Desk1”,
                        “content”:
                        {
                              “pages”: [pages],
                              “pageImages”: [page images],
                              “exercises”: [exercises],
                              “media”: [media]
                        }
```

```
                        },
                        {
                                "host_id": "Desk2",
                                "content":
                                {
                                        "pages": [pages],
                                        "pageImages": [page images],
                                        "exercises": [exercises],
                                        "media": [media]
                                }
                        }
                ]
        }
}
```

5.4.1.2.8 TerminateApp

URL

`POST https://cognitos.ics.forth.gr/api/v1/TerminateApp`

Terminates a specific application on one or more artifacts

Information

Request format	JSON

Parameters

Name	Required	Description	Example
app	required	Contains the app to be terminated	"My Book"
host_id	required	Contains the host(s) the app should be terminated at	"Desk1"

Example Request

```
{
        "terminateApp":
```

```
        {
                "app": "My Book",
                "host_id":
                [
                        "Desk1",
                        "Desk2"
                ]
        }
}
```

5.4.1.2.9 InitiateActivity

URL

```
POST https://cognitos.ics.forth.gr/api/v1/InitiateActivity
```

Initiates a classroom activity with predefined behavior

Information

Request format	JSON

Parameters

Name	Required	Description	Example
name	required	Contains the name of the activity to be initiated	"Examination"

Example Request

```
{
        "initiateActivity":
        {
                "name": "Examination"
        }
}
```

5.4.1.2.10 TerminateActivity

URL

```
POST https://cognitos.ics.forth.gr/api/v1/TerminateActivity
```

Terminates a classroom activity. The classroom immediately switches to ‘Studying’

Information

Request format	JSON

Parameters

Name	Required	Description	Example
name	required	Contains the name of the activity to be terminated	“Examination”

Example Request

```
{
      “terminateActivity”:
      {
            “name”: “Examination”
      }
}
```

In addition to the API’s functions, a number of events are also fired during a number of circumstances, with the most common being launching and terminating applications. These events can be exploited by CognitOS Core to further adjust the system’s overall behavior. Utilizing the CognitOS API, a developer, or even an educator, should be able to easily customize the system’s overall functionality, as well as its behavior from any external application.

Chapter 6

Extending CognitOS

6.1 Adding new applications to CognitOS

CognitOS is not a simple system, on the contrary, it is a highly expandable and editable system, hence its framework side. When adding a new application or a new application component, CognitOS receives a bundle containing the UI complete with the application logic, which includes rules in the form of conditional statements that define the application behavior and functionality based on the CognitOS's cases of use. The UI is added to the UI Server, while the application logic is managed by the Core Server. In order to launch – or use in general – the new application, the Core Server communicates with the AmI-Solertis to get environmental metadata, which in collaboration with the behavioral rules received upon addition of the application, define whether the application should launch, and if yes, display appropriate content.

6.2 Cross-device Interaction

In order to achieve a seamless intercommunication of devices, a number of handling services have been developed that can control data transferring and transitions between devices (Figure 43). These services handle data sharing and are also responsible for content manipulation, in order to fill the front end components with respective content, according to classroom parameters (e.g., current course and lesson). The handler service is responsible – among others – for informing all different modules about current classroom parameters, as well as identifying their changes.

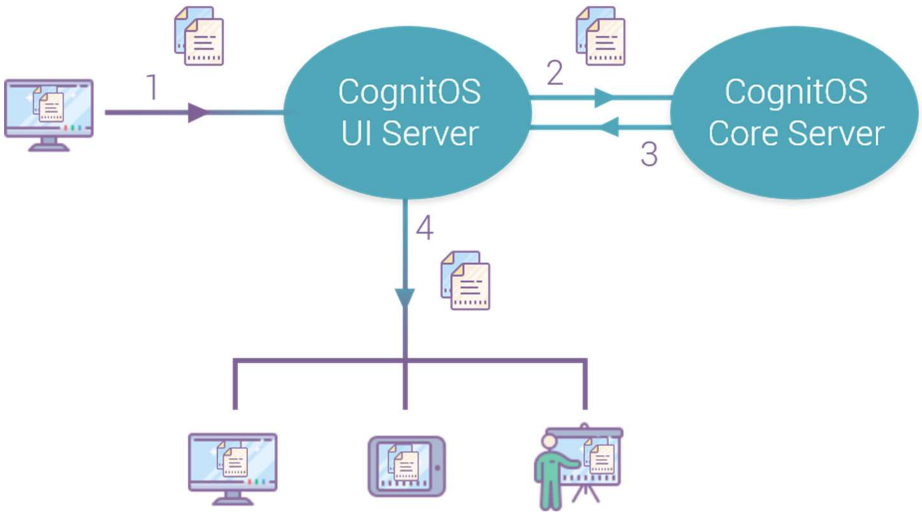


Figure 43. Data transferring between artifacts

Each device runs a different instance of CognitOS. In order to enable communications through the server between all clients, a socket service was created, which handles sending, receiving and processing of requests between the various instances. When one classroom artifact wants to communicate with another, HTTP requests are forwarded to the CognitOS UI server, which in turn is responsible for sending the request to the appropriate receiver. Figure 44 gives an example of broadcasting to all artifacts the current classroom status.

```
broadcastClassroomStatus(body: any) {
  console.log("sending classroom status: ", body);
  this.http.post(this.broadcastClassroomStatusURL, body).subscribe(
    res => {
      console.log(res);
    },
    err => {
      console.log('Error occured: ', err);
    }
  );

}
```

Figure 44. Socket service code snippet broadcasting classroom status

Each instance of CognitOS running on different artifacts contains a certain type of listeners, in this case called Observables. Observables are responsible for identifying and handling received requests and inform the front end for content changes.

Finally, through cross-device interaction, CognitOS is able to ensure the flow of data and interactions in a coherent sequence across devices. Students are able to pause an activity on one device (e.g., their desktop) and resume from the same point from another (e.g., their tablet), without any intermediate action required. An illustrative example is presented in Figure 45, where the desktop activity is transferred to the tablet, when the latter is undocked, thus locking the desktop and vice versa, when docking the tablet, the desktop unlocks and displays the activity previously present on the tablet, which in turn becomes an auxiliary device.

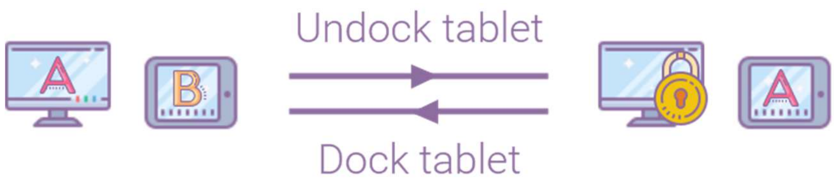


Figure 45. Activity transferring across devices

6.3 CognitOS Business Logic

CognitOS is not a one-sided system, on the contrary, its behavior can vary according to the different cases of use and the contextual parameters of the environment. System operation is closely depended on external parameters, in order to adapt its functionality according to the context of use. Since CognitOS can utilize a variety of artifacts, both inside a classroom environment, as well as in other places (e.g., at home), depending on the number of artifacts (as well as which) that are present, CognitOS alters each artifact’s functionality accordingly.

There are (currently) three different modes, corresponding to cases of use for CognitOS, as follows:

Studying. During studying, a student can be either in classroom or at home. While studying, students can have unlimited access to all resources of the system, browsing through exercises and course material freely. Study time can either happen when students are mostly at home, but in some cases in classroom as well (e.g., after class hours).

Lecture. Lecture time is during class hours. Since CognitOS knows which is the current course, as well as lesson, it has the ability to lock access to irrelevant material. For example, during the Grammar – Lesson 3 lecture, students can only view the corresponding book pages, since the book application adjusts automatically to launch the appropriate course. The same rules apply to all applications, i.e., 'My Exercises' allow access only to relevant assignments and 'My Media' displays only relevant multimedia content.

Examination. An examination can be initiated in various cases. Through AmI-Solertis, CognitOS can utilize classroom metadata, such as the class schedule, and initiate a scheduled examination. The educator can also choose to program an examination, through the CognitOS API, and initiate it at any point during a lecture. Since mini-quizzes are also a part of LECTOR's intervention techniques, an examination can be launched at any point a class-wide remedial action is required. When there is an ongoing examination, CognitOS is in lockdown mode and each artifact only displays specific content. In more detail, the Desktop displays the set of examination exercises and students can browse only between the different activities. The tablet locks to 'Utilities' (Figure 46), where only the utilities that the educator chose, when programming the examination, as appropriate regarding the examining course are unlocked (e.g., during a Mathematics examination only the calculator and media containing relevant material are available, whereas during a Grammar examination the calculator is locked and the thesaurus is available). In addition, the classroom public smart board displays a countdown timer, informing students about the remaining time, as scheduled by the educator, to finish the examination.

The aforementioned modes can be altered according to educators' preference. The level of strictness or lenience of the lockdown can vary should an educator suggest so. For example, during a lecture on Grammar where the context includes travelling to different destinations, an educator may allow access to the Geography course in order to give students the ability to explore these destinations. This behavior can be achieved through the functions available inside the CognitOS API. An educator can choose to create various rules for an examination, for example, depending on the different courses. These rules are used to adjust the level of lockdown strictness for the various classroom activities.

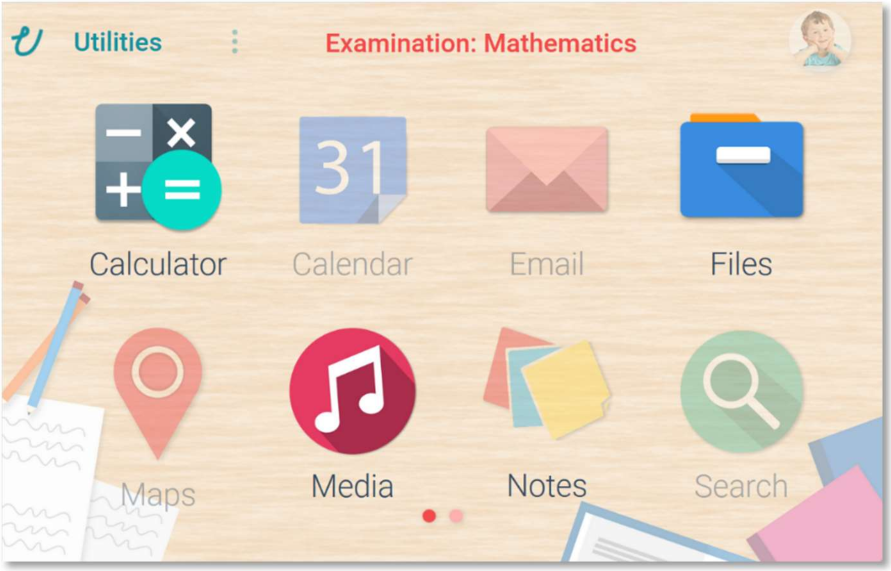


Figure 46. Tablet is locked to 'Utilities' during an examination

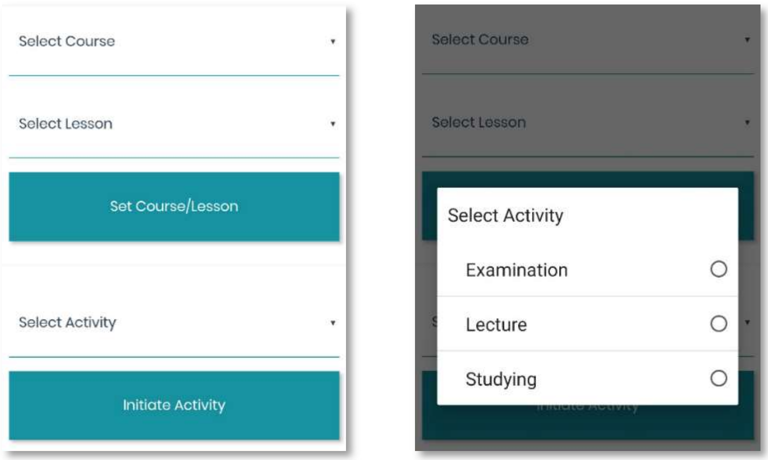


Figure 47. Controller mini-app for educator's smartphone

Apart from the system's ability to know, through AmI-Solertis, the current state of the classroom and initiate the appropriate mode (i.e., 'Examination', 'Lecture', 'Studying'), the educator can select a mode on demand at any point from her smartphone. An example controller mini-app for the educator's smartphone is presented in Figure 47, where she can select and set the current course and lesson, as well as the current classroom activity.

6.4 Intervention Dictation and Management

LECTOR [14] introduces a non-invasive multimodal solution, which exploits the potential of ambient intelligence technologies to observe student actions (SENSE), provides a framework to employ activity recognition techniques for identifying whether these actions signify inattentive behavior (THINK) and intervenes – when necessary – by suggesting appropriate methods for recapturing attention (ACT).

Whenever a stimulus is detected by the SENSE component, the THINK component initiates an exploratory process to determine whether the incoming event indicates that the student(s) has lost interest in the learning process or not. Finally, at the end of the exploratory process, if the result points to inattentive behavior, SENSE appropriately informs the ACT component which undertakes to restore student engagement by selecting an appropriate intervention.

CognitOS is the medium through which LECTOR displays interventions. As soon as an appropriate intervention is selected, LECTOR sends a command via an HTTP request to the CognitOS's server. When a request is received, CognitOS identifies whether it is a class-wide or individual intervention and proceeds to forward the request to the appropriate artifact(s). Each artifact is equipped with mechanisms able to display received content on demand, through CognitOS.

For example, the aforementioned toast notification (section 5.3.2) can display motivational messages on individual artifacts when one or a few students feel unmotivated or tired. This component receives, through an HTTP request, metadata regarding the target student or students and the appropriate content to display (Figure 48). When a request is forwarded to this component, it receives the appropriate information and launches immediately.

```
this.exerciseService.getMotivation().subscribe(() => {
  this.showNotification = this.exerciseService.getShowNotification();
  this.name = this.exerciseService.getName();
  this.notificationType = this.exerciseService.getType();
  this.message = this.exerciseService.getMessage();

  if (this.showNotification) {
    setTimeout (() => {
      this.closeNotification();
    }, 4000);
  }
});
```

Figure 48. Notification component receiving request data

6.4 Intervention Dictation and Management

Chapter 7

System Evaluation

7.1 Heuristic Evaluation on Prototypes

Before proceeding with testing CognitOS with actual users, an iterative heuristic evaluation experiment [106] was necessary, in order to avoid any major usability errors. In general, a heuristic evaluation process includes the examination of the system's User Interface (UI) by a number of expert evaluators. CognitOS was examined by five (5) experts, who also judged its compliance with certain recognized usability principles.

During this process of individual sessions, the main objective is to extract identified findings and merge them into a list of usability issues. Afterwards, each evaluator is asked to provide severity ratings for each list item individually. These ratings, which are used to prioritize the issues according to their severity, may range from zero (0) to four (4), where the first is categorized as "not a usability problem" and the last as a "usability catastrophe". The experiment on CognitOS revealed twelve (12) usability issues, none of which was rated with a four. In addition, the ease-of-fix scale – zero ("would be extremely easy to fix") to three ("would be difficult to fix") – revealed that the majority of issues required minimum amount of effort to fix.

The first iteration of this process conducted on CognitOS's prototypes, revealed a number of usability issues that were crucial to be resolved in this stage (prototyping), before proceeding with the actual implementation of the system. Afterwards, a second iteration was planned, to further investigate CognitOS's performance, before a user-based evaluation could be conducted.

7.2 Heuristic Evaluation on Implemented System

After completing the heuristic evaluation experiment on CognitOS's prototypes and resolving the issues identified, the system was implemented and a second iteration of the process was planned. The objective of this process was to further investigate the system's performance in its entirety, assessing its usability and adaptability to the actual target environment and whether it fulfills its educational goals.

7.2.1 The Process

Apart from further investigating CognitOS's possible usability issues, the second iteration of the heuristic evaluation included assessing the ease of interaction with the system and the comprehensibility of the UI (User Interface). In order to efficiently conduct this more detailed experiment, a set of actual use case scenarios containing a number of tasks were created, which were given to a number of five (5) expert evaluators, who were engaged in this process.

The evaluation experiment that was conducted was divided into five stages: (i) the preparation, (ii) the introduction, (iii) the actual experiment, (iv) the usability issue extraction and rating and (v) the usability issue ease-of-fix rating.

Since it was important to browse a big part of the system for this experiment, use case scenarios and tasks were created to lead users subconsciously into interacting with the biggest part of CognitOS. These scenarios were written in an electronic format and given to the users one by one in a presentation-like format. Before proceeding with initiating the experiment with the introduction stage, CognitOS was launched with respective UIs to all classroom artifacts, i.e., a desktop, a tablet, a board and a smartphone – which for this experiment was controlled by an assistant to the evaluation process acting as the educator, in order to simulate an actual classroom environment.

After the preparation stage was completed, the users were invited in the classroom, one by one in separate sessions, and were introduced to the purpose of the experiment, as explained in the previous section (**Error! Reference source not found.**), followed by a brief introduction to the system and its functionalities.

During the actual experiment, the scenarios were read aloud to each user as well as being displayed in an electronic format besides the experiment artifacts, so users can turn to read them again at their disposal. Since that was a heuristic-interaction evaluation experiment with experts in this domain, the was no need to offer hints or help and the session time was not written down. However, all users were prompted to speak aloud during each session, since their comments constituted a basis for identifying easier the usability issues of the system.

After finishing the session, each user was asked to extract system's usability issues based on certain recognized usability principles. These issues were afterwards merged into one list, which contained all unique findings. Each user was given a copy of this list to provide a severity rating to each issue, according to their preference. The final severity rating was produced by computing the average from each user rating.

Finally, the identified issues, along with the final severity rating, were forwarded to the development team, which ranked them according to the ease-of-fix scale, in order to designate the amount of effort needed to address them.

7.2.2 Use Case Scenarios and User Tasks

Each user was given four (4) scenarios, containing a total of nine (9) tasks. These scenarios represent possible use cases of CognitOS in its actual target environment and cover a wide range of the system's functionality.

Introduction

You are Johnathan Smith, a third grade student at the Intelligent Classroom of the HCI lab at FORTH. Inside the classroom, every student has a Desktop PC, a personal tablet, as well as access to the public classroom board. All artifacts are hosting the CognitOS system.

Scenario 1

Task 1

You are attending a lecture on Grammar on lesson three (3). Before beginning with the course, your teacher asked you to watch the video "Climate Zones of the Earth" on your desktop.

Task 2

After finishing the video, you proceed on the book to solve the Activity B of the current course.

Task 3

Class is now over, so you undock your tablet and head home.

Scenario 2

Task 1

You are now at home. You have assigned homework from the lecture you just attended (Grammar, lesson 3), specifically Activity C. Since you do not have a desktop PC at home, you will solve your assignment on the tablet.

Task 2

After you finish the exercise, you submit it to the teacher.

Scenario 3

Task 1

The next day at the classroom, the teacher requests your assistance at the board to solve a multiple choice exercise.

Task 2

While you are at the board, your classmates are distracted, loudly talking to each other. Immediately a quiz is initiated on every student's desktop PC and you go to your seat to answer the questions.

Task 3

You are confused on the answers on question number three (3), so you decide to take a hint. Later on, you are certain that you don't know the answer on the next question, so you use the provided solution.

(at this point, the system identifies that the student feels unmotivated to solve an exercise and relies on hints and solutions, so it displays a motivational message)

Task 4

After using the solution, you decide that you want to refrain from using any more help, so you watch a relevant video "Global Warming" on your tablet.

Scenario 4

Task 1

After a week, during class, your teacher informs you that your grade on Grammar's lesson three (3) examination has been added to your grades. You want to see how well you scored.

7.2.3 Evaluation Findings

The second heuristic evaluation process revealed twenty-seven (27) usability issues, which are categorized and sorted based on their severity ratings, as given by the expert evaluators. Table 4 presents the extracted issues, along with their severity and ease-of-fix ratings.

Table 4. Usability issues with severity and ease-of-fix ratings

Heuristic	**Severity Rating**	**Ease-of-fix Score**
Experts suggested the system to be more intelligent (e.g., when educators request students to watch a video, the system should only display the video)	3	2
The profile should provide different views according to the user (age based, parent view, educator view)	3	2

Experts suggested applications to immediately open on specific courses and lessons	3	1
An option to sort courses on tablet based on pending assignments should be provided	3	1
Each exercise should be accompanied by its individual due date and progress indication	3	1
Experts found the statistics on the profile overwhelming	3	1
There should be a different representation on profile regarding grades	3	1
Desktop PC should contain a title indicating current course	3	0
The menu on the board is not intuitive	3	0
The board should display more information (e.g., course remaining time)	3	0
The option for launching an exercise on book is not intuitive	3	0
When solving an exercise on book an option to apply answers should be provided	3	0
When launching 'My Exercises' the first incomplete activity should be preselected	3	0
An indicator on droppable components when dragging an element should be provided	3	0
Color coding on certain components should be redefined (e.g., red text on prefilled answers)	3	0
The profile should include more information on student academic progress (e.g., absences)	3	0
When submitting an exercise, a confirmation pop up should be provided	3	0
The examination on desktop should be better distinguished from a plain exercise	3	0

On 'My Media' home page, an indicator about the corresponding course to each video should be provided	3	0
The "launch application, then choose course" model should be "choose course, then launch application"	2	3
Experts felt that there was a big mental switch from the desktop UI to the tablet UI	2	3
More modalities on exercise solving should be provided	2	2
There were inconsistencies regarding menu navigation between some components	2	1
Dropdown drawer on tablet exercises should be accessed more obviously	2	0
A "reset all answers" button on the exercises should be provided	2	0
The toast notification on desktop should last longer	2	0
While watching a video, its title should be present	2	0

The final severity score for each usability issue was extracted by combining and averaging individual scores provided by each expert. The final score ranges between two (2) and three (3). Eight (8) issues were minor usability problems, whereas the remaining nineteen (19) were ranked as major issues and thus the most important to fix. As a last step, the development team designated the amount of effort needed to address these issues, by providing an ease-of-fix score. According to *Table 4*, fixing these issues requires a minimum to small amount of effort.

7.2 Heuristic Evaluation on Implemented System

Chapter 8

Conclusion and Future Work

8.1 Conclusions

During the past decade, the notion of the Intelligent Classroom has prevailed, equipped with technologically enhanced artifacts, a vision incited by the emergence of IoT and Intelligent Environments. In the recent years, many attempts towards the creation of student-oriented Intelligent Classrooms have been made. These approaches, enhanced with the concept of student engagement, resulted in the notion of Attention-aware Intelligent Classrooms. These environments mainly consist of a wide range of technologically augmented artifacts presenting multimodal and context-aware applications. However, in order to successfully enhance the educational process, these artifacts need to work seamlessly together. Their diverse characteristics raise a number of cross-device migration challenges of the available applications.

To address these issues, this thesis has proposed CognitOS, an attention-aware, student-centric working environment that hosts educational applications and instantiates a common Look 'n' Feel across the various classroom artifacts, thus transforming the group of isolated units into a unified environment. From an engineering perspective, CognitOS framework: (i) acts as a host of educational applications, (ii) supports the mission of the Attention-aware Intelligent Classroom by employing mechanisms to intervene – when necessary – to draw the educator's attention to the problematic situations and re-engage distracted or unmotivated students to the task at hand, (iii) offers tools for creating and editing applications, customizing its structure and functionality on demand and (iv) provides an API with useful functions to define behavioral rules according to different cases of use.

8.2 Future Work

Since CognitOS is a service, it already provides means to configure its behavior. The system's expandable API offers useful functions that can be utilized to achieve this goal. Also, a style guide has been created in order to facilitate the creation of new components and applications to be added to CognitOS and fit seamlessly with the existing ones.

However, apart from the behavior configuration, the customization of the service itself is worth further research. User-friendly means of customization – from adding new learning content to introducing new applications – should be provided to make the process straightforward and effortless.

8.2.1 Learning Content Editor

CognitOS's structure features data-independent components. In more detail, the components developed are dynamically filled on demand given any content that fits the requirements of each different type. For example, to fill a multiple choice exercise, a user (i.e., an educator or a developer) needs to provide a set of questions followed be the number of possible answers, including an indication of the correct one. This data is stored in a database connected with CognitOS. Currently, in order to provide more data or edit existing ones, a user needs to interact directly with the database. Therefore, developing an educator-friendly editor to add learning content is important.

In this editor an educator would find a list of all available components that utilize data, as well as a collection of all existing database entries. From the editor, an educator should be able to add new learning content and edit or remove existing entries.

8.2.2 Component Editor

Currently, CognitOS features a set of applications containing a variety of components. For example, 'My Exercises' application contains various exercise types, such as multiple choice and fill the blanks. However, any other developer should be able to externally add new components to expand the functionality of these applications. An editor should be able to offer tools to enrich applications with new components.

In order to do so, the editor should provide a list of existing applications, complete with their already registered components. Through this editor, a user would insert the new component under the application it applies. In addition, a user would be able to define the component's behavior based on the application's already existing rules, while creating specific conditions of use.

8.2.3 Application Editor

In order to expand the service to fit better the educational needs inside a classroom setting, new applications should be easily installed to CognitOS. To add a new application, it should firstly abide to the rules of the system's style guide.

Apart from the UI development, any new application should be accompanied by its behavioral rules. A user should be able to easily create such rules through an editor. This editor should utilize the API functions provided by CognitOS to facilitate the construction of conditions that define the behavior of the application, according to the various classroom activities. These conditions could be in the form of simple "if this, then that" statements that enable users to easily create the application logic.

Bibliography

[1] V. L. Tinio, *ICT in Education*. e-ASEAN Task Force, 2003.

[2] A. T. Bates, *Technology, e-learning and distance education*. Routledge, 2005.

[3] C. Brooks, J. Greer, E. Melis, and C. Ullrich, “Combining its and elearning technologies: Opportunities and challenges,” in *International Conference on Intelligent Tutoring Systems*, 2006, pp. 278–287.

[4] N. Cross and R. Roy, *Engineering design methods*, vol. 4. Wiley New York, 1989.

[5] P. Xu, G. Han, W. Li, Z. Wu, and M. Zhou, “Towards intelligent interaction in classroom,” in *International Conference on Universal Access in Human-Computer Interaction*, 2009, pp. 150–156.

[6] H. Kopetz, “Internet of things,” in *Real-time systems*, Springer, 2011, pp. 307–323.

[7] A. Aztiria, A. Izaguirre, and J. C. Augusto, “Learning patterns in ambient intelligence environments: a survey,” *Artificial Intelligence Review*, vol. 34, no. 1, pp. 35–51, 2010.

[8] S. S. Yau, S. K. Gupta, F. Karim, S. I. Ahamed, Y. Wang, and B. Wang, “Smart classroom: Enhancing collaborative learning using pervasive computing technology,” in *ASEE 2003 Annual Conference and Exposition*, 2003, pp. 13633–13642.

[9] E. K. F. Chan, M. A. Othman, and M. A. Razak, “IoT Based Smart Classroom System,” *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 9, no. 3–9, pp. 95–101, 2017.

[10] Y. Shi *et al.*, “The smart classroom: merging technologies for seamless tele-education,” *IEEE Pervasive Computing*, vol. 2, no. 2, pp. 47–55, 2003.

[11]J. R. Cooperstock, "The classroom of the future: enhancing education through augmented reality," in *Proc. HCI Inter. 2001 Conf. on Human-Computer Interaction*, 2001, pp. 688–692.

[12]M. Antona, A. Leonidis, G. Margetis, M. Korozi, S. Ntoa, and C. Stephanidis, "A student-centric intelligent classroom," in *International Joint Conference on Ambient Intelligence*, 2011, pp. 248–252.

[13]G. S. Partnership, "The Glossary of Education Reform," *The Glossary of Education Reform - Student Engagement Definition*, Dec-2013. [Online]. Available: https://www.edglossary.org/student-engagement/. [Accessed: 02-May-2018].

[14]M. Korozi, A. Leonidis, M. Antona, and C. Stephanidis, "LECTOR: Towards Reengaging Students in the Educational Process Inside Smart Classrooms," in *International Conference on Intelligent Human Computer Interaction*, 2017, pp. 137–149.

[15]C. Savvaki, A. Leonidis, G. Paparoulis, M. Antona, and C. Stephanidis, "Designing a technology–augmented school desk for the future classroom," in *International Conference on Human-Computer Interaction*, 2013, pp. 681–685.

[16]W. Watson and S. L. Watson, "An Argument for Clarity: What are Learning Management Systems, What are They Not, and What Should They Become.," 2007.

[17]M. Szabo, "Cmi theory and practice: Historical roots of learning managment systems," in *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, 2002, pp. 929–936.

[18]K. Gilhooly, "Making E-Learning Effective.," *Computerworld*, vol. 35, no. 29, pp. 52–53, 2001.

[19]K. A. Al-Busaidi and H. Al-Shihi, "Instructors' acceptance of learning management systems: A theoretical framework," *Communications of the IBIMA*, vol. 2010, no. 2010, pp. 1–10, 2010.

[20]Y. Kritikou, P. Demestichas, E. Adamopoulou, K. Demestichas, M. Theologou, and M. Paradia, "User Profile Modeling in the context of web-based learning management systems," *Journal of Network and Computer Applications*, vol. 31, no. 4, pp. 603–627, 2008.

[21]A. Albirini, "Teachers' attitudes toward information and communication technologies: The case of Syrian EFL teachers," *Computers & Education*, vol. 47, no. 4, pp. 373–398, 2006.

[22] N. Cavus and M. M. Ala'a, "Computer aided evaluation of learning management systems," *Procedia-Social and Behavioral Sciences*, vol. 1, no. 1, pp. 426–430, 2009.

[23] H. Coates, R. James, and G. Baldwin, "A critical examination of the effects of learning management systems on university teaching and learning," *Tertiary education and management*, vol. 11, pp. 19–36, 2005.

[24] D. Dagger, A. O'Connor, S. Lawless, E. Walsh, and V. P. Wade, "Service-oriented e-learning platforms: From monolithic systems to flexible services," *IEEE Internet Computing*, vol. 11, no. 3, 2007.

[25] T. Mayes and S. De Freitas, "Review of e-learning theories, frameworks and models," *JISC e-learning models desk study*, no. 1, 2004.

[26] S. Graf, *Adaptivity in learning management systems focussing on learning styles*. na, 2007.

[27] D. H. Jonassen and B. L. Grabowski, *Handbook of individual differences, learning, and instruction*. Routledge, 2012.

[28] J. C. Augusto, "Ambient intelligence: the confluence of ubiquitous/pervasive computing and artificial intelligence," in *Intelligent Computing Everywhere*, Springer, 2007, pp. 213–234.

[29] L. R. Winer and J. Cooperstock, "The 'intelligent classroom': Changing teaching and learning with an evolving technological environment," *Computers & Education*, vol. 38, no. 1–3, pp. 253–266, 2002.

[30] D. J. Cook and S. K. Das, "How smart are our environments? An updated look at the state of the art," *Pervasive and mobile computing*, vol. 3, no. 2, pp. 53–73, 2007.

[31] D. N. Rapp, "The value of attention aware systems in educational settings," *Computers in Human Behavior*, vol. 22, no. 4, pp. 603–614, 2006.

[32] R. G. Packard, "The control of 'classroom attention': A group contingency for complex behavior," *Journal of Applied Behavior Analysis*, vol. 3, no. 1, pp. 13–28, 1970.

[33] A. H. Johnstone and F. Percival, "Attention breaks in lectures.," *Education in chemistry*, vol. 13, no. 2, pp. 49–50, 1976.

[34] D. M. Bunce, E. A. Flens, and K. Y. Neiles, "How long can students pay attention in class? A study of student attention decline using clickers," *Journal of Chemical Education*, vol. 87, no. 12, pp. 1438–1443, 2010.

[35] D. A. Bligh, *What's the Use of Lectures?* Intellect books, 1998.

[36] D. Koller, "Death knell for the lecture: Technology as a passport to personalized education," *New York Times*, vol. 5, 2011.

[37] S. Khan, *The one world schoolhouse: Education reimagined*. Twelve, 2012.

[38] N. Unsworth, B. D. McMillan, G. A. Brewer, and G. J. Spillers, "Everyday attention failures: An individual differences investigation.," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 38, no. 6, p. 1765, 2012.

[39] K. K. Szpunar, S. T. Moulton, and D. L. Schacter, "Mind wandering and education: from the classroom to online learning," *Frontiers in psychology*, vol. 4, p. 495, 2013.

[40] J. V. GILMORE, "The factor of attention in underachievement," *The Journal of Education*, vol. 150, no. 3, pp. 41–66, 1968.

[41] R. L. Sprague and L. K. Toppe, "Relationship between activity level and delay of reinforcement in the retarded," *Journal of Experimental Child Psychology*, vol. 3, no. 4, pp. 390–397, 1966.

[42] D. H. Lloyd, "A concept of improvement of learning response in the taught lesson," *Visual education*, vol. 21, pp. 23–25, 1968.

[43] M. J. Kane, L. H. Brown, J. C. McVay, P. J. Silvia, I. Myin-Germeys, and T. R. Kwapil, "For whom the mind wanders, and when: An experience-sampling study of working memory and executive control in daily life," *Psychological science*, vol. 18, no. 7, pp. 614–621, 2007.

[44] W. McKeachie and M. Svinicki, *McKeachie's teaching tips*. Cengage Learning, 2013.

[45] D. R. Thomas, W. C. Becker, and M. Armstrong, "Production and elimination of disruptive classroom behavior by systematically varying teacher's behavior," *Journal of Applied Behavior Analysis*, vol. 1, no. 1, pp. 35–45, 1968.

[46] R. A. Winett and R. C. Winkler, "Current behavior modification in the classroom: Be still, be quiet, be docile," *Journal of Applied Behavior Analysis*, vol. 5, no. 4, pp. 499–504, 1972.

[47] V. M. G. Barrios *et al.*, "AdELE: A framework for adaptive e-learning through eye tracking," *Proceedings of IKNOW*, pp. 609–616, 2004.

[48] C. Merten and C. Conati, “Eye-tracking to model and adapt to user meta-cognition in intelligent learning environments,” in *Proceedings of the 11th international conference on Intelligent user interfaces*, 2006, pp. 39–46.

[49] J. L. Sibert, M. Gokturk, and R. A. Lavine, “The reading assistant: eye gaze triggered auditory prompting for reading remediation,” in *Proceedings of the 13th annual ACM symposium on User interface software and technology*, 2000, pp. 101–107.

[50] D. A. Slykhuis, E. N. Wiebe, and L. A. Annetta, “Eye-tracking students’ attention to PowerPoint photographs in a science education setting,” *Journal of Science Education and Technology*, vol. 14, no. 5–6, pp. 509–520, 2005.

[51] H. Wang, M. Chignell, and M. Ishizuka, “Empathic tutoring software agents using real-time eye tracking,” in *Proceedings of the 2006 symposium on Eye tracking research & applications*, 2006, pp. 73–78.

[52] M. Prince, “Does active learning work? A review of the research,” *Journal of engineering education*, vol. 93, no. 3, pp. 223–231, 2004.

[53] C. C. Bonwell and J. A. Eison, *Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports.* ERIC, 1991.

[54] L. A. Burke and R. Ray, “Re-setting the concentration levels of students in higher education: an exploratory study,” *Teaching in Higher Education*, vol. 13, no. 5, pp. 571–582, 2008.

[55] R. Hitz and A. Driscoll, “Praise in the Classroom.,” 1989.

[56] G. L. Morgan, *Improving student engagement: Use of the interactive whiteboard as an instructional tool to improve engagement and behavior in the junior high school classroom*. Liberty University, 2008.

[57] D. Szafir and B. Mutlu, “Pay attention!: designing adaptive agents that monitor and improve user engagement,” in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2012, pp. 11–20.

[58] L. Brown, R. Kerwin, and A. M. Howard, “Applying behavioral strategies for student engagement using a robotic educational agent,” in *Systems, Man, and Cybernetics (SMC), 2013 IEEE international conference on*, 2013, pp. 4360–4365.

[59] J. Chen *et al.*, “FishBuddy: Promoting Student Engagement in Self-Paced Learning through Wearable Sensing,” in *Smart Computing (SMARTCOMP), 2017 IEEE International Conference on*, 2017, pp. 1–9.

[60] S. Houben *et al.*, “Opportunities and challenges for cross-device interactions in the wild,” *interactions*, vol. 24, no. 5, pp. 58–63, 2017.

[61] C. Rowland, E. Goodman, M. Charlier, A. Light, and A. Lui, *Designing connected products: UX for the consumer Internet of Things*. O’Reilly Media, Inc., 2015.

[62] M. Wäljas, K. Segerståhl, K. Väänänen-Vainio-Mattila, and H. Oinas-Kukkonen, “Cross-platform service user experience: a field study and an initial framework,” in *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, 2010, pp. 219–228.

[63] A. Designs, “Acadly.” [Online]. Available: https://www.acadly.com. [Accessed: 15-Apr-2019].

[64] “Courselle - Create and Sell Engaging Online Courses,” *Courselle*. [Online]. Available: https://courselle.com/. [Accessed: 15-Apr-2019].

[65] “Kahoot! | Learning Games | Make Learning Awesome!,” *Kahoot!* [Online]. Available: https://kahoot.com/. [Accessed: 15-Apr-2019].

[66] “The easy way to differentiate your teaching,” *Quizalize*. [Online]. Available: https://www.quizalize.com. [Accessed: 15-Apr-2019].

[67] “Cloud-Based Lesson Delivery Software for tInteractive Whiteboards & Displays.,” *ClassFlow*. [Online]. Available: https://classflow.com/. [Accessed: 15-Apr-2019].

[68] “Classroom: manage teaching and learning,” *Google for Education*. [Online]. Available: https://edu.google.com/products/classroom/. [Accessed: 15-Apr-2019].

[69] “Moodle - Open-source learning platform | Moodle.org.” [Online]. Available: https://moodle.org/. [Accessed: 15-Apr-2019].

[70] “Quizlet,” *Quizlet*. [Online]. Available: https://quizlet.com/. [Accessed: 15-Apr-2019].

[71] “Learning Management System | LMS | Schoology.” [Online]. Available: https://www.schoology.com/homepage. [Accessed: 15-Apr-2019].

[72] “Seesaw,” *Seesaw*. [Online]. Available: https://web.seesaw.me/. [Accessed: 15-Apr-2019].

[73]“BookWidgets - The perfect content creation tool for teachers in the classroom,” *BookWidgets*. [Online]. Available: https://www.bookwidgets.com/. [Accessed: 15-Apr-2019].

[74]“Classtime: The clear path to student success,” *Classtime*. [Online]. Available: https://www.classtime.com/. [Accessed: 15-Apr-2019].

[75]“Edmodo,” *Edmodo*. [Online]. Available: https://www.edmodo.com/. [Accessed: 15-Apr-2019].

[76]“Socrative.” [Online]. Available: https://socrative.com/. [Accessed: 15-Apr-2019].

[77]“Padlet is the easiest way to create and collaborate in the world,” *Padlet*. [Online]. Available: https://padlet.com/. [Accessed: 15-Apr-2019].

[78]“Popplet.” [Online]. Available: http://popplet.com/. [Accessed: 15-Apr-2019].

[79]M. Antona *et al.*, *Ambient Intelligence in the classroom: an augmented school desk*. na, 2010.

[80]A. Leonidis, G. Margetis, M. Antona, and C. Stephanidis, “ClassMATE: enabling ambient intelligence in the classroom,” *World Academy of Science, Engineering and Technology*, vol. 66, pp. 594–598, 2010.

[81]M. Korozi, S. Ntoa, M. Antona, A. Leonidis, and C. Stephanidis, “Towards building pervasive UIs for the intelligent classroom: the PUPIL approach,” in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, 2012, pp. 279–286.

[82]M. Korozi, S. Ntoa, M. Antona, and C. Stephanidis, “Intelligent working environments for the ambient classroom,” in *International Conference on Universal Access in Human-Computer Interaction*, 2011, pp. 381–390.

[83]M. Korozi *et al.*, “Ambient educational mini-games,” in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, 2012, pp. 802–803.

[84]G. Mathioudakis *et al.*, “Real-time teacher assistance in technologically-augmented smart classrooms,” *Int. J. Adv. Life Sci*, vol. 6, no. 1, pp. 62–73, 2014.

[85]A. Leonidis, D. Arampatzis, N. Louloudakis, and C. Stephanidis, “The AmI-Solertis system: creating user experiences in smart environments,” in *Wireless and Mobile Computing, Networking and Communications (WiMob),* 2017, pp. 151–158.

[86] M. Korozi, M. Antona, A. Ntagianta, A. Leonidis, and C. Stephanidis, “LECTORSTUDIO: CREATING INATTENTION ALARMS AND INTERVENTIONS TO REENGAGE THE STUDENTS IN THE EDUCATIONAL PROCESS,” in *ICERI2017 Proceedings*, Seville, Spain, 2017, pp. 4486–4495.

[87] I. DIS, “9241-210: 2010. Ergonomics of human system interaction-Part 210: Human-centred design for interactive systems,” *International Standardization Organization (ISO). Switzerland*, 2009.

[88] J. Piaget, “Science of education and the psychology of the child. Trans. D. Coltman.,” 1970.

[89] J. Nielsen, “Teenagers on the web: 61 usability guidelines for creating compelling websites for teens,” *Nielsen Norman Group Report*, vol. 2005, 2005.

[90] L. Hanna, K. Risden, M. Czerwinski, and K. J. Alexander, “The role of usability research in designing children’s computer products,” in *The design of children’s technology*, 1998, pp. 3–26.

[91] A. Druin, B. Bederson, A. Boltman, A. Miura, D. Knotts-Callahan, and M. Platt, “Children as Our Technology Design Partners+,” 1998.

[92] N. S. Said, “An engaging multimedia design model,” in *Proceedings of the 2004 conference on Interaction design and children: building a community*, 2004, pp. 169–172.

[93] N. Kaplan *et al.*, “Supporting sociable literacy in the international children’s digital library,” in *Proceedings of the 2004 conference on Interaction design and children: building a community*, 2004, pp. 89–96.

[94] M. V. Covington, “Goal theory, motivation, and school achievement: An integrative review,” *Annual review of psychology*, vol. 51, no. 1, pp. 171–200, 2000.

[95] C. Midgley *et al.*, “The development and validation of scales assessing students’ achievement goal orientations,” *Contemporary educational psychology*, vol. 23, no. 2, pp. 113–131, 1998.

[96] T. C. Urdan and M. L. Maehr, “Beyond a two-goal theory of motivation and achievement: A case for social goals,” *Review of educational research*, vol. 65, no. 3, pp. 213–243, 1995.

[97] R. Kershner, N. Mercer, P. Warwick, and J. K. Staarman, “Can the interactive whiteboard support young children’s collaborative communication and thinking in classroom science activities?,” *International*

Journal of Computer-Supported Collaborative Learning, vol. 5, no. 4, pp. 359–383, 2010.

[98] N. Mercer, P. Warwick, R. Kershner, and J. K. Staarman, “Can the interactive whiteboard help to provide ‘dialogic space’for children’s collaborative activity?,” *Language and education*, vol. 24, no. 5, pp. 367–384, 2010.

[99] G. A. Miller, “The magical number seven, plus or minus two: Some limits on our capacity for processing information.,” *Psychological review*, vol. 63, no. 2, p. 81, 1956.

[100] J. W. Burgess and D. Spoor, “Seven faces in a crowd: Parallel or serial information processing,” *International Journal of Neuroscience*, vol. 12, no. 1, pp. 21–23, 1981.

[101] “INTERVENTION | meaning in the Cambridge English Dictionary.” [Online]. Available: https://dictionary.cambridge.org/dictionary/english/intervention. [Accessed: 23-Jan-2019].

[102] R. E. Johnson and B. Foote, “Designing reusable classes,” *Journal of object-oriented programming*, vol. 1, no. 2, pp. 22–35, 1988.

[103] M. Fayad and D. C. Schmidt, “Object-oriented Application Frameworks,” *Commun. ACM*, vol. 40, no. 10, pp. 32–38, Oct. 1997.

[104] G. Froehlich, H. J. Hoover, L. Liu, and P. Sorenson, “Hooking into object-oriented application frameworks,” in *Proceedings of the 19th international conference on Software engineering - ICSE ’97*, Boston, Massachusetts, United States, 1997, pp. 491–501.

[105] D. Brugali and K. Sycara, “Towards Agent Oriented Application Frameworks,” *ACM Comput. Surv.*, vol. 32, no. 1es, Mar. 2000.

[106] J. Nielsen, “Heuristic Evaluation: How-To: Article by Jakob Nielsen. Jan. 1995,” *URl: http://www. nngroup. com/articles/how-to-conduct-a-heuristicevaluation/(visited on 11/01/2014)*.