FLEXIBLE EDUCATIONAL ENVIRONMENTS: ORCHESTRATION OF LEARNING ACTIVITIES THROUGH SEMANTIC INTEROPERABILITY

Dissertation in fulfillment of the requirements for the academic degree
Doctor of Technical Sciences (Dr. techn.) in Computer Science
Submitted by

ROCAEL HERNÁNDEZ RIZZARDINI

Institute of Information Systems and Computer Media (IICM)
Doctoral School of Computer Science
Graz University of Technology
A-8010 Graz, Austria
March 2015

First reader: Univ.Doz. Dipl.-Ing. Dr. techn. Christian Gütl
Second reader: Prof. Vanessa Chang

FLEXIBLE LERNUMGEBUNG: LERNAKTIVITÄTEN ORCHESTRIERUNG MITTELS SMANTISCHE INTEROPERABILITÄT

Dissertation zur Verleihung des akademischen Grades
Doktor der Technischen Wissenshaften (Dr. techn.) in Informatik
vorgelegt von

ROCAEL HERNÁNDEZ RIZZARDINI

Institut für Informationssysteme und Computer Medien (IICM)
Fakultät für Informatik
Technische Universität Graz
A-8010 Graz, Österreich
März 2015

Erstbegutachter: Univ.Doz. Dipl.-Ing. Dr. techn. Christian Gütl
Begutachter: Prof. Vanessa Chang

ABSTRACT

The term Web 2.0 refers to a participatory Web where collaboration happens at different levels on the Web. Furthermore, Web 2.0 includes interactive and innovative features for Web tools, and it is becoming standard for those tools to have their own applications for mobile devices. Such tools use *Cloud Computing* to scale their services to a massive audience. In addition, many of the tools provide access to programmatically control their features through an open *Application Program Interface (API).* In this dissertation, such Web 2.0 tools that use *Cloud Computing* and have an open API are referred to as *Cloud-Based Tools (CBTs).* Likewise, the CBTs have been investigated regarding their use in education. Several comprehensive studies demonstrate the applicability, beginning from the learning theories that support the inclusion and going through *Collaborative Learning, Learning Orchestration, Educational Models*, and *Cognitive Taxonomies*. *Learning Orchestration*, in particular, identifies the capacity to have a *granular* management over the CBTs is required, with the ability to provide adaption, flexibility, intervention, assessment, and role management. Moreover, it is identified within *Learning Activities* that use CBTs, it is possible to promote higher-order thinking skills, such as analyzing, evaluating, and creating. Pedagogical research identified barriers for the adoption of CBTs, such as authority, computer literacy, effectiveness of use, and technological cohesion with current *Virtual Learning Environments (VLEs).* Thus, it has become clear that a *flexible* Web interoperability is required between the VLEs and CBTs that addresses the aforementioned issues. Thereby, Web interoperability technologies are examined in terms of simplifying the integration and maintenance of Web interoperability with CBTs. The results are that Semantic Web technologies present the best approach due to the ability to have self-described Web APIs that allow automatic machine-processes. In contrast, the current educational technologies for Web interoperability with third-party tools do not address the management issues described, nor do they use Semantic Web technologies.

This doctoral dissertation focuses on enabling a *Cloud Education Environment* that is capable of orchestrating CBTs through their open APIs. Therefore, the *Cloud Learning Activity Orchestration* system (CLAO) is created, which serves as an interaction interface for *Learning Activities* that use CBTs. To enable Semantic Web interoperability, the *Cloud Interoperability Service (CIS)* is introduced, as it is capable of automatically recognizing and processing CBTs Web API without custom programs written for each API. Furthermore, ontologies for Web interoperability were developed for two specific application domains, which serve to develop Semantic *Generic Vocabularies (GVs)* that represent such application domains. With this CBTs can use such GVs to describe their Web API, and they can customize that description as needed through their own *API Documentation*. All of these technologies have been evaluated in terms of functionality, scalability, usability, perceived emotions, cognitive learning strategies, motivation, and learning analytics. Over six thousand learners have used these technologies and have been evaluated in different educational settings, including *Massive Open Online Courses*. The outcomes obtained are highly positive for the *Flexible Educational Environment* created.

KURZFASSUNG

Der Begriff Web 2.0 bezieht sich auf ein partizipatives World Wide Web, in dem eine Zusammenarbeit auf unterschiedlichen Ebenen stattfindet. Das Web 2.0 umfasst interaktive und innovative Funktionen für Web Tools, welche mittlerweile fast standardmäßig als Apps für mobile Geräte verfügbar sind. Um die Dienste solcher Tools für eine massive Anwenderanzahl skalieren zu lassen, bedient man sich des *Cloud Computing*. Viele dieser Tools erlauben einen programmierbaren Zugang zur Kontrolle ihrer Funktionen über eine offene Schnittstelle (*Application Program Interface, API*). Im Rahmen dieser Dissertation werden Web 2.0 Tools, die Cloud Computing benutzen und eine offene API besitzen, *Cloud-Based Tools (CBTs)* genannt und innerhalb des Lernkontexts untersucht. Mehrere umfassende Studien zeigen die Anwendbarkeit von CBTs, von ihrer Einbeziehung in Lerntheorien bis hin zu *Collaborative Learning, Learning Orchestration, Educational Models*, und *Cognitive Taxonomies*. Insbesondere im Bereich *Learning Orchestration* weist man darauf hin, dass eine *granular* Steuerung der CBTs erforderlich ist, zusammen mit der Verfügbarkeit von Fähigkeiten zur Anpassung, Flexibilität, Intervention, Überprüfung von Wissen, und Rollenmanagement. Es ist ebenfalls identifiziert worden, dass die Nutzung von CBTs in *Learning Activities* die Förderung von höher-geordneten Denkfähigkeiten (z.B. Analysieren, Evaluieren, Erschaffen) ermöglicht. Pädagogische Forschung identifiziert Barrieren zur Adoption von CBTs, wie beispielsweise Computer-Kenntnisse, Effektivität der Benutzung, und technologische Kohäsion mit *Virtual Learning Environments (VLEs)*. Um die erwähnten Punkte zu adressieren, wurde somit klar, dass eine *flexible* Web-Interoperabilität zwischen VLEs und CBTs vonnöten ist. Technologien zur Web-Interoperabilität werden untersucht, um die Integration und Wartung von CBTs zu vereinfachen; als Ergebnis stellt sich heraus, dass Semantic Web Technologies den besten Lösungsansatz darstellen, weil sie selbst-beschreibende APIs definieren, und diese können automatisiert und maschinell verarbeitet werden. Demgegenüber, die existierenden Lerntechnologien für Web-Interoperabilität mit Tools von Drittanbietern unterstützen weder die Lösung der beschriebenen Probleme noch die Nutzung von Semantic Web Technologies.

Die vorliegende Arbeit fokussiert auf das Ermöglichen eines *Cloud Education Environment*, welches die Orchestrierung von CBTs durch offene APIs erlaubt. Folglich wurde das *Cloud Learning Activity Orchestration* (CLAO) System, welches als Interaktionsschnittstelle für *CBT - Learning Activities* dient, umgesetzt. Um Semantic Web-Interoperabilität zu erlauben, wurde ein *Cloud Interoperability Service* (CIS) eingeführt, denn es ist in der Lage, CBT Web APIs automatisch zu erkennen und zu verarbeiten, ohne jede einzelne API implementieren zu müssen. Des Weiteren wurden Ontologien für Web-Interoperabilität in zwei Anwendungsbereichen definiert; sie dienten der Entwicklung von semantischen *Generic Vocabularies* (GVs) zur Repräsentation dieser Anwendungsbereiche. Dadurch können CBTs solche GVs zur Beschreibung ihrer Web APIs benutzen und diese Beschreibungen können sogar entsprechend ihrer *API Dokumentation* angepasst werden. All diese Technologien wurden unter mehreren Aspekten evaluiert: Funktionalität, Skalierbarkeit, Benutzerfreundlichkeit, Emotionsempfindung, kognitive Lernstrategien, Motivation, und Lernanalytik. Über sechs Tausend Lernende (Studierende) haben diese Technologien benutzt und wurden in unterschiedlichen Lernszenarien evaluiert, unter anderem *Massive Open Online Courses*. Die gewonnenen Resultate sind höchst positiv für die umgesetzte *„Flexible Lernumgebungen“* (*Flexible Educational Environment*).

ACKNOWLEDGEMENTS

This doctoral dissertation has been a continuous adventure filled with discovery, which has been achievable because of the many people involved. First, I extend my gratitude to my supervisor, Christian Gütl—your commitment to the pursuit of excellence has inspired and taught me. The whole process has been such a great learning experience; thanks for your dedication, patience, and guidance. Thanks to Víctor García for your interest and friendship from the very beginning, which made all of this possible in the first place. None of this could have been achieved without the encouragement that my beautiful and smart wife, Vivian, has continuously given me. You were always there with the right words. Finally, this research is dedicated to my kids, Fabián and Josué, as I wish to inspire you both in your future endeavors. Remember, all dreams can come true with the corresponding enthusiasm and effort.

The research process began as a dream to build a world-class research team at Galileo University, in Guatemala, my home country—a research team that leverages the high quality of the team members at Galileo and produces relevant research outcomes for the entire country's benefit. And the dream is coming true, as we have already begun high-impact research projects in the GES department, and we have a team of researchers working on them. I would like to extend special thanks to all of those researchers who are putting their effort and heart into this: Héctor for his dedication; Miguel for his creativity; Byron, Isabel, and Alex for their technological abilities; and many others such as our professors, tutors, instructional designers, graphical designers, and other key players who have participated in the process. With their contribution, this research has been feasible. Furthermore, appreciation to all my research colleagues and friends from several countries, with you all this research has been a great experience, in every joint publication I learned a lot from you. Special thanks to my second reader, Prof. Vanessa Chang, for her constructive feedback.

It is wonderful to be in a dynamic and forward-thinking environment such as Galileo University. Thanks to the leadership, vision, and support of our president, Dr. Suger, as well as other authorities of the university, all of these projects have been possible. We are just beginning; the research at Guatemala will continue to grow further with the collaboration and effort of many more.

Finally, thank you to all of my friends and family for their support throughout this time, especially to my mom and dad.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources and resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, March 2015

TABLE OF CONTENTS

List of Tables

LIST OF FIGURES

LIST OF CODE LISTING

LIST OF ABBREVIATIONS AND ACRONYMS

ACG: Authorization Code Grant
ADT: Application Domain Type
AM: Arithmetic Means
AMOES: Attrition Model for Open Learning Environment Setting
API Doc: API Documentation
BLE: Business Logic Engine
CAM: Contextualized Attention Metadata
CBT: Cloud-Based Tools
CEE: Cloud Education Environment
CES: Computer Emotions Scale
CIS: Cloud Interoperability Service
CLAO: Cloud Learning Activities Orchestration
CPH: Communication Process Handler
CRUD: Create, Read, Update, Delete
CSCL: Computer-Supported Collaborative Learning
EGO: Extrinsic Goal Orientation
ER: Effort Regulation
FWI: Functional Widget Interface
GA: Google Analytics
GD: Google Drive
GV: Generic Vocabulary
HEI: Higher Education Institute
HOTS: Higher-Order Thinking Skills
ICT: Information and Communication Technologies
IG: Implicit Grant
IGO: Intrinsic Goal Orientation
IMM: Intrinsic Motivations Measure
IRI: Internationalized Resource Identifier
ISFA: Interoperability Service Framework Architecture
IVN: Institute Von Neumann
IWC: Inter-Widget Communication
LA: Learning Activities
LAMS: Learning Activity Management System
LAO: Learning Activities Orchestrator
LEC: Learning Environment Connector
LMS: Learning Management System
LO: Learning Orchestration
LOTS: Lower-Order Thinking Skills
MM: MindMeister
MOOCs: Massive Open Online Courses
MSLQ: Motivated Strategies Learning Questionnaire
MSR: Meta-Cognitive Self-Regulation
MVC: Model View Controller
OER: Open Educational Resources
OLAS: Orchestration Learning Activities System
OWL: Web Ontology Language
PLE: Personal Learning Environment
PUEU: Perceived Usefulness and Ease of Use
RDF: Resource Description Framework

REST: Representational State Transfer
SD: Standard Deviations
SDK: Software Development Kit
SNS: Social Networking Sites
SOAP: Simple Object Access Protocol
SRL: Self Regulated Learning
SUS: System Usability Scale
TAM: Technology Acceptance Model
TC: Tool Consumers
TP: Tool Providers
VLE: Virtual Learning Environment
W3C: Word Wide Web Consortium
WSDL: Web Service Description Language

1 INTRODUCTION

This Chapter introduces the research of this doctoral dissertation. First, the motivations for creating a flexible educational environment is elaborated. Based on this, the general research questions that lead to the target of this research are stated. Finally, the research methodology and the structure of the thesis is presented.

1.1 Motivation

Changes at the societal, cultural, and economic levels because of the extended incorporation of *Information and Communication Technologies (ICT)* have been discussed extensively. ICT includes any communication device or application, such as computers, televisions, mobile devices and phones, satellite systems, and corresponding applications like videoconferencing. Furthermore ICT is the integration of telecommunications, computers and the necessary software, middleware, storage and audio-visual systems, which enable users to access, store, transmit, and manipulate information. Therefore, technology-enhanced learning is defined as the influence of ICT on educational systems, also referred to as e-education. The current increase in the availability of Internet broadband has opened the possibility to consume rich media content and applications. ICT has significantly changed how we interact, communicate, and work. Along these lines, the term Web 2.0, attributed to O'Reilly (2005), refers to a more participatory Web, where people collaborate, share, network, and become co-producers and consumers. The new human interactions that ICT enables potentially can encourage innovation and the integration of ideas and move societies forward. Although challenges arise, such as determining how valuable the knowledge produced is, identity issues, ownership rights, copyright, privacy, and other problems, these challenges also nourish the new societal mix. Furthermore, new and innovative *Web 2.0 Tools* are common nowadays. Most of the tools are Web based, although many of them have corresponding specialized apps for multiple devices (desktops, tablets, smartphones, gaming consoles, smart boards, smart TVs, etc.), providing an enhanced user experience and keeping the application state synchronized everywhere. Those *Web 2.0 Tools* usually include features such as collaboration, sharing, remixing, repurposing, and networking. These tools commonly run over cloud computing infrastructures (Chao, 2012) to enhance scalability and ensure service availability. In addition, *Web 2.0 Tools* have begun to open their Web APIs, so clients can access tools and features programmatically and build and create their own experiences using the tool. Therefore, in this thesis, these tools are referred to as *Cloud-Based Tools* because of their cloud computing infrastructure and open Web API. All these features open the path to develop user experiences and learning services that are built from multiple tools, leveraging the usage of such tool ecosystem.

Learning theories have comprehensively determined that the educational use of *Web 2.0 Tools* is possible and that many models and frameworks that can leverage their utilization already exist. Collaborative learning through ICT has long been a topic in education and has brought attention to the fact that *Learning Orchestration (LO)* is crucial for learning to succeed. Such orchestration involves activities, individual and social processes, capabilities to adapt and be flexible, and crucial teachers' interventions, along with the corresponding design, planning, management, and correct use of learning models and frameworks. This paves the way to the creation of innovative *Learning Activities* by developing a variety of *Web 2.0 Tools*. However, because of the distributed nature of the resources created in *Web 2.0 Tools*, many challenges have been identified, such as ownership and management. This is even more critical in open and massive education, where resources are usually publicly available with thousands of interested learners attracted to the materials for learning and re-purposing intent.

In summary, the educational process can be experienced by means of tools that are accessible through a variety of devices, places, and contexts, thereby enabling new interactions among participants (Cochrame, 2012). In contrast to predominant monolithic educational systems (Dagger et 2007), many new systems and specifications have grown to enable interoperability for educational settings. Although prominent advances have been made for educational interoperability, the fundamental *Learning Orchestration* issues related to having a granular management of resources are still missing. This involves the ability to manage down to the smallest detail the features of each resource created for a Web 2.0 tool. Another issue is that the current state of the art in interoperability for educational systems is a static, with a contract-based approach. This means that interoperation with third-party tools is necessary to create custom programs to make requests and process responses from those tools. With the current semantic technologies, it is possible to create Web API descriptions that can be processed automatically by machines without human intervention and avoid the need to write specific programs to achieve interoperability. Semantic technologies foster maintainability and evolvability of Web APIs, thus reducing the inherent interoperability costs associated with creating a distributed educational environment.

Teachers, learners, and technologists are faced with the need to incorporate and use *Web 2.0 Tools* in education, thus a flexible educational environment that is capable of enacting granular orchestrated *Learning Activities* is required. Moreover, a semantic approach for interoperability ensures long-term system maintenance and evolution without the corresponding constraints, costs, and rigidity imposed by custom-made contracts.

1.2 Goals and Research Questions

The research that has been conducted for this doctoral dissertation encompasses the use of *Cloud-Based Tools* for e-Education by enabling a flexible educational environment that uses the tools' ecosystem to design new and improved educational experiences. In relation to these factors, the main research goals are as follows:

- **How can the Cloud-Based Tools be used to fulfill the current and future needs of e-Education?** A comprehensive literature survey has been conducted regarding the use of *Cloud-Based Tools* (e.g., *Web 2.0 Tools*) in education, including theories, models, new forms of learning, open and massive education, building *Learning Activities*, and tool classification. Furthermore, the research motivations, challenges, and problems are identified. Thereby, from the learner's perspective, *Cloud-Based Tools* are measured with respect to motivations, usability, usefulness, acceptance, cognitive learning strategies, and user behavior analytics.
- **How can we achieve a highly flexible education environment in terms of the inclusion and orchestration of Learning Activities that use Cloud-Based Tools?** Thereby, simplifying interoperability mechanisms in terms of authentication, communication, and processing of *Cloud-Based Tools*' Web APIs and providing interfaces for process automation, enabling what is known as a *Cloud Education Environments*.
- **What interoperability mechanisms design can enable machine-to-machine processable Cloud-Based Tools' Web APIs?** Such mechanisms do not require human intervention to write custom standard contracts between systems. Instead, based on interoperability ontologies, a pragmatic semantic description of the Web API can be created, both per tool and per application domain. With such description enable a process to automatically discover the tool's features.

1.3 Methodology and Structure

The research methodology to meet the desired research goals is structured as a three-phase process (see Figure 1.1). The first phase consists of a literature survey that encompasses the technology-enhanced learning related to the objectives of this dissertation. It begins with Chapter 1, which outlines the main motivations of the doctoral dissertation, along with the main research questions and goals.

Chapter 2 provides an overview of educational concepts and ICT, including the basic learning theories associated with the use of new and innovative tools (e.g., *Web 2.0 Tools*) for learning. It also gives a condensed view of educational models and frameworks. Then, it elaborates on specific methods and practices that are relevant to the research, such as *Computer-Supported Collaborative Learning (CSCL)*, which leads to the main components and challenges of *Learning Orchestration (LO)*, a key topic used throughout the doctoral dissertation. Furthermore, to implement LO, one instrumental component is the models, such as the *Five-Stage E-Moderation Model,* which serves as a pragmatic approach to LO. Also introduced is *Bloom's Revised Taxonomy*, which is used later to evaluate the acceptance of *Cloud-Based Tools (CBT)*. Additionally, new forms of learning and how the learning context has changed because of the emergence of new technologies and the highly connected society are briefly reviewed. *Learning Activities* and *Mediating Artifacts* are introduced and used as a foundation to model the flexible education environment. The previous topics open the path to address the use of *Web 2.0 Tools* for education and how the incorporation of such tools builds a *Cloud Education Environment (CEE)*. Moreover, a comprehensive open education overview is given, with a focus on massive and scalable education. Thereby a state-of-the-art on dropout and attrition model proving worthwhile for the evaluation phase is introduced.

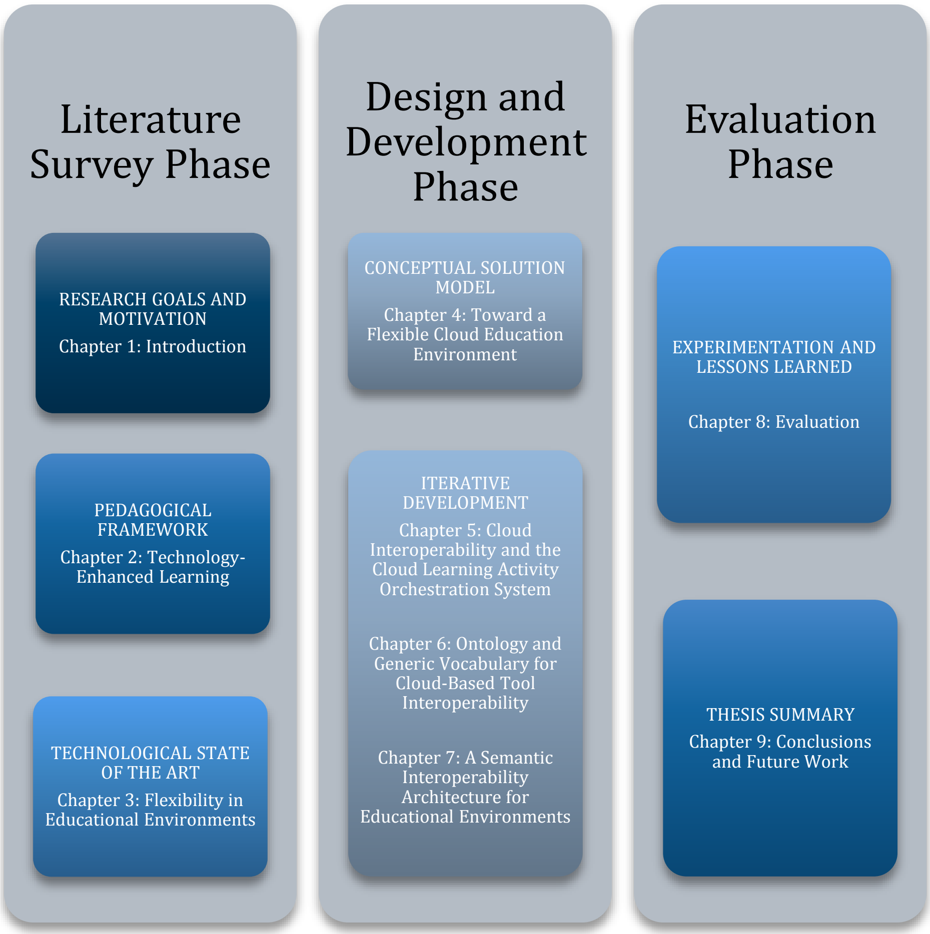


Figure 1.1 Research methodology

Chapter 3 elaborates on *Flexibility in Educational Environments (FEE),* first presenting the current pedagogical gaps and challenges identified for flexibility in educational systems. These gaps lay the foundation to focus on two areas, interoperability on the Web and interoperability in educational systems. First, interoperability on the Web is addressed though an overview of current technologies that enable a semantic approach for interoperability. Second, an overview of educational interoperability in building flexible environments with different specifications and standards is given, focusing on the IMS Learning Tools Interoperability as the state of the art. Other systems and frameworks that provide interoperability for educational settings by including third-party tools, such as *Web 2.0 Tools*, are also analyzed.

Based on the theoretical and technological background the development phase is organized in two stages. The first stage is to define a conceptual model for a flexible educational environment, and the second stage is to convert the conceptual model into a functional architecture and framework that then can be used for the evaluation of real learning scenarios. This second stage has been designed as an iterative development process, which allowed creation of the architecture and framework through a three-step process.

Chapter 4 presents a model for a *Flexible Cloud Education Environment (FCEE).* It begins by distilling the main problems and challenges for flexible educational environments. It elaborates on the application scope and use case of such an environment, setting specific objectives for the model to achieve from the infrastructure, technology, and pedagogical points of view. Thus, a *Conceptual Model for Flexible Interoperability* is designed that consists of 1) a semantic definition of the tools' Web APIs, first by elaborating an ontology for the tool domain (e.g., tool domains such as mind mapping, storytelling, etc.), to create semantic generic vocabularies for such domains, and then to detail for a given tool its Web API specifics. Furthermore, an *Interoperability Service Framework Architecture (ISFA)* is defined to process the previously introduced semantic definitions in such a way that they are machine processable without custom programming. To enable this type of interoperability with current tools' Web APIs, it is necessary to create a semantic proxy that maps these semantic definitions to real Web APIs. Finally, the *Orchestration Learning Activities System (OLAS)* is conceived to realize this interoperability technology in an educational setting.

Chapter 5 describes the first two steps of the aforementioned iterative development process. The first step of the iterative process is undertaken to build the first generation of the *Cloud Interoperability Service (CIS)*, which address issues by building services for authentication, communication, and business logic for interoperability with *Cloud-Based Tools*. Realized in a personal learning environment framework, several tools are included as advanced widgets. In addition, some initial learning analytics are introduced. The services built on this first generation prove interoperability and become the core services for further generations of the architecture. The second step for the iterative process focuses on two objectives. The initial objective is to build a *Cloud Learning Activities Orchestration (CLAO)* system that enables the realization of educational paths using *Cloud-Based Tools*. The complementary objective is to design the second-generation architecture CIS, including *Learning Orchestration* features such as learning paths with corresponding threshold controls. Furthermore, the whole CIS is modularized to be used more efficiently with third-party systems such as the CLAO. Interoperability with third-party tools is standardized through service bundles. Improved analytics are incorporated, and a data analysis dashboard is built atop a third-party service.

Based on the first two iterations of the CIS and on the model for flexible interoperability, Chapter 6 extends the development by detailing a semantic approach for describing the *Cloud-Based Tools*' Web APIs, and provides real examples of two *Application Domain Types*: mind map tools and an online document editor. Both are used later for the evaluation phase. This begins by classifying the tools' Web APIs for each application domain, enabling the construction of ontologies for interoperability. Then, the argument for the semantic technology to be used is elaborated, based on the technological background presented and the research objectives. From the ontologies *Generic Vocabularies* representing the application domain generic Web API are derived. In the final step, using the *Generic Vocabulary* as a base, the tool specifics can be semantically described, allowing the maximum possible flexibility for interoperability.

Chapter 7 presents the semantic interoperability architecture for educational environments, the third and final step of the iterative development process. This Chapter includes the semantic approach in the CIS and revamps parts of the multi-layered architecture to make it more suitable for the new framework. A detailed review of the architecture is given. This chapter looks at possibility to automatically process a Web API without custom coding contracts. To enable communication with current *Cloud-Based Tools*' Web APIs, a *Communication Process Handler (CPH)* is built that maps semantics to the Web API, a process that is simple to describe but rather difficult to implement.

Chapter 8 addresses the evaluation phase. It begins by introducing how the challenges, problems, and objectives described in the Chapter 4 have been identified and resolved. The lessons learned from the technical implementation are described, including relevant issues such as scalability tests and how the whole interoperability approach is more efficient and effective for developers compared to the current state of the art. The four complementary evaluation studies are presented, each one helps to validate the whole infrastructure and the main research questions. The evaluation encompasses a range of contexts from standard online courses to massive open online courses. It assesses learners' perceived potential acceptance of *Cloud-Based Tools* for education and further studies their perceived usability, motivations, and emotions when using them as standalone tools within a *Personal Learning Environment (PLE)* and in the CLAO using the CIS. Cognitive learning strategies are analyzed with respect to the whole proposed framework. Behavior analytics are part of the evaluations at different stages.

Finally, Chapter 9 contains a short summary and analyses the research results in light of the main research questions, and then open issues and future work on flexible interoperability in educational settings are discussed.

Part A: Literature Survey

2 TECHNOLOGY-ENHANCED LEARNING

Technology-Enhanced Learning has been a research area for many years that continues to evolve as new technologies enable new educational scenarios. This Chapter first presents in Section 2.1 an overview of educational concepts pertaining to ICT, with an special focus on *New Tools* available on the Web that allow learners and teachers to collaborate, communicate and share in new ways and these *tools* are called *Web 2.0 Tools*. The term "Web 2.0" is attributed to Tim O'Reilly (O'Reilly, 2005). It is related to a shift in a web of tools and practices towards a more participatory Web, where users have more interaction. The tools involve mechanisms for sharing, collaborating, networking, content media production, and others.

Following is an overview describing the major learning theories categories, in order to later present a structured summary of learning theories and perspectives, pedagogical approaches, and their applicability to online learning. Then the related models and frameworks that are currently being developed to realize such educational processes are introduced.

In the light of these *Web 2.0 Tools* and their enabling characteristics for a participatory Web, a review on *Computer-Supported Collaborative Learning* is presented because online learning is highly collaborative in nature, and as previously stated, the *Web 2.0 Tools* offer a large set of features to enable collaboration in educational settings. Moreover, *Learning Orchestration* challenges, characteristics and how the tools should be implemented are described, as means towards creating a flexible educational environment where management, interventions, and scaffolding are key pieces. Orchestration leads to models such as the constructivist model that is presented, as a means of leveraging the opportunities that *Web 2.0 Tools* offer, by following a 5-stage model where the learners are structurally guided through collaborative and communication tools. This 5-stage model first through a stage of access to technology, followed by stages of socializing, information exchange, knowledge construction and development of the acquired knowledge. These types of models prove to be fundamental in order to achieve highly effective online learning. Additionally, a review on the revised Bloom's taxonomy is given, which describes learning as a cognitive process and is used for further evaluation of these *Web 2.0 Tools*. To finish Section 2.1, a brief description of the impact that *Web 2.0 Tools* and technologies have had in education is presented, including how the learners has changed over time, what the current habits and behaviors are and how that might create changes in teaching practices. Section 2.1 follows the work presented in (Dillenbourg, 2002; Churches, 2008; Conole & Alevizou, 2010; Salmon, 2000) and others.

Section 2.2 introduces *Learning Activities* and presents an extensive taxonomy of *Learning Activities*. This taxonomy relates to the previously presented categories of learning theories and also elaborates on the context and tasks undertaken in a *Learning Activity*. Having this broad view of *Learning Activities* provides elements and concepts for using *Web 2.0 Tools* in education. Then, *Mediating Artifacts* are introduced using them can help teachers to more effectively and efficiently build *Learning Activities* without losing their path through the large amount of theories, concepts, approaches and technologies. *Mediating Artifacts* such as narratives, lesson plans, templates, wizards, models, and toolkits are also described.

Section 2.3 presents a typology of *Web 2.0 Tools* for education, as well as describe the *New Education Environments* that can be built by integrating such types of tools. Finally, Section 2.4 presents a historic and evolutionary perspective of open education, with a special focus on the revolution led by the rise of massive open online courses (MOOCs). This type of course is used to evaluate the research presented in this thesis, which includes a review of MOOCs' trends, problems and issues. Also, a novel model is introduced to describe the high drop-out numbers experimented in these types of courses.

This Chapter is partially based on previous publications (Hernández, Gütl, & Amado-Salvatierra, 2011; Hernández, Linares, Mikroyannidis, & Schmitz, 2013a; Hernández, Gütl, Chang, & Morales 2013d; Hernández, Gütl, & Chang, 2014d; Hernández & Gütl, 2015a; Hernández, 2015h; Hernández et al., 2015a).

2.1 Overview of Educational Concepts and ICT

As a main influencer of many changes in the social environment, ICT is defining new ways of interacting and communicating. It is beyond having access to the Internet. It has created new behaviors through the multiple devices that are available and the contexts in which the interactions occur. Young children are being (over) exposed to multimedia ICT. This exposition has resulted in many educational issues, as both devices and their corresponding applications are seen as means to create, innovate and further expand imagination, but also could have serious negative issues (Donaldson-Pressman et al., 2014). Education is not isolated to all these changes. Rather, Education it is highly affected because educational processes happen both individually and socially. Learning theories are constantly evolving to adapt to these new environments, although strong beliefs are held by different theorists and schools. This Section provides a review of learning theories in the current context of new technologies in our daily lives. The following is a short overview of the different learning theories that can be used as theoretical sources for the use *Web 2.0 Tools* in an educational environment.

Mayes and de Freitas (2004) describe three categories of learning theories:

- First, *Associative Theories* (a generalization of classical conditioning) involve the relationship between stimuli and behavior. Pavlov described that conditioned behavior is formed by pairing stimuli to conditions so that an animal gives a specific response. In other words, *"classical conditioning is a reflexive or automatic type of learning in which a stimulus acquires the capacity to evoke a response that was originally evoked by another stimulus"* (Pavlov, 1927). Aristotle defines the Law of Contiguity (the simplest form of classical conditioning) as: *"when two things commonly occur together, the appearance of one will bring the other to mind."* Simply speaking, associative theories involve learning through well-structured assignments.

- Second, *Cognitive Theories* involve using thinking to learn by understanding. The learner is viewed as being capable of processing information. Knowledge is seen as schema or symbolic mental constructions. Such theories are a response against the notion that people are programmed as animals that only respond to environmental stimuli. People are rational, and learning requires active participation, which is a consequence of thinking. The cognitive category can be divided into two specific categories: social cognitive theories and cognitive behavioral theories.
- Third, S*ituative theories* assume that learning is embedded in the context and the activity, as a social practice. They argue that learning is unintentional and happens in authentic activities, in a process called "legitimate peripheral participation" (Wenger, 1990).

Behaviorism, as part of the associative category, assumes that all behavior is caused by external stimuli. Thus, behavior can be explained without the need to consider what happens mentally or in the consciousness. However, it is commonly argued that behaviorism techniques where the student gets structured guidance towards the completion of a *Learning Activity* are not the best approach for learning with several distributed tools (Web 2.0). In contrast, *Web 2.0 Tools* can be used with an associative theory to effectively and efficiently perform learning tasks (Conole & Alevizou, 2010).

Cognitivism focuses on processing information and using it for further thinking and elaborating upon knowledge. People can select, reason about, make conjectures about and review knowledge, all through an internal thinking process that can be enhanced if it is expressed, which enables self-awareness. Therefore, the use of several *Web 2.0 Tools* can enhance this process (Chi, Pirolli, & Pitkow, 2000), as can the use of tools like mind maps, process organizers, blogs and social networks.

Constructivism considers learning to be an active process in which prior acquired knowledge and its corresponding understanding play a major role in the creation of future knowledge (Vygotsky, 1978). Cognitive constructivists focus on the learner's mind, individual understanding and intellectual processing by himself or herself. On the other side, social constructivism is situated in the context of the social learning process, such as through social networks, online comments on a resource, wikis, blogs, collaborative writing or any other of collaborative social activity. Recent studies (Roblyer et al., 2010) have shed light on how social networks can be useful learning environments and tools. Such findings suggest that *"teacher self-disclosure may lead students to higher levels of anticipated motivation and affective learning and lend to a more comfortable classroom climate"* (Mazer et al., 2007).

Cultural dimensions of learning must also be taken into account, as human beings have built their knowledge from private thinking to public sharing (DiSessa, 2001). Culture implies using tools and artifacts, and thus contributes to creating a space where knowledge can then be transferred by reasoning and problem solving.

Furthermore, effective learning behavior can be supported by using metacognition (experience rehearsal) and *mediating artifacts* (such as tools for reflection), where both teachers and learners can work together to build knowledge (Conole et al., 2008). Subsequently, communities of practice (Wenger, 1998) represent the concept of people who engage in a collective learning process and a well-defined and focused domain, and have shared goals and resources, where each learner has his or her own (learning) path, which is an individual and personal journey. As we examine later, the *Mediating Artifacts* play an important role in setting a learning experience because it moves the importance of the tool towards the knowledge acquired by performing a given action.

In the following Sections, relevant educational, ICT-related topics are introduced for this research, such as *Computer-Supported Collaborative Learning*, *Learning Orchestration*, the 5-stage e-moderation model and the revised Bloom's taxonomy.

2.1.1 Computer-Supported Collaborative Learning

Collaborative Learning is an umbrella term for a diversity of educational methods and practices where the key concept for positive learning outcomes is the interactions between students. In *Computer-Supported Collaborative Learning (CSCL)*, the reference to 'computer-supported' is about how to connect geographically distributed students and the types of technologies that help to build similar in-person interactions. Importantly, collaborative learning results from the cooperative construction of shared knowledge on the topic. Also, social interactions that are meant to be productive can be designed in special environments. Therefore, *Social Networking Sites (SNS)* are highly relevant as CSCL environments: "*collaborative activities are becoming integrated within comprehensive environments that include non-collaborative activities stretching over the digital and physical spaces and in which the teacher orchestrates multiple activities with multiple tools.*" (Dillenbourg et al., 2009).

Following is a summary of key CSCL concepts (Dillenbourg et al., 2009) that serves as the basis to further develop the research topics on this doctoral dissertation:

- The more interaction there is, the less individualization happens.
- There are two different perspectives: from the instructional and educational psychology point of view, educational activities that foster social interactions are merely methods to encourage and nurture individual's knowledge construction. From the socio-cultural perspective, social interaction is what matters most in cognition and is therefore the objective of learning (Wegerif, 2007).
- The limits between formal and informal instruction have vanished.
- Collaboration per se does not bring learning outcomes, rather results are tied to actual engagement in productive interactions.
- New technology, media and devices are not effective by themselves. One traditional example is the use of online forums in collaborative education: effective learning results have been obtained in specific cases (Schellens & Valcke, 2005). In contrast, other publications have reported very low communication, leading to no learning at all (Hammond, 1999; Goodyear et al., 2004). Following that example, SNS support similar communication

mechanisms to those that the online forum tools provide, but with a highly influential social context, where interactions are usually more frequent. The high intrinsic social motivation from continuously reviewing "*what is new on my social networks"* provides possible paths to increase participation.

- Shared understanding and knowledge are constructed collaboratively (Roschelle & Teasley, 1995).
- A similarity to face-to-face learning does not means that better learning is occurring. This (Haake, 2006) raises a fundamental question on "how technology can fulfill collaborative functionalities that are not available in face-to-face situations." An opposite view is possible, in that SNS covers many social interactions that were not available in traditional social settings.
- *Learning Activities* shape social interactions. If the learner internalizes the social interactions, they also shape the way the learner reasons about the domain. (Kuutti & Kaptelinin, 1997).
- Organizing the communication may not be effective (Dillenbourg, 2002).

2.1.2 Learning Orchestration

In the context of CSCL, several technologies can influence and change current educational boundaries, and sometimes vanish them. Nowadays, collaborative *Learning Activities* are carried out on wider learning environments that consist of multiple types of devices and occur at different social levels, such as in groups, class or public, and throughout different contexts, such as at home, in class, at work, at laboratories or through field trips. *Learning Orchestration (LO)* is defined in (Dillenbourg, 2009) as the process of productively coordinating interventions from learners across multiple *Learning Activities (LA)* (Beetham, 2013). The process of LO is based on teachers' functions, such as defining activities and evaluation rubrics, monitoring individual or group activities and adapting deadlines and workload. Dillenbourg et al. (2009) describes the challenges of orchestration as the following:

- Orchestrating activities at different social planes: This involves integrating *Learning Activities* at different social planes as a workflow, which is a flow of data between activities. The teacher should devote attention to monitoring interactions and adapting workload.
- Orchestrating scaffolds at different social planes: This is done to develop synergies for an effective strategy, such as by coordinating and supporting interventions, which come from many sources like peers, the teacher, the software or the learning materials.
- Orchestrating self-regulation and external regulation: This refers to the interplay between internal cognitive processes and external instructional collaboration scripts. The main idea is that learning processes and outcomes depend on the available regulatory information.
- Orchestrating individual motivation and social processes: Successful engagement fosters comfort in the learning process, in order to take risks and increase participation. Any learner can also play the role on motivation regulation. Motivation can also increase at the social level through collaborative knowledge construction.
- Orchestration requires adaptivity or flexibility: This refers to the level of external scaffolding required for a given learning process, which tends to decrease as the

student gains knowledge. This implies that educational environments should provide real-time information about interactions along with diagnostics, and systems should provide enough flexibility to adapt to current learning and individual conditions.

- The teacher conducts the orchestration: Through CSCL, knowledge construction is expected to be highly leveraged by peer interactions, thus transforming the teacher's role to orchestrating the learning experience. Therefore, monitoring, control and the flexibility to adapt the environment to the learning process are key issues.

A comprehensive review has been done by (Prieto, 2011) about *Learning Orchestration* in TEL and is used to extend the correspondent description of orchestration towards guiding this research. First are listed the main aspects found the by the authors, second the focus shifts to how it should be done orchestration in TEL.

Orchestration in TEL main aspects (Prieto et al., 2011):

- Design and planning: A key component of orchestration is designing and planning, which includes the traditional instructional design and instructional planning, and also encompasses the technological tools that move the learning process towards the desired learning outcomes. Therefore, it is fundamental to have a means of educational scripting, which refers to workflow control in an online environment.
- Regulation and management: A well-orchestrated learning environment provides a unified learning experience. Regulation—either external or internal—plays an important role, which includes time management, workflow, group management and individual paths. Authors argue that it can be greatly enhanced through technological means.
- Adaptation, flexibility and intervention: These include proper teacher participation to increase motivation, in addition to the act of adapting the original plans and design towards what the individual and group need. Therefore, flexible educational environments are required.
- Awareness and assessment: The concept of awareness is related to interventions, and is especially crucial in a distributed educational environment, as awareness enhances overall orchestration. Also, the use of assessments, both formative and summative, can provide helpful insights to learners about their own progress, and help teachers to adapt their flexible educational environment.
- Roles of the teacher and other actors: This refers to the teacher's new role of being a guide and then moving towards more active learner participation, like a learning-driven orchestration, with scenarios such that the main learning goals are set by the teacher and the learning tasks and coordination are managed by students.

Until now, the definition of *Learning Orchestration* has been discussed. In the following list, it is elaborated after Prieto et al. (2011) on how LO should be implemented in TEL:

- Pragmatism and practice: This means moving from conceptual and research-focused environments to wider implementation, by putting a flexible environment into place, which is required support the main aspects identified for LO.

- Alignment and synergy: This represents the coordination of elements, such as *Learning Activities*, social contexts, tools and scaffolding elements. The teacher must have the flexibility to adapt these elements through changing conditions, in order to obtain the learning outcomes.
- Models and theories: This represents the increased complexity related to orchestration, which requires well-designed *Learning Activities* based on robust theories and models. For that reason, the following describes a constructivist 5-stage model, which was adapted to TEL to orchestrate the learning process. Also, Section 2.2 describes more about designing *Learning Activities* with a more pragmatic approach.

2.1.3 5-stage e-Moderation Model

This Section is partially based on previous work published by Hernández, Guetl and Amado-Salvatierra (2011).

The model belongs to the situative learning theories category, where learning happens in real-life scenarios and has a strong social component. The 5-stage e-moderation model's underlying assumption is that *"learning includes an intricate and complex interaction between neural, cognitive, motivational, affective and social processes"* (Azebedo, 2002). Another assumption of the model is that *"participants learn about the use of computer networking along with learning about the topic"* (Wenger, 2002). The results show that it is more important to establish a proper environment and rules than to learn about the tool and networking interaction. Additionally, learning tools and how to network have changed tremendously due the familiarity of the learners with *Social Network Sites (SNS)* and many *Web 2.0 Tools*.

The model also scaffolds individual development. Scaffolding is a process that moves from directed instruction to a constructivist learning approach, from short-term to long-term learning necessities and from immediate learning to more holistic learning (McNaught, 2003; Wenger, 2002). The e-moderation proposed by Salmon (2000) consists of a 5-stage model (see Figure 2.1), where the learner goes stage by stage and learns through a constructivist approach, with specific expected participation and interaction, along with a given amount of interactivity through the conferencing system—traditionally an online forum within an online course, although many other sources are now possible. Brief descriptions of the five stages are given below, with some considerations within the context of SNS.

Stage One: Access and motivation. The main objective of the stage is to help the student become familiar with the learning environment, (e.g., to be able to login and use the system). Strong motivation and creativity are needed to overcome any type of technical problems. For *Social Networking Sites*, this stage has been proven to be less important, and the role of the e-moderator at this stage is further transformed, in order to indicate how the environment is used. This is in contrast to first-time online learners, who also require a considerable amount of technical support.

Stage Two: Online socialization*.* When using learning technologies for online conferencing without solid socialization, the scaffolding may be inadequate (Brown et al., 1999). For an SNS, the experience is all about socialization. Socialization is important since, for people working together, the two motivational factors are self-interest (due to extrinsic factors) and common interest (which requires trust and mutual respect). The concept of this stage is to create opportunities for socialization.

Stage Three: Information exchange. This stage is about sharing information, exploring, discovering and ensuring that each participant has a given role and is participating. In this stage, the e-moderator helps to manage the volume of information. *Badges, Stars* and *Likes* are also encouraging for learners at this stage.

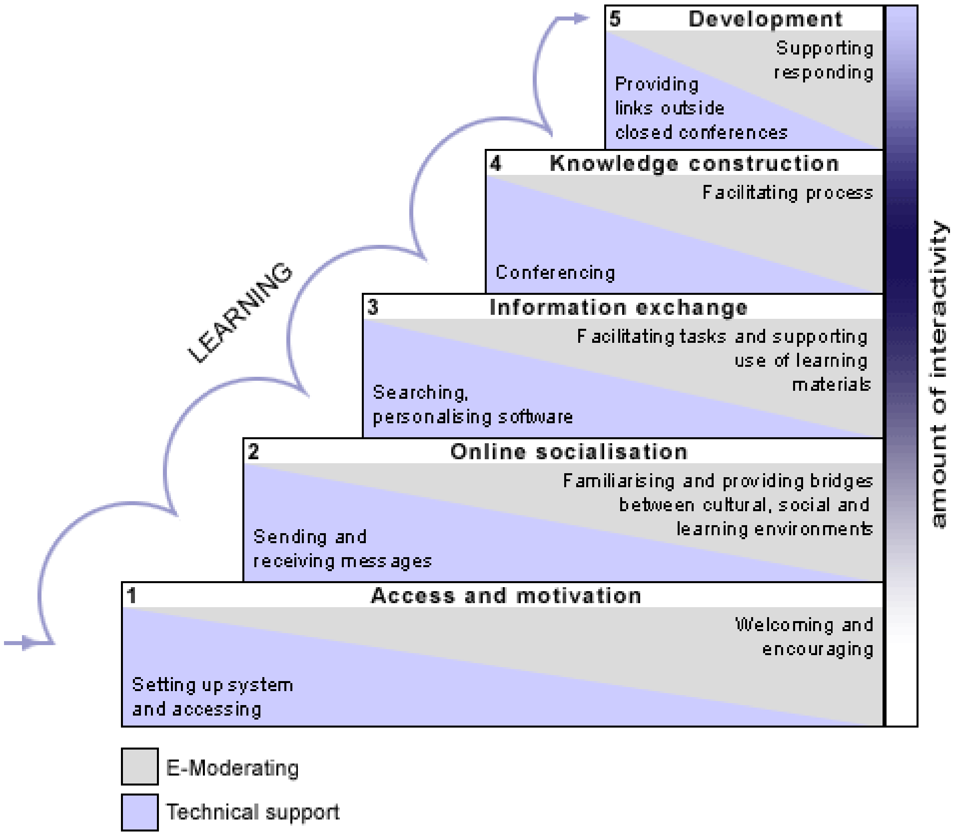


Figure 2.1 Five-stage model for e-moderation diagram taken from (Salmon, 2000)

Stage Four: Knowledge construction. The participants begin more exposed and participative interactions among themselves. Conferences and communications are unfolded and expanded, and some participants engage deeply in the conversations. Learning at this stage is more of a creative cognitive process of idea sharing, criticizing, expanding and reshaping through peer discussion (Rowntree, 1995). This stage becomes a communal constructivism by building knowledge and understanding through groups, experiences, etc. (Salmon, 2000). the author elaborate on this: "*Knowledge construction occurs when participants explore issues, take positions, discuss their positions in an argumentative format and reflect on and reevaluate their positions*" (Jonassen, 1995).

Stage Five: Development. "*Metacognition promotes integration and application of learning experiences*" (Salmon, 2000). This stage is about critical thinking, reflection and challenging thoughts. A constructivist learning approach allows learners to review their own ways of building their knowledge and thinking (Biggs, 1999). Reflection is a key concept at this stage, which is one of the most productive stages of the entire process. The participation of the e-moderator (being the course teacher) is fundamental for guiding learners through each stage.

2.1.4 Bloom's Revised Taxonomy

This Section is partially based on previous work published in Hernández, Morales, Medina & Barchino (2015a).

The Bloom's taxonomy is a framework for classifying statements of what students are expected to learn as a result of instruction. It classifies cognitive domain operations in six levels through a hierarchy, and it assumes that learners must master the lowest levels of the hierarchy before they can advance to the next level. Anderson and Krathwohl (1999) made some changes to the original taxonomy (Bloom, 1956). One of the key aspects of this review is the use of verbs, rather than nouns, for each category, and other aspect is changing the sequence of these within the taxonomy. They are organized in increasing order from lower-order thinking skills (LOTS) to higher-order thinking skills (HOTS).

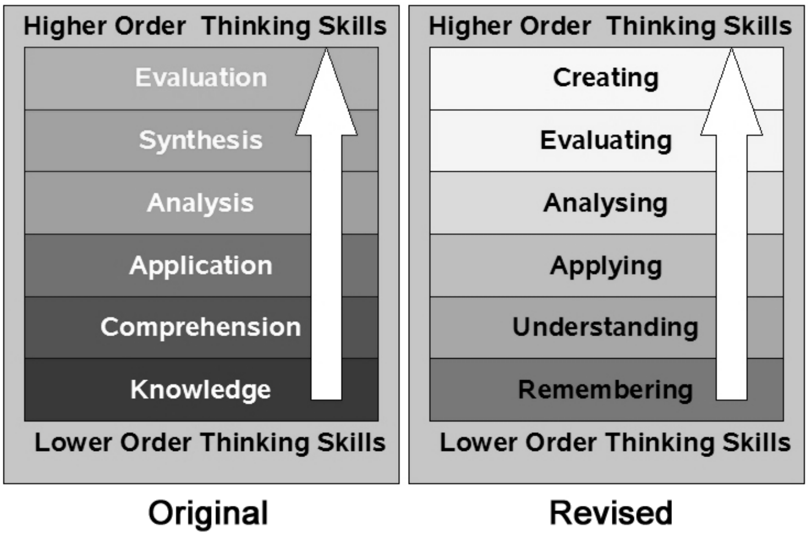


Figure 2.2 Representation of the new verbage associated with Bloom's Taxonomy taken from (Anderson & Krathwohl, 1999)

The new terms in the revised taxonomy are enumerated from 1 to 6 and presented in Figure 2.2: 1) Remembering, according to Anderson and Krathwohl (1999), is defined as retrieving, recalling, or recognizing knowledge from memory. Is when memory is used to produce definitions, facts, or lists or when it is used to recite or retrieve material. Remembering or recalling is reinforced by application in higher-level activities. 2) Understanding builds relationships and links knowledge. Students understand the processes and concepts and are able to explain or describe these. They can construct meaning from oral, written, and graphic messages through interpreting, exemplifying, classifying, summarizing, inferring, comparing, and explaining. 3) Applying is defined (Carrington, 2013) as carrying out or using a procedure through executing or implementing. Applying is related and refers to situations where learned material is used through products, like models, presentation, interviews, and simulations. 4) Analyzing is defined (Carrington,

2013) as breaking material or concepts into parts, determining how the parts relate or interrelate to one another or to an overall structure or purpose. Mental actions include differentiating, organizing, and attributing as well as being able to distinguish between components. 5) Evaluating means making judgments based on criteria and standards through checking and critiquing. Finally, 6) Carrington (2013) defines creating as putting the elements together to form a coherent or functional whole. It is related to reorganizing elements into a new pattern or structure through generating, planning, or producing.

2.1.5 New Forms on Learning and Contexts

The following Section briefly covers new forms of learning, new learners' characteristics and the changes in the teacher's role, as well as new forms of communication, collaboration, creativity, co-creation and richer contextualization. New *Web 2.0 Tools* are already available to support analyzing, interrogations and research. Thus, such tools may promote inquiry if they are used correctly.

The evolution of *Social Networking Sites* SNS and their leaders, such as Facebook, YouTube, Twitter and others, are well-known phenomena (Boyd, 2008; Hew, 2011). They offer a set of capabilities that can be very useful for online learning, although their full potentials do not only lie on the application side. Although it is very important and is actually what enables communication channels, the power of SNS lies in that they have become part of the social interactions for a large population: *"Social scientists have attempted to identify and recognize the use of Facebook by the younger generation to understand how this generation interacts online, communicates and identifies itself as a member of an online community."* (Dong, 2008). For example, Facebook is seen as a large commons for society, where people can stay connected in many matters including learning, while at the same time providing a comfortable environment for richer and more in-depth interpersonal connections and growth (Schwartz, 2009). Additionally, Web 2.0 social technologies such as blogs, wikis and others have proven to enable truly collaborative learning experiences (Pardo and Kloos, 2009).

Collaboration, co-creation and joint creativity are common in distributed environments such as the ones provided by *Web 2.0 Tools*. Participating in communities and having an audience can be very motivational, as participants can demonstrate their personal knowledge and obtain feedback. Both teachers and learners can collaborate and contribute in an educational setting. Consequently, such networked environments need to be carefully crafted in order for the collaborative engagement to flourish, through the use of heterarchical structures, flexible and exchangeable roles, effective skills for co-creation (such as not overwriting someone else's work), collective responsibility and even pride over the work done (Burgess, 2006). A richer context means that learners are allowed the freedom to organize their learning environment as required to their individual preferences, even the curriculum's flexibility is foreseen. All of these have affective and motivational benefits.

New learners' characteristics bring different technology usage patterns (2008 ECAR Survey about students' computer use for academics, and the OECE Report of Millennium Learners, OECD 2009). These survey and report show that libraries and VLEs comprise the largest ICT use for academia, although similar usage is seen in SNS, instant messaging, music and videos online. Millennial Learners are transforming into content producers (Hylén, 2007) who use media such as blogs, wikis and video and photo websites. However, the use of role-playing games (MMORPGS) and 3-D worlds such as Second Life is significantly less frequent (Pedró, 2009) despite their good expectations for educational purposes (Chittaro & Ranon, 2007).

A final word on the changing roles of teaching and teachers: Siemens (2009) defined the following teacher roles for networked learning environments: amplifying, curating, aggregating, filtering, modeling, way-finding, socially-driven sense-making and persistent presence. However, scaling these require strategic coordination and development support in designing, supporting and assessing learning. In contrast, problems such as the dominant culture in teaching, the lack of vision and experience and offering the appropriate set of incentives for teachers may reduce the corresponding adoption of *Web 2.0 Tools* (CERI, 2008).

2.2 Learning Activities and Mediating Artifacts

A *Learning Activity* is a set of tasks that achieve the desired learning outcomes through their completion (Beetham, 2013). The *Learning Activities* have three components (Conole, 2007): 1) *The Context:* the intended learning outcomes and the environment where the *Learning Activity* takes place (the following are part of the context: aims, pre-requisites, skills, subject, environment, time, difficulty). 2) *Educational Approaches*: Associative, cognitive, or situative (Mayes & De Freitas, 2004). 3) *Task Undertaken*: This includes the type, technique, interaction, roles, resources, tools, assessment and sequence.

A published *Learning Activity* taxonomy (Conole, 2007), serves as a guide for practitioners in developing *Learning Activities*. However, its use is relatively complex, in part because of the decision-making process regarding which theories and tools to use. In practice, teachers use different approaches when creating *Learning Activities*, and it is common to use *Mediating Artifacts* such as case studies, guidelines or narratives, or more abstract good practices such as models and patterns. A brief description of those *Mediating Artifacts* follows (Conole, 2007):

- Narratives and case studies: These are quite known for education and are usually contextually rich, which is an advantage but also represents a difficulty in trying to adopt them in a different context. Dialogic approaches tend to enable the discussion of ideas and peer dialogue.
- Lesson plans: These help to align theory and practice, and are usually helpful for *Blended Learning* (where education involves the use of both technology and face-to-face instruction). (Duncan, 2003).
- Templates and wizards: The typical example of this is a system such as the *Learning Activity* Management System (LAMS) that serves to guide teachers or instructional designers in building *Learning Activities*. The entire environment

is organized so that the user can pick (and mix) tools and fill templates. This is the category of how-to, walk-through wizards.

- Toolkits: These basically provide a theoretical background for decision making. Toolkits are a structured resource for organizing the *Learning Activity*. This includes detailed information layers that practitioners can follow, in a logically organized structure. The DialogPlus *Learning Activity* design toolkit is one example (Conole, 2005).
- Models and patterns: The 5-stage e-moderation model is an example of an abstract representation that the teacher can follow. Another example is Kolb's learning cycle (Kolb, 1984), in which learning is presented through an approach with a four-stage cycle (experience, reflection, abstraction and experimentation).

During the course of this thesis, some of these *Learning Activities* and approaches to evaluation are adopted, while at the same time constructing some *Mediating Artifacts* to enable the quick and more comprehensive deployment of a *Learning Activity* within an ecosystem of *Web 2.0 Tools*.

2.3 Web 2.0 Tools and Cloud Education Environments

As has already been described in the previous Sections of this Chapter, there is a large potential for the use of *Web 2.0 Tools* for education. Many publications and organizations have done classifications, such as Discovery Education (Education, 2014), which classifies tools as for: presentation, video, mobile, and community. Edudemic lists the top-100 tools suggested by teachers (Edudemic, 2011). A more comprehensive typology of *Web 2.0 Tools* by (Crook, 2008) is presented below:

- *Media Sharing*, with sites such as YouTube and apps such as Instagram, that involve creating and exchanging media with peers and audiences.
- *Media Manipulation*, such as mind maps, diagrams, online presentations and mashups.
- *Conversational Arenas*, which include instant messaging and chat. Nowadays, this is also included in a multitude of sites, especially in SNS such as Twitter, Facebook and Google Plus.
- *Online Games and Virtual Worlds* like Second Life, Openwonderland and gaming consoles that support online collaboration.
- *Social Networking*, both using common SNS and by creating private communities with standalone technologies.
- *Blogging*, an Internet-based journaling approach that has become an Internet phenomenon.
- *Social Bookmarking*, with a centralized and often shared space for bookmarks.
- *Recommender Systems*, which aggregate and *tag* user preferences in some specific domain. One example is the widely known Amazon recommender system (Amazon, 1996).
- *Collaborative Writing*, which was initially popular through the use of wikis and now with more robust tools, such as Google Drive document editor (Drive, 2014), among others.
- *Syndication*, by subscribing to feeds. Many apps have emerged for specific domains with already syndicated and sometimes curated content.

CLOUD BASED TOOLS AND CLOUD EDUCATION ENVIRONMENT

This Section is partially based on previous work published by Hernández, Linares, Mikroyannidis and Schmitz (2013a).

Cloud Computing is a major trend nowadays, with recent studies positioning it as one of the short-term adoption technologies for education (STEM, 2012). Cloud computing is essentially about expandable and on-demand services, tools and contents that are served to users via the Internet from specialized data centers. Cloud computing resources support virtualization, grow on-demand and collaboration. *Cloud-Based Tools (CBTs)* (known as *Web 2.0 Tools*, in this thesis the term *Cloud-Based Tools* and *Web 2.0 Tools* are used interchangeably) for collaboration have the potential to engage students, by allowing them to interact and brainstorm solutions, elaborate reports, and create conceptual designs. This approach has the potential to enable and facilitate both formal and informal learning. It also promotes the openness, sharing and reusability of learning resources on the web (Mikroyannidis, 2012). *Cloud-Based Tools (CBTs)* can interoperate with other systems, offering the possibility to orchestrate services that previously were seen as standalone *Web 2.0 Tools* and thus to create an ecosystem for a comprehensive and integrated learning experience.

Cloud-Based Tools are constantly evolving. Many of the so-called cloud services, have added an important element to the cloud landscape, which is interoperability features, opening their Web APIs to allow consumers to use their services in creative and innovative ways. This opens the possibility to create orchestrated services that provide learning experiences, which were not possible before. This also changes the paradigm from a monolithic architectural (Dagger et al., 2007) approach of education environments to a flexible, distributed and heterogeneous architectural setting for education environments, which is the aim of cloud education (learning and teaching) environments. This also maximizes innovation possibilities, allowing interoperability of the best and most appropriate cloud services based on learning needs, freeing up from a vendor specific approaches and limits, transforming the *Cloud Education Environment* (CEE) (Mikroyannidis, 2012) into a digital educational ecosystem of services and resources available for the practitioners, in contrast to a large amount of not interoperable software services that are difficult to manage and organize for a learning setting.

The CEE augments any educational setting or system by including the cloud as an ecosystem of applications, services, content providers, and computing power that does not belong to a particular educational institution. Therefore, this extends the range of possibilities to include many types of CBTs that can be used for educational objectives, some of them designed with that purpose and many others that might not be originally conceived for that but that fit well for certain learning scenarios. The CEE has the potential to enable new learning scenarios while simultaneously fostering educational actions that bring new pathways for learning (Malik, 2009; Mikroyannidis, 2012). On the other hand, CBTs offer a diversity of rich applications, features, and scenarios that can be used for education. Student-centered learning can be supported in the cloud (Chang & Guetl, 2010), as it promotes collaboration among students and instructors through setting a place to meet, interact, and conduct online *Learning Activities* using shared resources and processes (Berenfeld & Yazijian, 2010), while the cloud has been used to enhance higher education teaching and learning (Fox, 2009).

2.4 Open Education

This Section mainly republishes the work by Hernández, Gütl, Chang and Morales (2013d).

Over the last decades open licensing, open source and open content had made significant impact on educational approaches and settings (Martin, 2012). The open environment has encouraged knowledge sharing and knowledge exchange especially in the fields of science and education. Important factors reshaping modern education were influenced by Web 2.0 developments such as social media and web-based services (Gonick, 2013) as well as the use of cloud-based approaches (Gütl & Chang, 2008).

To capitalize on established technologies and tools, recent e-learning movement has progressed towards flexibility in accessing online courses and the use of collaborative tools. Restrictions and regulations on the participation of online learning have been lifted and MOOCs are now available to the entire population. The earliest record of the availability of an open course was in 2007, a course on 'Introduction to Open Education' by David Wiley at Utah State University (Downes, 2005; Vardi, 2012). Some 50 participants from eight countries participated in this course. At the same time Alec Couros offered an open course about 'Social Media and Open Education' (Chang & Gütl, 2010). In 2008 George Siemens and Stephen Downes developed a course about 'Connectivism and Connective Knowledge' (Downes, 2005; Rodriguez, 2013). This course was formally offered to 25 students from University of Manitoba and informally to some 2,200 students from around the world. The record number of students registered in the course led to the term *MOOC* (Daniel, 2012; Downes, 2005; Martin, 2012). About the same time when Wiley, Couros, Siemens and Downes offered their open online courses, Galileo University in Guatemala had also offered open online courses in 2005, 2006 and 2007. These courses attracted over 800, 1000 and 2000 students respectively (Hernández, Gütl, Amado-Salvatierra & Al-Smadi, 2012).

The early open learning courses not only enabled students from around the world to experience learning in a mass educational set-up, they also uncovered problems such as very high drop-out rates, student anonymity, insufficient support and issues with assessment and moderation (Lane, 2012). Regardless of these issues, MOOCs have continued to raise great interests and many well-known institutions have started to offer open online courses to the world. Motivations to join the MOOC movement varied, some institutions taking the opportunity to reach a greater learning group, yet others taking the advantage of offering courses at lower costs (Tseng, 2010). For students, they can sign up for MOOCs at no fees and if the course is completed successfully, they can earn credit for the course or if they choose a non-credit course, they can still participate in a variety of informal ways.

The number of students participating in some of the MOOCs is massive. For example, a course on 'Artificial Intelligence' from Stanford University attracted 58,000 students. MIT, Harvard and Berkeley have all joined forces and founded edX (edX, 2014). Other start-up companies such as Udacity and Coursera have also emerged, and these online education companies offer hundreds of courses and having hundreds of thousands of registrations (Daniel, 2012; Martin, 2012; Pisutova, 2012).

A description of a MOOC may include the following aspects: the course is open and free of charge. The course is participatory, contributions are shared by the learning community, and the course content, communication and collaboration are distributed over various resources and services (Martin, 2012). Different objectives, approaches and concepts have resulted in a variety of MOOCs and these MOOCs can be classified into two classes: *cMOOCs* and *xMOOCs*. cMOOCs are based on 'connectivism and networking' and the work of Siemens and Downes (Rodriguez, 2013) followed this approach (Rodriguez, 2013). *xMOOCs* are based on the 'behaviorist' approach and Stanford and edX followed this approach (Pisutova, 2012). Kay and Loverock (2008) extended this into three classes, with MOOCs having all three elements of network-based, task-based and content-based with the dominant element defining the class of the MOOC.

The *Network-Based MOOCs* are equivalent to cMOOCs where the main goal is conversation and socially contracted knowledge, and traditional assessment is difficult to apply (Lane, 2012). The main pedagogy of *Network-Based* is connectivism. Lane (2012) further defines *Content-Based MOOCs* as the equivalent to xMOOCS where the main goal is content acquisition, and traditional assessment (formative and summative) is more likely to apply. Course completion is more important with content-based MOOC than is networking. The last element is the *task-based MOOCs* which focus on the tasks the students have to complete to acquire the necessary skills. Learners are asked to complete certain types of work. For example a certain number and variety of assignments, and the skills required to accomplish the tasks. This class of MOOCs can be seen as a mix of instructivism and constructivism. Traditional assessment appears to be challenging.

A completely different view to describe and analyze MOOCs has also been introduced by Davis (1989). They describe a MOOC as a complex system which is characterized by self-organization, openness to information flow, turbulences and changes, and flexibility of interconnectedness of the various parts of the system. Over the last decades, emerging technologies and approaches have promised to revolutionize and also improve learning and teaching, however, evidence of progress and improvements in terms of effectiveness and efficiency is difficult to find (Gonick, 2013). MOOC is the current hottest topic of discussion in higher education and if the usual hype cycle is to go by, MOOC may end up in disappointment. There are also ongoing debates of whether this movement is going to be the next big bubble or a new development changing the way of learning and reshaping universities and other educational institutions (Buyya et al., 2008; Fini, 2009; Hernández, Pardo & Delgado, 2007; Tseng, 2010).

A brief look at the short history of MOOCs uncovers ideals and realities. Having a look at the early cMOOC, the idea of the concept was to learn about connectivism by exploring theory and experience (Rodriguez, 2013). The reality was described as mixed, in part *"positive and stimulating, and in part frustrating and negative"* (Lane, 2012). This is supported by the evaluation results on the key characteristics of connectionist-based course on autonomy, diversity, openness, and connectedness and interactivity. The majority of respondents rated the importance of *Learner Autonomy* very high, however, some learners also indicated a lack of confidence and preferred structures as well as guidance and assessment. The diversity of participants from various countries and different backgrounds and interests was mainly perceived as positive however an issue mentioned by the participants was the language barriers. The interpretation of openness by the respondents supports the concepts of sharing information, ideas and opinions freely, however, only a small fraction (14%) of learners participated actively. In terms of *Connectedness and Interactivity*, participants had the freedom to choose from a range of technologies, however on the negative side lack of clarity and moderation were emphasized (Downes, 2005; Lane, 2012). According to Rodriguez (Rodriguez, 2013) very little research has been done in xMOOCs and the following summarizes some findings on xMOOCs: *"x-MOOCs have shown impressive technology deployment, rapid course production, huge list of high standard partners and potentially disruptive and interesting certification alternatives [...]. They rely primarily on information transmission, computer marked assignments and peer assessment."* Siemens points out in his blog that *"cMOOCs focus on knowledge creation and generation whereas xMOOCs focus on knowledge duplication [...] learners from different parts of the world who find xMOOCs extremely beneficial as they don't have access to learning materials of that quality at their institutions"* (Pisutova, 2012).

On the positive side, MOOCs can make learning accessible regardless of social and cultural background. With the open environment, MOOCs bring together a diverse group of learners enabling them to converse, collaborate and learn autonomously. This type of learning compels student to learn in a self-regulated way and may choose tools of their choice. For institutions MOOCs might be a vehicle to reach a wider community and act as a strategic weapon for monetary advantages (Daniel, 2012; Hernández, Pardo & Delgado, 2007; Mackness, 2010; McAuley, 2010; Pisutova, 2012; Rodriguez, 2013).

On the negative side, MOOCs also raise a lot of issues. The MOOC approach is faced with a very high dropout rate and feeling of isolation and disconnect. Although active learning and control in learning are positively received, learners have also asked for guidance and pre-selection and filtering of their peers' contribution. One main criticism of MOOCs is the insufficient pedagogical approaches that are applied to design and run MOOCs. It is also highlighted that majority of participants are not prepared to control their own learning in a less structured environment. As assessments are mainly focused on computer-marked automated assessment and peer assessment, learners require sufficient guidance with these types of assessment. There are also issues with the certification and accreditation of completed MOOCs. On a bigger scale, business plans and the sustainability of MOOCs are ongoing issues. There is also the fear that the introduction of MOOCs is another way to reduce cost of tertiary education (Buyya et al., 2008; Daniel, 2012; Downes, 2005; Tseng, 2010).

Overall, MOOCs as an evolving phenomena have raised several issues and there is also room for research in several dimensions: on the educational side this would include research on appropriate pedagogical approaches and affective aspects. On the technological end, the research is concerned with the effective and efficient support of learning in an open and self-guided learning. On the administrative side, this include business models for sustainability and certification and accreditation solutions which benefit learners and educational institutions.

DROP-OUT AND AN ATTRITION MODEL

This Section mainly republishes the work by Gütl, Hernández, Chang and Morales (2014).

Dropout from different kinds of education, as higher education and lifelong learning is a scope of study trying to identify the nature of the dropout process. Initial research in MOOCs had focused on factors affecting drop-out. Given the rise in online open education that is scalable, more research is required to support and encourage persistence. Yang et al. (2013) emphasized that unlike research on attrition of other forms of online learning, MOOCs raises new research questions. This is supported with the freedom of the learners to choose what, when, where and how to learn. Learners can control their own learning, and in many settings they can personalize and select their own learning content and choose the preferred (learning) tools. Usually, there are no pre-requisites or financial burden to enroll in open courses. The entry barrier is low and no penalty is applied if one leaves the environment. The high drop-out rates reported by various sources is emphasized as one of the major drawbacks of this learning context. This finding is both disillusioning and misleading. A study by Jordan (Jordan, 2014) reported a median value of 6.5% completion rate across 39 courses with a broad range between 0.9% and 36.1%. The study also revealed a decreasing completion rate for increasing course length.

Some selected research on attrition in the context of open learning may shade some light on the reasons for the very high drop-out rates. Clow (2013) proposed the *Funnel of Participation Model* which is inspired by the 'marketing funnel' to model the process of a customer from taking notice to buying an asset. This happens usually in four phases: awareness, interest, desire, and action. In the context of MOOCs, the analogies to these phases are: *Awareness*, *Registration*, *Activity*, and *Progress*. Each phase is characterized with a large fraction of drop-out. The 'funnel' notion can be applied for density of contributions in the activity phase.

The effect can be illustrated by the actual attrition numbers of the MOOC. For example, some 5000 students were enrolled in the "Introduction to Infographics and Data Visualization" offered by the College of Communications at the University of Texas, Austin (Liu, 2013). Of the 5000 students, 44% interacted in the forum, 33% completed the first quiz, 26% the second quiz, and finally only 0.4% completed the course successfully. In particular in the context of the activity phase, researchers such as Balakrishnan (2013) and Yang et al. (2013) have looked into learners' interaction pattern on *Learning Activities* to predict drop-out or persistence with the learning tasks.

A completely different approach was followed by Adamopoulos (2013). Based on the grounded theory method, a content analysis (text and opinion mining) of user-generated online reviews has been performed. The proposed model for online course retention suggests the following categories: student course evaluation, course characteristics, university characteristics, platform characteristics, and student characteristics.

Kizilcec, Piech and Schneider (2013) criticized the monolithic view of disengagement in MOOC settings regarding attrition research and discussion. Based on the engagement in terms of interaction pattern, they have suggested a classification method and identified four classes of engagement: *Completing Class* groups learners who complete a majority of activities, and eventually finish the course. *Disengaging Class* describes patterns of students who take assignments in the beginning but stop over time and completely left the course or still consumed some content without taking further assignments. The *Auditing Class* is characterized by students taking infrequent assessments but engages by consuming learning content. The *Sampling Class* includes learners who selectively consume content.

Gütl et al. (2014) have conducted a study to understand reasons and factors for leaving the MOOC in an "Introduction to e-Learning" course offered by Galileo University in Guatemala. One thousand six hundred and eighty (1680) students enrolled in the course and only 143 (8.5%) participants completed the course, and a total of 1537 (91.5%) left the course. A questionnaire was sent out to the group of students who did not finish the course and 134 students completed the questionnaire. The respondents were 69 (51.49%) male and 56 (48.51%) female. As shown in Figure 2.3, a variety of reasons were uncovered in order to understand the motivation to enroll in the MOOC.

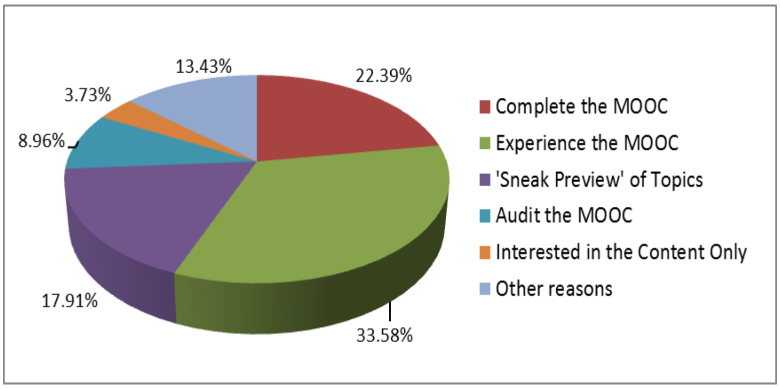


Figure 2.3 Drop-out students and the motivation to enroll in a MOOC taken from Gütl et al. (2014)

Interestingly, 22.39% of the respondents expressed their objective were to 'complete the course'. The rest making up 87.61% gave the following reasons: 33.58% indicated they wanted to experience the MOOC environment. About 17.91% users wanted to explore and have a 'sneak preview' into the topics. Some 8.96% indicated they wanted to audit the MOOC by learning only the content that they are interested in without having to finish the course. Only 3.73% were interested in the content without formally completing the course. Around 13.43% participants had given 'other reasons'. These reasons included 'having a quick view of the subject', 'deepening knowledge on a subject', 'contributes to my job activities', 'refresh and update the knowledge in a subject', and 'learn about the methodology'.

Inspired by various literatures, for example (Adamopoulos, 2013; Chyung, 2004; Clow, 2013; Kizilcec, Piech, & Schneider, 2013; Yang, 2013) on MOOCs' attrition, retention and completion rates, and our own research in the same area, an Attrition Model for Open Learning Environment Setting (AMOES) as shown in Figure 6 is proposed to understand and differentiate attrition reasons.

The AMOES model is divided into 3 sections (see Figure 2.4). These are the attrition and retention factors, the open online learner group classification and the funnel of involvement in an open learning setting. As the aim of attrition analysis is on the learners, AMOES groups learners according to 3 classes, these being healthy attrition, unhealthy attrition, and persistence learners.

Interlinked with these differential types are factors comprising external, internal and student factors that may contribute to a learner belonging to a healthy, unhealthy or persistence class of learners. Another contributing link is the administrative (awareness and registration) and pedagogical (activities and success) aspects of the MOOC which is term the 'funnel of involvement' in the learning setting. The final evolution of this model was based on a number of our own studies in MOOC up-take and drop-out and a deep analysis on attrition, retention and persistence of MOOCs over recent years.

Examples of external factors are competing courses that are offered in the MOOC space, varying technological infrastructure in different countries, cultural aspects and others. As these factors take place outside of a MOOC provider, institutions could identify strategies that may curb some of the external forces. Examples of internal forces are aspects relating to the organization of the MOOC that are under the control of the MOOC provider. Student factors are matters relating to individual student's desire to study a MOOC, prior knowledge of the study area with varying reasons. For example, some students enroll in MOOC because of their job, some for general interest, some for credential aspiration, and so on. Depending on the students' intention and motivation, ultimately they form different types of learners such as exploring user, content learner, restricted learner, disengager and completer.

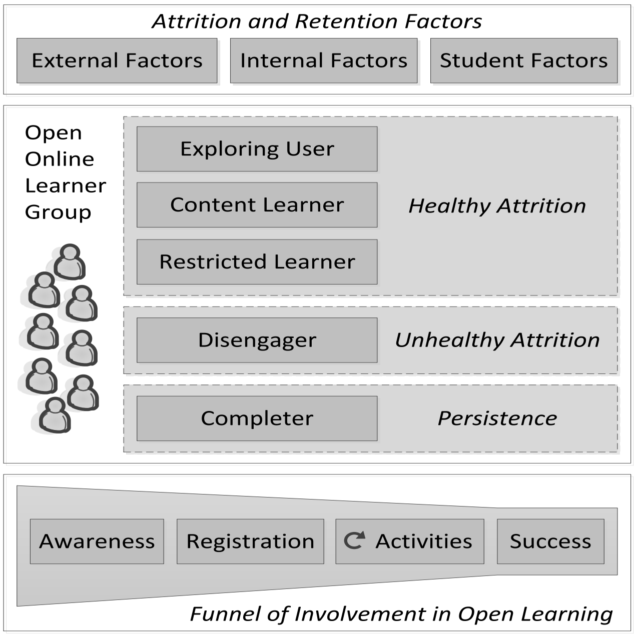


Figure 2.4 Attrition Model for Open Learning Environment Setting (AMOES) taken from Gütl et al. (2014)

The funnel of involvement in open learning is a modified 'funnel of participation' from Clow (2013). This last section of the AMOES model interlinked closely with the external, internal and student factors, along with the student differential classification types. 'Awareness' links closely with the 'External Factors' in that the MOOC must exist. This is followed by 'Registration' where students sign up and participate in the MOOC 'Activities'. At this stage, the MOOC provider (Internal Factors) plays a pivotal role in controlling the amount of activities that are balanced with interactive, engaging and contributing participation that would lead to a satisfying and ultimately successful experience. Implied in the 'Activities' funnel of involvement of the MOOC offering is the dependency upon the availability, compatibility and reliability of ICT which touch on both external and internal factors. Finally, the measure of 'Success' is based upon contributing student factors and the differential classes of healthy, unhealthy and persistence learners.

2.5 Discussion

Parts of the following discussion have been published in Hernández and Gütl (2015a).

The incorporation of *Web 2.0 Tools* in education appears to be one of the current important and large steps in education. Many studies and use cases have been done. Learning theories and pedagogical models presented in this Chapter support the inclusion of such tools, and *Learning Activities* can increasingly use them to foster engagement. Collaborative activities, new communication channels and new ways to represent knowledge and express ideas, share information and have large audiences are part of today's educational environments on the Web. Social interactions change the way online classrooms are perceived, and the teacher becomes more of a facilitator by intervening in the learning process, providing relevant, timely information, and constantly guiding learners through the learning experience. (Hernandez & Gütl, 2015b)

Among the large diversity of learning theories, pedagogical approaches, models and frameworks, CSCL originally provided key concepts to understand and support such collaborative online learning experiences. In addition, moving from designing to orchestrating collaborative activities has several challenges, such as monitoring and adapting interactions, workload, coordinating interventions, internal and external regulation, social learning and the continuous changes in the teacher's role towards conducting a flexible learning experience. Thus, practical approaches for orchestrating learning where all of these components are aligned and creating synergy among them can enhance the path towards obtaining learning outcomes. Flexible educational environments play an important role for adapting and controlling the learning process.

Practical experiences have revealed that collaboration per se does not provide learning outcomes. Thus, orchestrating collaborative learning experiences requires models and frameworks, one of which is the 5-stage e-moderation model. Although originally developed for traditional online forums, its structured stages can be applied in the use of Web 2.0 for learning. The two fundamental ground stages for successful online learning (according to the model) are to use the educational environment along with the correct levels of motivation and, second, online socialization. Both CSCL and the 5-stage e-moderation model support the inclusion of Web 2.0 into the educational process, and both are taken into account when elaborating upon the educational scenarios to evaluate the technology outcomes of this research.

In contrast, current topics such as online collaboration and socialization are missing from xMOOC experiences. Still, large attrition is seen in such courses, and applying the 5-stage model could be helpful, especially by emphasizing the stages of first securing correct manipulation and adaption to the educational Web environment, in order to later enable paths to successful online socialization. One possible socialization approach could be to separate the MOOC-enrolled learners into smaller (comparatively) groups (e.g., with 200 learners per group), and then activate online socialization activities that are incentivized and partially (with a rather soft presence) guided by a tutor within these smaller groups—this could potentially could lead to better or stronger social relations in the course.

The revised Bloom's taxonomy is used to evaluate the CBTs. Chapter 8 presents an acceptance model that uses the taxonomy to evaluate the learners' perspective on using these tools in education. Additionally during this Chapter has been described the main components of a *Learning Activity* that takes the use of tools into account, as just a means to achieve the learning outcomes. Clearly, the use of *Mediating Artifacts* is highly recommendable, because it provides a simple yet solid process for the teacher to create a *Learning Activity*. Chapters 4 and 5 introduce a model and system, respectively, to support the concept of *Mediating Artifacts* with the use of *Web 2.0 Tools*, while focusing on the templates and wizards approach.

There is an increased offer of *Web 2.0 Tools*, and challenges to using them properly arise from pedagogical, methodological and infrastructure points of view. The typology of Web 2.0 has served to identify the type of tools that are further used in this research: the *Media Sharing* type. A CEE can support any Web 2.0 tool, but the

Media Sharing type introduces simple yet innovative features. If structured within the proper *Learning Activities* using *Mediating Artifacts*, they result in a robust use case for CEE (presented in Chapter 4).

Web 2.0 Tools, such as mind maps and online office productivity suites, provide the flexibility to elaborate multiple types of *Learning Activities*. These tool types avoid the hurdles for most learners of first understanding a new technological tool concept, in order to later use it to express and acquire the learning outcomes. Nonetheless, the large ecosystem of *Web 2.0 Tools* (Chang & Guetl, 2007) implies that an online education environment needs to support such an ecosystem. Thus, in the introduced CEE concept, *Web 2.0 Tools* can be included not just by referencing them, but by enabling interoperability controls that allows the management of such a distributed learning environment. Therefore, the scope of this research is to enable a CEE that overcomes the inherent orchestration problems that multiple distributed *Web 2.0 Tools* have—such problems are elaborated upon in Chapter 4.

Further studies about *Learning Activities* using *Web 2.0 Tools* need to be performed, especially in the areas of motivation, emotions, usability and learning strategies. These studies, along with gathering information to reference learners' behavior in an orchestrated ecosystem, are the keys to deeply understanding a CEE and its impact on education, such studies are addressed in Chapter 8. Finally, the MOOCs present many of the same challenges of traditional online learning settings and bring new challenges, such as large drop-out numbers, along with raising new difficulties with enabling educational experiences in such a massive environment. Therefore, enabling the correct pedagogical structures and management controls is a key factor to enhance current practices in MOOCs. Chapter 3 presents the current technologies to enable a CEE, including the first approximations of CEEs along with the underlying interoperability technologies to achieve an enhanced orchestration. The Chapter begins by summarizing the current pedagogical gaps and challenges in education regarding the incorporation of such *Web 2.0 Tools* into education.

3 Flexibility in Educational Environments

There are gaps and needs for flexibility in online educational environments from the pedagogical point of view, which are reviewed in Section 3.1. Among the challenges for adopting *Web 2.0 Tools* for education are the new skills that any educational environment should foster, which lead us to an increasing dilemma between teachers and learners regarding knowledge production and consumption, control and management. This is followed by a discussion of important success factors for adoption of such tools are listed at the teacher and institutional levels. Furthermore, the complexity created by this tool ecosystem is summarized, as well as how all of these issues, gaps and problems faced can be related to the interoperability of Web 2.0 in a CEE. Hence, the discussion on *Flexibility in Educational Environments* focuses on the interoperability of a Web 2.0 tool ecosystem, with special attention given to the simplicity of incorporating and orchestrating such tools. Subsequently, in Section 3.2, the current technologies for Web interoperability are analyzed, with a special focus on how to enable semantic interoperability towards an automatic processing of Web APIs. The current interoperability technology approaches for flexible educational environments are described in Section 3.4. Accordingly, an overview of the relevant standards and specifications related to flexibility is given. Finally, the current systems and specifications for educational interoperability are described, including the IMS LTI specification and how other frameworks have addressed the orchestration of *Web 2.0 Tools* for education, along with the issues that are still pending with such technologies.

This Chapter is based partially on the following publications (Hernández, Linares, Mikroyannidis, & Schmitz, 2013a; Hernández, Gütl, & Amado-Salvatierra, 2014a; Hernández & Gütl, 2015a; Hernandez & Gütl, 2015b).

3.1 Pedagogical Aspects

The following Section includes parts of the publication Hernandez and Gütl (2015b).

This Section reviews some of the pedagogical gaps, challenges and issues to the inclusion of *Web 2.0 Tools* in an educational setting following the work by Conole and Alevizou, (2010). Hence, *Flexibility* is required in educational environments, which, in this research, is focused on the integration of a *Web 2.0 Tools* ecosystem (external tools to a VLE) and the management controls these tools provide to orchestrate them, in order to enable a CEE. Along those lines, it is relevant to mention that, for today's participatory culture, where students are used to social networks, to a large diversity of content sites, and to the interaction and sharing that being online produces, Jenkins (2006) defined 12 skills that educational environments should foster:

1. Play — the capacity to experiment with one's surroundings as a form of problem-solving.
2. Performance — the ability to adopt alternative identities for the purpose of improvisation and discovery.
3. Simulation — the ability to interpret and construct dynamic models of real-world processes.
4. Appropriation — the ability to meaningfully sample and remix media content.
5. Multitasking — the ability to scan one's environment and shift focus as needed to salient details.
6. Distributed Cognition — the ability to interact meaningfully with tools that expand mental capacities.
7. Collective Intelligence — the ability to pool knowledge and compare notes with others toward a common goal.
8. Judgment — the ability to evaluate the reliability and credibility of different information sources.
9. Transmedia Navigation — the ability to follow the flow of stories and information across multiple modalities.
10. Networking — the ability to search for, synthesize, and disseminate information.
11. Negotiation — the ability to travel across diverse communities, discerning and respecting multiple perspectives, and grasping and following alternative norms.
12. Visualization — the ability to interpret and create data representations for the purposes of expressing ideas, finding patterns, and identifying trends (Jenkins, 2006)

Jenkins (2006) suggest that these skills have high educational potentials, with opportunities for a diversity of learning scenarios. Therefore, having multiple CBTs that provide practice for such skills in any type of knowledge domain is highly important. However, as (James, 2008; McPheson, 2008) describe, there are associated tensions because the boundaries are blurring between the producers and users of such tools, including issues of ownership, authorship, openness, expert authority and amateur creativity. All of these require better organization for deploying *Learning Activities* with CBTs. This demands enhanced interoperability with the CBTs, by being able to control their deployment, in contrast to just launching the CBT or simply sending a link for learners to use it.

There is evidence that the success factors for the adoption of *Web 2.0 Tools* in education are (Conole & Alevizou, 2010):

- *Scaffolding and Guidance of Teachers*: An important factor is to provide teachers with effective support regarding how to use such tools in their courses. This includes guidance on designing courses, *Learning Activities* and what the role of the teacher should be, in moving from a traditional lecturer to a facilitator.

- *Strategic Alignment*: The institution as a whole has to have a widespread alignment in order for these initiatives have a chance of success.
- *Understand the Learner Experience*: The learner's affective and emotional perspective has to be taken into account, in order to enable the factors that engage them.
- *Appropriate Support Structures*: Effective support is a key factor, including simple means to enable the use of such tools. Currently, the support of such structures is weak within the current tools' interoperability specifications and frameworks. This leads the educational community to find better and more granular controls over the educational experience.
- *Staff Incentives and Rewards*: The feeling of ownership and control over the teaching practice is crucial.
- *Sharing Good Practices*: Mechanisms to share and adopt good practices among the educational community.

There are still many barriers when putting the use of CBTs into practice. One of them is knowledge expansion, which refers to that now it is possible to use, remix and repurpose with the content. On the one hand, it may lead to more knowledge (Surowiecki, 2004). On the other hand, the knowledge may become superficial (Keen, 2007). Another issue is the absence of hierarchy and control, expressed in terms of increased complexity with the loss of content integrity and de-contextualized content, which may lead to misinterpretations. Boundaries are continually blurring, and different technologies and contents are continually overlapping. In this sense, Cardon and Aguiton (2007) describe users' motivation to use *Web 2.0 Tools* as "*hybrid motivations, where the individualisation of the user's goals meets the opportunity of sharing personal expression and the performance of creativity in a public space."* Furthermore, all participants in this networked society are both individualists and, at the same time, mutually dependent (Ryberg, 2008). Finally, collective ownership and co-modification may become troublesome in an educational environment. Therefore, *Web 2.0 Tools* may support non-formal and informal educational contexts better where the role boundaries between teachers and learners do not have to be well defined. Unless more clearly defined roles and the corresponding authority over the *Web 2.0 Tools* are added, it is practically impossible to organize formal learning at an institutional scale. Thus, interoperability controls are required to support such roles and organization. In summary, the authors Conole and Alevizou (2010), have summarized issues presented in Table 3.1, which presents the educational dilemmas of incorporating *Web 2.0 Tools*.

This diversity of changes implied by *Web 2.0 Tools* (CBTs), from a pedagogical point of view, creates a barrier to adopting the changes. Among them are concerns of: 1) Authority (especially related to how the learners behave in a digitally networked environment). 2) Computer literacy issues, such as if the newly introduced tools can be adequately used for education. 3) The quality and effectiveness of their use. 4) The inherent issues with the legacy systems for long-term technological cohesion. All of these barriers can be decreased by having enhanced interoperability with CBTs, interoperability that enables control over and guidance regarding how the educational experience is going to happen, increasing the quality of the experience as a whole and lowering the literacy issues because the CBTs' management process and setup can be done automatically.

Cause	Education dilemma
Expansive knowledge	Challenges the role of the teacher
Hierarchy and control less meaningful, content can be distributed and located in different ways	Need to rethink the design process, offers the potential for new learner pathways
Increasingly complex digital landscape	Widening skill gaps between the "tech savvy" and the others
Power of the collective, collective intelligence	Potential for new forms of learning; digital and networked literacies

Table 3.1 Educational dilemmas arising as a consequence of new technologies taken from (Conole & Alevizou, 2010).

Finally, simple yet comprehensive interoperability to support the aforementioned capabilities that can be used by other legacy systems, such as the current VLEs, is still a work in progress, as presented in Section 3.3. However, new Web technologies (Section 3.2.5) may smooth the path to enable more granular control over the whole educational experiences.

3.2 Interoperability on the Web

A general overview of currently available Web interoperability technologies is presented in this Section, including the interoperability contracts on the Web, as well as a brief introduction to the semantic Web, the styles of *Web Services*, and the Linked Data implementation using JSON. Finally, this Section also introduces a Web API vocabulary built atop of Linked Data concepts, which might prove useful for description and therefore the automatic consumption of Web APIs in a CEE.

3.2.1 Interoperability Contracts on the Web

For systems to be able to interoperate, there has to be an specified communication protocol containing a diversity of definitions, such as how the requests and responses are made, what the data models to exchange are and the available services and operations that are known, in order to establish a *Contract*. The system interfaces define how *Contracts* have to be used, in which order, the dependencies between them and what the final results are that are obtained. This is the standard approach of the remote procedure call (XML-RPC), during which the client and server are tightly coupled, and is usually based heavily on the implicit state control-flow. Each message exchange between the two represents a change in the state and transitions of the systems. Since these requests to the server are hardcoded into the clients, the client breaks when the server changes its implementation until it has adapted to the new contract. The knowledge about the API is embedded into the programming done at the client side, leading to tightly coupled systems and creating impedance for the evolution of each system separately. This static, non-machine-readable definition makes it impossible to dynamically communicate changes to the clients, and makes the clients responsive enough to adapt to those changes.

The Web differs from the previous mentioned model, as it is based on media types and protocols. The media types define the data and processing models, as well as the serialization or payload formats. Also, the protocol defines the interaction model by linking relations. Thus, instead of relying on tightly coupled services, where every system interface is built custom made for each other, parts of the contract in the Web can be negotiated and interpreted at runtime, while the client keeps the state of the communication (Lanthaler, 2014).

3.2.2 Linked Data and Semantic Web

The Web is not only limited to documents—it can represent any type of resources. Tim Berners-Lee already considered this in Web's original proposal (Bernes-Lee, 1989). Thus, the Word Wide Web Consortium (W3C) created the Resource Description Framework (RDF) (Swick, 1999). The Semantic Web is a web of data that can be processed by machines. However, the Semantic Web community worked a lot on artificial intelligence in its early years, until (Shadbolt, Hall, & Berners-Lee, 2006) stated that the basic ideas of a Semantic Web were still mostly not yet done, which motivated the publication of the Linked Data principles (Berners-Lee, 2006).

The Semantic Web provides machine-readable data, as an extension of the Web. The RDF then defines a simple data model of triples, consisting of the subject, predicate and object. These triples are used to build a graph, and multiple graphs form a dataset. In contrast, RDF is often considered to be highly complex, mainly because of its RDF/XML serialization format (Beckett, 2004). In RDF, each concept is an internationalized resource identifier (IRI). These concepts or IRIs can be reused and are unique, and the concepts' meanings are defined by the IRI's owner (Jacobs & Walsh, 2004). If a set of concepts is built, it is often referred to as a vocabulary, or formally known as an ontology. The RDF Schema (RDFS) (Brickley & Guha, 2004) defines how to describe classes, properties and the data types, essentially like object-oriented programming. Then, the Web Ontology Language (OWL) (OWL, 2009) can be said to be an extension of RDFS but adding new concepts, thus making it a more expressive language. Both RDFS and OWL are for inferring new knowledge rather than for validating data.

All of these semantic Web technologies do not require IRIs to be deferenceable (capable of being referenced) to change that were postulated the Linked Data principles as defined by Berners-Lee (2006):"*a) Use URIs as names for things; b) Use HTTP URIs; then, persons can look for those names; c) When searching for an URI, provide useful information, using the standards (RDF, SPARQL); [and] d) Include links to other URIs, thus making it possible to discover more objects*". Linked Data mainly requires Web developers to identify their objects and concepts through an IRI, and also distinguish between resources that are information (e.g., documents) and resources such as things (Lanthaler, 2014) (the term *Object* or *Resource* is used interchangeably throughout this thesis).

3.2.3 SOAP and RESTful Services

For this thesis, Web services are defined as interfaces based on HTTP that enables machine-to-machine interoperability by exchanging structured data. The *Simple Object Access Protocol* (SOAP) (Lafon, 2007) specifies a messaging framework that uses an XML message format. The *Web Service Description Language* (WSDL) (Chinnici, et al. 2007) defines the interface for a Web service, while the *Universal Description Discovery and Integration* (UDDI) (Clement, Hately, Riegen, & Rogers, 2004) allows the discovery of services and its descriptions. Despite the promises of this type of technology, it has many flaws because it is based on RPC. For example, instead of using HTTP as an application protocol, it only uses HTTP as a transport protocol. Another problem is that the service interfaces published using WSDL often describe their own implementation specifications, thus being conducive to tightly coupled interoperability.

The previously stated problems and the relative complexity of SOAP-based services have shifted the attention of the developer community to lightweight solutions such as RESTful (REST API) services—about 3 out of every 4 Web services are based on this architecture (Vitvar & Musser, 2010).This does not mean that those RESTful services fully use the REST architectural style. Nevertheless, most of them do use HTTP as an application protocol and their resources have an IRI.

Representational State Transfer (REST) (Fielding, 2000)is a traditional *Client–Server* architecture, which means that the client can send requests to the server. It allows the creation of extensible, evolvable, maintainable and loosely-coupled distributed systems on the Internet. REST provides a specific design to improve performance, scalability, reliability and resource abstraction for hypermedia. One important aspect of the architecture is that the server is *Stateless* (e.g., the server does not maintain the session state). In REST, any concept can be thought of as a *Resource*—this is the key abstraction of information. Additionally, in REST, a representation is a resource with some metadata. It also enforces *Self-Descriptive Messages* that can be processed without anything else. Finally, it also refers to the use of hyperlinks as a way to navigate the state of an application, as well as hypermedia controls to advertise valid state transitions at runtime, instead of predefining static contracts. This brings the possibility of building loosely-coupled and evolvable systems.

A Web API may use parts of REST. However, unless it uses it all, it is not completely RESTful compliant. Web APIs usually use HTTP and XML or JSON as a serialization format (Crockford, 2006), the latter being the most popular format used in recent years, especially due to its heavy usage in AJAX (Garrett, 2005) related technologies, among others.

3.2.4 JSON-LD

JSON is a popular serialization format while JSON-LD supports transporting Linked Data using JSON. The initial goal is to make it easy for developers to use it by transforming their current JSON to JSON-LD (both are fully compatible). The design of JSON-LD focuses on the concept of “context,” which provides mappings between JSON and an RDF model. This means it links object properties of JSON to concepts described in an ontology—these are JSON properties to be mapped to IRIs, making them unique on the Web, and if possible, deferenceable. It is also possible to put identifiers to JSON objects through the `@id` keyword to express hyperlinks between resources. The context can be referenced either by embedding it in the document or by using the HTTP Link header, the latter option provides an easy path to upgrade current API without the need to change current JSON documents. Then makes it possible to define the `@type` properties, individual values and entities. It is possible to associate a class with an entity, and it is also possible to associate a data type. The most commonly used types are already standardized by using XMLSchema (Biron & Malhotra, 2004) and by reusing them to improve interoperability, meaning that no implementation details are presented and no hard coupling can exist because properties are already defined and widely used. In Listing 3.1 a simple example of JSON-LD is presented. It first uses `@content` to link to a vocabulary that describes what a person is and its properties, such as name, birthdate, or spouse (JSON-LD, 2014).

```
{
  "@context": "http://json-ld.org/contexts/person.jsonld",
  "@id": "http://dbpedia.org/resource/John_Lennon",
  "name": "John Lennon",
  "born": "1940-10-09",
  "spouse": "http://dbpedia.org/resource/Cynthia_Lennon"
}
```

Listing 3.1. A simple example of a JSON-LD document representing a person, taken from json-ld.org (JSON-LD, 2014).

One of the cumbersome issues with JSON-LD is to write full IRIs, which can be error prone, so to minimize this, there is a way to define prefix mappings in the context. With this prefix the vocabulary namespace is considerably shortened.

This makes the context more readable for developers. Another approach to minimize IRIs is by using the `@vocab` keyword that is used to define an implicit global prefix that can be used for properties.

The following is an extract from the JSON-LD 1.0 Data Model (Spurny, Kellogg, Lanthaler, & Lindström, 2014):

- The data serialized using JSON-LD has to form a graph, and it is possible to serialize the same data in different ways as a consequence of the graph data model and JSON itself.
- A JSON-LD document serializes a generalized RDF Dataset, which is a collection of graphs that comprises exactly one default graph and zero or more named graphs.
- The default graph does not have a name and may be empty.
- Each named graph is a pair consisting of an IRI or blank node identifier (the graph name) and a graph. Whenever practical, the graph name should be an IRI.
- A graph is a labeled directed graph (i.e., a set of nodes connected by edges).
- Every edge has a direction associated with it and is labeled with an IRI or a blank node identifier. Within the JSON-LD syntax, these edge labels are called properties. Whenever practical, an edge should be labeled with an IRI.

- Every node is an IRI, a blank node, a JSON-LD value, or a list.
- A node having an outgoing edge must be an IRI or a blank node.
- A graph must not contain unconnected nodes (i.e., nodes that are not connected by an edge to any other node). An example graph is modeled in Figure 3.1.

JSON-LD is a data interchange format, but it is not enough to represent a Web API. For that it needs a vocabulary that defines the concepts that later are serialized using JSON-LD. A vocabulary is capable of representing the hypermedia controls that are necessary to implement a truly RESTful Web service.

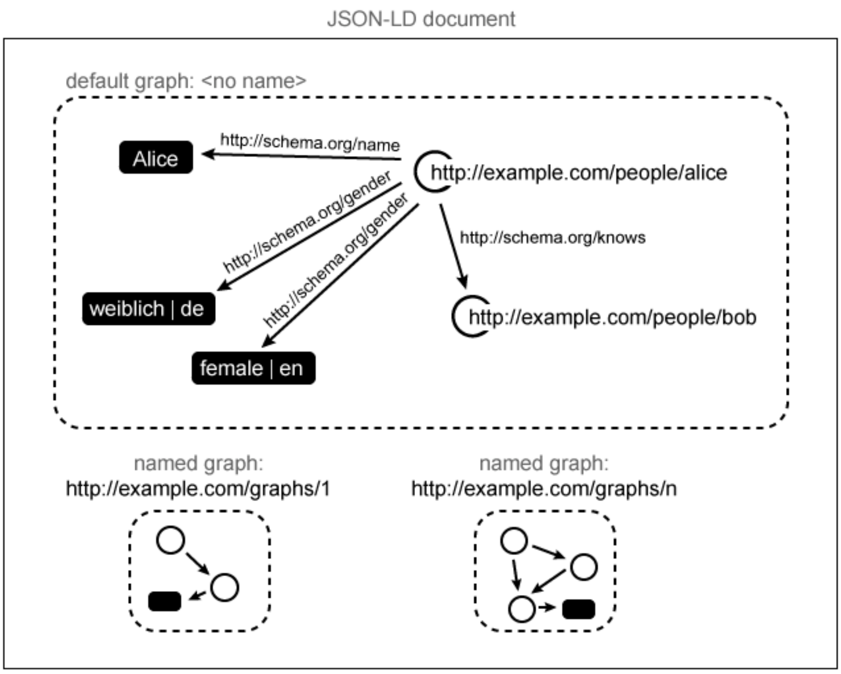


Figure 3.1 Illustration of the JSON-LD data model taken from (Spurny, Kellogg, Lanthaler, & Lindström, 2014)

3.2.5 Hydra

RDF has no inherent support for hypermedia, so a shared vocabulary to support the concepts that are usually found in Web APIs is needed, including full support of RESTful principles. This allows generic clients to process a given Web API description. For simplicity, a RESTful Web API includes interlinked resources, each identified by an IRI. A client needs to understand the semantics of a hyperlink (e.g., the relations between resources).

The Hypermedia-driven applications (Hydra) vocabulary serves to describe Web APIs using JSON-LD that are truly RESTful. According to Hydra's unofficial draft, *"The basic idea behind Hydra is to provide a vocabulary which enables a server to advertise valid state transitions to a client. A client can then use this information to construct HTTP requests which modify the server's state so that a certain desired goal is achieved. Since all the information about the valid state transitions is exchanged in a machine-processable way at runtime instead of being hardcoded into the client at design time, clients can be decoupled from the server and adapt to changes more easily."* (Lanthaler, 2014).

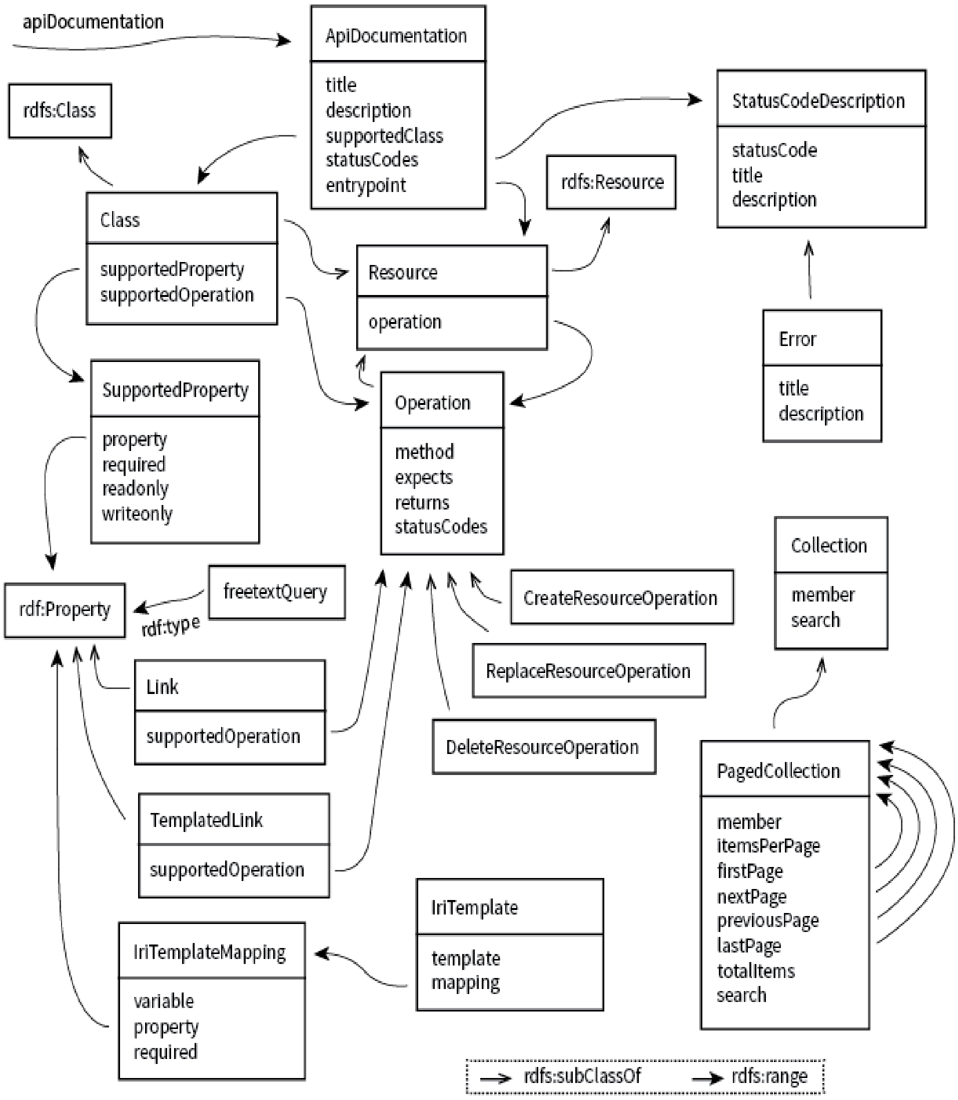


Figure 3.2 The Hydra core vocabulary, taken from (Lanthaler, 2014)

Figure 3.2 shows the Hydra vocabulary in a simplified graphic. At the top of the Figure 3.2 it is the `ApiDocumentation` class which serves to document a Web API. First it defines the main entry point, and then it documents classes, properties, and operations. It also permits one to document the HTTP status codes. It has a `Resource` class that is a subclass of RDF Schema's Resource, which is used to inform a client that the IRI is deferenceable and that it can be used to retrieve more information—this is Linked Data. As depicted in Figure 3.2, Hydra also supports the notion of classes (object or resources), properties and operations, which are key concepts to model a Web API. The `Operation` has the necessary information to construct HTTP requests, and it also describe what the Web API expects and how it is returned. This allows manipulating the server's state of a given `resource`. `Operations` do not have bound `properties`. Instead, `properties` are bound to `classes`, so `operations` are over a `resource`. In `supportedProperty` are enumerated `properties` for a class. A property can specify whether it is required, `readonly`, or `writeonly`. Figure 3.2 shows that the operation supports a basic set of CRUD operations, such as `CreateResourceOperation`, `ReplaceResourceOperation` and `DeleteResourceOperation`, but Hydra does not restrict the mapping of these `operations` to specific HTTP methods, so all these `operations` may simply use the HTTP POST method instead of alternatives such as HTTP DELETE. Another important class is the `collections`, which is quite commonly used in any system, and often used to contain a set of data of the same type.

```
{
  "@context": [
     "http://www.w3.org/ns/hydra/core",
      {
            "description": "rdfs:comment",
            "mm": "http://cis.galileo.edu/specs/mindmapping#",
            "supportedClass": "hydra:supportedClass",
            "supportedOperation": "hydra:supportedOperation",
            "supportedProperty": "hydra:supportedProperty",
            "vocab": "http://cis.galileo.edu/mindmeister/vocab#"
        }
    ],
    "@id": "http://cis.galileo.edu/mindmeister/vocab",
    "@type": "ApiDocumentation",
    "supportedClass": [
        {
            "@id": "vocab:EntryPoint",
            "@type": "Class",
            "description": "The main entry point or homepage of API.",
            "label": "EntryPoint",
            "supportedProperty": [
                {
                    "property": {
                        "@id": "vocab:EntryPoint/userInfo",
                        "@type": "Link",
                        "description": "User information",
                        "domain": "vocab:EntryPoint",
                        "label": "userInfo",
                        "range": "Collection",
                        "supportedOperation": [
                            {
                                "@id": "_:getUserInfo",
                                "@type": "hydra:Operation",
                                "description": "Get logged user information.",
                                "expects": null,
                                "label": "getUserInfo",
                                "method": "GET",
                                "returns": "vocab:User"
                            }
                        ]
                    },
                    "readonly": true
                }
            ]
        }
```

Listing 3.2. A mind map Hydra Web API example presenting the main entry point of the API available.

The resource presented in Listing 3.3 has its own human readable `label` and `comment`, and it defines the `supportedProperty` that it has, beginning with the *id*, where it says it is `readonly`. Then a second property called `username` is defined, notice that this is a reference to `mm:username`. This reference points to another vocabulary that actually describes the data type of this property—something like `xsd:string` could be possible. Finally a third property is described: the `User/maps` is a `Collection` of user maps. Notice that it has an operation related to it called `create_map` of the type `mm:NewMapOperation`, defined in another vocabulary. The operations can be invoked through the HTTP POST method, which is expected to return a `Map` resource. Listing 3.2 depicts a Hydra example for a mind map Web API. It starts with the `@context`, which contains a reference to the Hydra vocabulary.

```
{
    "@id": "vocab:User",
    "@type": "mm:User",
    "label": "User",
    "comment": "User representation",
    "supportedProperty": [
        {
            "property": "vocab:id",
            "readonly": "true"
        },
        {
            "property": "mm:userName"
        },
        {
            "property": {
                "@id":"vocab:User/maps",
                "@type":"Link",
                "domain":"vocab:User",
                "range":"Collection",
                "label":"user-maps",
                "description":"Maps of user",
                "supportedOperation":[
                    {
                        "@id":"_:create_map",
                        "@type":"mm:NewMapOperation",
                        "method":"POST",
                        "description":"Create a new map",
                        "label":"createMap",
                        "expects":"vocab:Map",
                        "returns":"vocab:Map"
                    }
                ]
            },
            "readonly":true,
            "writeonly":false
        }
    ],
```

Listing 3.3. Part of the proposed mind map Hydra Web API example, presenting elements as an example the supported operations of a user for map resources.

This example also features a reference to other vocabulary, expressed as `mm`. Then it expresses that it is the `API Documentation`, which is basically a description of the Web API. This comes with the definition of `supportedClass,` which contain each of the resources available. It first starts with the `EntryPoint` that gives the client processing the Hydra Web API a place to start using the Web API. It has a property that gives the user's information and has an operation associated with it, `getUserInfo` returns the `User` resource. Continuing with the example, in Listing 3.3, the `User` resource is described. It is of the type `mm:User`, which references another vocabulary.

As demonstrated in the previous example, Hydra's expressivity is sufficient to describe Web API resources, properties, and operations. And it can describe how these operations need to be performed, the data types used for each property, and the relations among all of them. With Hydra it is possible to define vocabularies that can then be used by other Web APIs, thus enabling reusable definitions. This can be highly important for several scenarios, such as an industry agreeing in a common vocabulary for describing common objects, properties, and resources. With Hydra the *Common or Generic Vocabulary* gives enough flexibility because although a *Generic Vocabulary* can be used by all industries, each implementer of a Web API can use it. But then it is necessary to write the API Documentation (or API Doc) that contains the specific modeling of the application. Thus with this approach the *Generic Vocabulary* can model the generics of an industry that can be the reused in the API Doc, which actually describes the specifics of the Web API, which may include new resources, properties, and operations.

3.3 Systems and Specifications for Educational Interoperability

Interoperability related to educational settings was defined in Moen and McCLure, (1994) as a "condition that exists when the distinctions between information systems are not a barrier to accomplishing a task that spans multiple systems" or in (Olmedilla, Saito, & Simon, 2006) as "the capability of different systems to share functionalities or data." It is clearly an important research topic for educational technologists. In terms of cloud computing, the author in (Lewis, 2012) identified the important role of standards for interoperability, and (Aroyo, et al., 2006) listed some of the most used standards (e.g., learning object interoperability framework (Simon, Massart, & Duval, 2004), content object repository discovery and resolution architecture (Kraan & Mason, 2005), Edutella (Nejdl, Wolf, & Qu, 2002), and learning tools interoperability (IMS, 2014).

There are many educational standards and specifications, and according to Shepherd (2006) and Al-Smadi, (2012), those can be organized as follows:

a) Authentication: mainly focused on single sign one.
b) Content packaging: providing sharable content and the transmission of it among systems.
c) Data definitions: providing a kind of schema (in XML or any other format) that has the corresponding content structure.
d) Data transport: to describe how data is transferred among systems.
e) Launch and track: how content and tools can be launched and afterward tracked.
f) Metadata: used for content description, search, and retrieval.
g) Philosophical: frameworks for describing a process, contents, tools.

At time of writing, the interoperability has many related standards or specifications, such as IMS Learning Information Services, the IMS Learning Tools Interoperability, the IMS Common Cartridge, IMS EDUPUB2, IMS Question and Test Interoperability, IMS Accessible Portable Item Protocol, IMS Learning Metrics Profile, SCORM, IEEE Learning Objects Metadata, IMS Learning Design, IMS Content packaging, and others (IEEE LOM, 2008; IMS CP, 2008).

There are many interoperability official standards, specifications, de facto standards (not an official standard but widely accepted and used by the industry), and reference models. All of them focusing on interoperability at different levels and from different perspectives, all of them helping to create a flexible educational environment in which many pieces of the big puzzle of an educational environment can be used as plug and play components. The focus in this thesis is researching interoperability with external tools in general, not for a specific application domain (e.g., for assessments, the focus probably is on the corresponding specifications such as IMS questions and test interoperability). This interoperability with external tools are presented in the following subSections and has been traditionally addressed by including widgets with the tool interface and (or) by launching the external tool and having basic communication between the VLE and the external tool (for this thesis, the terms *External Tool,* Web 2.0, and CBT are used interchangeably). Subsequently, the current technologies and their limitations in educational external tool interoperability are presented.

3.3.1 ROLE PLE

The ROLE project aimed to enable learners to assemble and use their own learning environments, which became advanced *Personal Learning Environments (PLEs)* (Friedrich et al., 2011). ROLE technology is centered on the concept of self-regulated learning, aiming at creating autonomous learners that are able to plan their learning process, search for suitable resources independently, and learn and then reflect on their learning process and progress. Using ROLE's techno-pedagogical infrastructure, a psycho-pedagogical setting adapted to the specific needs of courses and students has been built.

ROLE aims to include any type of content and tools with the possibility of the learner using a simple process to construct a learning environment. ROLE consists of a variety of preferences expressed in tools that are used for learning, and the same applies to any type of content or service.

So the inclusion of those contents and tools is through a widget-based approach. (OpenSocial, 2014; Cáceres, 2010). Many widgets are openly available, so for the ROLE project, it was the right approach. Apache (2014), the reference of an OpenSocial container, was used. Due to the lack of communication between widgets, *Inter-Widget Communication* using *Gadget Pubsub Channel* was used—it is a message bus that enables widgets to publish or subscribe to events using a message format (OpenSocial, 2014) with a unified and extensible JSON message format (Isaksson, 2010). Due to several limitations, the Extensible Messaging and Presence Protocol (XMPP) was selected (Saint-Andre, 2004; Saint-Andre 2009; XMPP, 2004).

Contextualized Attention Metadata (CAM) (Linden, 2003) allows recognition and identification of the context of learners based on theories such as competence-based knowledge space theory (CbSTK) (Albert & Lukas, 1999), which provides a path for adaptively assessing domain skills and carefully planning personalized learning paths. CAM could be applied in assessment systems, usage reflections, and usage pattern detection.

Highly related to ROLE are mashup personal learning environments, or MUPPLEs (Isaksson, 2010), which are said to guarantee a clearly arranged user interface. And because it is not possible to integrate all the services, tools, and contents in a single presentation layer, the PLE serves as a single entry point. The ROLE PLE also has a recommender system (Kirschenmann et al., 2010).

Finally, ROLE includes a software development kit (SDK) a very good idea for adopting the platform because it provides a consolidated start point for adopting this new technology by educational institutions. As of September 2014, ROLE had completed, with no significant adoption of the ROLE SDK. Therefore an SDK does not guarantee adoption by third parties of a new technology.

3.3.2 IMS-LTI

Some parts of the following Section have been taken from Hernández, Gütl and Amado-Salvatierra (2014a).

Learning tools interoperability has been a great way to enable further innovation in learning settings. The objective is to enable Tool Consumers (TC) and Tool Providers (TP) to exchange information. A relevant work related to the aforementioned scenario is the IMS-Global LTIv.2.0 specification, which has contributed to enabling interoperability, and it is currently state-of-the-art in interoperability for educational settings. LTIv1.0 allowed to mainly launch a TP from the TC as depicted in Figure 3.3.

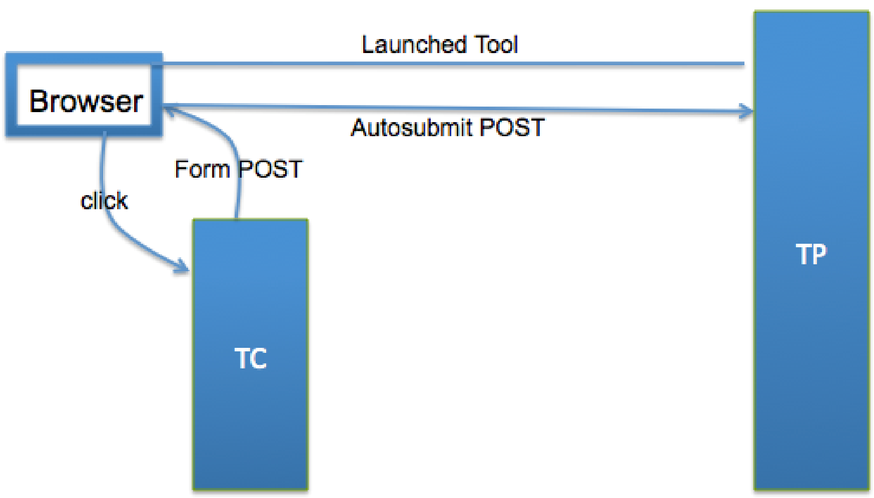


Figure 3.3 An schema of LTIv1.0 taken from (IMS, 2014)

LTI's version 2.0 allows launching any tool in a transparent way for the users. It handles automatic credential exchange and management, authentication, and authorization in a secure fashion, including the notion of context (e.g., a course) and respective user info and roles. It uses OAuth 1.0 protocol to securely sign messages. It defines a TC Profile, which is metadata that describes attributes and available services and supports capabilities of the TC through a REST service. TP Profile also exposes the supported capabilities and services it provides. In Figure 3.4 is depicted an overview of LTIv.2.0. The Figure is described in IMS (2014) as: *"is essentially provided a means of connecting two systems together: a "Tool Consumer" which "consumes" the tool, and a "Tool Provider" which "provides" the Tool to be used in the Tool Consumer. A Tool Consumer would typically be an LMS. Examples of Tool Providers include an externally hosted testing system or servers containing externally hosted premium content. The nature of the relationship established between a Tool*

Consumer and a Tool Provider via LTI is that the Tool Provider delegates responsibility for the authentication and authorization of users to the Tool Consumer. The Tool Consumer will provide the Tool Provider with data about the user, the user's current context and the user's role within that context. This data is provided in a secure manner so that the Tool Consumer may trust its authenticity.". Between the TC and TP can be two types of connections, one is LTI services and the other is messages. Messages are transported over HTTP usually to launch the TP. The LTI services is a direct connection between the TC and TP using JSON-LD format, the standard example given by LTI is that a TP send back to the TC the grades obtained by the learners at the TP.

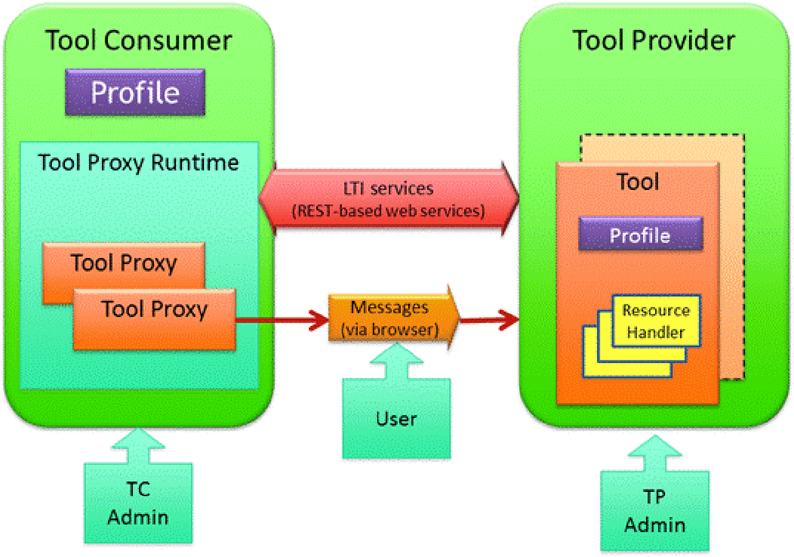


Figure 3.4 Overview of LTIv.2.0 taken from (IMS, 2014)

LTIv.2.0 defines a Tool Proxy, which determines a negotiated interface contract between a particular TC and TP. This includes tool details (TP Profile) and capabilities made available through this contract. This provides great flexibility to incorporate new services, without changing any core standard. It just needs to incorporate new service metadata. Also the notion of a resource handler has been incorporated into LTIv.2.0. This means that a tool may expose several kinds of resources (i.e., a library tool serving several books) that can be launched from the TC, and potentially each kind of resource has a different endpoint at the TP, with a custom set of parameters for launching a tool. LTIv.2.0 also provides REST services for server-to-server interoperability as architecture for bi-directional Web services, with resources as a basis for HTTP addressing, with appropriate use of HTTP intrinsic methods. In Figure 3.5 is depicted the TC and TP communication process that is initially negotiated with the Tool Proxy, thus the bidirectional communication is enabled.

Additionally, it supports media type definitions with the support of linked data and uses JSON-LD for payload. The primary drawback of the current LTIv.2.0. is that it does not offer the concept of basic CRUD operations (create, read, update, delete) over the resources. Nor does it offer support for any other type of operation over a resource. Thus, it limits itself to predefined communication between TC and TP without the ability to execute explicit operations that might be available on public API by the TP.

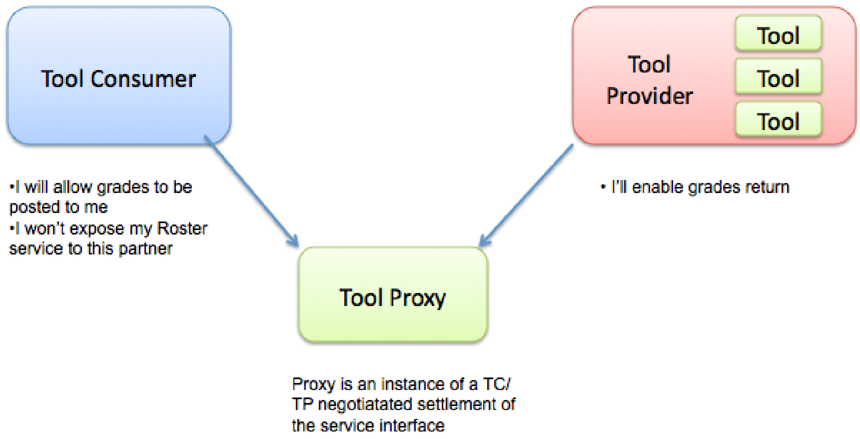


Figure 3.5 Tool proxy example, as an intermediary negotiated contract between the TC and TP taken from (IMS, 2014)

Using as an example a TP offering mind map services, LTIv.2.0 does not provide the ability to define the following operations: create a map, create an idea, connect the idea with other idea within a given map, comment over an idea, or assign permissions over a map to a student group that might collaboratively work on it. LTIv.2.0 is limited to the exchange information between the TC and TP, launching the tool from the TC, and providing context (a group, a classroom) to that tool. In simple terms, with LTIv.2.0, the VLE is able from a course to launch a tool, exchange predefined data with the VLE and the tool, and provide data such as a grade after using the tool. LTIv.2.0 does not use linked data for Web API description and discovery of properties and operations, nor does it have an ontology to describe those. As mentioned above, it is an automatic contract process for the exchange of information. There are interesting opportunities for improvement to achieve a complete interoperability within CBTs used in a CEE.

3.3.3 GLUE!

There are several architectures that support the integration with external tools, such as CBTs. A good example is GLUE! (Alario-Hoyos et. al., 2013), an architecture for the integration of external tools in VLE. It provides an architecture that is capable of creating, configuring, and assigning external tool instances. This architecture is capable to use external tool instances, update users that share control over the external tool instances. And, finally, delete the external tool instances. Additionally, the referenced GLUE! architecture implementation notes that there is a GLUE! core that handles all communication between the VLE and the external tool and processes the integration contracts. Those contracts are represented and materialized as adapters for both the VLE and the external tools. By using GLUE! to interoperate a VLE with an external tool, first an API interface, known as a VLE adapter, has to be created and programmed to connect the VLE to the GLUE! Then, each new external tool that is integrated has to be programmed through a new tool *Adapter* that communicates between the external tool API and the GLUE! This is a clean approach to interoperating VLE with external tools that allow a single intermediary to handle all communication logic in a standardized way. However, it requires manually developing and maintaining the tool adapters, which involves custom programming. And GLUE! does not support operations (e.g., CRUD), nor does it have the notion of resources and related properties. Thus it limits itself to launch and basic communication between the TP and TC. A more elaborate communication requires extending the GLUE! features. Figure 3.6 depicts the overall GLUE! Architecture. It is necessary to create VLE adapters to incorporate it into a VLE, and a tool adapter enables basic communication with the tool. Once all of this is in place, the teacher and (or) learner can launch a tool and use it. GLUE! supports four main use cases: First, the creation, configuration, and assignment of external tool instances. Second, the use of those external tool instances. Third, the updating of users' sharing of external tool instances. Fourth, the deletion of external tool instances. Further manipulation, such as their particular objects or resources, is not possible, nor is there further support for operations over those resources.

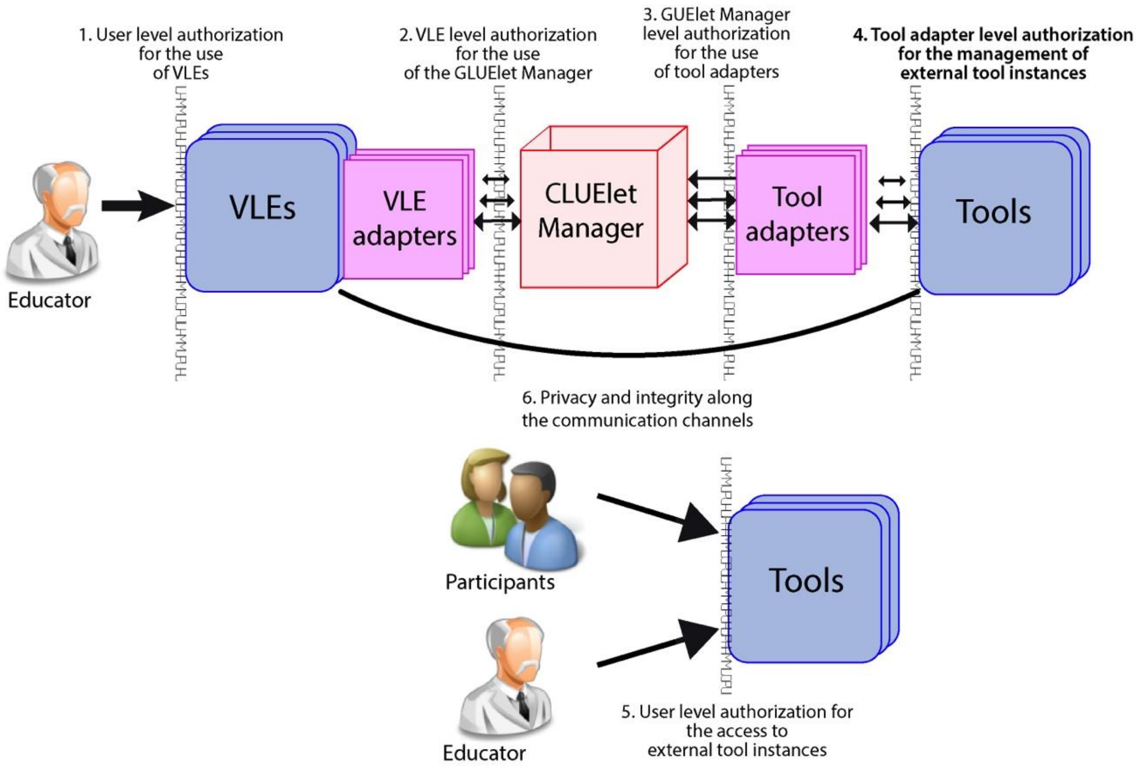


Figure 3.6 GLUE! Overview of the architecture taken from (Alario-Hoyos, et al, 2013)

3.3.4 Other Educational Interoperability Technologies

There are other available technologies and frameworks to enable interoperability. Recently, the edX LMS[1], specially designed for deploying MOOCs, published the XBlocks[2] API (in a pre-alpha version as of September 2014) to build courseware. It is an API for components such as the video player, LON-CAPA problems (LONCAPA, 2013), and compound components such as a learning sequence. XBlocks is a Python language-level API and provides interfaces for several things, such as storing data. XBlocks are embeddable within an edX LMS. An XBlock SDK that contains useful features for developers, such as the template XBlock generator, exists. XBlocks satisfies two objectives: it works together with other blocks to build a complete course, and it is independent of other blocks, allowing them to be combined with flexibility. Like any Web application, they maintain a storage layer, render information through views, and process user actions through handlers. They can only run within the proper run time environment: the edXStudio or the edX LMS. XBlocks integrate new content or tools through their interface, although all the desired behavior has to be specified through static web contracts to enable interoperability with external tools.

A similar approach is another LMS such as the commercial LMS Blackboard[3], which has a Blocks API[2], and through which it is possible to create web applications using Java or .NET. It uses an xml configuration file that defines security restrictions, among others. It is also possible to access, add, remove, and manipulate data within the LMS, including data such as users, courses, grades, calendars, content items, and more. Another commercial LMS such as Desire2Learn[4] and an open-source LMS like Moodle Blocks and Sakai Widgets (Desmet, 2008) have third-party static contract external tool capabilities. Another example, *Learning Activities Management Systems* (LAMS)[5], is capable of designing, managing, and delivering online collaborative *Learning Activities* while providing teachers with an intuitive and interactive authoring environment for creating sequences of *Learning Activities*. For connecting and integrating with external tools, LAMS has defined what is called Tool Adapters, which use LAMS Tool Contract for management issues such as authorization and authentication. The adapters are also known as *Wrappers* and can integrate CBTs. Those adapters do not have any business logic—they are just a bridge. All of these solutions are based on the static contracts that need to be made and maintained for each new CBT to be used and do not include inherent support for object manipulation.

3.4 Discussion

Parts of the following discussion have been taken from Hernández and Gütl (2015b).

Enabling a CEE that integrates CBTs is indeed necessary for new educational experiences. The current ecosystem of CBTs and multiple content providers force educational practitioners to use a distributed digital environment, in such a way that issues related to content production, ownership, and authorship arise. Furthermore, it is necessary to provide a simple yet powerful unified environment that includes CBTs while addressing challenges such as simplifying the adoption barriers for teachers, giving them best practices, allowing them full control over the educational

1 https://www.edx.org
2 https://github.com/edx/XBlock
3 http://www.blackboard.com
4 http://www.d2l.com
5 http://www.lamsinternational.com

experience, creating easy initial steps for the use of a new tool for the learner, providing support structures for both learners and teachers, and allowing institutional adoption. Thus, issues such as hierarchy and control problems, role definition and corresponding management of those roles, authority over resources created, integration with legacy systems such as VLEs, and lower literacy issues are created when using a new CBT for the first time. Such a unified environment that addresses the described challenges and issues can only be conceived if granular controls for interoperability are enabled between a central management system (such as a VLE) and the CBTs. This requires access to the CBTs' APIs, and the focus of this thesis is to solve specific issues of control, management, and authority over this distributed ecosystem of tools. Currently, many approaches (like the PLE widgets) use a lightweight approach for interoperability, which enables CBTs in the educational environment but fails to ensure real integration with the Web API, thus limiting the educational scenarios. It also fails to address the current pedagogical needs mentioned in Section 3.1. The widget approach like ROLE claims to have a single entry point and a unified interface for enabling the CEE. This is partially true in that it does enable a single educational environment, but it does not offer a unified or guided educational experience, and it lacks granular controls over the tools that it integrates.

When analyzing current educational interoperability specifications and systems, it is clear that they use a static standard contract-based approach for interoperability, and the construction and maintenance of a CEE have a cost proportional to the number of CBTs to integrate. That is the case for the GLUE! System. Although it is argued to be loosely coupled because of the interoperability between CBTs and VLEs, it is not tightly constructed. Instead it uses GLUE! as an intermediary and still requires one to create custom programs for GLUE! interfaces for each CBT. So it is tightly integrated with GLUE! instead of tight integration in each VLE. The same pattern of tight integration is followed by all *Block*-like solutions (see Section 3.3.4), which are traditional APIs interacting with the VLE internals. These can host newly programmed contracts to interoperate with CBTs on a one-to-one basis. This static Web contract approach requires one to create and maintain tightly coupled contracts between the CBT and the TC, so the creation and maintenance costs are proportional to the number of tools to integrate. On the other hand, even though IMS LTI v.2.0 uses JSON-LD, its Linked Data capabilities are used only for simple messaging and tool configuration, but not for tool orchestration and management. Thus, it is used mainly for automatically setting contracts, but not for Web API description and discovery.

Current specifications and systems reviewed in this Chapter for educational interoperability lack the ability to clearly define for each CBT the *Objects* and their corresponding *Operations* and *Properties*, so the management controls over CBTs are limited, as is inferred from Section 3.1. From a pedagogical point of view, *Granular Controls* over CBTs are required. Furthermore, those specifications and systems do not use current semantic technologies that are capable of enabling machine-processable definitions of Web APIs, which simplify interoperability efforts. In the following Chapter, we introduce a semantic interoperability model to enable a flexible CEE with semantic definitions for machine-to-machine processing of semantically described Web APIs.

This review concludes that due to the current state of semantic technologies, one of the best options when using semantics is through Linked Data, and strong options are JSON-LD and Hydra. The first is a well-known serialization format, which eases adoption of developers of Linked Data. The second is a simple yet complete vocabulary that is capable of describing a Web API. Hydra enables full Web API description and discovery, which could potentially decrease the interoperability costs for a CEE, because interoperability happens at a higher and simpler abstraction level. The costs are expressed in terms of time to construct and maintain interoperability with the Web APIs. In a CEE, it becomes even more important to reduce costs because potentially it could have large and evolving tools to interoperate with. If a semantic approach is going to be used to create interoperability with CBT Web APIs, it is necessary to have a semantic vocabulary that is capable of describing such Web APIs, so it is a matter of using something such as Hydra or creating one for this specific research. This approach has the additional requirement of creating generic parsers for such a vocabulary. Hydra provides such a basic vocabulary processor.

Finally, one decision to make when using a given technology is to balance how widely it has already been adopted and its potential. Because even if the specification comes from an international consortium, it may not evolve to gather enough traction from the development community. For instance, the author of this thesis implemented the IMS Enterprise specification (IMS, 2013) for interoperability with universities' administrative systems (e.g., handle the catalog of courses, learners, and professors) in the .LRN LMS. Now such specifications have been superseded or replaced by Learning Information Services (IMS, 2013). Another example is the ROLE SDK. Although ROLE has a well-designed and robust SDK and architecture, it does not have a healthy adoption among any but a few users. On the other hand, JSON has worldwide adoption. Thus, carefully selecting the appropriate technology to solve the desired problem and creating the potential for larger adoption are important.

Part B: Design and Development

4 Toward a Flexible Cloud Education Environment

This Chapter proposes the architecture for a flexible *Cloud Education Environment (CEE).* Section 4.1 focus first on the problems and challenges presented when using distributed cloud tools for learning purposes and describe the interoperability gap with current technology. It then elaborate on how this approach can enable a CEE with a faster implementation time while lowering interoperability costs. Section 4.2 describes several applications related to a flexible CEE and shed light on the specific objectives of this research. Section 4.2 presents a use case that then is used in the next Chapters as a basis for technology and scenario development and based on it presents corresponding evaluation. This leads to a proposal for a conceptual model for flexible interoperability in Section 4.3, which presents an interoperability service framework that introduces the definition of application domain–type Web API through a vocabulary representation. It also demonstrates how to implement a tool in this semantic interoperability model. Then the *Interoperability Service Framework Architecture* is presented with a conceptual architecture that is capable of interoperating with *Cloud-Based Tools* (CBTs) by overcoming the problems and challenges already mentioned. This includes the use of *Linked Data*, introduced in the previous Chapter, as the main approach to build the interoperability solution. And it is designed as an intermediary pathway while tool providers (TPs) begin to use this semantic Linked Data technology, thus enabling current technologies to benefit from this approach. Then an *Orchestration Learning Activities System* is introduced that in conjunction with the *Interoperability Service Framework Architecture* enables a true CEE. Finally, a summary of the conceptual flexible environment is given to guide the reader through the next Chapters.

This Chapter provides a consolidation of the work published in Hernández, Gütl and Amado-Salvatierra (2014b), which include previous steps toward a proposal for a flexible CEE, and it is partially based on Hernández and Gütl (2015a).

4.1 Problems and Challenges

The following Section mostly republishes the work by Hernández and Gütl (2015a).

In Chapters 2 and 3 several challenges and problems were identified to enable a truly CEE. Here, such gaps are summarized and addressed by this research. The gaps are organized in four groups: 1) how to enable a flexible CEE (Section 4.1.1), 2) how to use open APIs to create an enhanced educational orchestration (Section 4.1.2), 3) the problems of current interoperability approaches (Section 4.1.3), and 4) how to create an *Application Domain Type* definition of a tool that can be interpreted at run time by the machine without human intervention (Section 4.1.4).

4.1.1 Enable a Cloud Education Environment

Today many software tools are available for everyday life tasks. The apps for the mobile platforms, such as iOS (AppleApp, 2014) and Android (Google, 2014), have thousands of them available. In September 2014, data from AppBrain Stats revealed more than 1,300,000 applications for Android (AppBrain, 2014), and AboutTechnology reported similar numbers for the same month (Costello, 2014). There are also thousands of tools available online (in the cloud) that are web based and that fully run over an Internet browser (Conole & Alevizou, 2010). Thousands of tools are traditionally available for desktop computing. In September 2014, data from MetroStore Scanner reported more than 170,000 applications (MetroStore, 2014). And other environments are also growing or consolidating their app ecosystem, such as smart TVs and gaming consoles. Current learners typically have multiple devices, use multiple apps through them, and now experiment with scenarios and applications that were neither available nor possible before this ecosystem was in place. In this current reality, the standard monolithic environment approach for VLE is still predominant in education. Thus, the challenge is a distributed, non-monolithic environment because is not possible to limit educational settings to just one environment. The aim is to create an educational environment based on a distributed set of services and contents available in the cloud of apps and devices. Along with fostering new skills (see Section 3.1) and addressing challenges described such as adoption barriers for teachers, allow teachers full control over the educational practices that are created, simplify learners' use of CBTs, and make such a CEE institutionally used and adopted. Another benefit of using CBTs is that many of them are running over cloud computing, which is highly scalable in terms of computing to support thousands of active requests. All of this, in conjunction with the nature of a distributed environment for performing the learning experience, brings a highly scalable environment.

4.1.2 Enhanced Orchestration Through Open APIs

Learning Orchestration requires flexible educational environments (see Section 2.1) where design and planning can be reflected in an educational workflow. This requires full administrative control over all the components of the educational experience. *Learning Orchestration* also requires interventions, adaptation of the learning paths, scaffolding knowledge and experiences from one activity performed in a CBT to the next one. Subsequently for educational purposes, it is not enough just to use in an educational setting new tools that are available on the cloud and through many devices, because that comes with hurdles such as the following:

- Multiple login registrations.
- Difficulties for the teacher to follow up and verify learners' performance in the third-party tool.
- Inability to pre–set up the learning process as designed (e.g., create and prepopulate tool instances to be used by the learners), requiring the learner to first understand and discover how to administer the tool and then set up the

tool instance as required, thus increasing the cognitive load in nonessential, nonrelated educational tasks.

- Group activities may require extra setup that cannot be easily deployed by the teacher.
- Multiple tools used in a given educational experience exponentially increase the setup and management problems. This is even worse in MOOCs, as presented in Section 2.4.
- Utilization of results from one tool as an input for the activity to be done in another tool is not integrated. For example, using a mind map to create a story in another tool requires going back and forth between the two tools (in contrast to interoperable tools that are capable of information flow).

For obtaining educational success when using CBTs is necessary, as described in Section 3.1, to have *Granular* management and setup controls to ensure smooth use of multiple CBTs. Granular management refers to the ability to control as many resources as possible within the CBT.Issues such as hierarchy and control, role management, authority over CBTs and integration with institutional systems such as VLEs need to be solved. These can be achieved by using tools' open APIs to enable orchestration, which is explained in the following Sections.

A current trend is that many of the available tools have open Web APIs that tool consumers (TCs) can use to manipulate them. Mashery API network provides a collection of more than 70 open services that offer APIs for different kind of tools (Dev, 2014). Such APIs allow a TC to perform different types of operations over the tool. Operations such as creating an instance of the tool, assigning that instance to a user, giving read and write permissions, creating objects within tool instances, and reading data of what has been done with the tool instance are now possible. Different tools support different sets of possible operations. It is up to the TP to decide what to open through their API. It has been noted (Alario-Hoyos & Wilson, 2010) that the more mature and popular tools usually have a more open and robust API in terms of quantity and quality of operations for manipulations of their tool instances.

4.1.3 Overcome the Static Contract-based Interoperability Approach

To interoperate with the tools' Web API, it is necessary to create in the TC a custom interface. The same approach is used in other specifications, such as IMS LTI v.2 (see Section 3.3.2) (IMS, 2014) or the GLUE! System (Alario-Hoyos et al., 2013)(see Section 3.3.3). The custom interface requests the published operations of the API, and it processes the payloads of those operations. This custom interface is as large and complex as the tool's Web API is. The more operations the API permits, the larger the TC interface needs to be. This model also requires the inclusion of a new tool, a new corresponding custom interface at the TC needs to be built. As a result, any change or update in the tools' Web API (TP) represents that all custom interfaces for each one of the TPs at the TC needs to be updated, tested, and (once ready), deployed. This also has a collateral effect: during the update phase, some services may be unstable or unavailable, depending on how the TPs manage their update cycle. Maintenance of custom API interfaces has a high cost in terms of time to build and update and human resources necessary to build them through a software engineer.

4.1.4 Tool Type Generic Definition and Machine Processable APIs at Run Time

An *Application Domain Type (ADT)* definition is a generic description of a Web API for a given application domain. An application domain such as mind maps may have several company products or TPs. Another example of an application domain is an online document editor, which already exists in many TPs that provide such a tool. The ADT definition is a generic and formal representation of the Web APIs for such tools (e.g., mind maps), and this formal representation have a description of the most common features that an application domain may support through its Web API. For example, a mind map tool have several TPs, those different TPs products can have very similar set of features, or different features. Thus, with the multitude of tools available, there is no formal approach to describe an API, nor is it possible to define a common set of objects, operations, and properties for an ADT (e.g., mind maps). It is necessary to formalize a specification of the ADT. This serves to identify that API1 operation 1 is the same as API2 operation 2. With this specification it is possible to design systems that can interoperate others by linking this generic semantics (the ADT definition) to the TP specific semantics exposed in their Web API. And moreover, it helps to specify what an ADT should provide in its Web API, and with this information, the TC may choose what to support and include when considering the inclusion of an ADT.

Still, there is no standard approach for machine-processable APIs at run time. How can we create such a definition of a tool that can be interpreted at run time and avoid custom program interfaces for each new tool? A semantic approach for interoperability educational systems does not exist, although IMS LTI v.2 uses JSON-LD (JSON-LD, 2014), which can embed semantics, but it does not use these features at its full potential. A semantic approach leads to the discovery and identification of the available objects, operations and properties a tool has—all at run time, and all machine-processable. This clears the hurdles of custom interface programming for each tool. This improves the scalability of building, extending and maintaining tools.

4.2 Application Scope and Use Case

The following Section mostly republishes the work by Hernández and Gütl (2015a).

Enabling a CEE as described in Section 4.1.1 have several applications through which the educational process benefits ranging from process automation through educational orchestration to create completely new educational experiences. The main applications and research objectives identified for such an environment is listed as follows:

- **Objective 1: Interoperability for orchestration of CBT through automatically recognition and operation of Web APIs.** A VLE such as an LMS is able to use tools available on the web that have published APIs. It is possible to manipulate those tools to orchestrate an educational experience having the VLE as a starting point instead of having it as monolithic system. The educational experience can be highly distributed in several services as required, but backstage orchestration can be managed centrally.

- **Objective 2: CBT's Web API description is constructed at a higher level by semantically describe the objects, properties, operations and their relations, rather than writing standard contracts and requests.** The developer in charge of integrating the CBT in a VLE can apply a simplified process to integrate a CBT into the VLE. This eliminates the custom programming used with current interoperability processes. It is mainly a configuration process to enable a given CBT in a VLE.
- **Objective 3: Creating application domain–type specifications serves as a generic description of Web APIs**: individual TCs, the local and global learning technology consortiums, organizations, educational ministries and departments are able to create specifications of *Application Domain Types* (e.g., a specification of what a mind map should support, or a storytelling tool, a document editor, etc.). Such specification consists of 1) a set of objects to be represented in the *Application Domain Type* (e.g., objects such as a mind map instance or an idea of a mind map), 2) operations available over the object (e.g., create an object, update it) or 3) properties of the object (a mind map might have a title, a description, an owner's list, etc.). This type of specification could be based on current, future and custom needs of each organization.
- **Objective 4: TPs may choose to support partially or totally such application domain–type specifications, and the TPs have the possibility to extend such specifications.** TPs choose what parts of their API to expose. A single semantic vocabulary is used for Web API description.
- **Objective 5: TCs administrators are able to define which objects, operations and properties are supported by the TP and can be used.** TCs choose what features of the CBT enable to their users (e.g., learners and teachers of a VLE). Therefore, a limited and controlled set of features and actions are available for the users.
- **Objective 6: Teachers are able to use and orchestrate educational experiences based on cloud tools.** Full granular control and management is enabled to teachers. The teacher, in a simple form, manages authority over resources created in the CBTs, with full control over the created educational experience. The CEE is fully interoperable with the VLE, thus adoption barriers are lowered both institutionally and individually. This type of CEE provides inherent support structures to deploy *Learning Activities* using CBTs. Educational workflows can be implemented, using different pedagogical theories and models. Teachers can perform necessary interventions, changes to learning paths (individually or groups), enable interplay of several resources or CBTs, change roles, scaffold results from a CBT to the next one, use different social planes (individual, group, public, others).
- **Objective 7: Teachers and learners are able to launch a given tool within their VLE** and realize their learning experiences using those tools while interoperability with these tools is automatically orchestrated. This simplifies the usage of CBTs by both, teachers and learners. No configuration steps need to be done, the whole educational environment and scenarios can be created, all tools are used and behave as configured by the teacher. Either by choosing a specific brand tool (e.g., a Google document editor (Drive, 2014) or just by choosing the *Application Domain Type* to be used, the teacher or learner have the freedom to choose from the available tool brands. Learner focus on using the tool to perform a *Learning Activity* and do not care about the internals of

management and delivery of the assignment, thus lowering the literacy issues related to use a new CBT for the first time.

AN USE CASE

As an example use case of the interoperability of CBTs with a VLE, a teacher in a course designs a *Learning Activity*. First the teacher assigns each learner to research a topic and present a document with the required information. This document is built using the Google Drive document editor (Drive, 2014) (or any other tool available). The VLE provides automatic document instance creation for the students and assign proper document edition rights for each participant. The third step is that, based on the research outcome in the document, learners must represent the main ideas in relation to a mind map. The teacher creates a set of basic ideas, thus learners can build on the ideas and relations from that starting point. The same learner has a map created in the MindMeister tool (MindMeister, 2014) assigned to them. Each map already has a set of basic ideas on it. Once the *Learning Activity* due date has passed, the document and mind map resources is read-only for the learners, preventing further changes while the teacher assess the learning assignment. This can only be achieved if the VLE is capable of creating the document and mind map resources (each learner have his or her own independent resources). The VLE must then be capable of sharing editor permissions with the learner resources to later change those permissions to read-only. Thus, the VLE and the CBTs in this example are loosely coupled. They require not just the ability to launch the cloud-based tool from the VLE but also to manipulate the creation of resources and administer read and write permissions to groups and learners. This requires communication between the VLE and the CBTs' Web API. Furthermore, collaborative *Learning Activities* can be also modeled, using the same approach, through sharing the CBTs instances to groups of learners. This collaborative scenario brings an issue of concurrent edition and the corresponding versioning of the CBTs instances. This issue is out of the scope of this research dissertation, due the fact that versioning is dealt by the CBT provider. Some CBTs provide extensive versioning, such as with Google Drive, in contrast MindMeister provides a different approach for versioning due the nature of the tool.

4.3 A Conceptual Model for Flexible Interoperability

The following Section mostly republishes the work by Hernández and Gütl (2015a).

The conceptual model that defines a solution to the problems and challenges within the application scope presented in previous Sections is a combination of an interoperability service framework. Such framework first defines a semantic vocabulary for representing the ADT web APIs, and how each specific tool implementer can use and customize it. The corresponding interoperability architecture to support such vocabularies includes machine-to-machine communication for automatic service recognition. Finally, a conceptual approach for an orchestration of services within the context of *Learning Activities* is presented, a piece of the solution that is highly relevant to enabling CEE experiences.

4.3.1 Sematic definition for CBT Web APIs

Improved support for interoperability of Web APIs needs to be provided based on the problems and challenges, and on the scope of this research. First, each available Web API method can be identified as an operation to an object or resource. An *Object* can be anything within a given tool. It can also be the learner, the teacher, or a piece of the tool. For example, if is used a story telling builder as a tool, which has the following features: ability to create a story, add characters, acts and stages to the story. Then one story can be an object. Within the story can be additional objects such as characters, stages, acts, etc. Each of those objects has a set of *Properties* that helps to describe and utilize it. As an example, the story object may have a set as properties such as a title, a description of the story, and the owner of the story. It may include a collection of acts related to the story and how they are going to be presented. Thus, it is possible to identify a given tool with objects, their properties, and how those objects are related. Additionally, actions can be performed on an object. These are called *Operations*. Such operations could be performing a given act. This may imply running an animation that was built for that act. Figure 4.1 describes the schema of a tool with its objects and their correspondent properties and operations.

If this schema is used to identify all possible objects, properties and operations in an ADT such as storytelling, it is possible to define characteristics that such an application should have, which can then be translated to a Web API. This can also be represented as an ontology. In the ontology, an object can have properties, and through those properties can link to another object. For instance, the learner can have multiple stories, and each story may contain multiple acts. The operations exists but are not be associated to a particular object, due to the RDF restriction, because they are not conceived to support the relation of an operation to an object. This ontology could be used to identify the main objects, properties and operations an application should have for interoperability.

With the ontology that describes a particular ADT Web API, it is possible to create a *Generic Vocabulary (GV)* using semantic Web approaches (e.g. Linked Data). The GV is used to model an ADT (e.g., storytelling tools). The GV is useful to support basic definitions in a domain. For instance, a storytelling tool should have a basic number of objects such as learner, story and act. For each one, a set of properties could be defined. Over each object, an operation might be performed. Once this definition is done, using a generic, then the TP is able to reference to that GV to describe its own tool Web API. The GV can serve the TC to identify the ADT and use the GV as a base for further interoperability with tools. The TC may choose to only accept providers that conform to the GV. The TC may even choose which objects learners can manipulate and which properties learners can access. The TC may also choose to support extra features that the TP exposes, even if those features are not defined in the GV. Having a GV may help institutions, local organizations, state or province ministries, and governmental organizations to define what an ADT should include. The TP may choose to adhere to that GV by referencing it but not necessarily by changing its own Web API vocabularies.

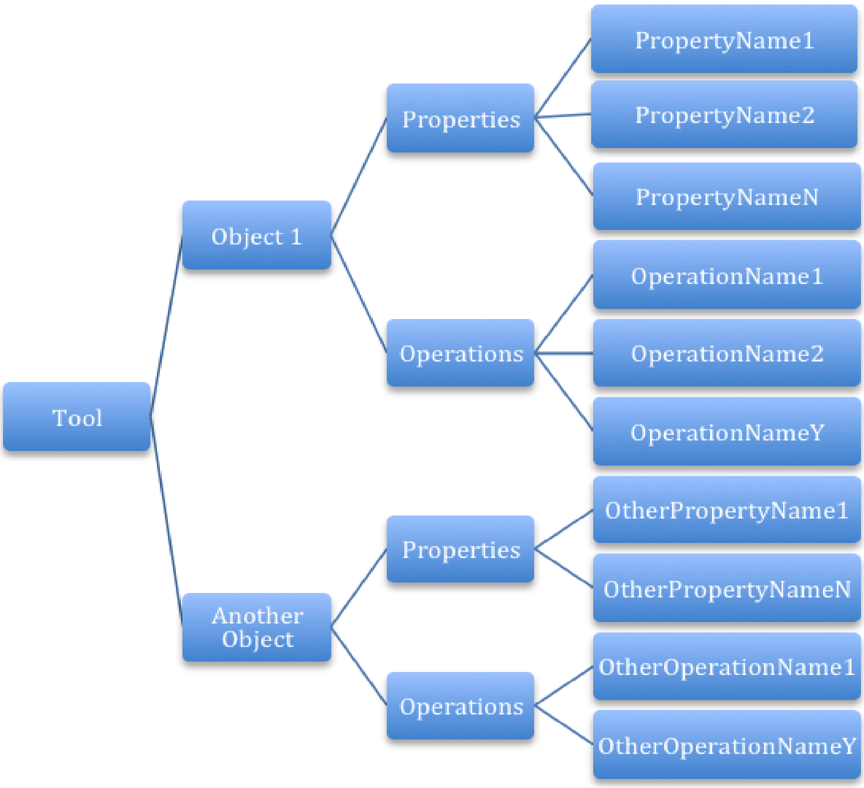


Figure 4.1 A simple schema of a tool with object: its properties and operations taken from (Hérnandez & Gütl, 2015)

Once the GV is defined, it can be used for building the specific API vocabulary. The API vocabulary can be built using the GV. It describes all the objects, properties and resources that are specific to a particular tool. Therefore, it is possible that a tool may have all the correspondent objects defined in the GV, but it might not have the same properties and operations over the objects. This hypothetic tool may also feature extra objects and corresponding operations. Following the storytelling example, the tool may have a character object which represents a character that performs something within an act. The character object may have operations like initialize performance and others. This object might not be specified in the GV but supported by the tool, and thus specified in the API vocabulary. Then, the API vocabulary also defines the properties that the object has and the type for each property. This might be something that is already defined in the GV but overridden in the API vocabulary. For the operations within the API vocabulary, it contains the specifics of what is expected such a property and what it returns, such a collection, a string, an object, what HTTP method it uses, and others. These elements are specific for each tool, therefore is not necessary to include it in the GV.

The concept of a GV has already been introduced, including how it is used by the API vocabulary. The GV is built based on the previously introduced examples of ontologies. Figure 4.2 is an example of an object, property and operation mapping among the three. As it is depicted, one ADT has its own set of necessary objects and properties, which are represented in the ontology. Then those operations and properties usually have a one-to-one representation in the GV (both represent the generics in the tool domain). Then the API Doc may have all of those, partially support them, and (or) include new ones.

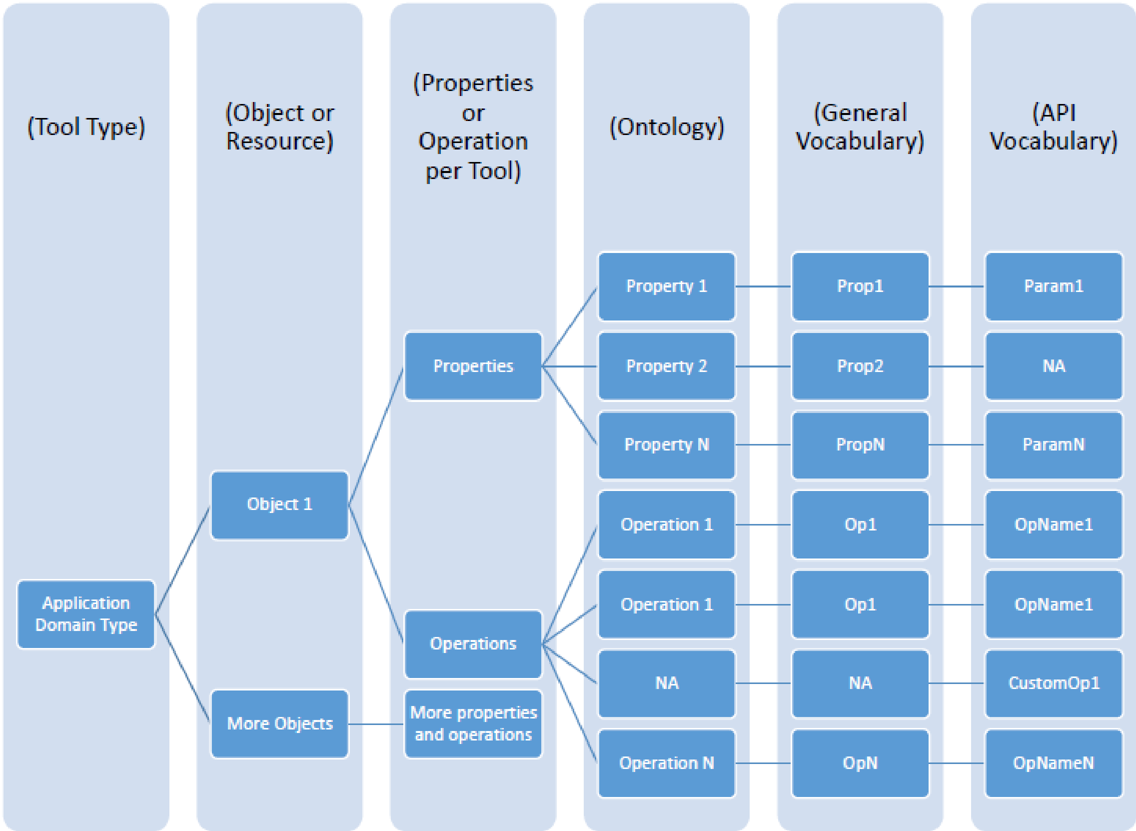


Figure 4.2 Example Ontology, GV, API Doc mapping of object, property and operations for a tool taken from (Hérnandez & Gütl, 2015)

4.3.2 Interoperability Service Framework Architecture

Now, with the defined vocabularies, both the generic and the specific per-tool (API Doc) vocabularies, it is possible to design an *Interoperability Service Framework Architecture (ISFA)* that is capable of interpreting these API vocabularies definitions and process them in run time with a generic processor. Therefore, it is not necessary to build a specific API programming interface for each tool, the semantics embedded using these vocabularies is enough to search, discover and build communication, authentication, create requests, process responses, create user interfaces, and others. All of these happening automatically at run time. The designed ISFA interoperability process has four components:

1. Semantic Runtime Interpreter
2. Authentication Handler
3. Communication Service
4. Templating System

The four components are presented in Figure 4.3, also the architecture specific components are described in Figure 4.4.

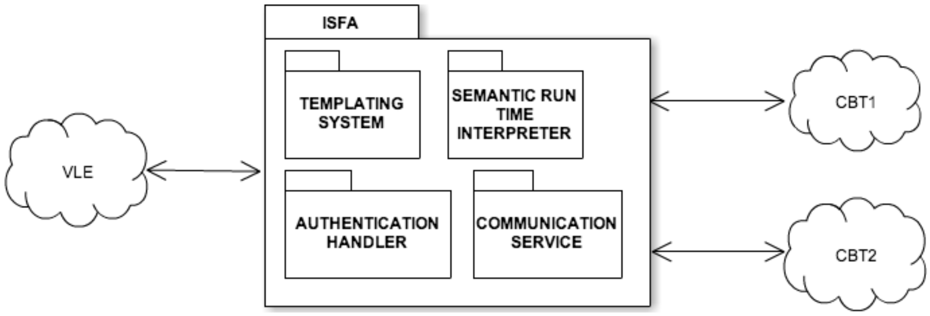


Figure 4.3 ISFA interoperability process from (Hernández & Gütl, 2015a)

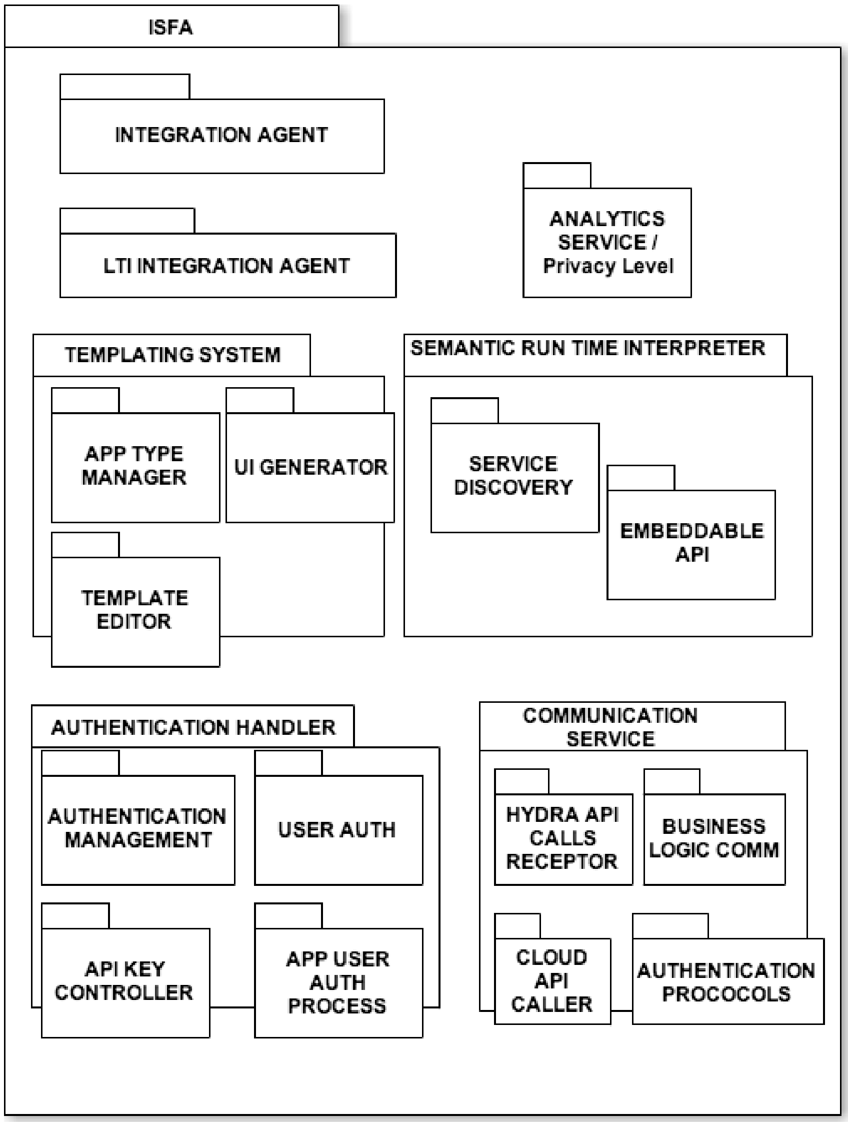


Figure 4.4 Interoperability service framework architecture (ISFA) components. Embeddable taken from (Hernández & Gütl, 2015a)

4.3.2.1 Semantic Run Time Interpreter

This is the core of the ISFA. It features a *Service Processor* with the capability to analyze an API vocabulary and the corresponding GV. Furthermore, it is capable of understanding the CBT Web API automatically at run time, including what objects, operations and properties are available and how they are related to each other. With this processed information, it is possible to create the business logic components exposed by the TP semantic Web API. In this architecture, following the current trend in composing multiple services, a current common approach to invoke a service is through the use of program utilities that can be invoked in a Web environment. Thus, a widget approach is utilized (Olmedilla, Saito, & Simon, 2006), as other architectures have chosen to integrate with third-party TPs. In the ISFA, this approach is materialized by creating an *Embeddable API* that can be invoked through a browser. In addition this *Embeddable API* can be executed independently (not within a browser). This *Embeddable API* should encapsulate and represent all the business logic identified by the *Service Processor.* Once this is available, it can be used within a browser as with a widget implementation, or in a particular system, as a third-party library to be invoked.

4.3.2.2 Authentication Handler

Every TP has an authentication protocol in place, whether it is a custom process or it is based on a standard or specification. In this ISFA, it is avoided to impose a specific authentication mechanism, such as the OAuth 2.0 authorization framework (IETF, 2012) that is token-based. Nevertheless, support for multiple specifications might be needed. As interoperability experiences have demonstrated, any specification implementation by a TP may contain specific tweaks that may require customization. Therefore, an *Authentication Management* needs to be in place to be able to define multiple protocols. This may include invoking a *User Authentication*, which may require an *Application User Authentication Process*, something that happens on the TP side. Also, it is common nowadays that an available TP Web API requires the TC to identify itself before initiating communication. This usually means that the TP gives the TC an *API* key that is used prior to any communication by the TC.

4.3.2.3 Communication Service

This service includes support for processing requests and responses from the TP semantic Web API. It is a centralized process, with a single point for communication back and forth between the TP and TC, in order to maintain authentication integrity with cross-domain issues (Armbrust, 2010). Also, it is linked to the authentication protocols and inject the necessary information into the request to keep authentication in place in a REST architectural style for web services.

4.3.2.4 Templating System

The main functionality of this system is to be a *UI Generator* that automatically creates the UI, materialized as web forms of the objects, its properties and corresponding operations. Thus, an object has a web form with all the properties as field entries, and operation over that object may allow the editing of object properties. This uses the *Service Discovery* described in the *Semantic Run Time Interpreter* to automatically create such a web form, and uses the *Embeddable API* for enabling the web form to interoperate with the TP. A *Template Editor* allows the enabling and (or) disabling of objects, properties and operations for different user types, which means that the TP may expect a property and the TC may send it automatically without requesting a user input. For example, if the property is an ID for the object, the TC chooses not to request the user to input that ID and, instead, generates it automatically. Also, the *Template Editor* is capable of basic placement and style of Web form elements (through CCS3) to provide a more consistent UI, independently of which TP is instantiated for use.

Related to the above is the *Application Type Manager*, which presents how each TP is supporting the GV, and it allows administrator to configure whether to allow or not to use TP additional features, or even limit GV features, for all TPs or on a TP-by-TP basis.

4.3.2.5 Other services

It is important to notice that, with the ISFA, it is possible to use the *Application Templates* and *Embeddable API*, as can be done in any other system, such as a VLE. The *Integration Agent* acts as a simple, yet highly secure, authentication mechanism between the ISFA and the VLE.

Other module service is the *Analytics Service*, which stores relevant learning analytics data for further processing. This may include privacy-level management. The *Analytics Service* implies that the TP semantic Web API provides such data. Finally, although is out of the scope of this research, the ISFA can have a built-in bridge to IMS LTI, thus leveraging its capabilities, such as group and user management, among others.

4.3.3 The Semantic Proxy

The ISFA interoperability process, depicted in Figure 4.6, takes into account that the CBT Web API is capable of responding using a semantic technology, it uses a, and has its own corresponding API vocabulary. But the current reality is, as of 2014, barely few CBTs uses semantic technologies. In the near future, the CBTs Web API is expected to remain as it is, and the probability of short-term adoption of semantics vocabulary as payload is very small. However, yet to support the ISFA interoperability process, it is necessary to have a *Semantic Proxy* to handle requests and responses that goes back and forth between the ISFA and the current CBT's Web API. The semantic proxy is, therefore, an intermediary layer as presented in Figure 4.5.

First the semantic proxy gets the entire request set from the ISFA and map them directly to the corresponding Web API method of the CBT. For instance, the ISFA may request an operation and sends the corresponding data. Then the semantic proxy recognizes that request, and map that operation to the corresponding method exposed in the CBT's current Web API (that does not support a semantic approach, but may support REST, JSON, XML, SOAP, etc.). Once the CBT returns the payload in any applicable format, the semantic proxy is able to process that response and, with its content, construct a payload using semantics that are processable by the ISFA. That is the main and simple purpose of the semantic proxy, yet it is a key part in this design until semantic technology is widely adopted by TPs. The semantic proxy also needs to support a standard definition to map operations and properties over the Web API and its corresponding singularities. This includes proper management of extra properties that may be needed in the back-and-forth communication. Also, the proxy needs to support different type of payloads, communication protocols, etc.

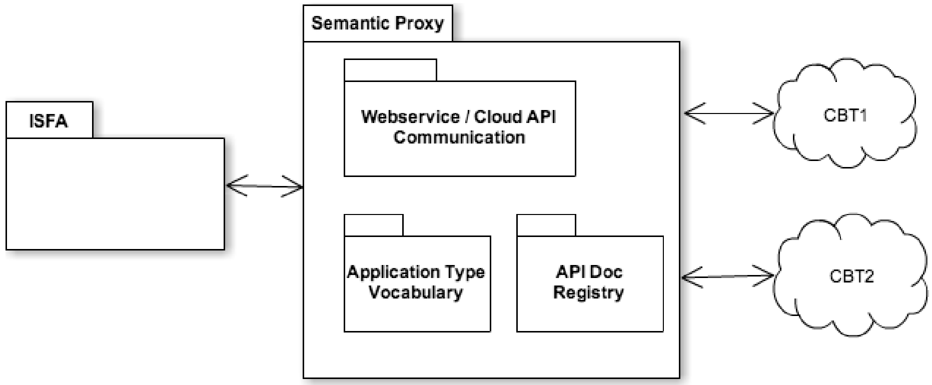


Figure 4.5 Semantic proxy to enable communication between ISFA and the CBTs. taken from (Hernández & Gütl, 2015a)

The Semantic proxy features a way to register new Web APIs. Firstly, a *Service Designer* has a GV for the ADT in place. Then, the *Service Designer* creates the specific API vocabulary for the new CBT Web API to support. This API vocabulary matches with the available methods on the current CBT Web API. Once the API vocabulary is defined, a *Formal Mapping Description* is created to match API vocabulary definitions to CBT Web API methods. This includes authentication type, method URI, response format, response object and properties, error codes, etc., each with its corresponding mapping to the API vocabulary. This *Formal Mapping Description* may be done manually or with a user interface.

4.3.4 Orchestration Learning Activities System

To enable a CEE to use the CBTs through the *Interoperability Service Framework Architecture (ISFA), an Orchestration Learning Activities System (OLAS)* has been designed, which easily enables the use, configuration, management and integration of the CBTs available. The OLAS is an interdependent system, which works between the VLE and the ISFA to enable the usage of CBTs for education in the context of a *Learning Activity*.

Figure 4.6 depicts how the OLAS could work along with the VLE, whether the VLE is a PLE, LMS, LCMS, etc. The OLAS launches the TPs through the ISFA. First, it acts as the user interface for educational settings of the ISFA by providing templates of the integrated CBTs. This is a lightweight system or package within a VLE, as is depicted in Figure 4.6 (a). The approach is that the VLE contains a simple bridge to send information of the teachers and learners from the VLE to the OLAS standalone system, where the *Learning Activity* using the CBTs is configured, performed and delivered. In this case, within the VLE exists only a native package implementation sending back-and-forth single sign-on (SSO) user credential authentication and corresponding data that needs to be exchanged, such as user and group information, results from activities, etc. For option (b), the OLAS do not run as a stand-alone system, but it does run within the VLE as a native package. This leads to simplified communication between the OLAS package and the rest of the VLE, and it do not require a SSO. Both options (a) and (b) have the same set of functionalities, although option (a) requires a lightweight integration to the VLE, enabling faster adoption by VLE products and easy maintenance. However option (a) requires the setup of another system, the OLAS to support its deployment. Also, it do not have user interface space constraints because it only displays itself. In option (b), the OLAS runs within the VLE and thus is expected to present its interface within an already-created layout or template (e.g., a block in Moodle, a portlet in .LRN or a widget), thus limiting the available space, which is important when launching the TPs' interfaces. It is possible to overcome that trouble by adding a modal view (an emergent window within the browser current view) that can use the full space available without constraints, such as dealing with CSS, or by simply opening a new browser window. The (b) approach requires a larger custom package construction for each VLE to enable it, as it requires native framework programming, database management, etc. User and group context are native for this approach. Both of the approaches are plausible. For this research, we have chosen the (b) solution because it requires less custom programming for each VLE to integrate with. First, OLAS has an *Orchestration Service* that provides the creation of *Learning Activities* and the corresponding CBTs' resources that are present within the educational context grouping, such as class, group class, or individual assignments, which are known as *VLE Contexts*.

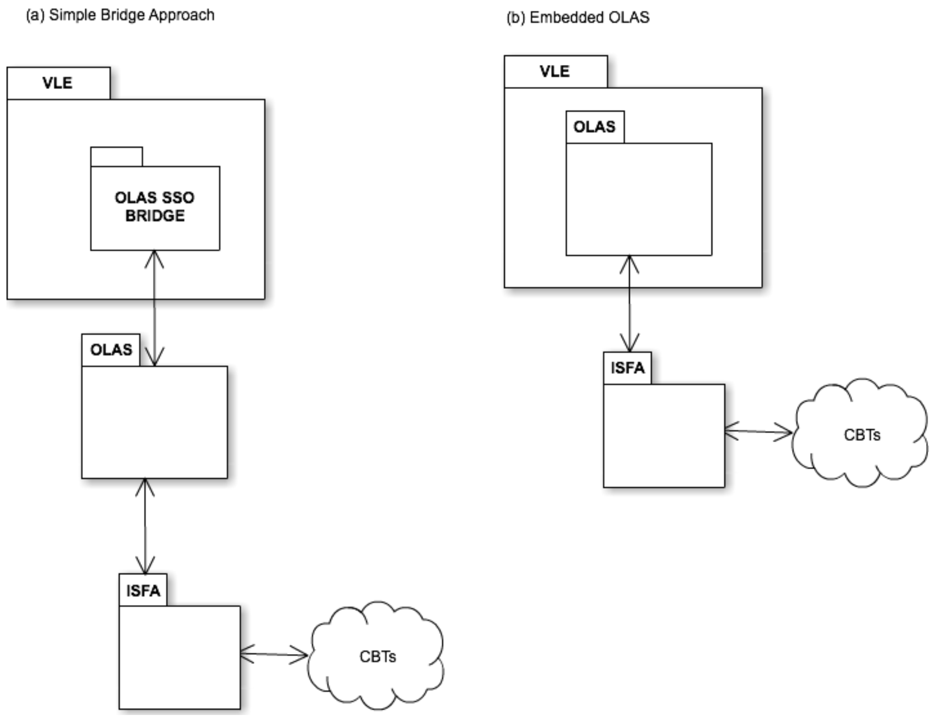


Figure 4.6 The VLE & OLAS integration approaches taken from (Hernández & Gütl, 2015a).

Figure 4.7. In an OLAS *Learning Activity*, it is possible to create CBT instance resources for the learners to work with, and those instances can be a copy of a CBT instance already created, which therefore acts as a template for other instances. For example, it is possible to create a document with a given index to be filled out by the learner or a mind map with the main ideas to be extended. Also, it enables the user to automatically manage CBT instance creation. An instance in this case would be the resource that would be given to the learner to work with—for example, a mind map resource created only for an individual learner. Each learner linked to a *Learning Activity* may have his or her own mind map to work with.

This involves creating the resource instance and assigning the proper *Editor* permissions. This may also extend to revoking permissions and giving a *Read*-only permission, which is suitable in the event that the *Learning Activity*'s due date has already passed and therefore the learner cannot longer modify the resource instance and can only view it.

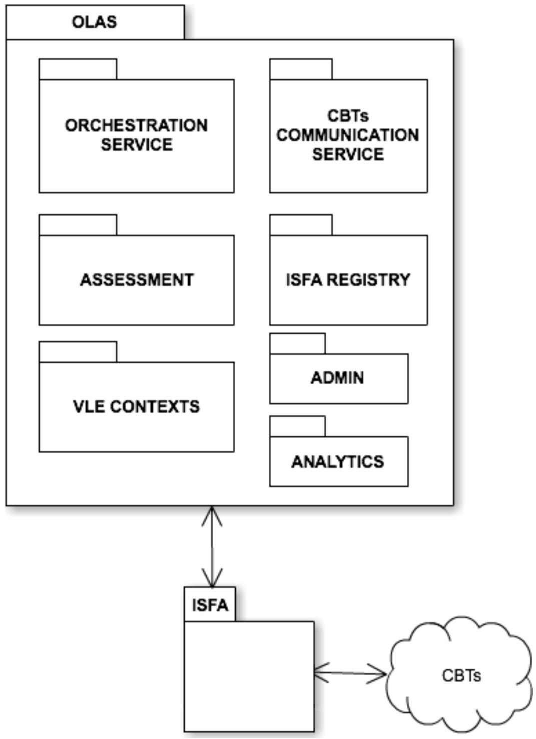


Figure 4.7 The Orchestration Learning Activities System (OLAS) Architecture taken from (Hernández & Gütl, 2015a)

The OLAS has the capability to create *Learning Activities* that are composed of using multiple CBTs (e.g., first make a summary in an online document, then extract main ideas into a mind map and finally represent it using a storytelling tool). These CBTs' orchestrations may include communication between them. Thus, they require a *CBTs' Communication Service*, such as the interwidget communication like in the known examples in a PLE (Kirschenmann et al., 2010). This helps to automate the information back and forth between CBTs, for instance, from the previous example, extracting titles from a document to automatically create a mind map of ideas. Then, from the mind map structure, the user can create a corresponding structure using storytelling tools, such as acts and scenes.

The OLAS includes the association of teachers and learners, enabling a common environment to perform and access the *Learning Activities*. Once a given *Learning Activity* piece has been designed, the system allows the teacher to deliver it to the learners.

Once the learner has completed the *Learning Activity*, the OLAS connection results from the *Learning Activities* to the VLE assignments and or gradebook solution (or other correspondent control system that requires such an input for further grading), thereby sending results from the CBTs directly to the VLE. The results may be in a format that needs to be assessed (e.g., a mind map tool delivers a mind map instance that has been assigned to a student or group of students, and this mind map then is assessed and graded by the teacher, a teacher assistant or a similar person using a manual process). Or it might also be conceivable to send the current grade obtained by performing a practice in a CBT, which may have an automatic grading process. All this process may include the launch of corresponding assessment methods available in the VLE, CBTs and (or) third-party systems.

The aim with the OLAS is to be simple and flexible enough to not impose any pedagogical or methodological paradigm, approach or model because the purpose is to create a workspace environment to use CBTs for learners and teachers, rather than leaning toward pedagogy.

There is VLEs for *Learning Activities* management, using corresponding specifications, such as the IMS Learning Design (IMS, 2013). Nevertheless, it is out of the scope of this research to use them because it would require possible extensive adaptations to the systems or the building of new layers. Therefore, it is proposed a simple yet powerful interface to use in conjunction with the ISFA. The OLAS enables simple workflow management with basic thresholds to pass from on part (e.g. tool) of the *Learning Activity* to the next one. The thresholds are represented by a quantitative indicators that must be met before allowing the learner the go to the next step of the workflow (e.g., a minimum number of ideas in a map or a minimum length of a document counted in number of words). These workflow thresholds can be obtained and calculated from CBT Web API.

There is an *Administration* interface for VLE managers to configure what CBTs services and capabilities can be used. Such configurations are provided and stored in the ISFA but accessed via OLAS. One of the final features is to enable a bridge to better capture *Analytics*, especially user behavior within the OLAS user interface. To use the ISFA, the OLAS is required to have an *ISFA Registry* that identifies and authenticate the use of the ISFA. This emulates the API key for a TC to a TP.

4.4 Discussion

The following Section is partially based on Hernández and Gütl (2015a).

Currently, to enable interoperability between a VLE and Web API of CBTs, custom-made interfaces for each tool are required to be integrated, that is programming interfaces of each Web API. This implies a considerable effort and costs to build and update each interface. The presented solution enhances that approach, as it converts Web APIs into APIs powered by semantic technologies. The conceptual model describes a robust way for the TC to control and administer tool objects, defining its properties and possible operations. At the same time, it simplifies interoperability efforts, enables a general abstraction for an ADT (e.g., a mind map) through the GV and permits the tool to be discoverable—for example, to find all operations available for a tool at run time without the customized code becoming involved. And it is Web APIs' discoverability that reduces the necessary effort and costs to deploy it. Instead of manually programming Web contracts between the TC and TP, it is just a matter of defining it as described in this Chapter.

The presented conceptual model simplifies interoperability for education and allows access to the complete set of features offered by the Web API. Thus, it permits the users to create tailored learning experiences and utilize VLE management controls that are not currently available, as described in the use case. With this interoperability approach, it is possible to enable *Learning Orchestration* mechanisms (Hernández, et al. 2013a; Hernández, et al. 2014c) that can enhance the learner experience through the use of CBTs.

Another very promising capability is that once the API vocabulary of a CBT is defined, it is possible to design an automatic user interface constructor through the ISFA templating system. Furthermore it is possible to provide custom UI controls for operations, and those operations can also enable the automation of educational course management, such as learners' group creations, roles' management, etc. Analytics services can be embedded into the interoperability process, and correspondent service operations can be defined using semantics, both in GV and in the API vocabulary.

A large majority of CBTs do not use semantics yet, so there is a need to create an intermediate stage. Once the intermediary is in place (the semantic proxy), a TC that is capable of using semantics is capable to consume any standard Web API,. The ISFA is able to automatically process any CBT semantic Web API through its semantic run-time interpreter, and it then provides the necessary controls for the VLE and the OLAS to manage and orchestrate educational experiences. All authentications are handled as well by ISFA both at the user and application-to-application level. Then, the OLAS, by using ISFA for interoperability, is enabled the creation and orchestration of *Learning Activities*, such as the one described in the use case of Section 4.2. As a result, the use case is fully supported.

With all of the previously mentioned architecture components, it is possible to create a CEE in the current VLE (Hernández et al., 2014b) and add new CBTs relatively quickly because no further custom programming is needed either on the TC or the TP side. Tool description is done at a higher level, simply using semantic technology.

The software development of the conceptual model presented in this Chapter for flexible interoperability is planned as an iterative process. To address the interoperability by building the ISFA three iterations are planned. The first iteration (presented in Chapter 5) is to build the ISFA by means of test the interoperability with current CBTs Web API, this includes all the aspects such as authentication, communication, etc. Thereby in this first iteration a third-party environment system is chosen for display the user interaction interface. Then as a second iteration (also in Chapter 5), new enhancements are made to the ISFA, and is included the construction of the OLAS. Before the last software development iteration, it requires to first defining the semantic technology to use and correspondent semantic vocabularies to be used for CBT Web API (to be elaborated in Chapter 6). Finally, the third iteration, in Chapter 7, is presented the ISFA where the semantic interoperability concepts and technologies are completely implemented, along with the semantic proxy required to enable communication with CBTs.

5 Cloud Interoperability and the Cloud Learning Activities Orchestration System

In this Chapter the *Cloud Interoperability Service (CIS)* architecture is presented, which is a middleware service that is capable to interoperate with current CBT Web API. Then the *Cloud Learning Activities Orchestration System (CLAO)* is described, which works along with the CIS. The objectives are that the teachers are able to use and orchestrate educational experiences using CBTs, and both teachers and learners can enact integrated educational experiences from any VLE.

First in Section 5.1 the technology used for the described systems is presented. This includes the software development framework, database and data processing technologies, along with cloud infrastructure and third-party systems used.

For the software development presented in this Chapter the research methodology described in Section 1.3 is followed, in particular referring to the *Iterative Development Process,* introducing the first and second iteration of the process. The first software development iteration, includes communication, authentication and analytics, see Section 5.2. For the first iteration of the development process, a *Personal Learning Environment (PLE)* named ROLE (ROLE, 2013) is presented, which has been chosen as the user interface to concentrate on the CIS architecture. The CIS first generation focuses on a main infrastructure to enable the use of *Cloud Based Tools (CBT)* in a personal learning environment. It also considers the use of those CBTs within the PLE widget based environment. This first generation contains an *Analytics Layer* that is then used for evaluation on real test-beds. However, in this Chapter the semantic interoperability approach is excluded, to address later that challenge.

Therefore, in Section 5.3, with this first generation CIS built, a second software development iteration creates the CLAO, as designed in Section 4.2. The CLAO serves as a user interface for creating and orchestrating *Learning Activities*, and is fully integrated with a VLE. Along with the CLAO a second generation of CIS architecture is introduced, with a revamped model towards building the final architecture. This CIS second generation introduces the *CIS Services Modules*, which are built for each CBT, implementing interfaces to interoperate. A more fine-grained analytics layer was incorporated, allowing accessing and analyzing data from the CBT usage itself.

This Chapter is based on research published in (Hernández, Linares, Mikroyannidis, & Schmitz, 2013a; Hernández, Gütl, & Amado-Salvatierra 2014c).

5.1 Technology Infrastructure

The following Section presents the different technologies used for building the conceptual model and framework architecture described in Section 4.3. It consists on several pieces, a software development framework and its corresponding programming language, third-party libraries and systems, databases, service infrastructure and others.

SOFTWARE DEVELOPMENT FRAMEWORK

For the development of the systems within this doctoral dissertation, it was planned that one of the technologies to be tested is JSON-LD[1], therefore it become highly important to find such a development framework capable of automatically processing JSON-LD. At the time of the system building, there exist only on *JSON-LD Processor* (Lanthaler, 2014)., which is built to be used in the *Symfony Web Application Framework*[2], which uses the PHP language. Furthermore, PHP support the *object oriented programming* paradigm (Rentsch, 1982), something highly relevant to be able to better build the target architecture.

Symfony is PHP framework for creating Web sites and Web applications, using Symfony components. The framework consists on two major parts: (1) a *Toolbox*, which are a set of already built and easy to integrate software components. The advantages are writing less code and therefore make software less error prone. (2) A *Methodology*, that works as an "assembly diagram" for applications. This methodology is a structured approach to efficiently and effectively built complex applications, while ensuring application stability, maintainability and upgradeability.

Symfony supports the *Model View Controller (MVC)* paradigm (Krasner & Pope, 1988), which is an architectural pattern for creating interfaces. A given application is divided in three parts, first the *model*, which reflects the application business logic, the problem domain, independently of the user interface. The *model* manages the data, the logic and rules of the application. The second part is the *view*, which is in charge of presenting the information. The third part is the *controller*, which deals with the inputs and converts them to commands for the *model* or *view.* Having the MVC paradigm in place also helps to better build the architecture, where is clearly separated the user interfaces from the business logic, as is described along all the software iterations of this thesis.

1 http://www.json-ld.org

2 http://symfony.com

Additionally, for the user interaction and interface layer is heavily used JavaScript (Flanagan, 2002), the de facto standard programming language for running applications within a Web browser. This provides two main advantages, first, browser providers ensure to correctly and efficiently support JavaScript, and thus it performs with high efficiency. Second, libraries built atop of JavaScript permits to easily be adapted by third-party users, which is needed for the architecture and is further elaborated in the following Chapters.

DATABASES, DATA PROCESSING AND VISUALIZATION

As for database has been chosen the PostgreSQL database[3]. Which is a widely known database, with high performance and scalability. It provides advanced replication and load balancing techniques, which are essential for educational environments such as MOOCs where scalability is highly relevant. This database is used as the backend for the Symfony framework.

Additionally, for the *Learning Analytics* supporting technology, in particular for storing, processing and visualizing large amounts of data two different technologies were selected, both based on their technical capabilities and scalability. (1) The *Google Fusion Tables*[4], which is an experimental data visualization Web application, where is possible to create data tables and manipulate them to extract information and visualized it. Furthermore, it provides a Web API, thus permitting to automatically feed the table with data from other systems, this data feeding could happen at real time.

(2) The second technology used for *Learning Analytics* is *Google Analytics* (Google, 2014), which provides means to measure user interactions across a diversity of devices and environments, while having the Google computing speed and scalability built in. The different tools provided by *Google Analytics* (GA) enable to obtain new insights of user behavior, and with that valuable information optimize the whole educational experience. GA comes with a Web API and a *Software Development Kit* which makes easy the integration and usage of the entire GA, and also provides means to automatically feed information into it.

THIRD-PARTY SYSTEMS, CLOUD COMPUTING AND OTHERS

As is described in the following section 5.2, the first approach for enabling a *Cloud Education Environment*, is to use as educational environment a *Personal Learning Environment (PLE)* through the ROLE system (ROLE, 2013), see also Section 3.3.1. To implement the required system new widgets (OpenSocial, 2014; Cáceres, 2010) for each CBT needs to be built. A group of widgets is denominated as widget bundle, and some widget bundles need to be selected to enable the educational experience. Additionally, the *Inter-Widget Communication* to send messages between widgets needs to be implemented.

---

3 http://www.postgresql.org

4 http://www.google.com/fusiontables/Home/

5 http://git-scm.com/

The course material is delivered to students at the Galileo University using a customized version of the *.LRN Learning Management System* (LMS) (Hernández, Pardo, & Delgado, 2007). Student-to-student communication is also supported through dedicated online forums. Teachers and instructional designers create and upload all teaching and learning material into the LMS. The architecture to be built integrates with any LMS, and the examples presented in further Chapters integrate with .LRN.

The education environments were fully enabled on a cloud computing infrastructure. The cloud infrastructure of Amazon's Elastic Compute Cloud (EC2) (Barr, Varia, & Wood, 2006) for hosting the whole deployed systems. Cloud technologies can be dynamically adapted allowing optimum resource utilization and provides availability, flexibility and scalability (Leavitt, 2009), and important factor for further deployment of the system on a university wide scale. Thus, the implemented environments can serve as true test-beds for cloud-education environments.

Finally, as source control system for all the software pieces to be developed and used, it has been selected GIT[5], which is a distributed revision control system and provides support for distributed, Web-based workflows.

5.2 Cloud Interoperability within a Personal Learning Environment

The following Section is partially republishes the work published by Hernández, Linares, Mikroyannidis and Schmitz (2013a).

Towards a cloud interoperability and orchestration architecture for educational environments, the research methodology (see Section 1.3) designed for this doctoral dissertation focuses on 3 development cycles, where each iteration improves on the previous one. The objective is that the interoperability service can be used in different types of VLEs. Therefore, the first approach was to investigate the potential of using a cloud-based infrastructure in order to enable a *Personal Learning Environment (PLE)* using the ROLE system (ROLE, 2013). This is the first iteration of the software development process. To enact the CBTs within the PLE it is necessary to build an interoperability service, which is introduced in the following Section.

5.2.1 Cloud Interoperability Service First Generation

To address the interoperability of different CBTs with the ROLE system (ROLE, 2013), it was necessary to create an integration service that acts as a backend service to access the RESTful based APIs of the CBTs used for the widgets. This is the *Cloud Interoperability Service (CIS)* that serves as an interoperability architecture, it hosts many services as are described, and address the *Same Origin Policy* restriction (W3C, 2014). In Figure 5.1, is depicted how does the CIS integrates with the ROLE PLE, which due its loosely-coupled architecture through the use of widgets, allows to simply embed the created widgets into ROLE, while the widget itself have the corresponding business logic to interact with the CIS.

As shown in Figure 5.1, the architecture built for the CIS consists of the following layers: the first layer is a *Business Logic Layer* that contains a *Functional Widget Interface (FWI)* JavaScript-based library (Soylu, Mödritscher, Wild, De Causmaecker, & Desmet, 2012). It contains all the business logic necessary to implement the required behavior of the widget. This layer is used directly by the widget within the ROLE system.

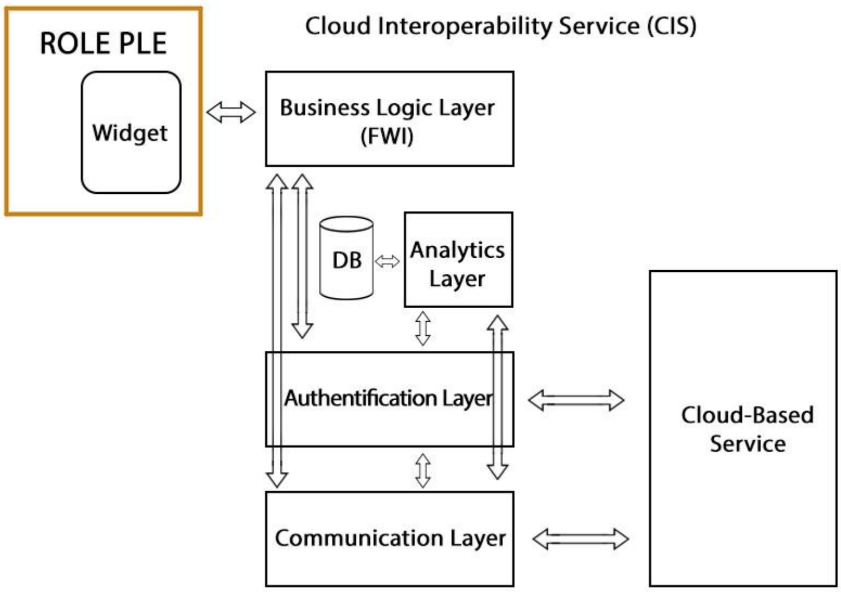


Figure 5.1 The Cloud Interoperability Service taken from (Hernández et al., 2013a)

As a second layer, the *Analytics Layer* is used to record user behavior and interaction data from the CBTs, and send it to an analytics database for further processing. The recorded data provide useful information for usage analysis, collaboration level and behavioral patterns in CBTs. All actions that the CBT makes available are retrieved via this layer, i.e. for the MindMeister CBT we collect information on how many *Ideas* were created, who created and edited them, at which date and time, among other relevant information. Thereby, this layer provides the ability to collect information on the CBTs. The combination of this information with further information collected from the ROLE Contextualized Attention Metadata (Schmitz, 2011) renders a more accurate image of user behavior.

The third layer of the CIS is the *Authentication Layer,* that handles the required tokens exchange for application and user authentication: the tokens are used for future RESTful API calls. Authentication takes place between the CIS and the CBT: first the CIS is authenticated to the CBT itself with a registered application key (api-key) provided by the CBT which is used to identify the CIS for accessing the CBT API, and second the users are authenticated using the login url provided by the CBT with the corresponding parameters for credentials and requested access permission. After obtaining the authentication token this layer makes it available to other layers as needed or for third party services where the CIS is being used. (these CBTs are accessed via widgets within the ROLE system). Although ROLE provides mechanisms for OAuth, either as a consumer or provider behavior, the tokens-based authentication is not supported, therefore it is included in the CIS and it serves for future CIS generations that have to work standalone of the ROLE system. Figure 5.2 shows the CIS first-time authentication mechanism: it begins with having the learner login into the CBT, and grants access to the CIS and thus the ROLE system. This token is stored in the Google Fusion Tables for further sharing it with the ROLE system.

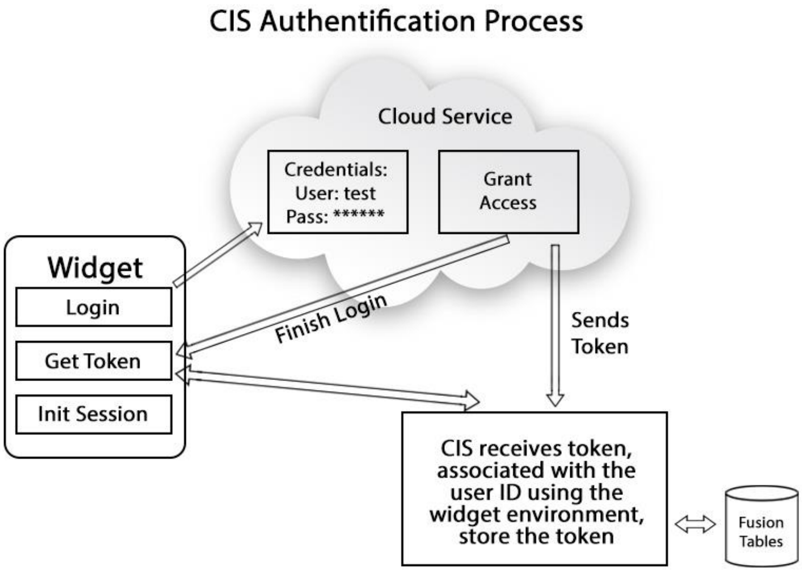


Figure 5.2 CIS First-time Authentication Process taken from (Hernández et al., 2013a)

As the fourth layer, RESTful requests to interoperate with the CBT is handled via a *Communication Layer*. The *Business Logic Layer* uses the *Communication Layer* and the *Authentication Layer* to enable the use of CBTs.

5.2.2 CEE: ROLE and Two Widget Bundles

In order to create a comprehensive learning experience for learners (Hernández et al., 2013a), it has been decided to select two different widget bundles: the first widget bundle consisted of the following six widgets: ObjectSpot, Binocs Media Search, MediaList, EtherdPad, MindMeister Mind Map and Facebook. The second widget bundle included three widgets, namely Google Drive, MindMeister Mind Map and Facebook (ROLE, 2013).

The *ObjectSpot* search widget allows learners to find online resources from a variety of bibliographic sources, including CiteSeer and Google Scholar. *Binocs* focuses on media search, allowing users to search for learning content in various Web 2.0 platforms like YouTube, SlideShare, and Wikipedia. Additionally, both widgets provide access to repositories of Open Educational Resources (OER), containing free learning material of high quality. Some of these repositories are OpenLearn (OpenLearn, 2014), OpenScout (OpenScout, 2009), and Globe (GLOBE, 2014).

The *Media List Widget* allows the user to create custom media lists based on the search results from the Binocs widget. The *EtherPad widget* is a text editor that allows users to write a document collaboratively with their peers in real-time. When multiple authors edit the same document simultaneously, any changes are instantly reflected on everyone's screen. This is particularly useful for meeting notes, drafting sessions, education, team programming, and more.

In the second widget bundle, it was added a *Google Drive Widget* (see Figure 5.3) that enables the users to interact with their files. Google Drive allows the user to store files, share exactly with whom they want and create documents using the Google Drive suite. The 'widgetification' of this service renders these actions available within the widget environment, that is, the PLE. The widget was developed based on the OpenSocial specification, using the Google Drive SDK and the Google Drive Web API technologies, as well as the JavaScript client library provided by Google. This library provides functionality for authenticating through OAuth, obtaining user information and interacting with files and documents. Because Google provides this library, it was not necessary to use the CIS in this case, since all the CIS provided functionalities are included within the library.

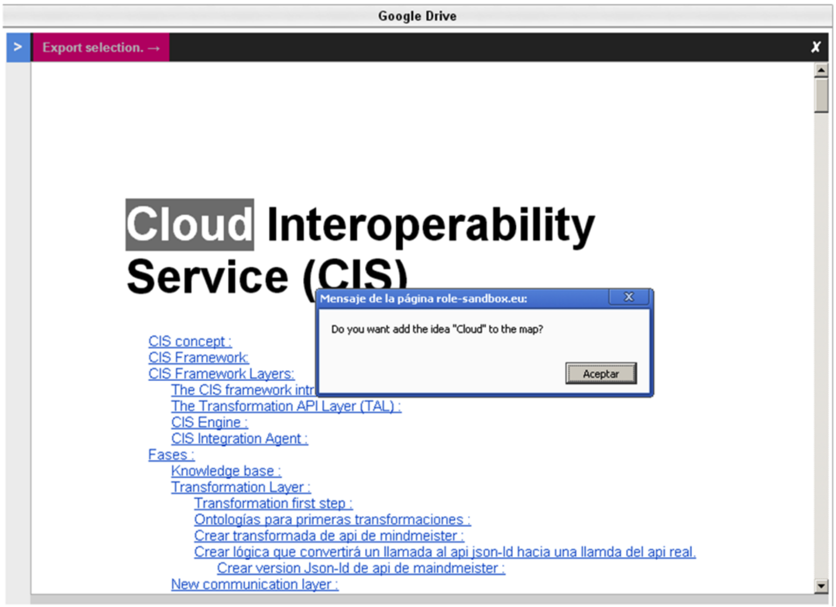


Figure 5.3 The Google Drive widget in a selection mode to send text to the mind map taken from (Hernández et al., 2013a)

Inter-Widget Communication (IWC) functionalities enable the user to select a text from the document and send it as an IWC event (also as an OpenApp event) to create a new mind map node. Getting the selected text within a document was performed by getting the static HTML version of a document through the Google Drive client library and inserting it into the widget as a "preview" version. Since this view was already available in the widget, the text could be accessed through the CIS.

5.2.3 Evaluation and Lessons Learned

The possibility to include and manage several CBTs and creating learning paths opens a large set of opportunities for enhancing and expanding the construction of learning experiences. What still remain to be addressed is how to assure the student follows the learning path. Ensuring that the student do not gets lost, confused or distracted in the CBT.

Further improvement to CBTs analytics is required, such as gathering and analyzing data from the CBT environment itself, not just about the use within a VLE or a PLE, thus further enhancements for better analytics are needed.

The CIS architecture was proven to be a key component to allow interoperability between CBTs and VLES such as the ROLE PLE. It was foreseen enhancements in the CIS including automatic service composition for improving interoperability and inclusion of a templating system for building user interfaces and more tailored learning path.

The mind map widget is a tool that delivers the functionality to create collaborative mind maps and reuse previously created maps as learning resources. The mind map widget uses the OpenSocial specification, as well as the MindMeister Web API (MindMeister, 2014).

The mind mapping editor enables the user to create and edit maps, ideas, nodes, and other actions. To achieve the required (or specified) operation and to receive elements from other widgets and incorporate them automatically into the map the Open Application specification is utilized (OpenApp, 2012). It was designed and implemented using business logic and authentication (or communication) libraries in order to interoperate with the MindMeister RESTful services using the CIS. This allows systems communication for interacting with the mind map (instead of a simple map embed) for the publication and listening of widget events (i.e. add an item to a map, a map published for discussion, etc.).

The Facebook discussion widget was implemented according to the OpenSocial Gadget specification (Facebook, 2012). The widget offers a comments area for collaboration and communication about a mind-map or document.

To further progress on the envisioned ROLE real-time communication and collaboration infrastructure, it was necessary to provide communication between widgets, especially for sending events originating in various widgets (ObjectSpot, Binocs Media Search, MediaList, EtherdPad) to the mind-mapping widget. Likewise, in the case of the ObjectSpot widget, it was necessary to enable communication through events according to the Open Application specification. In the case of the MediaList widget, however, it was required to be able to add a new broadcast event to send items stored in a list to the mind-map and reflect them as new nodes in the map. Figure 5.4 shows the first widget bundle architecture and Figure 5.5 a screenshot of that first widget bundle in action.

The CIS architecture has proven to be successful for enabling interoperability with CBT Web API, including the different authentication and communication protocols. This serves as a base for future architecture improvements and next software development iterations. Although key shortcomings have been identified, such as it still needs a general approach for business logic processing. Additionally the ability to generate the user interface and interaction controls, in contrast to just focus on generating widgets for the ROLE PLE. Moreover, was determined that processing analytics data can be a burden. This due the fact that the raw data obtained needs to be organized, presented (usually with the help of graphics) and finally evaluated. A new analytics tool needs to be selected to consequently feed it with the raw data, otherwise analytics may be too difficult.

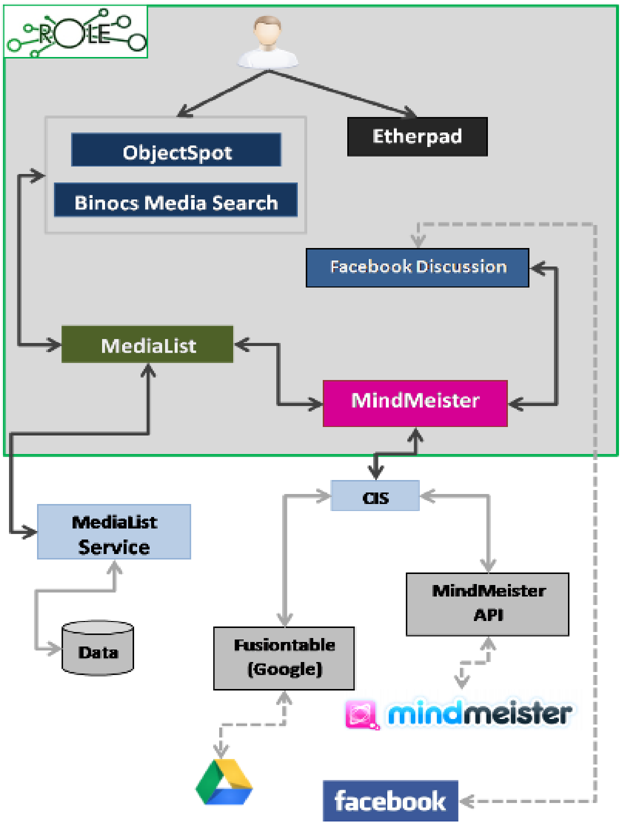


Figure 5.4 The first widget bundle architecture taken from (Hernández et al., 2013a).

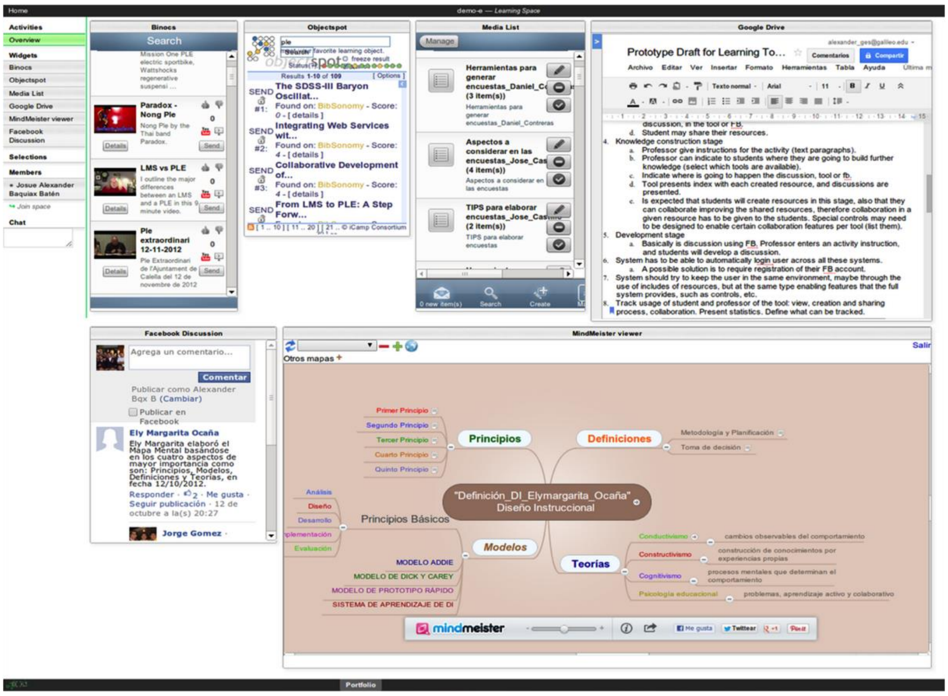


Figure 5.5 The widget bundle in action taken from (Hernández et al., 2013a)

5.3 The Cloud Learning Activities Orchestration System

The following Section includes parts of the publication by Hernández, et al. (2014c).

Coordinating interventions from learners across multiple *Learning Activities* to enhance productivity is a process known as *Learning Orchestration*. The teacher designs *Learning Activities*, assessments, rubrics and monitors group or individual activities. The teacher also distributes workload and sets deadlines among others. As an example, the teacher designs a set of steps to perform a *Learning Activity* that might require using a mediating artifact. A mediating artifact is as the use of *"different tools and resources can provide support and guidance on the context of a learning activity, the choice of pedagogy, the creation of associated learner tasks or any combination of these. They range from contextually rich illustrative examples of good practice (case studies, guidelines, narratives, etc.) to more abstract forms of representation that distil out the 'essences' of good practice (models or patterns)"* (Conole & Alevizou, 2010). In the use case in Section 4.2, the mediating artifact is the modeled use of the document editor and the mind map produced by the learners while they are doing the *Learning Activity*.

To create a CEE that is capable to orchestrate *Learning Activities* using CBT, a *Cloud Learning Activities Orchestration (CLAO)* has been created in Hernández et al. (2014c), which is an infrastructure that allows teachers and self-organized learning groups to perform the aforementioned *Learning Orchestration* scenarios using CBT. In the following Subsection is presented the CIS which handle all the interoperability between the CLAO and each CBT. And in the Subsection is introduced the CLAO.

5.3.1 Cloud Interoperability System Second Generation

The following Section presents the second iteration of the software development process for building the CIS. The CIS is capable to integrate, reuse and personalize each of the CBTs that can be added to CLAO. It achieves this interoperability through a definition of services and through definition of a common interface of communication. The CIS is based on (Symfony, 2014), a Web development framework (PHP), refer to Section 5.1. CIS extends Symfony2 by implementing a custom bundle.

The CIS is divided into four layers (see Figure 5.6). First the *Communication Layer*, which in CIS identifies each CBT that can be used for learning, and prepares for each of these CBTs a custom integrated service communication bundle. The *Communication Layer* includes a Symfony controller that administers each CBT service bundle, the controller also communication related data with the assignments module that sends information back and forth to the LMS. The CBT service bundle contains all related logic to enact communication with the CBT (further details are given in the following pages). Within this layer, tracking data is sent to be stored and used by the *Analytics Layer*. This layer main objective is to perform all the API requests between the CIS and the CBT public API.

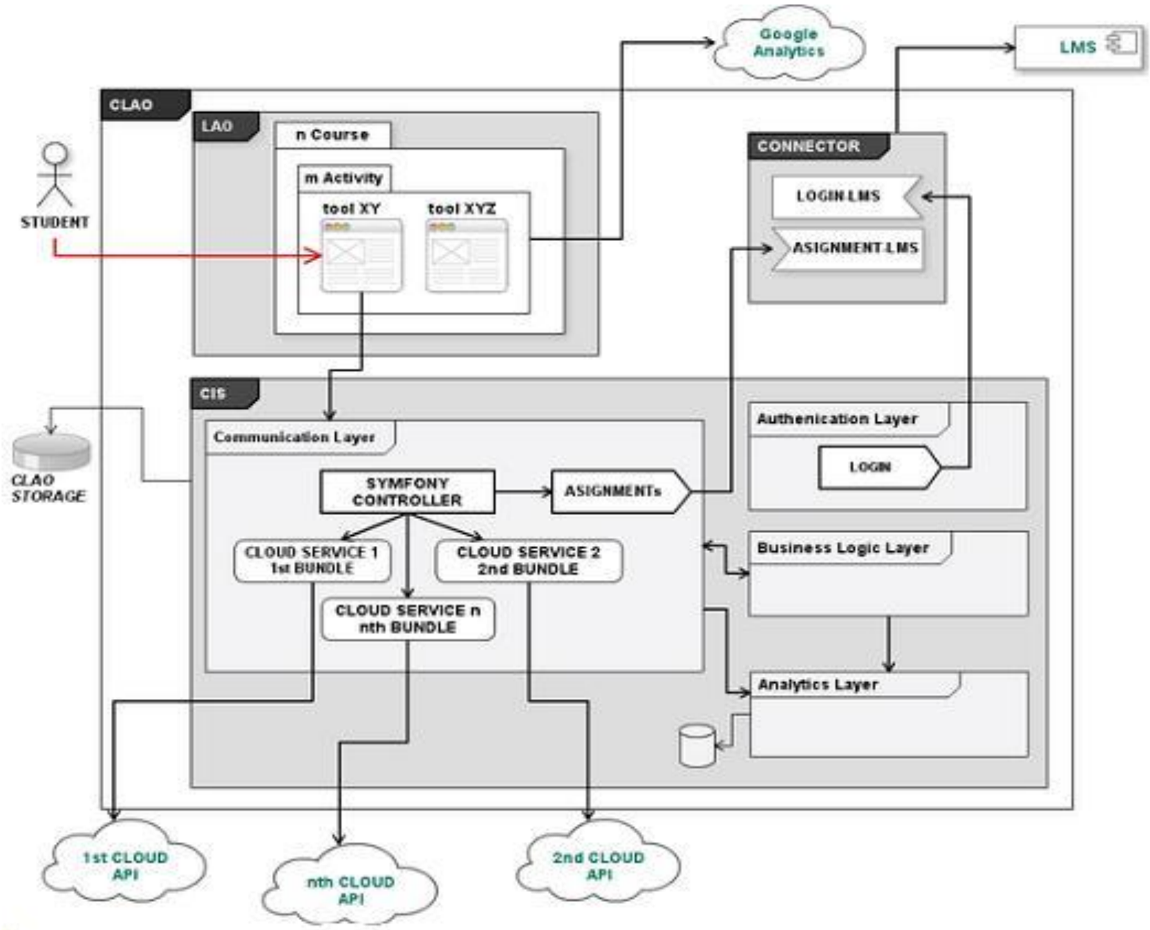


Figure 5.6 Cloud Learning Activities Orchestration (CLAO) Architecture taken from (Hernández et al., 2014c)

As second layer, the CIS architecture (contained in Figure 5.6) has an *Authentication Layer,* which handles the required tokens exchange for application authentication, as well as the correspondent learner authentication towards the CBT.

The third layer is the *Business Logic Layer* (see Figure 5.6), creates the necessary controls for interaction between the CBT interface and the *Learning Activities Orchestrator (LAO)* (introduced in the following Subsection 5.3.2) interface presented to the learner. This layer identifies the business logic of each CBT. This business logic is then used by the *Communication Layer*—for instance, it is possible to add an *Idea* to a mindmap through the CBT public Web API when a UI is presented at the LAO for this purpose. The *Business Logic Layer* also handles the CLAO storage, for management of assignments, tools and user relationships. Finally, is relevant to note that CLAO architecture runs over a cloud infrastructure of Amazon's Elastic Compute Cloud (EC2) (Barr, Varia, & Wood, 2006).

As the fourth layer, the *Analytics Layer*, records user behavior and interaction data from the CBT, and sends these data to cloud-based storage (Google Fusion Tables) for further analytics processing.

User sessions and activities within CLAO are also recorded by the services provided by *Google Analytics (GA)* by feeding of customized data related to the *Learning Activity* and learner. GA was selected for two reasons: Firstly instead of creating a new analytics interface, reuse a tool for analytics already available because analytics is not the main research focus. Secondly because it provides a mature and powerful interface that fits well for the research objectives. Each time a user enters into the CLAO, a session is recorded in GA (according to GA, a session currently stands for uninterrupted activity for no longer than 30 minutes). Additionally, it has been added a recording script at LAO that sends information of a *Learning Activity* in a per tool basis, by tracking user window focus in a given tool. The recorded data provide useful information about usage analysis, collaboration level, and behavioral patterns in the CBTs. All actions that the CBTs makes available are retrieved via this layer. The customized data feed made enables GA to provide time spent per session per tool.

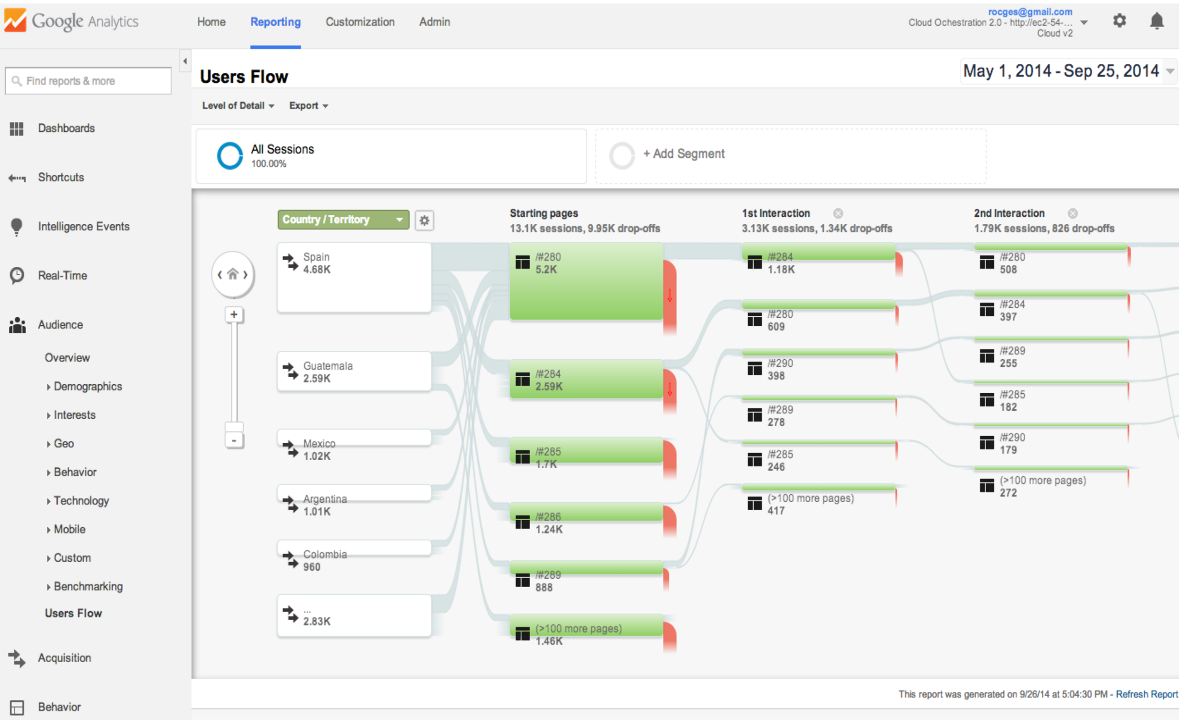
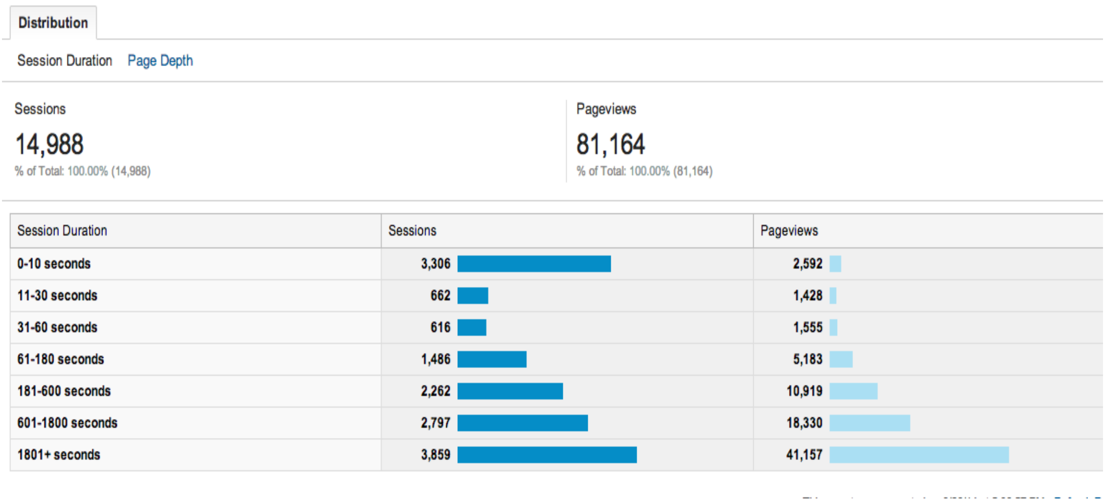


Figure 5.7 User flows graphic provided by GA.

The *Users Flow* in Figure 5.7 graphical representation by GA, shows each CBT related to a given LA, and how the learners behavior flows over the CBT and LA. GA presents in Figure 5.8 the *Engagement* data by means of number of sessions, page views, session duration.



Distribution

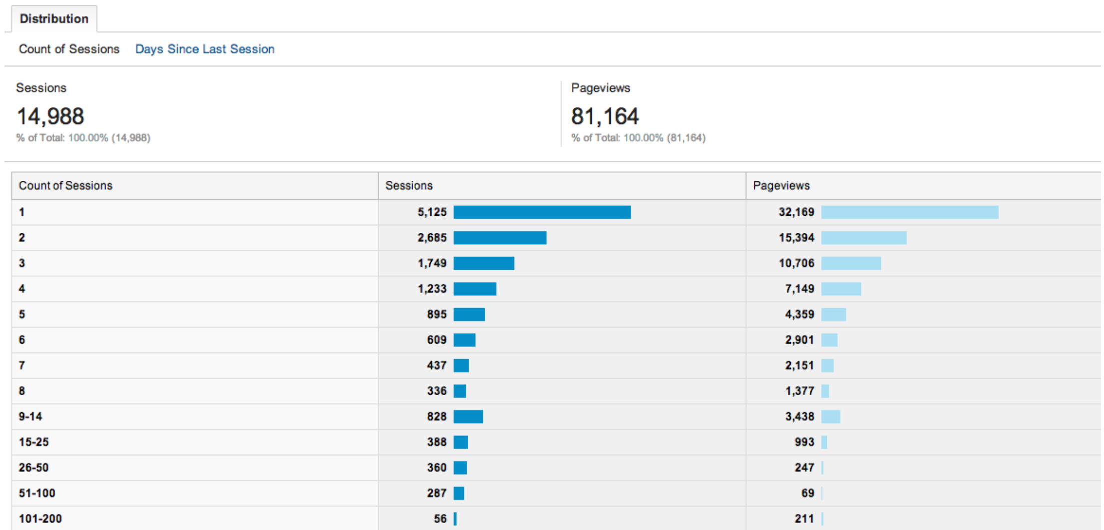
Session Duration | Page Depth

Sessions	Pageviews
14,988	81,164
% of Total: 100.00% (14,988)	% of Total: 100.00% (81,164)

Session Duration	Sessions	Pageviews
0-10 seconds	3,306	2,592
11-30 seconds	662	1,428
31-60 seconds	616	1,555
61-180 seconds	1,486	5,183
181-600 seconds	2,262	10,919
601-1800 seconds	2,797	18,330
1801+ seconds	3,859	41,157

Figure 5.8 Engagement statistics provided by GA.

Among many other type of reports that include demographics (due Google's ability to link their user accounts data), interests (market segments), Geo-location and language and other type of reports, the users' sessions and page views frequency report is shown in Figure 5.9 that simply shows the number of sessions and correspondent page views ordered and grouped by frequency.



Distribution

Count of Sessions | Days Since Last Session

Sessions	Pageviews
14,988	81,164
% of Total: 100.00% (14,988)	% of Total: 100.00% (81,164)

Count of Sessions	Sessions	Pageviews
1	5,125	32,169
2	2,685	15,394
3	1,749	10,706
4	1,233	7,149
5	895	4,359
6	609	2,901
7	437	2,151
8	336	1,377
9-14	828	3,438
15-25	388	993
26-50	360	247
51-100	287	69
101-200	56	211

Figure 5.9 Sessions and page view frequency provided by GA.

Furthermore, in Figure 5.6 is depicted the *Learning Environment Connector (LEC)*, that is explained in the following Section 5.3.2.

CIS Services Modules are the single components of the CIS architecture. These modules are built for each one of the identified CBTs used for *Learning Activities*. Each module is a bundle built with the Symfony2 framework, implementing the functionality of CBTs interfaces. A CIS service module is prepared with the elements described by the public API of the CBT (e.g. Dipity, SoundCloud, Cacoo, MindMeister). A CIS service module adopts the functionality provided by the CBT Web API to enable teachers to define the various activities in their own learning environment.

Each CIS service module uses each of the layers provided—communication, business, authentication, and analytics—and communicates with the service tool independently. Authentication is prepared according to the service method (e.g. OAuth, URL Tokenized). The *Business Layer* executes every CRUD operation upon the activity type (e.g. creation of a map). Figure 5.10 shows the representation of each cloud service represented by a bundle, describing interaction.

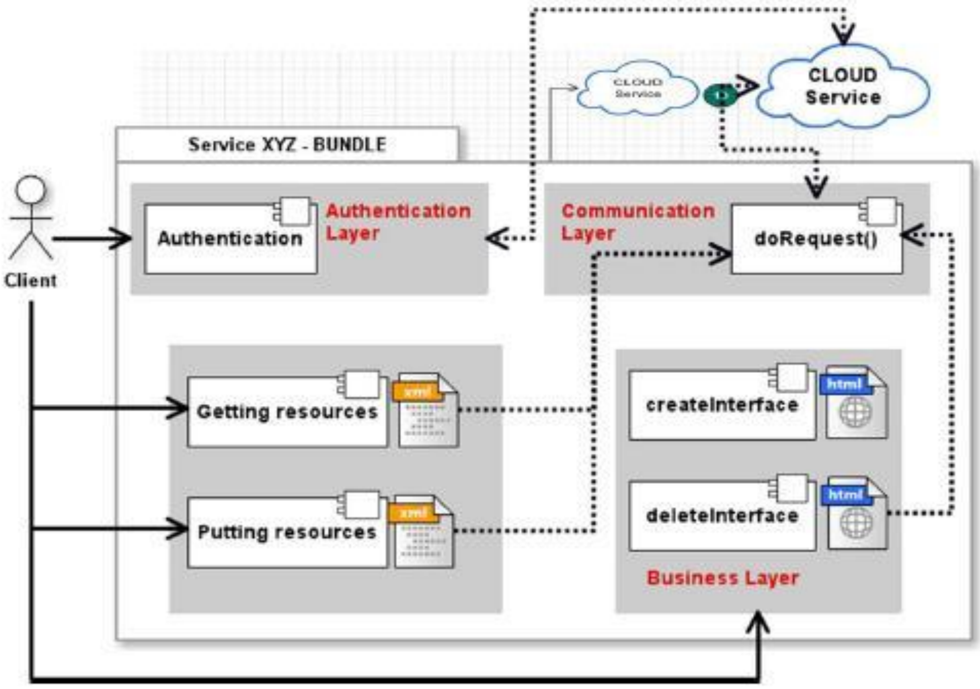


Figure 5.10 CIS second generation service modules. Each CBT represented by a bundle taken from (Hernández, et al. 2014c).

5.3.2 CEE: The CLAO Architecture

The CLAO provides a unified workspace environment that enables teachers and learners to use CBTs. In addition, CLAO handles communication with CBT, as well as authenticates and provides opportunities to create and orchestrate *Learning Activities*. The CLAO consists of the following layered architecture (see Figure 5.6): a) *Learning Activities Orchestrator (LAO)*, which constitutes the user interaction layer (interface and interaction). b) *Learning Environment Connector (LEC)*, which enables interoperability between the LMS and the CLAO. c) *Cloud Interoperability System (CIS)*, which is the core component of the interoperability architecture. It does the interoperability with each of the CBTs used in CLAO as presented in the previous Section 5.3.1. Finally, it is relevant to note that CLAO uses a cloud-computing infrastructure that utilizes the cloud infrastructure of Amazon's Elastic Compute Cloud (EC2) (Barr, Varia, & Wood, 2006).

Learning Activities Orchestrator (LAO): This component constitutes the user interaction layer of the CLAO architecture (interface and interaction). It presents the "one-stop shop" for learners with a description of the *Learning Activity (LA)* and an entry point to the CBT (e.g. MindMeister, Google Drive). Once the user is identified in the system, he or she can begin exploring the assigned LA in each course. Each LA presented to the student has a description, assessment rubric, examples, and entry point to the assigned learning CBT. LAO user interface (see Figure 5.11) creates a visual interface that is connected to the CBT, including features allowed by tool Web API (e.g. in Google Drive online document, to embed its editor into the LAO and main controls such as 'create a document'). Tools and activities are configured by a teacher who is also capable of creating students groups within a course. Assignments completed in such a tool can be sent to the LMS. Once students have finished their assignments with a tool, they can send them to the LMS for the teacher to assess them. Figure 5.11 presents the LAO user interface. The process is organized as follows: it begins with the *Learning Activity* description, and then is given a list of CBT to be used for the LA. When the learner wants to use a CBT, a resizable window within the LAO comes up to operate with tool, allowing easy return to LAO's main UI. It is even possible to maintain more than one tools open in a multi-window presentation, although TP restrict user to just one window per tool. Having multiple windows helps students to go back and forth easily from CBTs while working in LA that require the use of multiple CBTs.

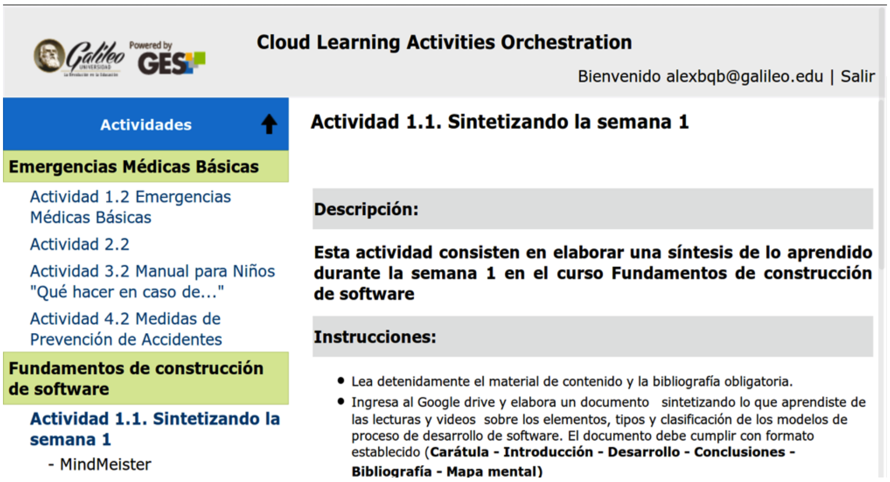


Figure 5.11 LAO Interface for students linking with Learning Activities taken from (Hernández et al., 2014c)

A simple workflow is enabled, in order to enact the CBTs according to the designed flow. Simple thresholds can be activated to allow learner to only work on the next CBT once the threshold has been achieved. For example, a threshold in a document can be the total number of words, and in a mind map can be the total number of ideas, in Figure 5.12 it is the configuration interface for the teacher.

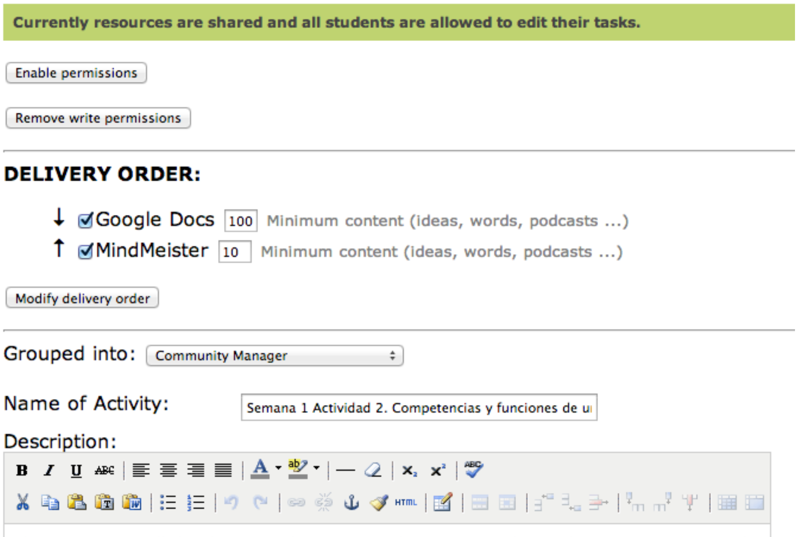


Figure 5.12 Part of the LA editor's interface, with simple workflow and thresholds.

Learning Environment Connector (LEC): This component is used for integrate into the CLAO architecture and the monolithic learning management system or VLE, providing a single user authentication. Examples of interaction between systems are user authentication (single sign-on), session management and assignments. The LEC provides an API to create custom integration within the CLAO and a VLE. This includes two main services: (1) a single sign on service between the CLAO and the VLE. (2) The assignments management, to link grade results, from a *Learning Activity* performed in CBT, to the VLE assignments management tool.

5.3.3 Evaluation and Lessons Learned

A new interface for deploying *Learning Activity (LA)* was created through the CLAO, and also an improved CIS architecture for interoperability with better modular support for each CBT contract, and connection to GA for user behavior analytics.

With the CLAO, in a given LA the CBT resource instance belongs to the learner, with the following consequences: (1) a learner could modify an assignment even after submitting it, (2) a learner could erase the resource and all its history and so keep it from further analysis, and (3) a learner had to manually share resource editing rights with the CLAO in order for allow it to be able to perform automatic analytics.

The solution applied to CLAO in this software development iteration to these previously stated consequences is twofold. Firstly, to include an automatic process to send the assignment via the LAO user interface (this was the only way to send the assignment). Secondly, to have the CBTs resources created automatically by the CLAO, and then shared to the learner. The proposed solution requires that the learner to work in a specific MM or GD instance and keeping the resource ownership with the CLAO. This also had the advantage of remove editing rights to the learner once he or she had submitted the assignment, making it impossible for learners to improve their assignments after the deadline.

Although with the new GA integration and data processing many useful reports were available, there were some limitations, for example GA can give back user information only in an hour frame (i.e. from 12:00 to 13:00 hours), thereby GA cannot yield a more granular style. Also it was not possible to model in detail the browsing behavior patterns when a learner was in CLAO. For example, a learner is in MindMeister, then goes to the LMS, then goes back to Google Drive, then to MindMeister, then changes to another activity, and so on. The reason the CIS cannot model such behavior in detail is that GA does not provide granular data for that kind of analysis. It keeps those data within itself and shows the data through a UI of *User Flow,* which can be used to identify browsing behavior patterns. Although for GA, all is under the same URL request, making it impossible to create a browsing pattern for it. This needs to be improved by having RESTful-style URL requests in the next software iteration, making it possible to create browsing patterns within GA.

5.4 Discussion

The first and second generation of the CIS created an improved architecture for CBTs interoperability, which now uses service modules for each CBT WEB API to manually coding the respective contracts. The second generation introduced the notion of CRUD operations, which highly relevant for further enabling a semantic interoperability approach, which is elaborated in the next Chapters. Is foreseen that the services modules introduced in the CIS second generation are no longer used when the semantics are in place, because an automatic API discovery process is present, thus enabling not to code each new CBT contract, instead use a generic engine to process the API.

Additionally, it has been identified that each CBT has different types of payloads such as XML, JSON or others. Furthermore the CBTs have slight different variations of the same authentication protocol (e.g. OAuth) or completely different and very custom approaches. The same issues are raised with CBT that have different variations of RESTful services implementation. Not all the CBT support the same type of access to their resources through the Web API, for instance some permit to create ideas within a mind map and other do not. In summary all these issues are served as a foundation to build the Semantic Proxy described in Section 4.3.3. Consequently the heterogeneity of CBT Web API requires identifying and comparing them when thinking on creating a generic semantic representation, which is introduced in the next two Chapters.

The CLAO has been built using the OLAS described in Section 4.3.4 and it fully integrates with the CIS. During the development phase it become clear that CLAO needs to fully automatize the creation and sharing of CBTs resources, because otherwise require learners to configure the CBTs. If CLAO remains as the *Owner* of the resource, it can control who can edit, view, etc., which avoids administrative problems over the resource, thereby such improvements are already included.

Regarding user behavior analytics the GA integration provided a powerful user interface for processing, although as already mentioned, still need improvements on how the information is fed to enable GA to produce a better *User Flow* charts. Another problem faced was tracking sessions and time spent when learner decided to use GD or MM standalone, using the same resource that was shared through the CLAO but—instead of using it through the CLAO—entering and using the resource through the CBT website. Possible solutions are workarounds such as embedding scripts into the shared resource. These analytics improvements go into the next software development iteration, described in Chapter 7, although, first in Chapter 6 is described the semantic technology to be incorporated.

6 ONTOLOGY AND GENERIC VOCABULARY FOR CLOUD-BASED TOOLS INTEROPERABILITY

As already mentioned in previous Chapters, the *Web 2.0 Tools or Cloud-based Tools* (CBTs) for educational settings have a large potential and acceptance for both, students and teachers (Hernández, Morales, Medina, & Barchino, 2015a). The aim of this chapter is to create interoperability to address the problems and challenges introduced in Section 4.1, and to target the research objectives outlined in Section 4.2. The Objective 1 is interoperability for orchestration of CBT through automatically recognition and operation of Web APIs. This requires the selection of semantic technologies that allow describing such Web APIs to later process them automatically. The Objective 2 (see Section 4.2) is that CBT's Web API description is constructed at a higher level by semantically describe the objects, properties, operations and their relations, rather than writing standard contracts and requests. For this, first are selected two *Application Domain Type (ADT)* for consistency with the use case specified in Section 4.2 (the ADT refers to a given type of CBT, where all its CBT providers that have the same set of core functionality). The selected ADT are mind maps and online document editor. A review of several classified CBT Web APIs for each ADT is presented Section 6.1. Then an ontology for interoperability of CBTs is contributed (see Section 6.2), in this case two ontologies are created, each one specialized per ADT. The ontology serves to formally describe the semantic definition for interoperability with CBT Web APIs as introduced in Section 4.3.1.

The Objective 3 (see Section 4.2) focus on creating ADT specifications that serves as a generic description of Web APIs. Following the discussion of Web interoperability technologies in Section 3.4, has been selected JSON-LD and Hydra to support the semantic model. Thereby, ADT *Generic Vocabularies (GV)* are presented (see Section 6.3) based on the introduced ontologies and using the selected semantic technology.

In Objective 4 (described in Section 4.2), *Tool Providers (TPs)* may choose to support partially or totally such application domain–type specifications, and the TPs have the possibility to extend such specifications. This achieved through the use of the *API Documentation (API Doc)*, that describes precisely the Web API of a particular CBT using the GV and the semantic technologies, see Section 6.4. This API Doc will be different between several CBTs that belong to the same ADT, but all of them will use the same GV as a base to describe the Web API. To prove this, is built the API Doc for two different CBTs per each one of the two-selected ADT.

This Chapter is based partially on the following publications (Hernández, Gütl, & Amado-Salvatierra, 2014a; Hernández, Gütl, & Amado-Salvatierra, 2014b; Hernandez & Gütl, 2015b).

6.1 Cloud Based Tools and Web API Classification

The following Section includes parts of the publication by Hernández, et al. (2014a), Hernandez and Gütl (2015b) and, Hernández, Gütl and Amado (2015b).

For this research two *Application Domain Type (ADT)* are selected. (1) The mind map CBT was chosen for implementation. (2) The document editor, with a focus in online document editors due the cloud-based nature of this research.

At Galileo University, where the evaluation happens, the mind maps and online documents editor are increasingly used as a resource for the teacher and learners, therefore both were chosen. From the vast amount of CBTs, for this research exemplary it is used only these two ADTs. In the following Subsections are described the reviewed CBTs for each ADT. Then, each CBT is evaluated based on their Web API functionalities, by identifying the supported objects, operations and properties. This serves to later build the corresponding educational interoperability ontologies.

6.1.1 Mind Maps

Mindmapping is a powerful pictorial technique defined in Buzan and Buzan (1993) as “a well-known technique used in note taking and is known to encourage learning and studying, mainly used for representing knowledge, concepts and ideas” (Sarker, Wallace, & Gill, 2008). A mind map is defined in Siochos and Papatheodorou (2011) as “a graphical technique, which is used to represent words, concepts, tasks or other connected items or arranged around central topic or idea”.

For the mind map CBTs the selection process was as follows: a first rank of 31 identified tools produced as is shown in Table 6.1. From these 10 were not considered due the fact that they are desktop applications, which work standalone and do not have connectivity to the Internet or to a cloud service. Other 12 were discarded due the fact they do not provide an API to interact with, therefore making impossible to create an interoperability bridge with them.

Table 6.2 summarizes the rest of the CBT important in the context of this research, selected based on their Web API maturity, which means that the selected are those that have a larger and more sophisticated set of operations available in their Web API as is further described. Some of them are mind map applications while others are Web CBTs with a structure, purpose, and (or) graphical representation similar to those of a mind map. These CBTs were classified in groups of three main interoperability features: interoperability response types (e.g., CSV, XML, or JSON), CBT API maturity level, and authentication mechanism (e.g., OAUTH, OpenID, Auth URL). In terms of API maturity, CBTs were classified in four levels, with Level 4 (L4) the highest maturity level for an open API, allowing service consumers to do almost anything with the CBT, in a mind map is manipulating the maps and its resources or objects. Level 3 (L3) stands for an intermediate maturity level in which reading almost everything within a mind map is possible, although add, update and delete operations to ideas are not possible. Level 2 (L2) identifies a CBT that allows embedding the web-based editor and player of the CBT, and through its API it can manipulate the general objects, such as map names. Finally, Level 1 (L1) implies a poor API, simple authentication, and a simple embed of html editor or player. CBTs identified with API Level 1 were discarded for this study.

Table 6.2 present the 4 CBTs list characterized with the previously defined structure. CBTs in Table 6.2 were classified based on three resources (user, map, and idea), including supported operations and properties.

Tool	Tool URL	Scope	API
MindMeister	http://www.mindmeister.com/es	Web-based	Yes
Mind42	http://mind42.com/	Web-based	Yes
Twiddla	http://www.twiddla.com/	Web-based	Yes
Scribblar	http://www.scribblar.com/	Web-based	Yes
Cacoo	https://cacoo.com	Web-based	Yes
Lucidchart	https://www.lucidchart.com/	Web-based	Yes
Gliffy	http://www.gliffy.com/index-f.php	Web-based	Yes
Creately	http://creately.com/	Web-based	Yes
Spicynodes	http://www.spicynodes.org/index.html	Web-based	Yes
Bubbl	https://bubbl.us/	Web-based	No
Spiderscribe	http://www.spiderscribe.net/	Web-based	No
Mindomo	http://www.mindomo.com/	Web-based	No
Drichard	http://drichard.org/mindmaps/	Web-based	No
Pearltrees	http://www.pearltrees.com	Web-based	No
Wisemapping	http://www.wisemapping.com/	Web-based	No
Coggle	https://coggle.it/	Web-based	No
Mindmup	http://www.mindmup.com/#m:new	Web-based	No
Mapul	http://www.mapul.com/	Web-based	No
Examtime	https://www.examtime.com/es/	Web-based	No
Draw	https://www.draw.io/	Web-based	No
Popplet	http://popplet.com/	Web-based	No
Visual-mind	http://www.visual-mind.com/download.php	Desktop-based	No
Freemind	http://freemind.sourceforge.net/wiki/index.php/Download	Desktop-based	No
Semantik	http://code.google.com/p/semantik/	Desktop-based	No
Recallplus	http://www.recallplus.com/index.php	Desktop-based	No
Labyrinth	https://people.gnome.org/~dscorgie/labyrinth.html	Desktop-based	No
Insilmaril	http://www.insilmaril.de/vym/	Desktop-based	No
Thebrain	http://www.thebrain.com/#-110	Desktop-based	No
Mindnode	http://mindnode.com/	Desktop-based	No
Xmind	http://www.xmind.net/	Desktop-based	No
Cmaptools	http://cmaptools.uptodown.com/	Desktop-based	No

Table 6.1 Group of 31 identified mind map tools reviewed

Supported features per resource or object for each CBT			Mind-Meister1	Cacoo3	Scribblar2	Creately4
user	Operations	login	x	x	X	X
		list maps	x	x	X	
		change user	x		X	
	Properties	id	x	x	X	
		username	x	x		
		firstnames	x		X	
		lastnames	x		X	
		email	x		X	x
map	Operation	create map	x	x	X	x
		update map	x		X	
		delete map	x	X	X	x
		get map	x	x	X	x
		share map	x			x
		copy map	x	x		x
	Properties	Id	x	x	X	x
		Title	x	x	X	x
		creationdate	x	x	X	
		Revisions	x			X
idea	Operation	create idea	x			
		delete idea	x			
		update idea	x			
		get idea	x			
	Properties	id	x		X	
		map id	x		X	
		title	x			
		parent	x			
		posx	x			
		posy	x			
		userid	x		X	
API maturity level			L4	L3	L3	L2

Table 6.2 Selected mind maps CBTs with interoperability features taken from (Hernández et al. 2014b)

6.1.2 Online Document Editor

The document editor, which is currently an indispensable tool in a learning environment, for this research, it was focused on the online document editors, which is a common technological trend in major office productivity suites such as Microsoft Office, which has an online version of its desktop suite, known as Office Online and stored in OneDrive (formerly called SkyDrive) (One Drive, 2014). New players such as Google initiated with its own suite of office productivity CBTs, which is originally named Google Docs, and now is Google Drive (Drive, 2014). Although both identified solutions encompass many CBTs, this research focuses only in the online document processing CBTs available on the Internet.

Office productivity suites are one of the most mature software available at this moment, but still many new brands and versions come out to the market often. As noted earlier, the most used suite is desktop version of Microsoft Office, but for this research it was clearly identified that the online version of this suite is suitable for the purposes exposed in the previous Chapter. Firstly 21 tools were identified and are shown in Table 6.3, several of these were not considered because they are desktop applications and do not have a Web API available. It is relevant to mention that research exposed here can include desktop applications, or any other type of application for that matter, with the only requirement to have an openly available Web API. Following with the depicted table, other 6 were discarded because of the lack of available Web API to interact with.

Tool	Tool URL	Scope	API
Gdrive	https://drive.google.com	Web-based	Yes
Zoho	https://docs.zoho.com	Web-based	Yes
Onedrive	https://onedrive.live.com	Web-based	Yes
Etherpad	http://etherpad.org/	Web-based	Yes
Hackpad	https://hackpad.com	Web-based	Yes
Crocodoc	http://personal.crocodoc.com/	Web-based	Yes
Draftin	http://draftin.com	Web-based	Yes
Pangurpad	http://pangurpad.com/	Web-based	No
Livedocuments	http://livedocuments.com/	Web-based	No
Collabedit	http://collabedit.com/	Web-based	No
Freeoffice	http://www.freeoffice.com/	Web-based	No
WriteUrl	http://www.writeurl.com/	Web-based	No
Firepad	http://www.firepad.io/	Web-based	No
Lotus	http://www-03.ibm.com/software/products/en/lotusymp	Desktop-based	No
Dataviz	http://www.dataviz.com/DTG_home.html	Desktop-based	No
Thinkfree	http://www.thinkfree.com/main.jsp	Desktop-based	No
Wordperfect	http://www.wordperfect.com/	Desktop-based	No
Libreoffice	http://www.libreoffice.org/	Desktop-based	No
OpenOffice	https://www.openoffice.org/es/	Desktop-based	No
NovusOffice	http://www.novusoffice.com/es/standard.aspx	Desktop-based	No

Table 6.3 Group of 21 identified document editor tools reviewed

Table 6.4 has the rest of the CBT that are both online but have an open accessible Web API. Some of them are standalone document editor, but others are full office productivity suites. These CBTs were classified in groups of three main interoperability features: interoperability response types (e.g., CSV, XML, or JSON), CBT API maturity level, and authentication mechanism (e.g., OAUTH, OpenID, Auth URL).

In terms of API maturity, CBTs were classified in four levels, with Level 4 (L4) the highest maturity level for an open API, allowing service consumers to do almost anything with the CBT, being able to manipulate almost anything within a document.

Level 3 (L3) stands for an intermediate maturity level in which still has access to most of the properties of a document, importantly has the ability to edit the content of the document itself and just a few properties are not accessible and the update resources operation is not available. Level 2 (L2) identifies a CBT that has access to the basic properties of a document and some of the basic operation, but it cannot update the content within a document. This is particularly important feature because allows to create specialized scenarios where is possible to automatically manipulate the content of the document, transferring content back and forth from the document to the VLE and other CBTs. Finally, Level 1 (L1) implies a poor API, simple authentication, and a simple embed of html editor or player. There are no CBTs that classifies into this category.

Supported features per resource or object for each CBT			Gdrive	Draftin	Etherpad	Zoho	Onedrive	hackpad	Crocodoc
User	Operation	Login	X	x	x	x	x	X	x
		list files	X	x	x	x	x	X	x
	Properties	Id	X	x	x	x	x	X	x
		Username	X	x		x	x		
		email address	X	x		x		x	
		picture url	X						
File	Operation	create and upload	X	x	x	X	x	x	x
		Delete	X	x	x	X	x		
		update resource	X	x		X	x		
		update content	X	x	x			x	
		Copy	X	x			x		
		Get and download file	x	x		X	x		
		share file	x	x	x			x	
		export (plain text, text/html...)	x	x	x			x	x
	Properties	Id	x	x	x	X	x	x	x
		Title	x	x	x				x
		Labels	x						
		Owners	x	x	x				
		created date	x	x	x				
		Editable	x						
		last modifying	x	x					
		parent_id	x	x					
API maturity level			L4	L4	L3	L2	L2	L2	L2

Table 6.4 Selected document editor CBTs with interoperability features taken from (Hernandez & Gütl, 2015b)

Table 6.4 presents the seven CBTs characterized with the previously defined structure which have more features and are of further interest in the research. CBTs in Table 6.4 were classified based on two resources (user and file), including supported operations and properties. The avid reader may notice that this classification is on purpose and uses the Hydra vocabulary represented in a Web API (resources, operations, properties, and others). This is later be used as requirements to create an adequate ontology to represent a mind map CBT for interoperability.

6.2 Proposed Ontologies for Interoperability

The following Section includes parts of the publication by Hernández, et al., (2014a) and, Hernández, Gütl and Amado (2015b).

Ontologies used for educational settings have been developed in different research and development projects. Initiatives as presented in the IntelLEO project defined a working reference framework with different ontologies, mainly the learning context (Jovanovic & Gasevic, 2011) and activity ontology (Jovanovic & Gasevic, 2012). Another example of an ontology implementation in learning context is presented in (Kolas, 2006). For the specific case of mind maps, an equivalent definition was evaluated, the Topic Maps. Topic Maps are defined with a formal ontology in ISO/IEC 13250 standard (2006). These ontologies were analyzed as building blocks to define ontologies to be translated to semantic technologies with the aim to complete interoperability between TC and TP.

A proposed general ontology for each ADT is presented, namely a mind mapping, and a text editing. The ontology formally models the possible operations that the CBT possibly has, represented as isolated classes. Additionally the resources that the CBTs are built on also are represented as classes, and the relations between the classes, including its corresponding properties. As expected, among the vast possibilities of CBTs brands and approaches, there are a large number of different operations and resources available for each provider (as depicted in
Table 6.2 and Table 6.4) therefore it has been decided to only represent those general and most common operations and resources that are in Level 4 interoperability API maturity.

The ontology for a mind map ADT is presented in Figure 6.1 (using UML to represent OWL). This general ontology for mind maps models common resources, properties, operations, and relations identified in Table 6.2. The objective is to model maps and their main resources such as maps, ideas, and users along with the correspondent properties for each resource. Finally, the operations are also represented. From this ontology, the semantic GV for a mind map CBT is derived.

Figure 6.2 presents the interoperability ontology for document editor ADT. This includes modeling the *User Class Resource*, with its properties. Then the *Document Class*, its properties, and further relation to the *User Class.* Then the six operations that are common and relevant for interoperability manipulation and automation. This general ontology for document editor models common resources, properties, operations, and relations identified in Table 6.4. As in the previous ontology, this as well serves as a base to derive the GV.

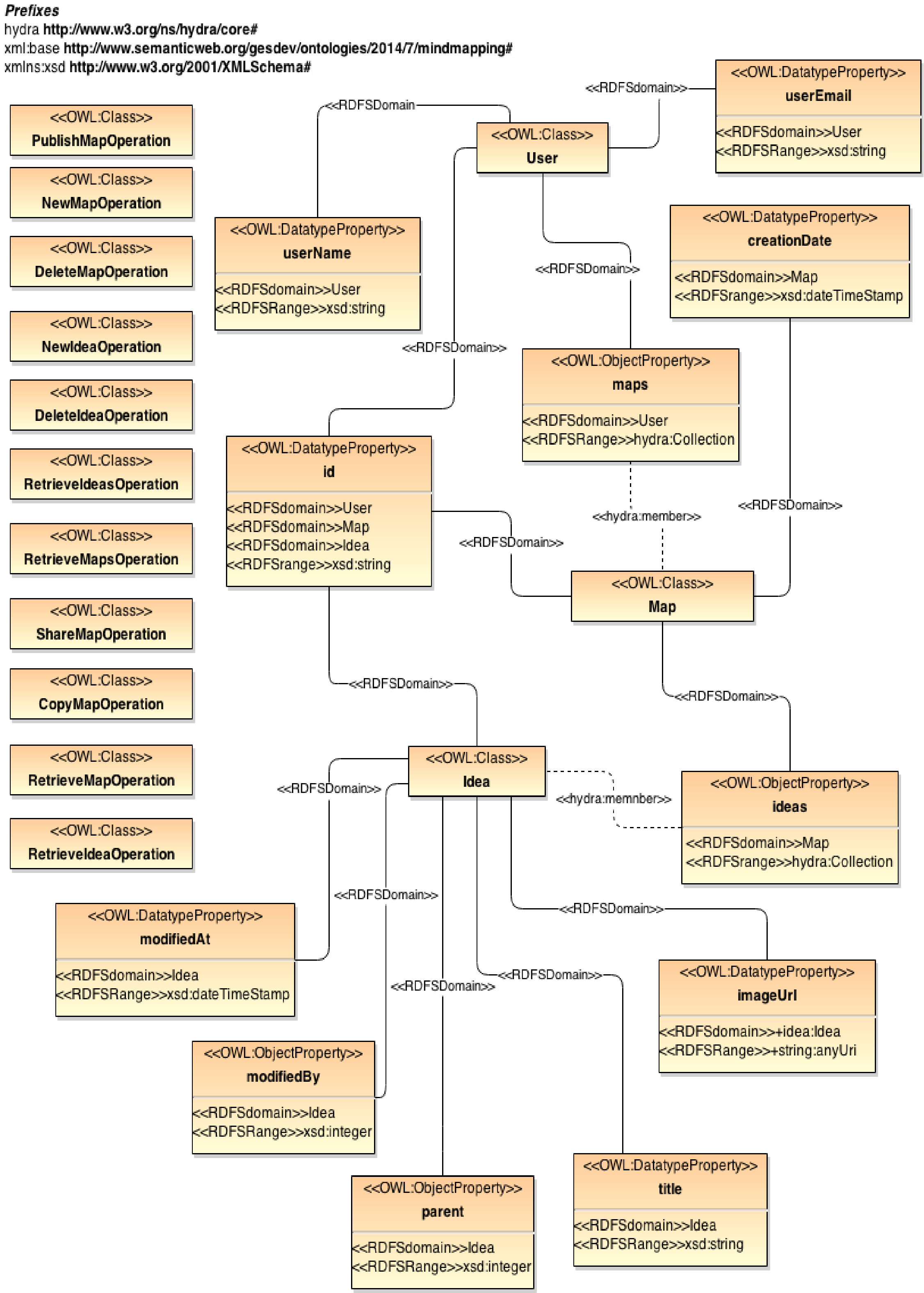


Figure 6.1 Ontology for mind map educational interoperability (OWL:Class is equivalent to rdf:Class) taken from (Hernández et al. 2014b)

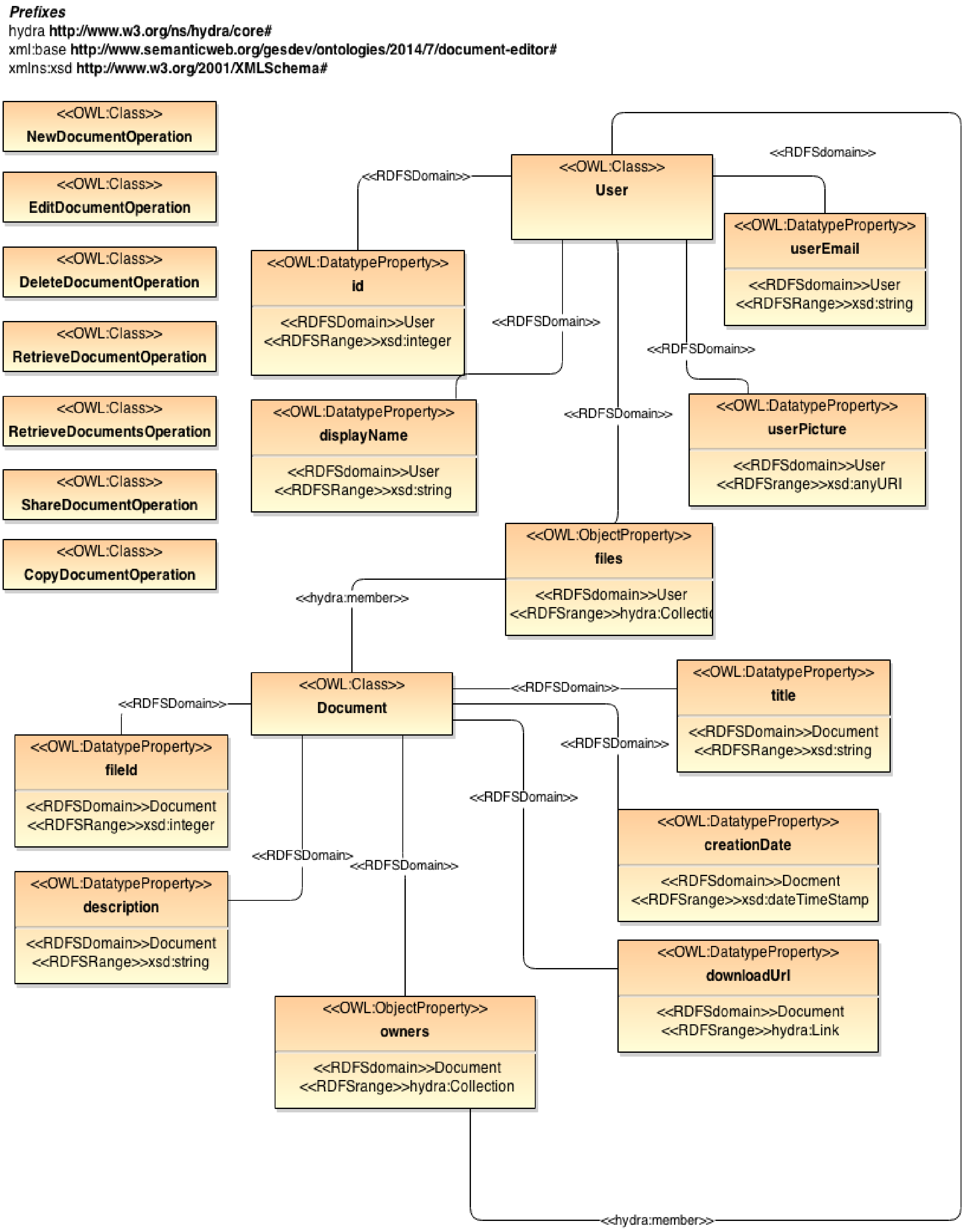


Figure 6.2 Ontology for document editor for educational interoperability (OWL:Class is equivalent to rdf:Class) taken from (Hernández et al. 2014b)

6.3 Generic Vocabularies for CBTs Interoperability

The following Section includes parts of the publication by Hernández, et al. (2014a), and Hernández, Gütl and Amado (2015b).

This Section introduces the *Generic Vocabulary (GV)* developed for each ADT, based on the previously constructed ontologies. To achieve that, it is necessary to select semantic Web technologies that support the outlined objectives. JSON-LD and

Hydra (see Section 3.4) are selected to support the semantic model for Web interoperability. JSON (see Section 3.2.4) itself is a widely used payload format thus is quite straightforward to further adoption of the proposed solution. Many TPs already have JSON payload, then those TPs are able to adopt JSON-LD without too many changes over their existing Web API (for instance using the HTTP Link header for adding reference to other documents context). Furthermore RESTful services are highly common in Web APIs, thus supporting them correctly is necessary. Hydra is a small but powerful addition to the current Web APIs to describe CBTs semantically using Linked Data and using a robust REST implementation. This is a powerful mechanism that enables discovery at run time of the Web API. Hydra provides the possibility to create generic descriptions of an ADT, highly relevant for this research to support the semantic model. In addition, Hydra also provides de flexibility for the TPs extend and support their own set of objects, operations and properties, a prerequisite to the flexible interoperability model.

Using Hydra (see Section 3.2.5) vocabulary it is possible to create self-descriptive, hypermedia-driven Web APIs. They can fully use Linked Data expressivity with REST's major benefits, such as scalability, evolution, and loose coupling. Hydra models represent resources, properties, and operations, among other useful classes for describing Web APIs. A GV is formed with subclasses of Hydra classes, and it is intended to model a specific ADT (i.e., a mind map GV can be used as a basis to model any mind map CBT Web API). The ontologies presented in the previous Section are used, leading to create a mind map GV that can describe basic properties and operations any mind map CBT may have (Section 6.3.1), and the same applies to a document editor (Section 6.3.2). This GV serves as the basis for interoperability while leaving CBT-specific details at implementation time.

6.3.1 Mind Map Generic Vocabulary

For a mind map the GV starts with the notion of a map, its properties, and its relation to ideas (the map can have a collection of ideas). A user can have a relation to a collection of maps. The idea can have a relation to another idea, forming the connections within a mind map. The supported operations for creating and deleting resources are inherited from Hydra's modeled operations. Both are available for a map and (or) an idea resource. It has been also extended to support new operations not modeled in Hydra, for retrieving, publishing, and sharing. The share operation is crucial to assign the proper permissions to a resource, which follows the use case describe in Section 4.2. A common set of operations and properties in the GV was used (see Figure 6.3). This gives the developer more flexibility and allows a specificity of details at the implementation time, as are referenced later.

Listing 6.1 describes exemplarily the API's GV that it is constructed with JSON-LD and Hydra. This piece of the GV describes the user (a learner or a teacher in this context) as `mm:User`, and its properties (the first property is `mm:userId`). This description process continues with the rest of the resources (maps and ideas) and corresponding properties.

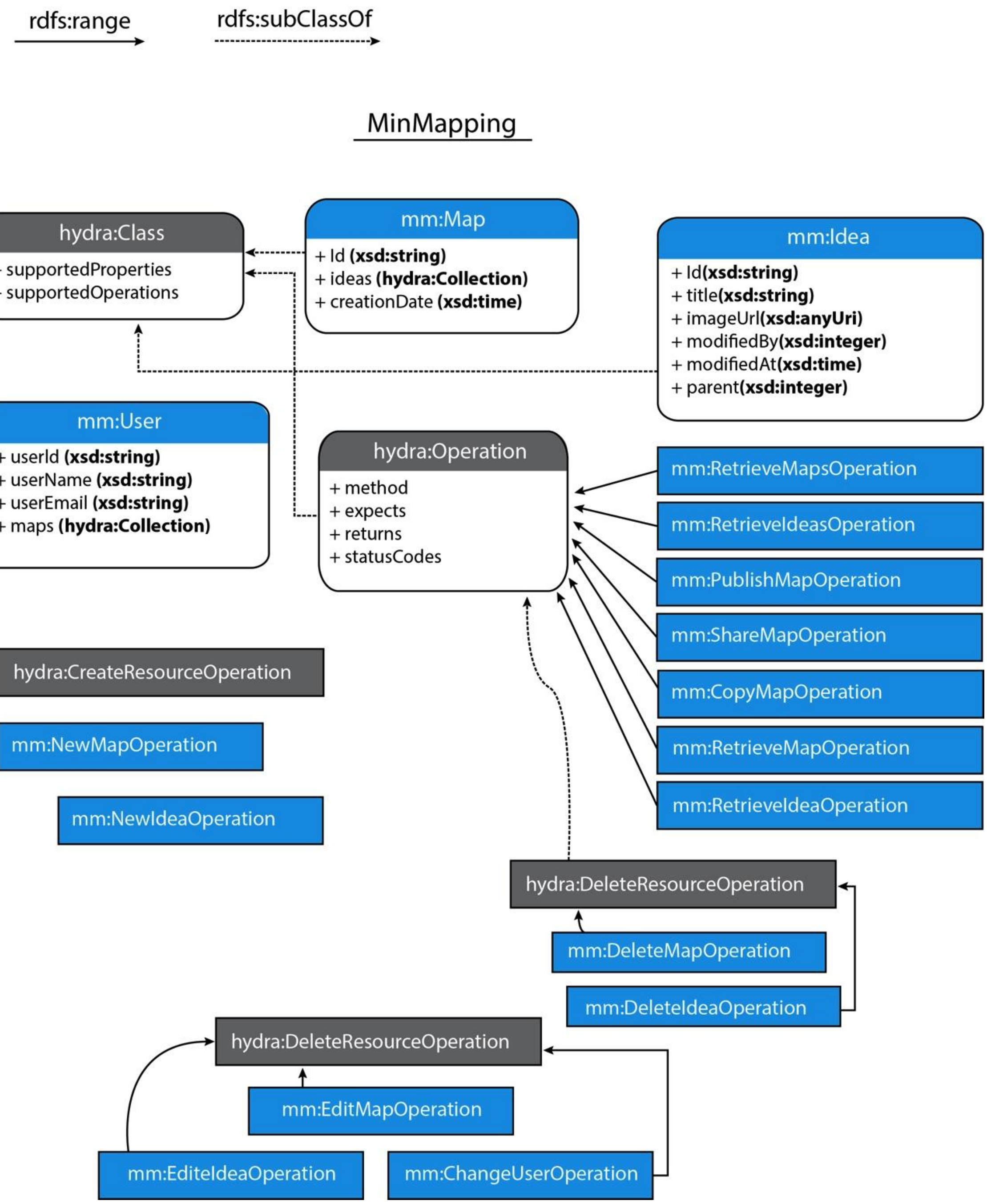


Figure 6.3 Mind map generic vocabulary represented as class diagram taken from (Hernández et al. 2014b)

Listing 6.2 contains the part of the GV that describes a user resource can have a collection of maps (`mm:maps`) related to it. The same type of description relates to a map with a collection of ideas (`mm:ideas`), meaning that a map has multiple ideas related to it.

```
{
"@id":"http://cis.galileo.edu/specs/mindmapping",
  "@type":"owl:Ontology",
  "label":"Generic Mind map Vocabulary",
  "comment":"A lightweight vocabulary for mind-maps services api.",
  "defines":[
   {
     "@id":"mm:User",
     "@type":"hydra:Class",
     "comment":"User representation",
     "label":"user",
     "status":"testing"
   },{
     "@id":"mm:userId",
     "@type":"rdf:Property",
     "label":"user-id",
     "comment":"User identifier",
     "domain":{
      "@id":"mmUser"
     },
     "range":{
      "@id":"xsd:integer"
     }
   },
```

Listing 6.1 Mind map generic vocabulary, a resource and its property example

```
   {
     "@id":"mm:maps",
     "@type":"rdf:Property",
     "label":"maps",
     "comment":"Maps of user",
     "domain":{
      "@id":"mm:User"
     }
     "subClassOf":["hydra:Collection"]
   } {
     "@id":"mm:Map",
     "@type":"hydra:Class",
     "comment":"General representation of a mind map.",
     "label":"mind-map",
     "status":"testing"
   } {
     "@id":"mm:ideas",
     "@type":"rdf:Property",
     "label":"ideas",
     "comment":"List of ideas",
     "domain":{
      "@id":"mm:Map"
     },
     "subClassOf":{
      "@id":"hydra:Collection"
     },
     "status":"testing"
 }
```

Listing 6.2 The generic vocabulary with, one to many relations examples, for maps and ideas.

The operations are modeled as a subclass of `hydra:CreateResource-Operation`. Listing 6.3 presents an operation to create a new map (`mm:NewMapOperation`), other operations create an idea (`mm:NewIdeaOperation`), and another deletes a map (`mm:DeleteMapOperation`). The other operations depicted in Figure 6.3 were also modeled but not included in these examples, but can be seen in Hernández (2015h).

It is worth to mention that in the GV, the operation but not its specific input and output types are modeled because they vary for each mind map CBT. Therefore, the Web API specifics are then modeled in the API documentation at implementation time and use the GV as structure for modeling the Web API. The API documentation contains specifics about the CBT, such as specific properties and operations naming, which operations and properties are supported, the data types, the http methods to be used, extended operations, and properties of the CBT itself.

```
{
  "@id":"mm:NewMapOperation",
  "@type":"hydra:Class",
  "label":"new-map",
  "comment":"Create a new map of ideas",
  "subClassOf":[
   "hydra:CreateResourceOperation"
  ],
  "status":"testing"
} {
  "@id":"mm:NewIdeaOperation",
  "@type":"hydra:Class",
  "label":"add-idea",
  "comment":"Add an idea to the ideas map",
  "subClassOf":[
   "hydra:CreateResourceOperation"
  ]
} {
  "@id":"mm:DeleteMapOperation",
  "@type":"hydra:Class",
  "comment":"Delete a map",
  "label":"delete-map",
  "subClassOf":[
   "hydra:DeleteResourceOperation"
  ]
}
```

Listing 6.3 Listed some of the generic vocabulary operations.

6.3.2 Document Editor Generic Vocabulary

The GV for a document editor is simpler compared to its equivalent to a mind map CBT, because the main resource or object is the document itself. Thus as depicted in Figure 6.4 the document resource has its set of properties. The GV also have the associated operations that can be performed towards a document, such as retrieve the document, new (create), edit and delete operations that inherit from the correspondent Hydra classes. It was purposefully left apart the management of folders (containers of documents), and document organization, because it is out of the scope of the research objective and suits better for a full description of the office environment, or to a storage CBT description.

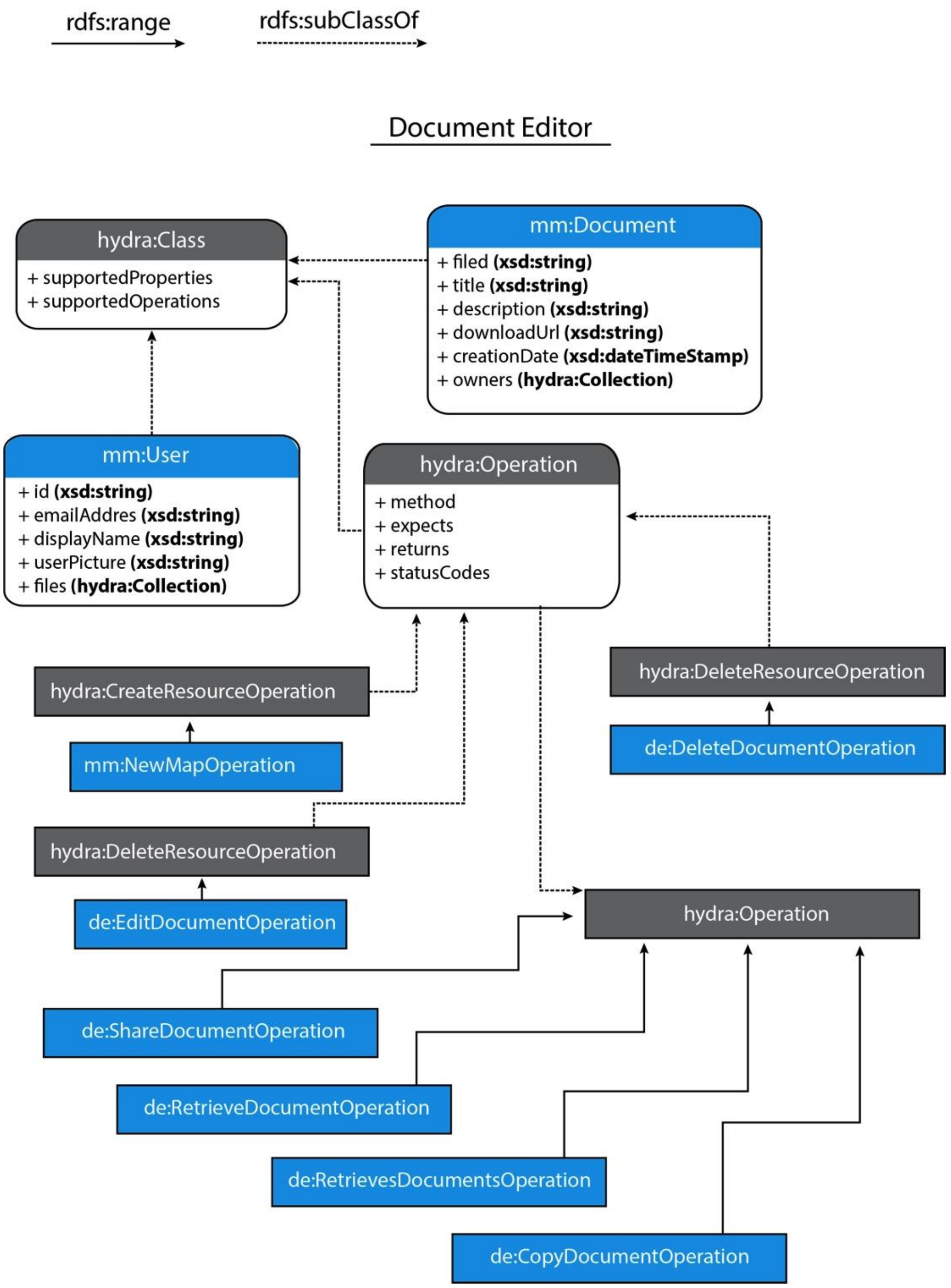


Figure 6.4 Document editor generic vocabulary represented as class diagram taken from Hernández and Guetl (2015b)

Listing 6.4 describes the initial part of the GV for a document editor. This part describes the *Document* as `de:Document`, and its properties that were define previous (e.g. title, description). It is also describe the *User*, although this is included in Hernández (2015b). Furthermore, it describes that a document can have multiple owners, which are users that have editor rights for the document.

```
{
    "@id": "de:title",
    "@type": "rdf:Property",
    "label": "title",
    "comment": "Document title",
    "range": {
       "@id": "xsd:string"
     }
} {
    "@id": "de:description",
    "@type": "rdf:Property",
    "label": "description",
    "comment": "Document description",
    "range": {
        "@id": "xsd:string"
    }
} {
    "@id": "de:owners",
    "@type": "hydra:Link",
    "label": "files",
    "comment": "Owners of file",
    "domain": {
        "@id": "de:Document"
    },
    "range": {
        "@id": "hydra:Collection"
    }
} {
    "@id": "de:Document",
    "@type": "hydra:Class",
    "label": "document",
    "comment": "General representation of a document",
       "supportedProperties": [
        {
            "property": "de:fileId",
            "readonly": "true"
        },
        {
            "property": "de:title"
        },
        {
            "property": "de:description"
        },
        {
           "property": "de:downloadUrl"
        },
        {
            "property": "de:creationDate",
            "readonly": "true"
        },
        {
            "property": "de:owners",
            "readonly": "true"
        }
       ]
}
```

Listing 6.4 Document editor generic vocabulary, the main resource and some properties examples

The operations modeled for this GV are described in Listing 6.5: create a new document (`de:NewDocumentOperation`), edit (`de:EditDocument-Operation`), delete (`de:DeleteDocumentOperation`), share (`de:ShareDocumentOperation`). Also exist but not included in the Listing 6.5, but in Hernández (2015b), `de:CopyDocumentOperation`, `de:Retrieve-DocumentsOperation` and `de:RetrieveDocumentOperation`.

```
    {
        "@id": "de:NewDocumentOperation",
        "@type": "hydra:Class",
        "label": "new-document",
        "comment": "Create a new document",
        "subClassOf": [
            "hydra:CreateResourceOperation"
        ]
      {
        "@id": "de:EditDocumentOperation",
        "@type": "hydra:Class",
        "comment": "Edit document",
        "label": "edit-document",
        "subClassOf": [
            "hydra:ReplaceResourceOperation"
        ]
       } {
        "@id": "de:DeleteDocumentOperation",
        "@type": "hydra:Class",
        "comment": "Delete a document",
        "label": "delete-document",
        "subClassOf": [
            "hydra:DeleteResourceOperation"
        ]
       } {
        "@id": "de:ShareDocumentOperation",
        "@type": "hydra:Class",
        "comment": "Share map",
        "label": "share-map",
        "subClassOf": [
            "hydra:Operation"
        ]
     }
   }
```

Listing 6.5 Document editor generic vocabulary operations

6.4 API Documentation

The following Section includes parts of the publication by Hernández, et al. (2014b), Hernandez and Gütl (2015b) and, Hernández, Gütl and Amado (2015b).

The GV presented in the previous Section contains the common properties, data types and operations that a CBT may have. A Hydra *API Documentation (API Doc)* describes all the current resources, operations, properties and data types that a CBT supports through its Web API. An API Doc first inherits from the GV the common operations and properties related to a resource (e.g. a *Map, Idea* or *User*). Thus the API Doc is defined for each specific mind map CBT, it uses the GV to define its own classes, as an example, the API Doc for a mind map CBT may define a `retrieveMap` operation that is a subclass of the GV `RetrieveMapOperation`. The API Doc describes the IRI, HTTP method, the expected data and return values when the TC requests an operation. In the following Subsections are introduced the API Docs for four CBTs selected for this research.

6.4.1 Mind Map CBTs API Doc for MindMeister and Cacoo

From existing mind map CBTs analyzed in Section 6.1.1, it has been decided to select two of them for this research, the ones with a higher maturity level, which are MindMeister (MindMeister, 2014) and Cacoo (Cacoo, 2014). Both CBTs were chosen because they had an open Web API. MindMeister provides a rich Web API covering mind map manipulation in the same way as using the MindMeister editor (i.e. create a map, add an idea, add relations between ideas, etc.). Cacoo offers a mid-level API (i.e. is possible to create a map, assign collaborators to it, add comments, but is not possible to add ideas and relations). Therefore only a map container can be created but not the ideas within the map that has to be done manually through the use of Cacoo UI.

The API Doc from the MindMeister CBT represented in Figure 6.5 describes the three main resources of a mind map CBT: user, map and idea. A *User* resource (learners and teachers in this context) inherits from the GV class: User, the properties: `userId, userName, userEmail`. It also describes operations and properties for these components. Additionally, it has one supported operation, which is `retrievesUser`, this gets all related information from a user. A *User* can be related to multiple *Maps*, therefore the user property named `User/maps` is a collection of *Map* resources that belongs to a *User*. That property as well defines a set of supported operations related to it: `retrieveMaps` and `createMap`. Both operations inherit from the GV `RetrieveMapsOperation` and `NewMapOperation` respectively, although defines the correspondent HTTP method to be used, and what it expects and returns in the request.

The *Map* resource has the following CRUD operations: `createMap, retrieveMap, editMap, deleteMap`, and some additional operations such as the `shareMap, copyMap` and `publishMap`, all of them define its own HTTP method explicitly, and the expected input and return values. As noted earlier, all operations inherit from the GV. As depicted in Figure 6.5, the map can have collaborators, which is a collection of *Users*. This is supported by the operation `retrieveCollaborators`, which is a MindMeister specific operation, not described in the GV.

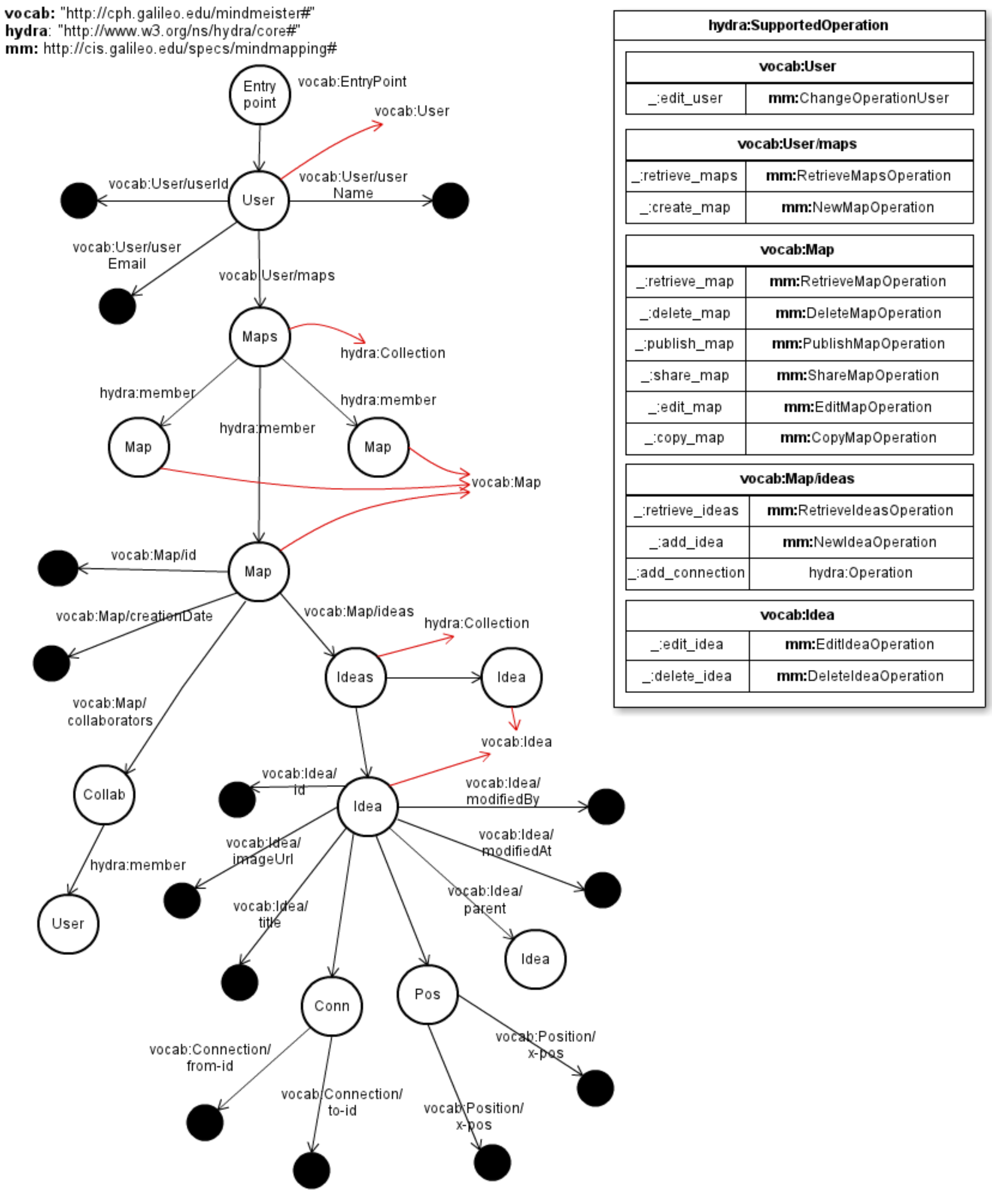


Figure 6.5 Mindmeister API Doc graph representation based on Hydra.

The *Idea* resource has the following properties: `id`, `title`, `parent`, `modifiedAt`, `modifiedBy`, `imageUrl`, inherited from GV. The operations `addIdea`, `editIdea` inherits from GV. To create parent-child relation between ideas it uses the parent property, but there is an operation named `addConnection`, it uses the connection class that describes the correlation between two ideas, that class has two properties, which are from-id and to-id, this is Mindmeister specific and not comes from the GV. Also it has the Mindmeister specific defined properties position, connection. The position is a class itself, that describes X and Y position therefore it has as two supported properties, first x and second y, this server to position an idea within the map. As described, the MindMeister Web API has a very rich set of available RESTful methods that allow to control and interact with it in a service-to-service way. The map also has a relation to *Ideas*, it can obtain a collection of its own ideas through the `retrieveIdeas` operation.

The second chosen mind map CBT is Cacoo (see Figure 6.6). Compared to MindMeister, it has a larger set of features, because it is not only a mind map CBT and therefore covers other functionalities such as UML and diagram editing. It also has a very rich usable API, although it is more restricted for handling granular resources (such as ideas and connections between them), and currently it is not possible to implement the full set of operations defined of the mind map GV. Cacoo implements the resources of user and map, and both are extended to some new properties. The operations supported are: create, retrieve and delete a map. Also, some new subclasses of maps (in Cacoo named diagrams) were created: comments (which has a relation to user class) and sheet (which is a concept of pages or containers of maps in a diagram), both with its own set of properties. No further operations are supported, this indicates that Cacoo offers a more limited interaction with the mind map CBT. It is not possible to interact on the level of ideas within a map, which, however, is the core concept of a Mind map CBT. Therefore, Cacoo is not as mature in terms of interoperability. Cacoo still requires using its own user interface for most common operations.

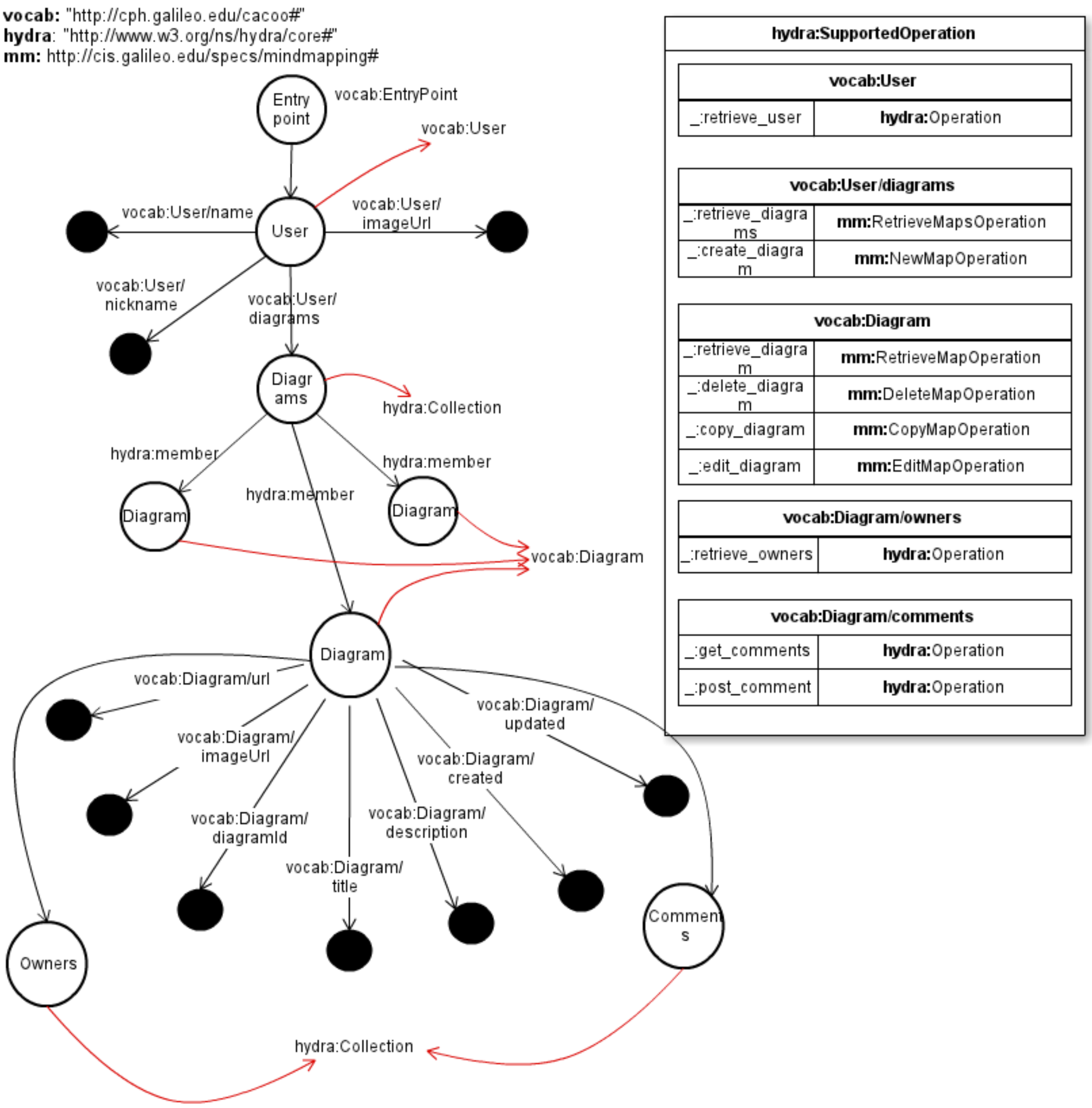


Figure 6.6 Cacoo API Doc graph representation based on Hydra

In Table 6.5 it is depicted a mapping between the ontology, and the GV, and the CBTs API Doc implemented operations. This helps to compare what is currently available for each CBT and how it relates to the GV and ontology. It is important to mention that due the scope of this thesis use case, a diverse set of operations available in the selected CBTs Web API has not been described neither in the API Doc nor in the GV, but the current approach can support the description of those operations and the related properties.

MINDMAPPING			
Ontology	GV	MindMeister	Cacoo
NewMap	NewMap	_:create_map	_:create_diagram
NewIdea	NewIdea	_:add_idea	X
RetrieveMaps	RetrieveMaps	_:retrieve_maps	_:retrieve_diagrams
RetrieveMap	RetrieveMap	_:retrive_map	_:retrieve_diagram
RetrieveIdeas	RetrieveIdeas	_:retrieve_ideas	X
RetrieveIdea	RetrieveIdea	X	X
PublishMap	PublishMap	_:publish_map	X
ShareMap	ShareMap	_:share_map	X
CopyMap	CopyMap	_:copy_map	_:copy_diagram
DeleteIdea	DeleteIdea	_:delete_idea	X
DeleteMap	DeleteMap	_:delete_map	_:delete_diagram
EditMap	EditMap	_:edit_map	_:edit_diagram
EditIdea	EditIdea	_:edit_idea	X
ChangeUser	ChangeUser	_:edit_user	X
		add_connection	retrieve_user
			retrieve_owners
			get_comments
			post_comment

Table 6.5 Comparison of ontology, GV and API Doc available operations

Table 6.6 describes the properties mapping, and then describes exactly its relation between the ontology, GV and CBT specific API Doc. For the full API Doc definition see Hernández (2015c) for MindMeister and Hernández (2015d) for Cacoo.

MINDMAPPING				
Ontology		GV	MindMeister	Cacoo
User	userId	userId	id	
	username	userName	userName	User/name
	userEmail	userEmail	userEmail	
	Maps	Maps	User/maps	User/diagrams
				User/nickname
				User/imageUrl
Map	Id	Id	Id	Diagram/diagramId
	Ideas	Ideas	Map/ideas	
	creationDate	creationDate	creationDate	Diagram/created
				Diagram/url
				Diagram/title
				Diagram/description
				Diagram/updated
				Diagram/owner
				Diagram/comments
				Diagram/imageUrl
Idea	Id	Id		
	Title	Title		
	imageUrl	imageUrl		
	modifiedBy	modifiedBy		
	modifiedAt	modifiedAt		
	Parent	parent		

Table 6.6 Comparison of ontology, GV and API Doc available properties.

6.4.2 Online Document Editor CBTs API Doc for Google Drive Document Editor and Draftin

From existing online document editor CBTs reviewed in Section 6.1.2, it has been decided to select two of them for this doctoral dissertation. The chosen CBTs have a higher maturity level. Those are Google Drive and Draftin. Both CBTs were chosen because they had a very extensive and robust Web API.

For the Google Drive online document editor API Doc represented in Figure 6.7 is described the two resources of a document editor CBT: *User* and *Document*. A *User* resource (learners and teachers in this context) inherits from the GV class for the *User* resource, the properties: `userId`, `displayName`, `emailAddress`. The `userPicture` property that is defined in the GV is not available. Then it contains a collection of *Files*. A *File* represents an instance of a *`Document`*, in the graph referred as Doc. Over the *User* resource there are two direct operations `de:RetrieveDocumentsOperation` and `de:NewDocumentOperation` both defined in the GV as well.

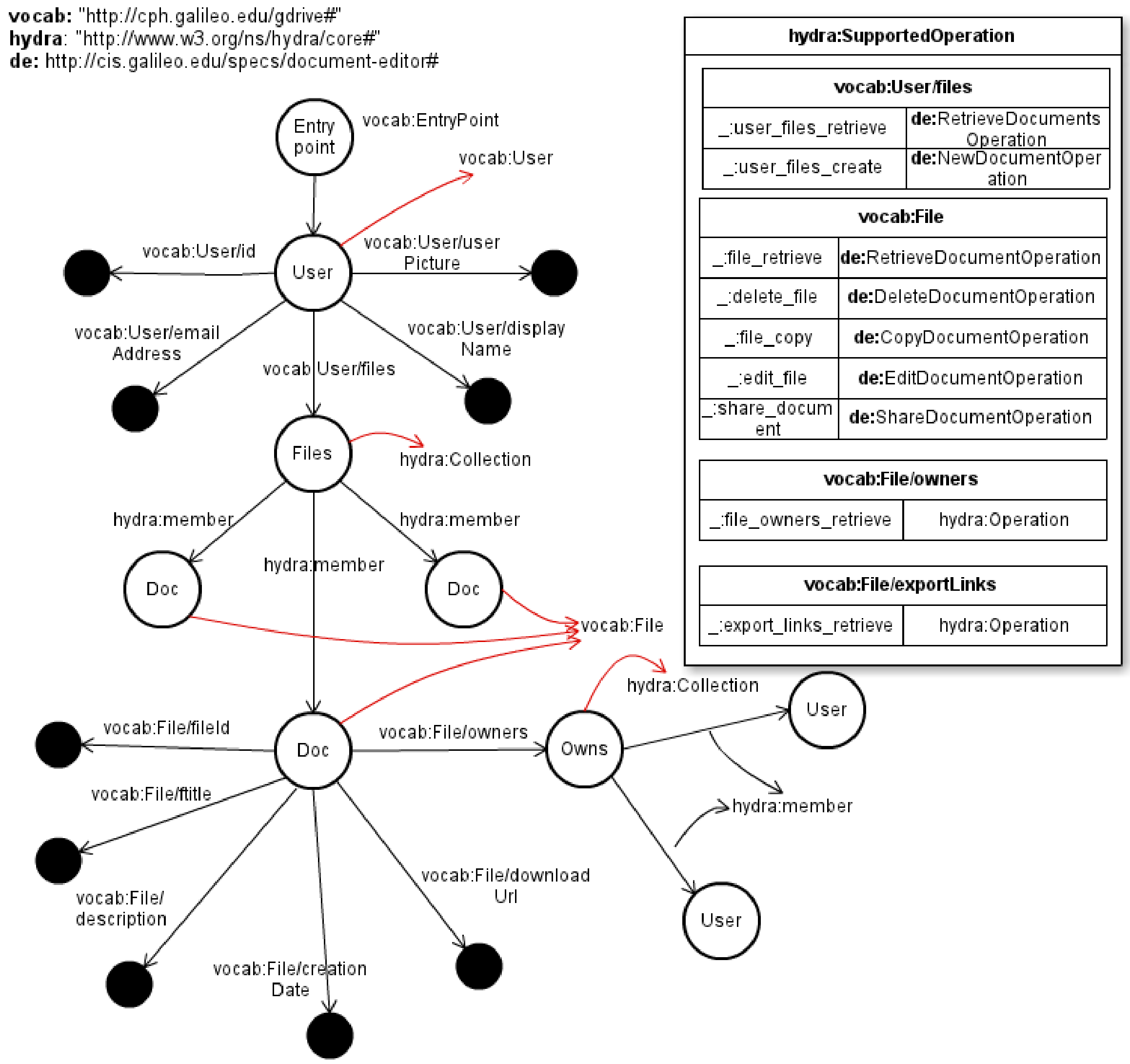


Figure 6.7 Google Drive Document Editor API Doc graph representation based on Hydra. "Doc(s)" as an abbreviation of Document(s) taken from (Hernandez & Gütl, 2015b)

A Doc has the properties of `fileId`, `title`, `description`, `creationDate`, `downloadUrl` and `owners`, which is a collection of *Users*. In this case all properties match to the GV definition. And the document resource has all the operations represented in the GV, and two additional operations: `owners_retrieve` and `export_links` operations.

For the second chosen CBT for online document editor, which is Draftin, the API Doc is depicted in Figure 6.8. It is highly similar to the one of Google Drive Document Editor, as it describes the two resources of a document editor CBT: *User* and *Document*. A *User* resource (learners and teachers in this context) inherits from the GV class, for the *User* resource, the properties: `userId (id)`, `emailAddress (email)`. The `displayName` property is not supported, and instead it has its own substitution with `firstName`, `lastname`. The `userPicture` property that is defined in the GV is not available. Then it contains a collection of documents. A *Document* represents an instance of a *Document* resource, and as a difference with Google Drive Editor, this is named *File* instead of *Document*, that is because the real Web API uses different naming conventions but represent the same, therefore the GV serves to identify that it is the same type of the resource. The *User* resource there are two direct operations `de:RetrieveDocumentsOperation` and `de:NewDocumentOperation` both also defined in the correspondent GV.

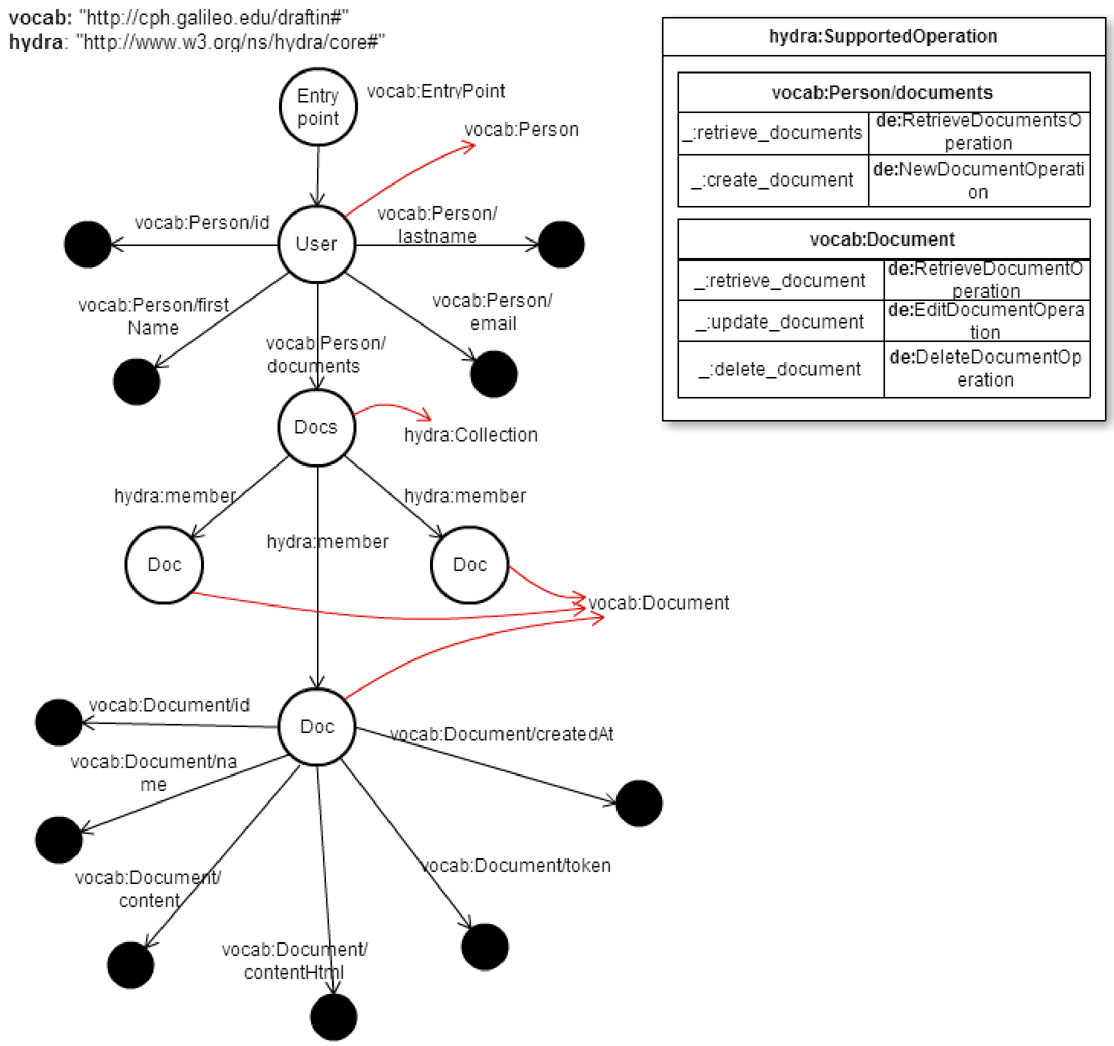


Figure 6.8 Draftin Document Editor API Doc graph representation based on Hydra. "Doc(s)" as an abbreviation of Document (s)

A Doc has the properties of `fileId (id)`, `title (name)`, `creationDate (createdAt)`. And is missing from the GV description, `downloadUrl` and `owners`. But it has some other new introduced properties such as content and contentHtml. And the document resource has almost all the operations represented in the GV, but missing two operations: de:CopyDocumentOperation and de:ShareDocumentOperation operations.

As in the previous Section, Table 6.7 and 6.8 covers a comparison between the ontology, and the GV, and the CBTs API Doc implemented operations. As in the previous ADT, it has not been used all the Web API available, and it is just focused on the necessary operations for supporting the use case. This is particularly notorious in the document editor ADT, because as it does with the Google Drive, it conforms more than just a document editor, this includes other type of applications, such a presentations, file storage, spreadsheets, etc. The properties mapping and their relation between the ontology, GV and CBT specific API Doc is depicted in Figure 6.8. For the full API Doc definition see Hernández (2015e) for Google Drive and Hernández (2015f) for Draftin.

ONLINE DOCUMENT EDITOR				
	Ontology	GV	GDRIVE	DRAFTIN
User	Id	Id		Person/id
	emailAddress	emailAddress	User/emailAddress	Person/email
	DisplayName	DisplayName	User/displayName	
	userPicture	userPicture	User/userPicture	
	Files	files	User/files	Person/documents
				Person/firstname
				Person/lastname
Document	fileId	fileId	File/fileId	Document/id
	Title	Title	File/title	Document/name
	Description	Description	File/description	
	downloadUrl	downloadUrl	File/downloadUrl	
	creationDate	creationDate	File/createdDate	Document/createdAt
	owners	owners	File/owners	
				Document/content
				Document/contetHtml
				Document/token
				Document/createdAt

Table 6.7 Comparison of ontology, GV and API Doc available properties

ONLINE DOCUMENT EDITOR			
Ontology	GV	GDRIVE	DRAFTIN
NewDocument	NewDocument	_:user_files_create	_:create_document
DeleteDocument	DeleteDocument	_:file_delete	_:delete_document
EditDocument	EditDocumenht	_:file_edit	_:update_document
RetrieveDocument	RetrieveDocument	_:file_retrieve	_:retrieve_document
RetrieveDocuments	RetrieveDocuments	_:user_files_retrieve	_:retrieve_documents
CopyDocument	CopyDocument	_:file_copy	X
ShareDocument	ShareDocument	_:share_file	X
		owners_retrieve	
		export_links	

Table 6.8 Comparison of ontology, GV and API Doc available operations

6.5 Discussion

The following Section includes parts of the publication by Hernández, et al. (2014a) and Hernández, et al. (2014b).

The proposed ontologies have modeled the mind map and online document editor CBTs Web API, thus serving to create a *Generic Vocabulary (GV)* for each *Application Domain Type (ADT)* using Hydra. The ontology covers the common operations and properties of the CBTs, and can enable the correct interoperability for such CBTs in the required scenario. Hydra provides the ability to create a GV as representation for a Web API published by CBTs. It includes properties, operations, input, and output data types.

As pointed out in Section 4.1, there is a large ecosystem according to Chang and Güetl (2007) of educational CBTs, many with published API. In order to avoid duplicity for ADT (i.e., implementing multiple TPs definitions for a CBT type, such as a mind map), GVs, such as the depicted in Section 6.3, has been created. It allows the creation of API Documentation for each TP by using general classes from a GV. As has been described, the GV derives from the ontology presented in Section 6.2, leaving the special features of each CBT to be included at implementation time in the API documentation. This work allows any TP Web API to be discoverable. The discovery happens at TC and starts from a single entry-point, and from there it uses introspection to discover all functionalities exposed and provided by CBTs. Hydra enables the creation of a Web API definition at a higher level of abstraction. With a Hydra-enabled runtime environment (i.e., a processor of Hydra-enabled Web APIs), it is possible to make educational CBTs interoperable in a VLE without custom coding for a given CBT. So when TP publishes newer versions of the API (with new and extended functionalities. A TP can deprecate or replace operations and (or) properties), it is easier to update that API without the burden of custom coding for each CBT. To update to a CBTs Web API in Hydra is as follows. Add, update, or delete any resources, properties, operations, or data types described in the API documentation, using the GV classes if possible. The TP provides its Web API using JSON-LD and Hydra. The next time the TC requests services for the TP, it discovers (through a linked data Hydra-enabled engine) new, updated, or deprecated parts of the Web API. Then the TC automatically adapts interoperability communication and features offered to the teacher and (or) learner. There is no human intervention when interoperability happens, nor is a need to program custom interoperability contracts on the TC side. The TC may decide to review the features if it detects changes in TP services.

The GV in conjunction with API documentation of CBTs enable TC to perform operations over the CBTs. A mandatory component in a CEE is a public Web API from the CBTs. Currently, some of the CBTs support RESTful API and JSON payloads (as depicted in

Table 6.2 and Table 6.4). But as Hydra is a new vocabulary, none of those CBTs uses it. This limitation makes it necessary to have an intermediate layer, such as the CBT proxy defined in LTIv.2, with the capability to take a TP Web API and publish the translation in Hydra format. With this Hydra proxy (as in our semantic proxy described in Section 4.3.3), it is possible to take any Web API, create a proxy Hydra model of it, and enable the TC to process Hydra based Web APIs. And in the future, when the TP incorporates Hydra based responses, then the Hydra proxy is not necessary, the Hydra proxy is addressed in the following Chapter.

While the ADT CBTs available evolves, and the correspondent Web API become more mature, it is possible to enhance and expand the correspondent ontologies, therefore the GV has changes as well. For interoperability, using the presented Hydra approach, changes in Web API as described, have no impact in the interoperability with educational systems such as the VLE, because as it is presented in the next Chapter, a Hydra enabled run time environment has the capability to adapt to those changes, and administrators are able to choose whether to include or not certain features, etc.

The interoperability use case using JSON-LD and Hydra enables a high granularity management level of the VLE over the CBT, enhancing automation of course administrative task over the CBT (i.e. create a map of each group in a class, assign read permissions to comment over other group's map, etc.), and allows full access to the TP Web API. As presented in Section 6.4, it requires only writing the definition of a Web API to include a new mind map CBT. This can be achieved rapidly, without a large technical effort or complexity, although a big challenge is to convince web developer to use these technologies, providing SDK or programming languages libraries that easy the incorporation of these technologies might be a good approach.

When sharing CBT resources, for example a mind map CBT, the teacher is able to manage a very granular set of properties and operations, for example, limit edition access to the map until a given assignment deadline, create role-based learning approaches, allow collaboration between learners, etc. This enables very custom designed learning experiences in a CEE.

The approach of using a GV and the API Doc enables first to have generic services definition for an application type, while the TC (i.e. VLE) can choose to support only that predefined generic service resources, operations and properties (described in the GV) or allow to support extended features provided by the TP that are not described in the GV but in the API Doc.

There are other benefits of this approach, with the discovery at run-time of the application specific properties and operations, it can lead to automatically create user interfaces, input data validations, etc., which is presented in the next Chapter. The GV brings the benefit of a simple soft definition of the CBT at the design time. Thus leaving CBT specific interoperability details at the time of implementation, through a formal yet simple representation using the API Doc.

This interoperability approach for educational CBTs enhances and simplifies the integration of new CBTs in a VLE. The Hydra approach could enable further interoperability features such as the ones cited in this study, built on specifications such as IMS LTIv2, or complementing its current capabilities.

7 A SEMANTIC INTEROPERABILITY ARCHITECTURE AND SYSTEM FOR EDUCATIONAL ENVIRONMENTS

This Chapter presents an innovative architecture and the corresponding system implemented towards the semantic interoperability of educational systems. To cover an interoperability scenario of *Virtual Learning Environments (VLEs)* and CBTs, this creates a truly *Cloud Education Environment* (CEE) (Hernández, Linares, Mikroyannidis, & Schmitz, 2013a). The layered architecture (see Section 4.3.2) takes a semantic approach toward the inclusion of CBTs. This approach provides a way to interoperate with machine-processable CBT Web APIs without custom-made interfaces. The architecture, due to its cloud technology nature, is capable of handling thousands of learners. In this Chapter the *Cloud Interoperability Service (CIS)* is presented, the third generation and last iteration of development cycle. The new CIS is presented in Section 7.1, which is a multi-layered architecture that includes the use of Hydra and JSON-LD. It permits automatic machine processing of CBTs Web APIs. In Section 7.2 a *Communication Process Handler (CPH)* is implemented to invoke these CBTs' APIs (based on the conceptual model described in Section 4.3.3). Currently, available CBTs do not use Hydra, therefore this intermediary layer, transform communication between the CIS third generation and the CBTs is necessary.

This Chapter is based on previous work published in Hernández (2015h) and Hernández, Gütl, and Amado-Salvatierra (2014b).

7.1 The Semantic Cloud Interoperability Service

The following Subsections describe both the architecture and corresponding system implementation details of the semantic CIS. Including a general overview of the architecture, with its different system layers. The Section finishes with a summary of the CIS and the lessons learned while constructing the system.

7.1.1 *Cloud Interoperability Service* Third Generation Architecture

The following Section includes parts of the publication by Hernández (2015h).

The CIS second generation, which was presented in Hernández, Gütl, & Amado-Salvatierra (2014c) and described in the previous Section 5.3 make use of CIS service modules. Such service modules are built upon specific CBT Web APIs to interoperate with *Learning Activities*. Each module is a Symfony2 (Symfony, 2014) framework *Bundle* that enables the CBT Web API to communicate. The *Bundle* uses the CIS second generation services for authentication, communication, and business logic (see Section 5.3.1). This approach has proven to be capable of enabling interoperability, but it has major drawbacks in light of semantic technologies. First, it scales linearly, meaning that for each new CBT to interoperate, it requires a new custom-made bundle implying software programming. Another issue is maintenance of *Bundles*, which requires human intervention. This happens when the CBT Web API gets an upgrade. Furthermore, it takes time to make the upgrades, which may cause service interruption until the *Bundle* is updated. The whole original CIS architecture has been revamped to a semantic technology Hydra approach and is presented in this Section, see also Figure 7.1. This new CIS approach eliminates the aforementioned drawbacks of custom programming and custom maintenance. Thus, integration of new CBTs to CIS can be fully automatized. This implies no custom effort to make use of CBT.

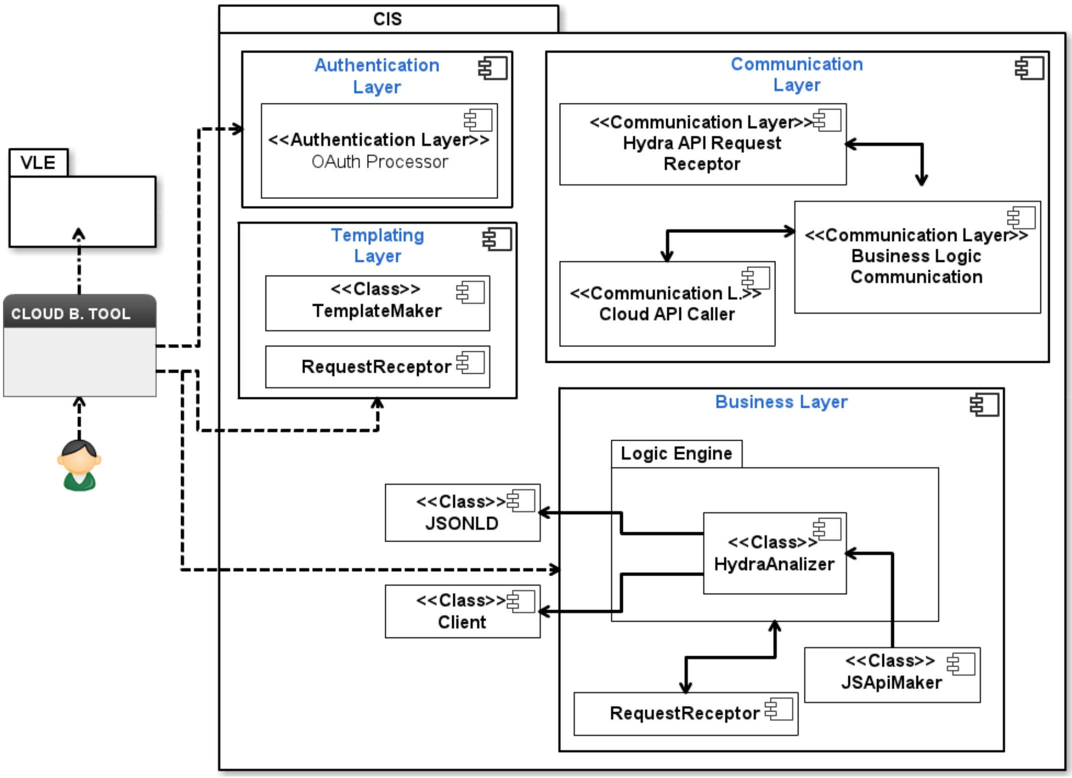


Figure 7.1 The CIS third generation architecture taken from (Hernández et al. 2013a).

The new CIS is built upon the Symfony2 framework, which is comprised of a layered architecture. Selected implementation details are covered in Section 7.1.2 to 7.1.5. The core layer is the *Business Logic Layer*, which is capable of processing the Hydra Web API. This layer provides the structure for operations, properties, data types, etc., as well as an embeddable JavaScript API for the *Template Layer*, enabling a user interface to the Web API. It uses the *Authentication Layer* to manage token-based identification, CBT Web API identity management and others. The *Communication Layer* processes all requests and responses between the CIS and the CBT Hydra Web API. Finally, the *Analytics Layer* has been extended to overcome the identified issues in Chapter 5. In the following Subsections, the layers are introduced.

7.1.2 Business Logic Layer

The following Section includes parts of the publication by Hernández (2015h).

The *Business Logic Layer* is the core of the CIS (see Figure 7.1), wherein the Web API sematic definition is processed. Here, the Hydra GV and API documentation are processed to automatically discover all operations, properties, etc., of a Hydra Web API. It has four main components: a) *JSON-LD Processor*. b) *Hydra Client*. c) *Business Logic Engine*. d) *Embeddable API*, as depicted in Figure 7.2.

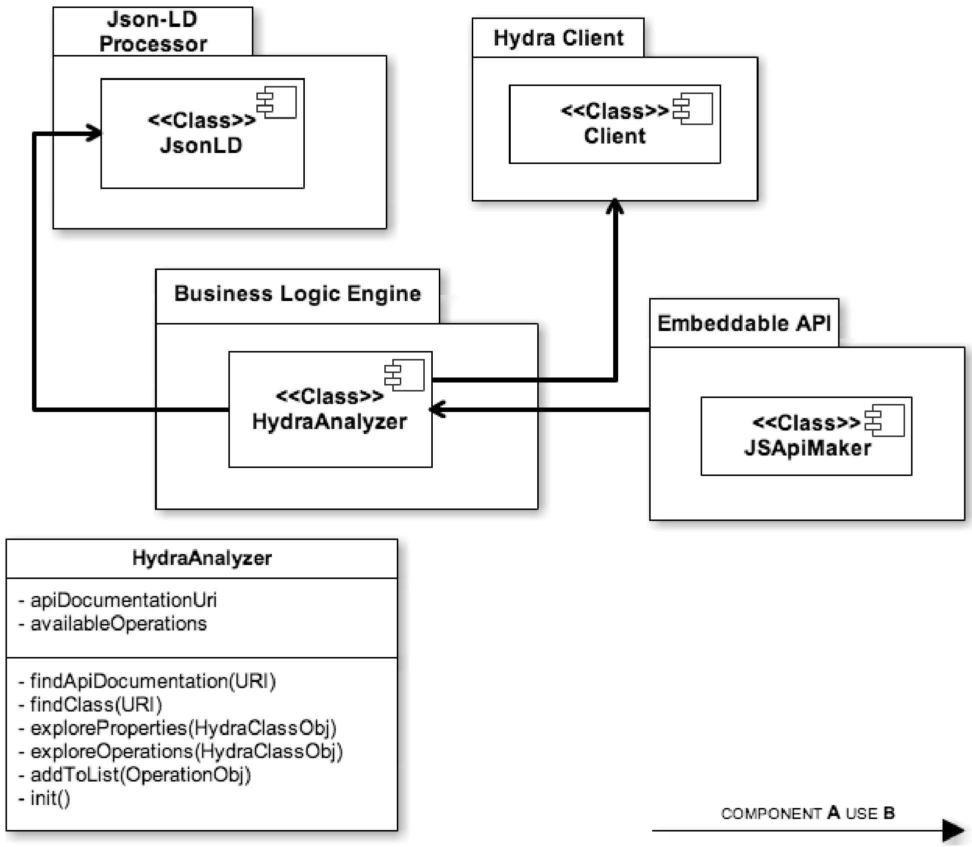


Figure 7.2 Business logic layer components

The *JSON-LD Processor* (JSON-LD, 2014) is used to deserialize a valid JSON-LD and then create PHP objects. These *Objects* contain all information related to that particular resource. For instance, a *Map* or an *Idea Resource*, with its corresponding *Operations*, *Properties*, etc. This *Processor* has been released and is available (Lanthaler, 2014; Hydra-Project, 2013). Thus, it is used as a third-party library.

The Hydra client is also available as a Symfony class. It is used to process the Hydra Web API. For example, it is able to get the API documentation URI of a Hydra Web API response. A set of enhancements has recently been done to support the latest Hydra vocabulary (as of June 2014).

The *Business Logic Engine (BLE)* is designed to discovering all operations and properties of a CBT Hydra Web API. The main purpose of the BLE is to build an operations list for the CBT Web API. It all starts with the entry point of the Hydra Web API. The BLE makes a request to that entry point, which gives a response in JSON-LD using Hydra. This contains a directive `@context`, which leads the BLE to the API documentation, which contains all the descriptions of the Hydra based Web API. This document is a JSON-LD document as well, and it is used the JSON-LD processor to convert the description into PHP objects which, in this case, are available within CIS. The created objects have all the descriptions of the Web API. The JSON-LD processor has built-in methods that allow searches within a JSON-LD document, which enables the BLE to use the results to further explore the document (using its own class, named HydraAnalyzer). At this point, the BLE begins a recursive search of the API documentation classes. It keeps discovering classes, as well as their properties and related operations (if there are any) until it finds primitive types (such as xsd:integer, xsd:date, etc.). Each newly discovered class is stored for future reference and use. This process is depicted in more detail in Figure 7.3

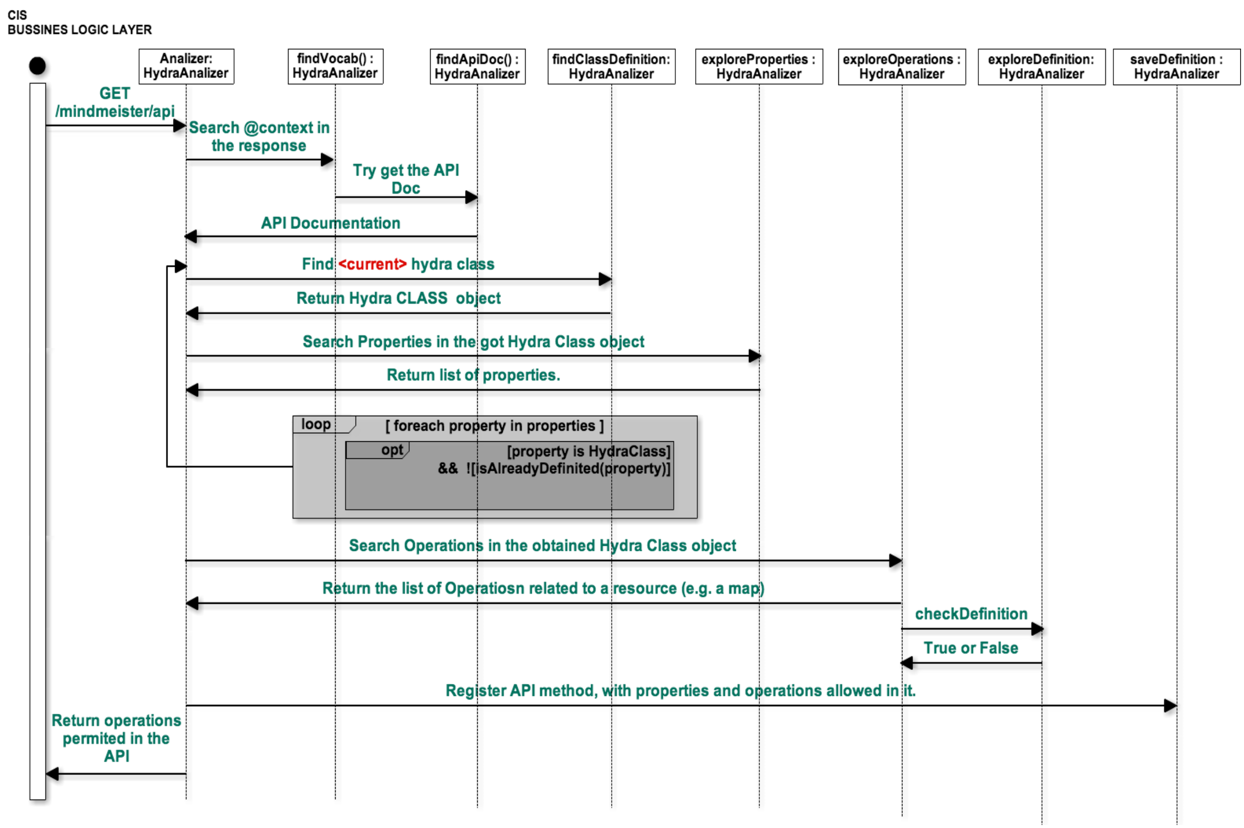


Figure 7.3 ***Business Logic Engine*** **sequence diagram**

The *Embeddable API* component (see Figure 7.4) is used to generate a JavaScript API that represents the Hydra API. It uses JavaScript because is designed to run this API in a browser, which opens the possibility to embed it. Embedding the API becomes highly relevant in supporting the *Functional Widget Interface* (Soylu, Möditscher, Wild, De Causmaecker, & Desmet, 2012) and opening the path to embed other systems such as personal learning environments (Friedrich et al., 2011). With the detailed description of the Hydra Web API that is provided by the BLE, the *Embeddable API* component (Figure 7.4) builds JavaScript-signed AJAX functions. These functions enact processes with the *Communication Layer* and manage the *Expect* and *Return* definitions. Ways have also been conceived to add extra pieces of code that also can be embedded for other purposes, such as analytics code that collects relevant behavior information.

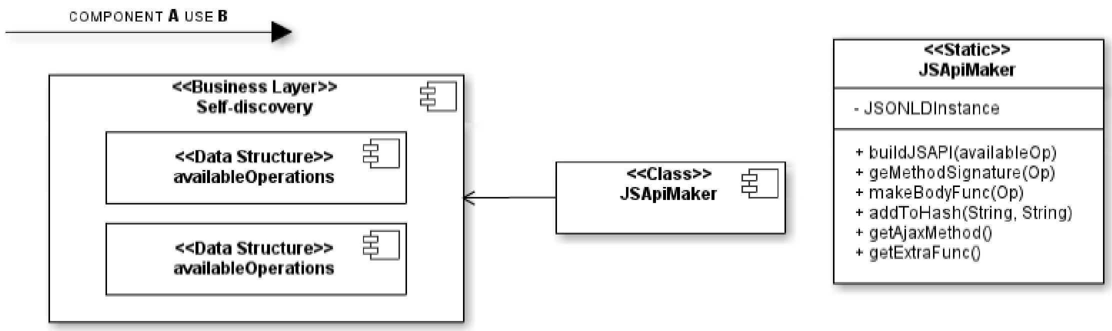


Figure 7.4 Embeddable API components

The *Embeddable API* module creates the embeddable JavaScript that represents the CBT Web API. As an example, *Embeddable API* generated (see Listing 7.1) defines a namespace, and then it contains the related operations, such as `mindmeister.createIdea`, which uses the properties required by that action. When it is needed to refer to another resource, it uses a JSON object for further description. Then, `doAjax` sends related requests to the *Communication Layer*. Keep present that the source for create the *Embeddable API* is the Hydra JSON-LD document.

```
(function (mindmeister, undefined) {
   mindmeister.doAjax = function(hydraUrl, method, hydraClass, params) {
       ...
   };
   ...
   mindmeister.createIdea = function (mapId, title, ... , position) {
       /* mapID and title are primitive type, integer y string respectively
          position is a JSON object. */
    mindmeister.doAjax(mindmeister.hydraUrl,"POST",mindmeister.apiDoc.idea,{"mapId":\
 mapId, "title": title, "position": {"x": position.x, "y": position.y}});
      ...
   }
}) (window.mindmeister= window.mindmeister || {});
```

Listing 7.1 The generated JavaScript from Embeddable API module based on Hydra Web API

Figure 7.5, outlines the sequence diagram beginning with the users' browser, which has the user interface presented for the CBT that usually includes an HTML form that uses the generated embeddable JavaScript. Thus, this HTML form uses the proper JavaScript functions for each CBT operation, which consequently send requests to the CBT Web API through the *Communication Layer*.

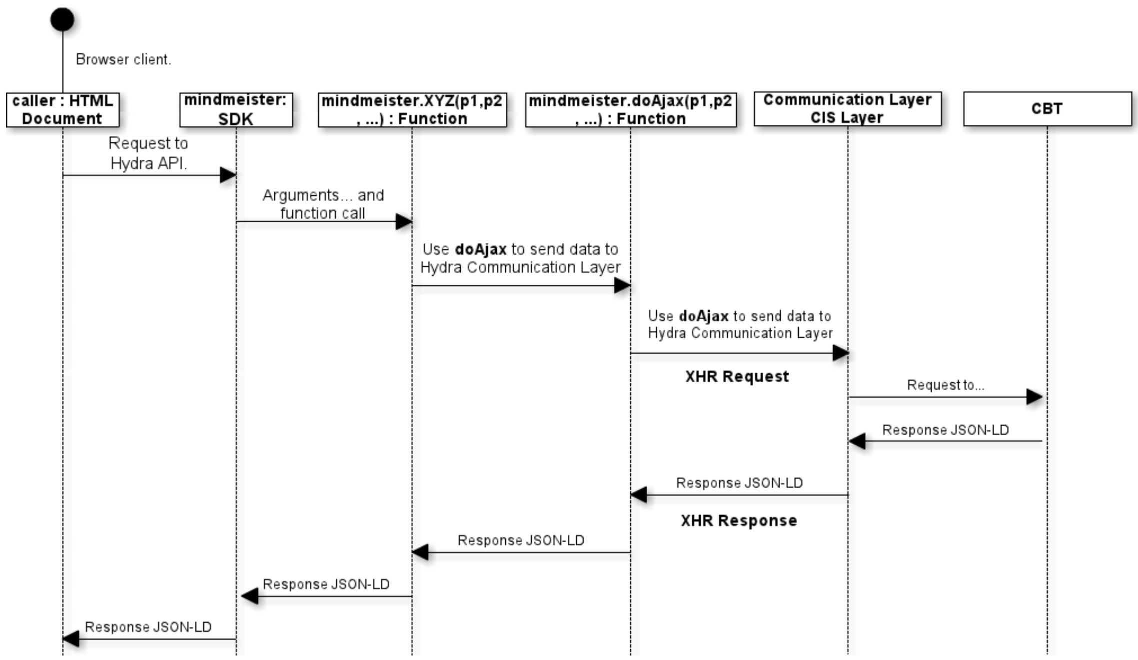


Figure 7.5 Sequence diagram from the client request to the CBT and the response.

7.1.3 Authentication Layer

The following Section includes parts of the publication by Hernández (2015h).

As with the previous version of the CIS architecture, the *Authentication Layer* uses a token-based authentication mechanism using OAuth 2.0. The OAuth 2.0 authorization framework *"enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its own behalf"* (IETF, 2012). It has been chosen as the basic authentication because it is a widely used mechanism, and many of the CBTs to be used already support it. OAuth 2.0 has four authorizations grant types. For the interoperability requirements, the *Authorization Code Grant (ACG)* and the *Implicit Grant (IG)* types are the most suitable for the CIS architecture (the others are for desktop or mobile scenarios). Both ACG and IG request and get an "*access token*" that is used to send authenticated requests to the Web API. The access tokens are defined as "*credentials used to access protected resources. An access token is a string representing an authorization issued to the client […] Tokens represent specific scopes and durations of access, granted by the resource owner, and enforced by the resource server and authorization server*" (IETF, 2012).

The ACG process starts with a user accessing the CLAO, which tells the user to login to the CBT (e.g., MindMeister, Google Drive, etc.). Then the CIS sends its `client ID` to the authorization server. After the user is successfully logged in, the user is asked to grant access to the CIS, then is directed to the CIS again, and then the authorization server send an authorization code, to which the CIS responds by sending back the code with a *Client Secret* and it receives the access token. IG is quite similar to ACG, except that the access token is given back to the CIS once the user has completed the authorization. The main difference between IG and ACG is that IG directly receives the access token and ACG requires an extra step. Firstly, it gets an authorization code that is used in a second step to get the access token. Both ultimately return an access token, which then is injected into the headers of each REST-based request.

To enable either the ACG or IG, the CIS manages the following data structure: a `hydra_api` that contains the cloud-based tool Hydra API identifier. Then it defines the *Authorization Grant Type* (e.g., ACG or IG). Regarding the `authorization_uri`, for IG it returns the access token. In the case of ACG, it returns an authorization code that is then used to request the access token. The CIS is modeled to support sending dynamic parameters in this request. The *`code_name`* is the parameter used for the authorization code for ACG to further request the access token. The *`token_uri`* that only serves the ACG is the URI to get the access token. The *`return_uri`* contains the URI to which the user is returned once the initial authorization has been completed. Finally, the *`token_name`* has the parameter name that contains the access token. The CIS, ACG and IG data structure is depicted in Listing 7.2.

```
hydra_api: 01235,
authorization_type: IG | ACG a ,
authorization_uri: {
    uri: http[s]://www.mindmeister.com/api/auth,
    method: POST | GET,
    params: {
 secret_id: {
       optional: false,
       value: 0123456
   },
client_id: {
       optional: false,
       value: ABCDEF
   },
},
dynamic_params: {
   “sig”: {
        “calculateBy”: “md5(secret_id${secret_id}client_id${client_id})”
   }
}
}
code_name: “<code_param_name>”,
token_uri: {
   uri: http[s]://www.mindmeister.com/api/auth?method=mm.auth.getToken,
   method: GET,
   params: {
       “code”: hydra_api.code_name
   }
},
return_uri: http[s]://miapp.hydra.com/api/auth,
token_name: “<token_param_name>”
```

Listing 7.2 The CIS ACG and IG JSON data structure for authentication

With this structure, the authentication mechanisms are organized in the following steps: First, the user, through the Web interface generated by the *Template Layer*, requests the URL to get the authorization using OAuth 2.0. This request is channeled through the *Communication Layer*, which sends a request to the cloud-based Hydra API, which returns the URL that is used for the OAuth 2.0 process. With this URL, it sends a request that is ultimately performed by the *Authentication Layer*. This layer, which builds all the negotiation parameters, is required for the authentication. It performs the necessary steps for the ACG or IG authentication process. Once the process is authenticated, it registers the access token for further request to the Hydra Web API that is done by the CIS. The whole authentication process is depicted in Figure 7.6.

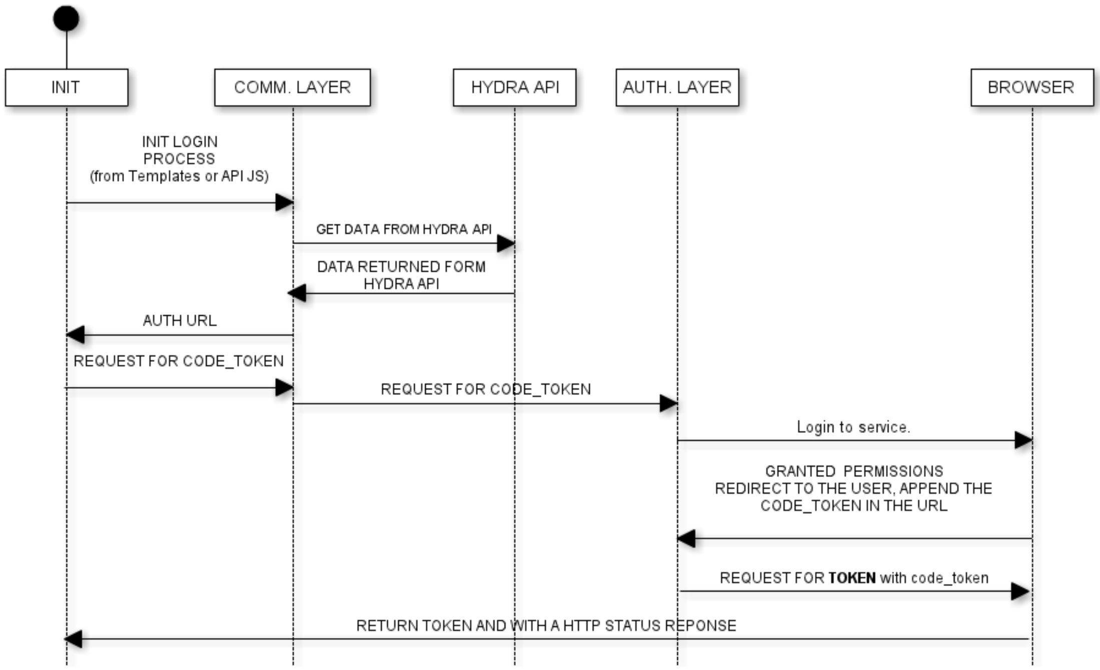


Figure 7.6 The authentication process for requesting and getting the access token.

7.1.4 Communication Layer

The following Section includes parts of the publication by Hernández (2015h).

The *Communication Layer* acts as the communication channel between the CIS architecture and the CBT Hydra Web API. The client sends a request to the Hydra Web API, which only needs to send the URI, the operation and its parameters, and the HTTP method. All this is obtained from the *Embeddable API* layer. This *Communication Layer* also helps generate a single request point to the CBT Web API, instead of direct requests from clients, which are something that is likely to be rejected by CBT security policies. Also, this single request channel serves to eliminate the *Same Origin Policy* restriction (W3C, 2014). This restriction does not permit a JavaScript to make requests to webpages that are allocated to other domains. Although cross-origin resource sharing allows overcoming that issue, it is still not widely supported by all browsers. JavaScript requests to other servers prove to be useful if it is planned to use the *Embeddable API* in a widget, as has been done in previous versions of the architecture (Hernández, Linares, Mikroyannidis, & Schmitz, 2013a).

Figure 7.7 shows the 3 components of the *Communication Layer*. The first component is the *Hydra API Request Receptor*, which is in charge of receiving a request from the client (as either the *Template Layer* or the *Business Logic Layer*), and it verifies that the request comes with all required data. Then it is sent to the *Request Constructor and Serializer*, which builds the real Hydra Web API serialized request. The *Cloud API Caller* injects the authentication data provided by the *Authentication Layer*, sends the final request to the cloud service and processes its response. The response comes in a JSON format, with the HTTP status code indicating request status.

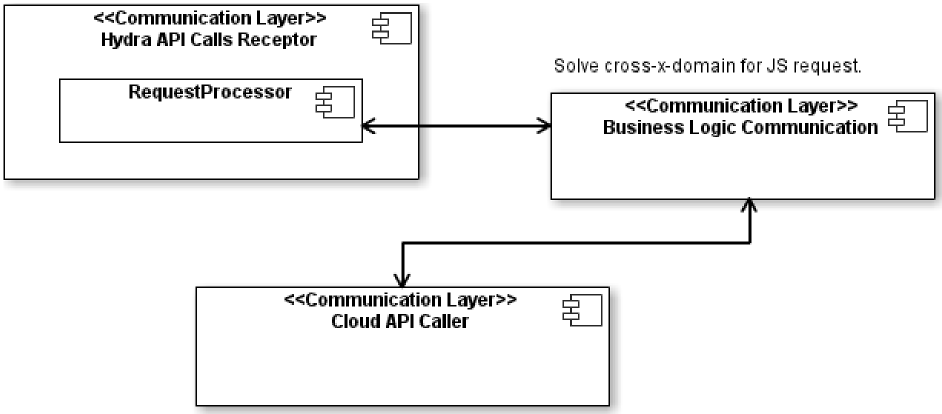


Figure 7.7 Communication layer modules.

7.1.5 Template Layer

The following Section includes parts of the publication by Hernández (2015h).

The aim of *Template Layer* is to create the HTML-based interface that corresponds to each of the available operations and provides the correct input types. This enables the user (teacher or learner) to manipulate or control these operations from the CLAO. Following the use case of Section 4.2, the teacher might give an independent mind map for each learner. Then, through the CLAO, finds an interface to automatically create thousands of maps and assign specific permission for each learner to his or her own map instance. Browser interfaces to do this administrative task on behalf of the teacher are automatically generated and provided by the *Template Layer*.

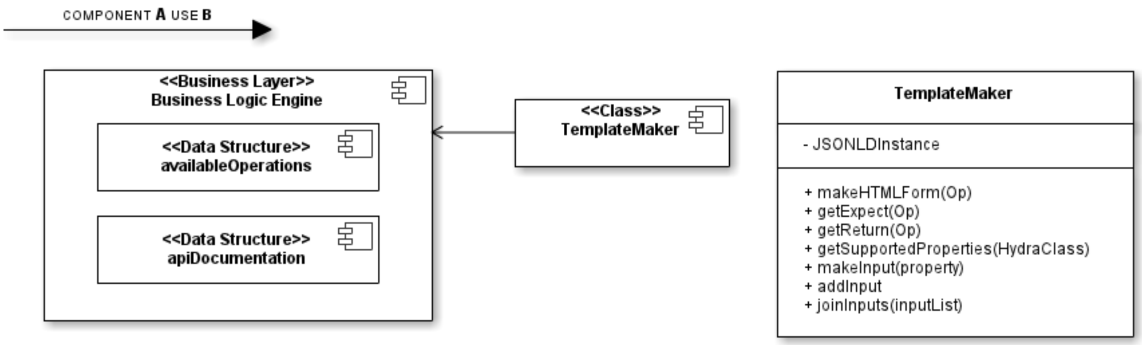


Figure 7.8 Template layer components and use of business logic layer

The *Template Layer*, depicted in Figure 7.8 uses the *Business Logic Layer*. More specifically, the BLE, as a base to automatically create Web interfaces for each of the operations with its correspondent properties and input types if they are available. The avid reader may notice that a property that is `readonly`, therefore, is presented as non-editable. An HTML form is constructed by each operation, which contains all the related properties needed for that operation. Based on Listing 7.1, the HTML result is depicted in Listing 7.3. The HTML form includes both editable and non-editable properties that are configured accordingly. The form generation also takes into account the use of the specified Web form method (POST or GET) for the operation and uses a form action to call the *Embeddable API* (generated at the *Business Logic Layer*), which then sends the request to the *Communication Layer*.

```
<form method=”<[hydra:Operation]method>” action=”[hydra:Url]”>
  <label>Title</label><input name=”map.title”></input><br/>
...
  <div>
      <span>POSITION</span>
      <label>X</label><input name=”map.position.x” type=”number”></input><br/>
      <label>Y</label><input name=”map.position.y” type=”number”></input><br/>
  </div>
</form>
```

Listing 7.3 HTML automatically generated by the template layer from the CBT Hydra Web API

7.1.6 Summary and Lessons Learned

The following Section includes parts of the publication by Hernández (2015h).

The CIS third generation is able to handle all the requirements for interoperability with a CBT using a semantic approach with Linked Data. That approach does not requires custom coding for each CBT, since the CBT Hydra Web APIs are self-described, and the *Business Logic Layer* is able to process and utilize all their services and use the *Communication Layer* to further interact with the CBT. The authentication issue is solved as well by the CIS, and the *Template Layer* of the CIS automatically generates administrative user interfaces for the CBT operations.

One of the problems that have arisen while using MindMeister CBT is that it provides only 3 map instances with the free account. To overcome that problem, the Galileo Univerisity main paid account had unlimited map creation, and the CIS only needs to share the map instance created with the main account to a given learner. This gives control over what has been shared with each learner. The ownership of the map instance resource that belongs to the VLE is highly fundamental, as it has been identified in previous studies (Hernández, Amado-Salvatierra, & Gütl, 2013c). Therefore, the VLE can manage the access control to it, which is quite important, because it is important to not lose any of the maps and to make such map instances read-only after the assignment deadline.

The CIS *Authentication Layer* proved to simplify the authentication mechanism using OAuth 2.0. Additionally, the CIS analytics layer now is capable of representing correctly the user flow, this due an addition of better URL representation (REST style) in order to identify to which resource it is referring to in the Google Analytics interface.

A drawback of this CIS third generation architecture is that it did not manage Hydra Web API versions, a version indicates new and changed features in a Web API. Therefore, it was not possible to automatically determine if the current request is dealing with a Web API that has been already identified and cached (in memory for performance reasons) in a previous request. The cache is cleaned every day to update it to the latest version of the Hydra Web API. Thus, it was conceived a more effective and efficient approach, that is create versions numbers of the CBT Hydra Web API. This allows the CIS to automatically identify and update the cached Hydra Web API information when needed. Relevant to mention that according to Hydra's author, it is conceived to process the Web API in each single request, although also some cache mechanisms might be in place to enhance performance.

Regarding the CIS *Template Layer*, it provides to the CLAO environment a user interface structure that is automatically built upon the service definition. That initial approach works well, but still is needed an interface organizer for operations, resources, etc., to enable better user interfaces and automated administrative processes. Consequently the automatically generated user interface is in need of an improvement to be configurable to present a simpler and more coherent interface to the user (e.g., hiding properties that are not meant to be seen by users, such as a map instance ID). This includes providing a user interface for system administrators to verify which are the available operations and configure which of those operations have a correspondent user interface control that is available for the user (teacher or learner) to manipulate when using the CBT.

7.2 Communication Process Handler to Support Semantic Interoperability

The following Section includes parts of the publication by Hernández, et al. (2014b).

As Hydra is a very new approach for JSON-LD based Web service interoperability, it can be expected that currently available Web APIs do not make extensively use of it yet. However, many of the most mature and popular tools and services have already rich open APIs. That is, for example, the case for MindMeister which has a RESTful XML based API, and Cacoo which offers a JSON/XML based API, and is the same for all the used CBT in this thesis. Given that, a *Communication Process Handler (CPH)* has been implemented to invoke these services' APIs (based on the semantic proxy defined on Section 4.3.3). Thus communication can be based on Hydra, the CBT domain type *Generic Vocabulary*, and the specific API Docs created for each CBT respectively. The CPH permits a TC to invoke CBT using Hydra APIs, and processes their responses.

Figure 7.9 illustrates the conceptual model of the *Communication Process Handler (CPH)*. It depicts the request and response process model, which mainly consist on the following four steps: 1) a TC makes a request to the TP (the CBT). 2) The CPH has a collection of invoked service API methods equivalents for each possible request. Therefore it determines which is the correspondent method to perform the request for a given CBT. 3) The TP, for example the MindMeister or Cacoo tool, offers responses in two formats: XML or JSON. 4) The CPH with the payload obtained from the TP constructs the Hydra-based response that the TC is able to process. It is worth to mention that each Hydra response comes with the "`@context`" directive, which adds semantic to the response through referencing to the API Doc, and the corresponding data types to be used. This means the TC is able to introspect the response, as depicted in Listing 7.4. This example provides the context for a map referring the API Doc of MindMeister, and the semantics for the `ideas`, `collaborators` and `creationDate`.

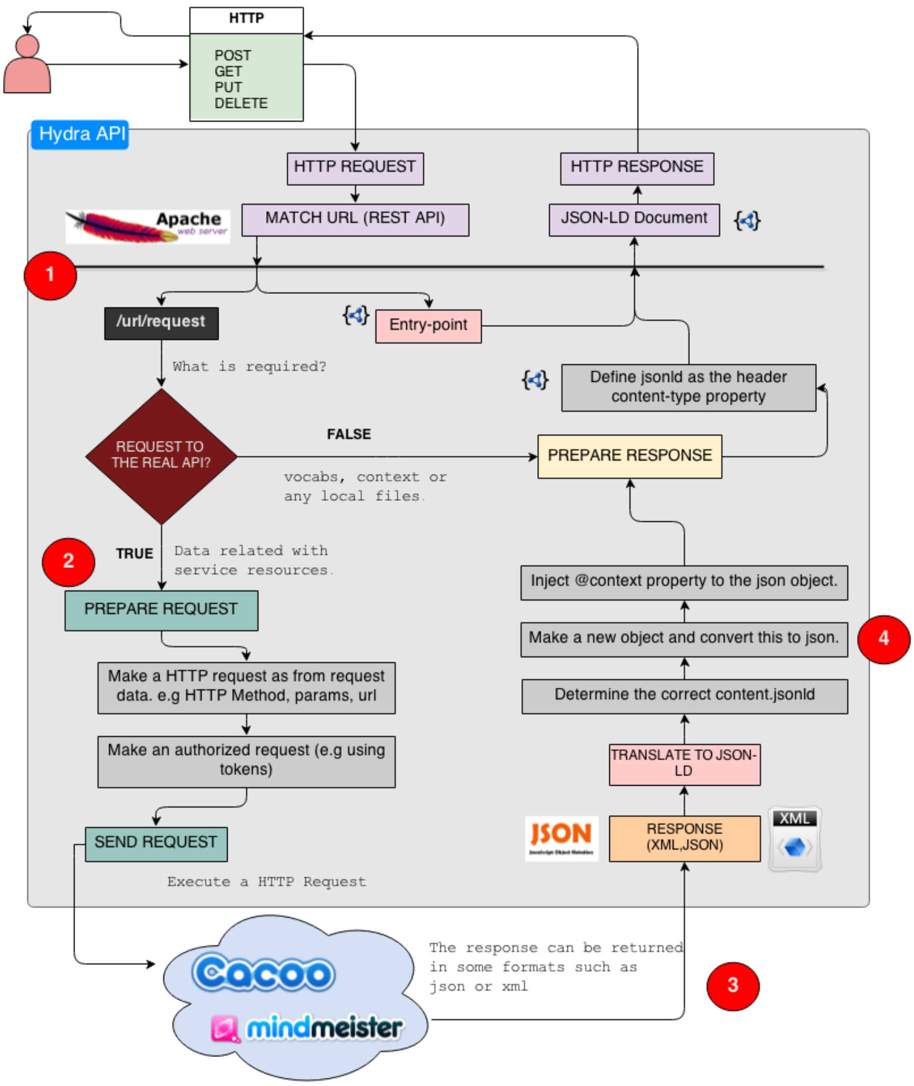


Figure 7.9 The CPH request and response process model

```
"@context": {
            "vocab": ”mindmeister-api-doc#”,
            "hydra": "http://www.w3.org/ns/hydra/core#",
            "ideas": {
            "@id": "vocab:Map/ideas",
                    "@type": "@id"
            },
            "collaborators": {
            "@id": "vocab:Map/collaborators",
                    "@type": "@id"
            },
            "creationDate": "vocab:Map/creationDate"
        }
```

Listing 7.4 Hydra context for a map resource

Furthermore, linked media types can be used for service discovery and introspection. For example the `retrieveMaps` TP operation for MindMeister. The response depicted in Listing 7.5 represents a collection of maps (as seen in the Listing, it contains two maps: `map id` 363606106 and 353929271).

```
{
  “@context”: ”context-maps”,
  “id”:   http://mindmeister.api/<user_token>/maps,
  “@type”: “Collection”,
  “members”: [
     {
        “@id”:”/<user_token>/maps/363606106”,
        “@type”: “vocab:Map”
     },
     {
        “@id”:”/<user_token>/maps/353929271”,
        “@type”: “vocab:Map”
     }
   ]
...
```

Listing 7.5 Hydra response to retrieveMaps operation

With the response in Listing 7.5 it is possible to discover further information of the CBT. It is inferred to use the `retrieveIdeas` operation because the @type describes it as a *Map* resource, such as described in Section 2, in the API Doc the map resources has related operations.

A request for `retrieveIdeas` for map (map id 363606106) receives a response from the TP with the *Ideas* collection presented in Listing 7.6. All these Hydra enabled communication is translated and performed by the CPH.

```
{
  “@context”: ”context-map”,
  “@id”:   "http://mindmeister.api/<user_token>/363606106",
  “@type”: "Collection",
  “member”: [
     {
        “@id”:”/<user_token>/maps/363606106/363606107”,
        “@type”: “vocab:Idea”
     },
     {
        “@id”:”/<user_token>/maps/363606106/363620037”,
        “@type”: “vocab:Idea”
     }
   ]
}
```

Listing 7.6 Hydra response to retrieveMaps operation

In Listing 7.6 the *Map* (`map id` 363606106) has two associated *Ideas* (`idea id` 363606107 and 363620037). As it has been exemplified it is possible to discover further resources by getting detailed information about the *Ideas*, etc. This discovery process of resources, its properties and operations, can be applied recursively, until related information of a given object (a *Map* instance) has been obtained and processed. With a Hydra-enabled run time environment capable of this discovery process, no specific coding for each tool is required. Only the Hydra *Generic Vocabulary* and API Doc descriptions have to be applied.

Figure 7.10 outlines the CPH architecture and maps the steps described in the CPH request and response model (refer to Figure 7.9): 1) TC Hydra request (Hydra API call receptor), 2) CPH does request to TP (HTTP request maker and caller), 3) TP returns response (HTTP response processor and maker), 4) CPH returns Hydra response to TC. First a request receptor component attends requests to the cloud-based Hydra API, serving as proxy to the communication. Based on the TC request is obtained the correspondent CBT API Doc. Then this is handled to the request processor, which extracts all related information that needs to go through for the final request to the CBT. Information such as expected payload, properties to be sent with the request, etc. The request is formed based on the TC request, but translated to the current Web API. That final request to the tool includes the use of Hydra *Template Links*, which serves to model automated URLs queries to the TP. This gives the CPH a template to construct the URL at runtime, includes the query parameters to be sent with the request.

The Listing 7.7 presents the Hydra *Template Links* use. This represents that the template has the actual URL to use, each `mappings` maps a query variable used in the template to a property and may optionally specify if it is required or not. The *Template Links* becomes useful in cases such as sending parameters in a RESTful request, parameters such as api-key, tokens, etc. that are only known by the TC, but the requests are already defined with the *Template Link.*

```
{
 "@context": "http://www.w3.org/ns/hydra/context.jsonld",
 "@type": "IriTemplate",
 "template": "http://api.example.com/issues{?q}",
 "mapping": [
   {
     "@type": "IriTemplateMapping",
     "variable": "q",
     "property": "hydra:freetextQuery",
     "required": true
   }
 ]
```

Listing 7.7 Template Link example

Once the HTTP request is made by the CPH, the *Response Processor* module takes the payload from the TP (CPH request and response model step 4), validates the response. Correct format is verified for both JSON or XML responses. Then the response is Hydra formatted and serialized using the *Response Maker* module, see Figure 7.10, and finally the CPH sends the response to the original TC. Some response properties from the CBT (TP) do not automatically match with the Hydra defined properties that the TC requires, therefore a configuration interface module is provided to match all properties between API Doc and CBT. The property match process is done as an initial setup *Contract* for a given TP, and only needs to update that *Contract* if the TP changes its response properties.

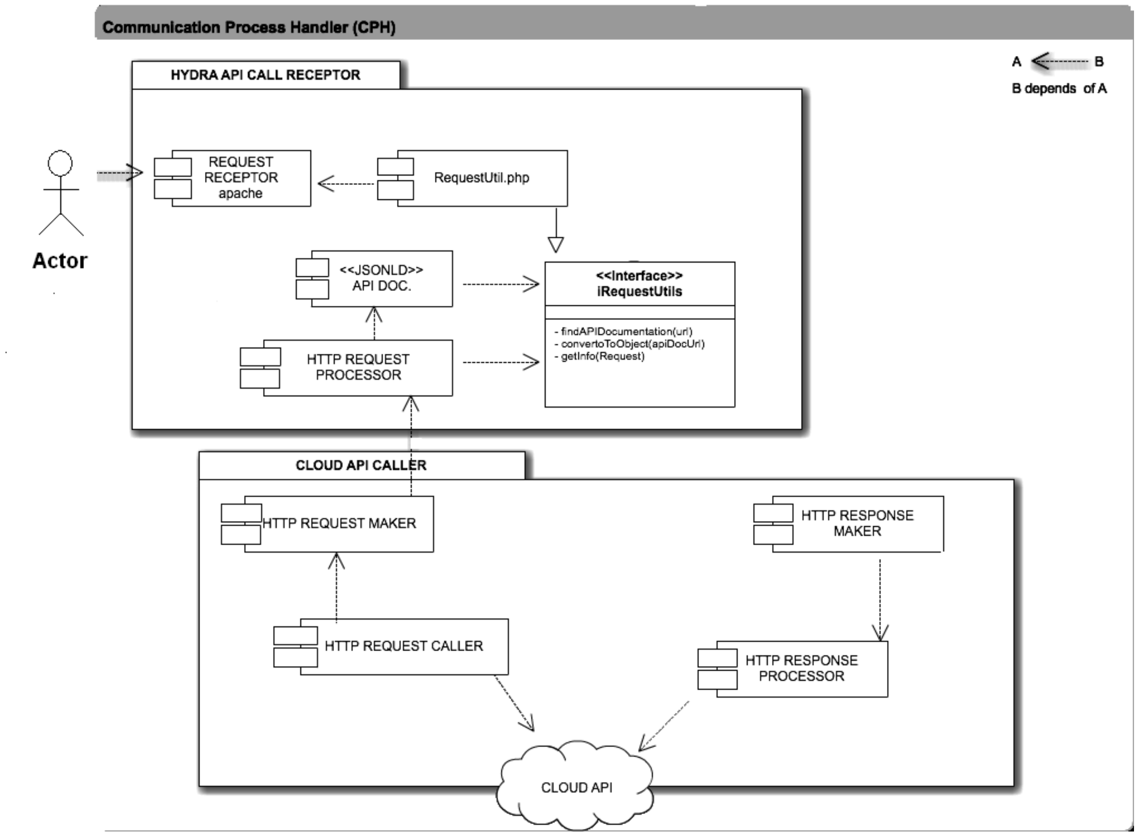


Figure 7.10 Communication Process Handler architecture

SUMMARY AND LESSONS LEARNED

The CPH communicates with the CBT API by using Hydra definitions. Instead of writing code directly based on the underlying API of a mind map (or any other tool for that matter), all requests go through the CPH harmonizing the APIs. The advantage is that interoperability becomes much more efficient and the programming can happen on a higher level of abstraction. The whole system is built on a layered architecture that helps to concentrate the transformations on a single layer instead of having to spread them throughout the whole code base. As soon as the third-party CBTs API adopt JSON-LD and Hydra themselves it becomes possible to completely eliminate that intermediary layer that is named the *Communication Process Handler (CPH)*.

Adding another CBT requires to manually define the API Doc for the CPH, which covers the CBT specific details of their supported (current) Web API. This process requires no coding, it is just a JSON-LD Hydra definition of the available API. This involves to review the CBT API, identify manually each available method, and to write its equivalent as Hydra, then register that definition in the CPH. It is necessary to define the objects, its properties and corresponding operations, and reflect them in the newly created API Doc for new CBT.

The API Doc is used to resolve the URL, structures, responses, operations and its properties. At the CPH, once it matches the request of the TC with an operation defined in the API Doc, it is built the correspondent real CBT Web API request. Once the cloud API returns a response, the CPH process it. Thus CPH has in place a match mapping for each valid response with the correspondent API Doc's valid properties. Once the CPH is configured, it automatically interprets all further responses from the CBT Web API. The CPH may need further extension as long as new CBT are added, the main reason is due the fact that many of the current CBT Web API implement slight variations on their own way of interpreting and implement REST, or authentication mechanisms.

7.3 Discussion

The following Section includes parts of the publication by Hernández (2015h), and Hernández, et al. (2014a).

The CIS architecture has been a three-phase iterative software development process, this Chapter has described the CIS third generation architecture, which enables a real CEE using CBTs. This innovative approach, which uses a semantic description of a CBT Web API along with Linked Data with the Hydra vocabulary for Web API descriptions, yields a robust interoperability process. There is no further need to custom-program API interfaces between systems. Instead, using Linked Data makes the Web API discoverable. Thus, with a Hydra Web API, the CIS is capable of processing it at run time. In addition to that, the CIS also brings enhanced control to CBTs by interacting with their available operations (e.g., CRUD operations). Neither Web API discoverability nor operations management is yet supported by interoperability frameworks and international specifications.

To enable this scenario, a *Communication Process Handler (CPH)* (Hernández, Gütl, & Amado-Salvatierra, 2014b) has been created and established to deal with the services that are not currently exposing their Web API using Hydra. Due the fact that Hydra is still is a W3C working group (Hydra, 2014), it is not in use by any of these CBTs. The CPH allows the conversion communication back and forth between the CIS and the CBT, using Hydra, so it acts as a communication-processing platform to register Web APIs that use JSON or XML payload and instead return a Hydra response. On the way back, any Hydra request is transformed properly into the Web API's accepted request formats. Many CBTs have different and heterogeneous Web APIs approaches and implementations, thus mapping of the CBTs Web API to achieve an intermediary layer through the CPH that emulates the Hydra Web API has proven to be a challenge due the many scenarios to support. Furthermore, scalability is not an issue with this semantic interoperability approach, as is proved in the following evaluation Chapter.

Finally, three interesting paths still need to be followed. First and foremost is how to use the CIS architecture and Hydra vocabularies with the IMS LTI v.2 to make the most of both and avoid duplicate solutions on certain aspects, such as authentication. Second, the *Template Layer* has a lot of potential to evolve to more sophisticated controls that enable the construction of elaborated management tools in the CLAO. Third, the CIS architecture opens the door to gather *Learning Analytics* from the e-Learning ecosystem of the CBTs, thus providing rich data to further analyze and enhance the learning process.

Part C: Evaluation

8 EVALUATION

This Chapter presents an overall evaluation of the flexible interoperability model along with the corresponding studies that elaborate the results of using several *Cloud-Based Tools (CBTs*). It begins to confirm (Section 8.1) that the *Cloud Interoperability Service (CIS)* and the *Cloud Learning Activities Orchestration (CLAO)* meet all the objectives specified in Chapter 4. Furthermore, the lessons learned from a technical implementation, the corresponding problems, issues, workarounds, and advantages that have been faced are presented. Moreover, a comparison between the semantic interoperability approach and the standard static Web service contract approach is elaborated. Additionally, the scalability of the CIS architecture is tested and the results are demonstrated.

In the second part of this Chapter a series of evaluation studies regarding CBTs, and the use of CLAO to enable a CEE are introduced. First, Section 8.2 presents results of the evaluation of emotions, motivations, and usability when using CBTs for *Learning Activities* in a MOOC are reported. The subsequent study (Section 8.3) evaluates the first-generation CIS architecture (see Section 5.2), thus enabling CEE in a PLE. First, the test beds are described with their corresponding *Learning Activities*. Then, the students' performance, perceived usefulness and ease of use, and emotional aspects are evaluated. Furthermore, this study addresses the use of ROLE's *contextualized attention metadata* with a *user interaction registry* that provides the first learning analytics results regarding the CIS usage.

Section 8.4 presents an evaluation that focuses on the experimentation to describe and identify the motivational factors and learning strategies of CLAO's impact on a MOOC by using the Motivated Strategies Learning Questionnaire (MSLQ) (Pintrich, Smith, Garcia, & McKeachie, 1991). It first measures student´s motivation: intrinsic and extrinsic goal orientations and the value given to a particular learning task. Furthermore, it elaborates on learners' expectations, such as beliefs, self-efficacy, performance, and anxiety. Second, it evaluates the learning strategies: cognitive and meta-cognitive (rehearsal, elaboration, organization, critical thinking and self-regulation), and resource management strategies such as time management, study environment, effort regulation, peer learning, and abilities to find help.

Finally, a short analytics evaluation (Section 8.5) of the learner's behavior analytics over the CLAO is presented. These experiences are assessed over three MOOCs and elaborated along with the use case presented in Chapter 4. Current learners' behavior results and limitations of the LA approach are discussed.

This Chapter proves that the CIS provides flexible interoperability for educational environments that can be applied to several learning settings, as the evaluation presented ranges from individual courses to the use of PLE to test the CLAO in several MOOCs. The Chapter is based on work published in Hernández et al. (2013a), Morales et al. (2014), Gütl et al. (2014), Hernández and Gütl (2015a).

8.1 Lessons Learned from Technical Implementation

The learner's experience can benefit by using the *Flexible CEE* by introducing innovative *Learning Activities* that use CBTs, opening new educational experiences that were not previously possible with a monolithic VLE approach (addressing challenge *to enable a CEE*, (Section 4.1.1)). This also means that the technical deployment of the learning environment (e.g. MOOCs) is easier, the inclusion of new CBTs is simpler, and the scalability needed to use and orchestrate those CBTs with hundreds of thousands of learners can be successful and manageable. The presented model (see Section 4.3) simplifies interoperability for education and allows access to the complete set of features offered by the API. Thus, the presented model permits the users to create tailored learning experiences and utilize VLE management controls that are not currently available, as described in the use case (Section 4.2). With this interoperability approach, it is possible to enable *Learning Orchestration* mechanisms (Hernández, Gütl, & Amado-Salvatierra, 2014c) that can enhance the learner experience through the use of CBTs (meeting the challenge of *enhanced orchestration through open APIs* (Section 4.1.2)).

Currently, to enable system interoperability between a VLE and a CBT's API, custom-made interfaces for each tool must be integrated—that is, programming interfaces for each API are required. This implies considerable effort and costs to build and update each interface. The presented solution enhances that approach, as it converts APIs into APIs powered by semantic technologies. The model describes a robust way for the TC to control and administer tool objects, defining its properties and possible operations. At the same time, it simplifies interoperability efforts, enabling a general abstraction for an ADT (e.g., a mind map) through the *Generic Vocabulary* and permitting the tool to be discoverable—for example, finding all operations available for a tool at run time without the customized code becoming involved. It is the APIs' discoverability that reduces the necessary effort and costs to deploy them. Instead of manually coding program contracts between the TC and TP, it is just a matter of defining them as described in this Chapter (overcoming challenge of *the static contracts* and enabling *the machine processable APIs*, (Sections 4.1.3 and 4.1.4)).

8.1.1 Objectives

The technical evaluation of the technology architecture implementation in light of the objectives set in the Section 4.2 is summarized in Table 8.1. Further details about the implementation itself and how each objective is covered are discussed in corresponding Chapter referred.

Objective	Short Description	Achieved Through
1	Interoperability for orchestration of CBT through automatically recognition and operation of Web APIs.	Is completely met through the *Generic Vocabulary*, the API Doc for each CBT used (see Chapter 6), and processing through the third generation CIS (Chapter 7).
2	CBT's Web API description is constructed at a higher level by semantically describe the objects, properties, operations and their relations, rather than writing standard contracts and requests.	The semantically defined Web API using JSON-LD and Hydra enable full compliance with this objective.
3	Creating application domain–type specifications serves as a generic description of Web APIs.	Through the *Generic Vocabulary*, introduced in Chapter 3, and then implemented in Chapter 6, permits the creation of the *application domain type* definition of a Web API, as in the CBTs selected for this research: mind maps and an online document editor.
4 and 5	(4) TPs may choose to support partially or totally such application domain–type specifications, and the TPs have the possibility to extend such specifications. (5) TCs administrators are able to define which objects, operations and properties are supported by the TP and can be used.	Accomplished using the *Generic Vocabulary* and the API Doc. The *Generic Vocabulary* does not determine the nature of the TP Web API. Instead, the approach is flexible enough to override everything that is defined in the *Generic Vocabulary*. At the same time, the TC can use the *Generic Vocabulary* as a means to determine automatically or manually what to support and (or) require of the TP.
6 and 7	(6) Teachers are able to use and orchestrate educational experiences based on cloud tools. (7) Teachers and learners are able to launch a given tool within their VLE.	Having the CIS is not enough to realize the educational experience. Therefore, it is necessary to enable a true CEE with precise and granular management controls to overcome the pedagogical issues described in Section 3.1. Thus, Chapter 5 introduces the CLAO, a CEE that uses the CIS to interoperate with CBTs.

Table 8.1 Objectives achieved through technical implementation

8.1.2 Implementation Experiences

CIS AND RELATED TECHNOLOGIES DEVELOPMENT EXPERIENCES

Some of the issues, gaps, problems and advantages encountered throughout the technological implementation, along with scalability tests are discussed in the following paragraphs. During the review of several CBTs, it was clear that not all have the same Web API maturity level (see Section 6.1). Therefore, not all the CBTs are ready for robust Web interoperability. For instance, MindMeister has a more comprehensive Web API in the mind map ADT, and after some conversations with the MindMeister developers, it became clear that this same public Web API was in use by their own mobile apps. Thus, it is clear why it is complete. In contrast, other tools have a very limited Web API that usually does not add value.

One non-planned change in the architecture was that, because the *Authentication Layer* is part of the CIS architecture but all communication with the real Web API occurs through the CPH, it was decided to move this layer to the CPH. This was a required change because the real authentication must happen at this stage, where the communication with the Web API is being done. Instead of creating a substitute bridge, it was simpler just to move the correspondent layer to the CPH, which implied a few adaptions to the layer. Once the CPH is no longer required (e.g., when the CBTs' Web API starts to support JSON-LDN and Hydra), then this layer can be returned to its original place.

Regarding the JSON-LD Processor introduced in Section 3.2.4, it was determined that it did not support the use of generic vocabularies. This means, for instance, that, if a given property was referenced in the API Doc in the *Generic Vocabulary (GV)* and was in the GV where the data type actually resided, it was not possible to determine the corresponding data type for that property because of the limitations of the JSON-LD Processor. The first approach was to do a type of denormalization, which means that, instead of using a GV and the API Doc, the API Doc contained all the necessary information about the Web API. In practical terms following the previous example, this implied not referencing the GV for a data type definition but instead defining it in the API Doc itself. With this simple change, the whole infrastructure worked and could be used in several of the experiments presented in this Chapter. Consequently, the JSON-LD Processor was modified to fully support the use of generic vocabularies and thus use the whole model defined in Section 4.3.1 and elaborated in Chapter 6.

The whole CIS architecture is very solid, and it fully encompasses the conceptual model for flexible interoperability in educational environments defined in Chapter 4. Moreover, this architecture could be extended for a generic interoperability use scenario for access and management of CBTs through the CIS. Indeed, it has great potential for enabling interoperability in general. This also demonstrates that the semantic JSON-LD and Hydra technologies are suitable for flexible yet formal and robust interoperability scenarios. Further work may include use the CIS architecture in other types of interoperability scenarios that have nothing to do with education, such as business cockpits, project management, etc.

A future identified enhancement is related to the combination of GV and a CIS templating system that provides several synergies. First, it could be easily configured to automatically support only whatever is defined within GV. This might help in developing a straightforward compliance method. Second, through a future interface builder for the CIS, the objects, their properties, and corresponding operations could be arranged in such a way that the desired interfaces could be constructed, along with the corresponding Web forms that represent the operations. In addition, the GV could be used as a visual identifier of the complaint features.

Creating the semantic proxy, the CPH, described in Section 4.3.3 and materializing it as the CPH proved to be a challenge because each CBT Web API has its own set of specific implementation details. Thus, supporting small or large differences requires a large set of specific support cases that are difficult to maintain in the long run, especially if the number of CBTs supported by the CPH grows. Therefore, it is highly recommended to start conversations with the CBTs' TP to explore JSON-LD and

Hydra support. One initial step has been taken with the MindMeister core team by presenting this technology to them at both the technical and managerial levels. It is believed that these types of technology adoptions are directly proportional to an economic drive at several levels. One is the perception of implementation difficulty for the technical developers who have constructed and maintained the Web API. To lower the barriers, the JSON-LD and Hydra communities, along with projects such as the one presented in this thesis, must use examples and tools that can demonstrate to the average developer what the benefits are, the ease of use and how smooth the implementation in their current Web APIs could be (especially if they already support JSON payloads), and how this semantic approach can be consumed by third parties in an easy and more straightforward way. This thesis work is one step further, possibly the first most extensive use of such technologies.

One important topic to mention is that an *Iterative Development Process* has been straightforward and enabled to learn about the different pieces and to continually improve the overall architecture. The first-generation CIS allowed to test basic interoperability with the CBTs' API. At the same time, it gave a real experience using a widget approach for presenting the CBTs in a unified environment. This widget approach later became one of the key designs in the whole architecture. Specifically, to make something to work in a widget, one needs, among other things, JavaScript libraries that can be embedded within it that contain the necessary business logic for the widget to work along with the CBT. This same approach was extrapolated to the final CIS architecture because it was anticipated that the CIS templating system could be embedded in many system types (widgets or not), so a simple yet robust embedding capability is necessary. In the second-generation CIS, the architectural foundation was developed, serving as the core to further extend the CIS to support JSON-LD and Hydra. At that point, the second-generation CIS and the newly created CLAO were tightly integrated, so separating them in the next iteration was a design requirement. Although the separation required some work, it proved to be straightforward, and probably the key factor to making the separation smoother was the widget approach followed in the CIS templating system because the CLAO only needed to consume the templates, and then each specific template had a configurator script that manipulated what to enable and disable in a given interface. Concerning *Learning Analytics*, each iteration and evaluation brought new insights that not only helped to improve the gathering of analytics but also brought some enhancements to the whole infrastructure to yield a system that was easier to use for both teachers and learners.

SYSTEM SCALABILITY

The subject *Use Case* (Section 4.2) is fully covered using the *CIS Architecture*. The CLAO and CIS has been tested for scalability by creating a set of three *Learning Activities* that represent that use case, with slight variations between them. In other words, it was simulated that the teacher assigned to the learners a map instance to work with. Each map instance contains a few initial ideas to be created automatically through the CIS by a definition given by the teacher. Once the learners completed the work with the map, they would submit it to the CLAO, and the map would be displayed in read-only mode for the learner to prevent further changes. It was conceived to create more than 10,000 individual resources (map instances), one instance for each learner, since the learner would do each *Learning Activity* independently. This means that, for the given MOOC test environment, it was performed that 10,000 learners would do the *Learning Activity*. At the time they were to begin the activity, the correspondent resource was to be automatically created and set up for the current context. This included the creation of a mind map and online document using a CBT and the correspondent permissions given to the learners. This proved to be highly efficient because of the CBT cloud infrastructure, and there was no noticeable delay while performing the process when accessing the *Learning Activity* resources. The performance results were within the range of 100 to 1,000 milliseconds for the following operations of map instance manipulations: creating map instances, adding ideas to maps, editing permissions rights over a map, creating online document instances, and assigning proper permissions. This performance range is true for the following scalability tests: first over sequential map instance manipulation, second over blocks of 100 simultaneous map instance manipulations, and third over blocks of 1,000 simultaneous map instance manipulations. The test results indicated a non-noticeable performance impact for system users with the different scalability tests. These tests enable real-life scalability scenarios, such as having 100, 1,000, or multiple sequential access to the map instances, thus simulating learners doing their *Learning Activities*.

The prototype evaluation indicated that the architecture scales well for any course type, including MOOCs. The CIS performs the discoverability of the Hydra Web API efficiently and effectively. The need for the intermediary CPH between the CIS and the CBT Web API is necessary because tool providers have not yet enabled their Web APIs to use Hydra. Some improvements would be useful, such as a better orchestration between CBTs within a *Learning Activity*. This includes communication between the CBTs themselves, which, though it was initially introduced and evaluated in the first-generation CIS using the ROLE PLE, has not been extended into the latest CIS architecture and thus is an open research area.

The learner's educational experience is benefited from using the CLAO and CIS by introducing innovative *Learning Activities* that use CBT, opening new educational experiences that were not possible before with a VLE monolithic approach, as presented in the following Sections. The MOOC technical deployment is easier, the inclusion of new CBTs is simpler, and the scalability to use and orchestrate those CBTs with hundreds of thousands of learners is proven to be successful and manageable.

AUTHENTICATION

As described in Section 7.1.3, the *Authentication Layer* covers the security through the use of standard, industry-accepted credential management and authentication protocols. It can be said that this is a well-addressed problem with the current Web technologies. However, user privacy issues have not been directly addressed in this research. Although the full framework is built atop a secure communication and authentication protocol, it is outside the scope of the presented work to address privacy. With the ability to read a great deal of information from the CBTs' Web API, a TC such as a VLE is able to gather information on user behavior as it can do in its own VLE. Therefore, the same user privacy issues faced in a traditional VLE is confronted in a CEE. In contrast, some CBTs may also have specific privacy contracts with users (learners and teachers) and with institutions that consume their Web APIs, thereby limiting that the TC can *read* about the user behavior, adding another layer of protection in privacy terms.

8.1.3 A Simplified Web Interoperability Process for Developers

One of the common questions of the scientific community concerning this research is, how moving the technical complexity from the standard Web contract integration to the proposed Semantic Web definition of CBTs, is actually really simplifying the overall process. It is a crucial question to increase the adoption the interoperability results of this research. There are two scenarios to support for software developers.

- In the *First Scenario*, from the TC point of view, such as enacting a CBT from a VLE to interoperate between both, a developer has to first code the corresponding requests and authentication mechanisms. This means that, to understand the CBT Web API, one must determine what the available methods are, identify how each one is connected to the others, create an application workflow (e.g., based on our use case to automatically create and assign a map to a learner) that makes sense for its ultimate purpose, create the respective user interfaces, and so on. This work is proportional to the size of the available CBT Web API, and its complexity may increase exponentially with the size as well. However, using the CIS, the TC developer simply needs to configure and use the CIS, and for each new CBT to be used in this framework requires only configuration of the interfaces (not creating them, but rather specifying what to use or not) and, if needed, creation of the necessary workflows to perform the desired behaviors. Therefore, the interoperability cost is reduced to zero, and it is just a matter of organizing the presentation layout and building the workflows, something that would be necessary even if the tool were the VLE's native module. Software development costs related to interoperability are eliminated.
- The *Second Scenario*, from the TP side, providing JSON-LD and Hydra-based payloads for their consumers, requires the following:
    a. Conceptualizing their objects, properties, and related operations (see Section 4.3.1). For in house developers of a CBT's Web API, this must be fairly easy because the developer most probably has a clear knowledge of the domain.
    b. Creating a semantic *graph* (not necessarily graphically, but logically) of the relations and workflow order of their Web API. For example, one can add an idea only if one already has a map for where to put it. This is the business logic behind their CBT Web API, something that the developer must also be quite familiar with. Graphs are presented in Section 6.4. It is a good approach to identify a Web API semantically, as it helps to clarify the interoperability process, in contrast to the common belief that the sematic Web is too complex.
    c. Writing the API Doc. This uses the JSON-LD and Hydra vocabulary, is based on the *graph,* and is created in the API Doc. This requires knowing the syntax, which is simple, and with some major example guides, it is actually easy to follow how to build a graph into JSON-LD and Hydra.

An API Doc playground is provided by Hydra's author, which can help to test whether the API Doc is syntactically correct or not.

d. Implementing the use of JSON-LD and Hydra payloads using the API Doc instead of the current payloads. If the CBT Web API already uses JSON (which is quite common), the path is straightforward. Otherwise, it requires the implementation of such a payload, which is as easy as using JSON instead of using XML payloads.
e. Ensuring that the CBT Web API complies correctly with a RESTful Web service. In summary, if the CBT Web API already has JSON payloads and is RESTful, then it is fairly straightforward for the TP developers to create a *graph* and then implement it as an API doc and using it to augment their payloads with this syntax. It is not about building a new Web API but instead just adding semantics to it, something that can be done in a fraction of the time required to build a Web API in the first place.
f. In the third scenario, if the CBT Web API still does not support JSON-LD and Hydra (none does at the time of this writing), then it is necessary to use the CPH, which probably is done by the TC developer. This has the following work:
g. Conceptualizing objects, resources and operations, as in the second scenario.
h. Writing the graph of the business logic, again as in the second scenario. In this case, the developer needs to learn about the CBT Web API, something that would need to be done anyway in any other interoperability approach.
i. Writing the corresponding API doc, as in the second scenario.
j. Creating a JSON structure that maps API Doc to the corresponding CBT Web API methods (with its parameters) and payloads. In special rare scenarios, the CPH may need to be extended.

In summary, TP developers adopting this technology need to implement their already available technologies using some extra description semantics, something that happens at a higher level of abstraction compared to coding software that deals with every implementation detail, thus presenting a straightforward path for adoption. For TC, it is all about automatic interoperability, and then managing a layout level for the interface and (or) building workflows for the desired behaviors using an automatically created JavaScript library. Whether the developers adopt the semantic approach and technologies is something that remains to be seen. The question of whether to adopt the CLAO or not, is a matter of preference for each institution or project owner, but the CIS actually works as a standalone component. Therefore, there is an open path to integrate it with any VLE or with other educational editors, such as an IMS-LD-based editor.

8.2 Evaluating Emotions, Motivation and Usability in CBTs

The following Section includes significant content from the publication by Hernández, Gütl, Chang and Morales (2013b).

This study is focused on a MOOC learning experience with *CBTs* for deployment of the *Learning Activities (LA)*, and a discussion of the findings based on usability, emotional and motivation aspects. The presented CBTs will be used for further integration into the CEE in next studies, thus, the present study demonstrates that the CBTs can be used to create LAs, and enable a CEE. The learning setting was designed based on the early MOOCs experiences of Galileo University (Hernández et al., 2013b) but was also influenced by a MOOC on *"Artificial Intelligence"* by Peter Norvig and Sebastian Thrun in late 2011, and MOOC sites such as Coursera and Udacity. The subject chosen focuses on *"e-Learning Introduction"* and the content was prepared in Spanish to cater for Spanish speaking participants from different countries. The research interest was to design a MOOC methodology and to evaluate the MOOC learning experience considering emotional, motivational and usability aspects while at the same time reviewing the use of *CBTs* for LA. The xMOOC approach based on the cognitive-behavioral teaching model was the focus of the MOOC.

8.2.1 Course Description and Setup

LEARNING OBJECTIVES AND LEARNING ACTIVITIES

The learning objectives of the MOOC can be summarized as to acquire knowledge of e-learning theory and technology as well as to apply the knowledge to design and create online courses. It was also important for students involved in the MOOC to have skills with online collaboration and peer assessment. The learning experience was based on self-guided individual learning and individual assignments combined with peer discussion and rubric-based peer assessment. Instructions were provided to guide peer assessment and the assigned tutors were given the responsibility and autonomy to manage their own groups.

Previous experiences and the literature revealed that students should be supported with the usage of learning tools, they should be provided with ample guidance, and be restricted from choosing their own tools. For these reasons, students who registered for this MOOC were constrained to a number of *CBTs*. Also, to prevent fragmented communication activities, group discussion and collaboration were confined to one communication channel. Two tutors were assigned to monitor participants within the forums, answering questions, clarifying concepts, moderating discussions, providing technical assistance and other duties. The educational concept, content structure, assessment activities and performance expectations were given to the students at the start of the MOOC experience.

The MOOC was designed with four learning topics. For each topic, short videos representing the main learning content were provided to the learners. Complementary readings of pre-selected documents and hyperlinks were provided to the students. Each topic had a set of LA and assignments supported by a selection of *CBTs*. Appropriate *CBTs* were selected based on the learning and instructional objectives. Students were supported with video-tutorials and written instructions. Table 8.2 summarizes the learning topics, learning and instructional objectives, activities and selected tools.

Learning Topic	**Instructional Objectives**	**Activity and CBTs**
"Talk about e-Learning" Overview of main concepts and methods of e-learning, identifying main stakeholders and their roles, advantages and issues.	• Content acquisition • Demonstrate an understanding of unit contents • Structure for knowledge representation	• Videos and documents access by .LRN[1] • Summarize in a Word Processor • Mind-map creation using Mindmeister[2]
"Technological platforms for e-Learning" A quick review of some of the common aspects of the LMS platforms, its strengths and limitations, and an introduction to the standards that are required for its development.	• Content acquisition • Create their own LMS • Analyze, Evaluate: organize, outline, structure the concepts of an LMS, the learning-teaching process, critically evaluate different types of LMS	• Videos and documents access by .LRN[1] • Basic configuration at LMS instance at Milaulas[6] • First create a mind-map using one mind mapping tool: Mindmeister[2], Cacoo[5], Bubble.us[7] • Second create a presentation and publish it using Slideshare[3]
"How to create a fascinating e-learning course" Description of the main elements of effective e-Learning design, best practices, methods, learner's context, design and processes to achieve learning outcomes.	• Content acquisition • Create, Analyze: outline, design and produce online learning units using the guidelines provided for high quality e-learning courses.	• Videos and documents access by .LRN[1] • Mind-map of student's first learning unit built using Cacoo[5], then create actual learning unit filling word processor templates. • LA designed and built with Educaplay[4]
"Developing an e-Learning course" Detailed guidance and structure for design and develop online instructional materials, use web tools, and a review production practices.	• Content acquisition • Create: Produce the online course based content templates, design and build a new introductory unit that includes a welcome video-message.	• Videos and documents access by .LRN[1] • Using the LMS instance at Milaulas[6]

[1] .LRN (.LRN, 2014)
[2] Mindmister (MindMeister, 2014)
[3] Slideshare (http://www.slideshare.net)
[4] Educaplay (Educaplay, 2014)
[5] Cacoo (Cacoo, 2014)
[6] Milaulas (http://www.milaulas.com)
[7] Bubble.us (https://bubble.us)

Table 8.2 MOOC learning topics, instructional objectives and selected CBTs, taken from (Hernández, Gütl, Chang, & Morales, 2013a)

Participants collaborated through the use of online forums. Active participation contributed to the overall assessment. To overcome lurking and to motivate active participation, a gamification approach was added where medals were awarded for student contributions and achievements. For instance a question marked as 'favorite' by at least 25 students that earn a 'stellar question' medal. The best answers voted as positive and relevant by at least 25 students that earn a 'great answer' medal. A 'contributor' medal is given to students with at least 10 contributions. Each medal is added up in the student profile and those awards are featured in the front page for community recognition.

For the grading of the LA, a peer-assessment approach was applied to every LA. In order to grade the peers, the students first had to submit their assignments for the corresponding LA, before they were able to randomly select a blind peer-assessment. Performance of peer assessment was also counted towards the overall performance of the course. Students were also given grade for the final project and the grade reflected the overall knowledge acquisition of the course learning objectives.

TECHNOLOGICAL ASPECTS

From the technical perspective, the xMOOC learning experience was designed to restrict the learning setting to a number of pre-selected tools and cloud services. This decision was made because of earlier experiences where learners had asked for seamless and integrated learning among their groups and that the use of different tools had impeded their learning. The main challenge was to manage the hundreds of learners and to keep pace with peak loads of requests with the tools.

The central access point for the MOOC was a *Learning Management System (LMS)* developed at and for Galileo University and is based on .LRN LMS (Hernández et al., 2007b). Several enhancements were developed for the MOOC. The templates were developed for the MOOC specific structures, interfaces and the same 'look and feel'. The MOOC has a main page and also has a counter showing the total number of subscribed learners. It also enables user to create accounts and log onto the system. For convenience, the participants can also register and login from Facebook. Each of the *CBTs* used for LA (see also Table 8.2) required their own credentials. No interoperability or look and feel adaptations were implemented for this study, and this remains an open field for future research.

Following previous experiences in online discussion and experiences from Hernández, Linares, Mikroyannidis and Schmitz (2013a), and a review of the Udacity portal, it was decided to integrate the online collaboration tool called the Open Source Questions and Answers System (OSQA, 2014). OSQA has proven capability for managing large group sizes and motivate collaboration within large groups. A seamless integration was realized by using the template mechanism and by developing a single sign on facility. This integration enabled students to go back and forth between the LMS and OSQA. Also a portlet was developed to inform students of recent and highly relevant contributions.

For the peer assessment activities, a new tool was created and integrated into the LMS. This assessment module included a rubric-based feature, where the instructors can create rubrics for the assessment activities. Learning products from peers were assigned randomly and anonymously for the peer-assessment activities. The MOOC facilitator can use the average results to moderate the activities or scale the grades. Students can also view the peer-assessment results. The same technologies presented are used MOOCs experimentation in Section 8.4 and 8.5.

Implementation information includes that this study uses the CBTs as standalone, not yet integrated within one of the presented environments of Chapter 5 or 7. The CBTs used are Mindmister (MindMeister, 2014), Milaulas (http://www.milaulas.com), Slideshare (http://www.slideshare.net), Educaplay (Educaplay, 2014), Bubble.us (https://bubble.us). The objective is to prove that with these CBTs is possible to create successful LAs. Furthermore, it proves the need to have a CEE that integrates and orchestrates such LAs that use CBTs, due the issues already identified in Section 3.1.

EXPERIMENTATION SETUP AND METHODOLOGY

The overall goal of the research was to gain insights on motivational, emotional aspects and usability issues as well as learning effectiveness and efficiency. In this paper, are highlighted some of the important and interesting findings.

The experimentation procedure included the following steps: (1) students enroll in MOOC, (2) students complete a pre-questionnaire to gather demographic details, (3) students undertake an orientation week in the first week of the course to familiarize themselves in the MOOC learning environment, (4) students access four weeks of LA, participate in online collaboration and complete assessments (5) students complete a post-questionnaire to evaluate their own performance and the overall MOOC experience. The instrument included the MOOC tools, content, the CBTs (see Section 6.1), the surveys, user behavior and user collaborative contributions over online forums, data entries from the peer assessment process, views and experiences from the instructors and professors, and interviews with the tutors and students.

With the research on emotional, motivational and usability aspects, the Computer Emotions Scale (CES) (Hyman, 2012), Intrinsic Motivations Measure (IMM) (Siemens, 2012) and the System Usability Scale (SUS) (Brooke, 1996) were used. For CES and IMM, a four point Likert scale was used and a five point Likert scale for SUS. The pre-questionnaire contained questions on demographics and the motivations to enroll in a MOOC. The post questionnaire contained the standard measurement instruments as listed above. Open-ended questions captured the learners' opinions about CBT and the overall MOOC experience and outcomes.

8.2.2 Evaluation and Lessons Learned

The MOOC learning experience was offered in October 2012. One thousand six hundred and eighty (1680) learners from 30 countries enrolled in the course. The majority of participants were located in Guatemala, see Table 8.3.

Only 143 participants or 8.50% of the enrolled users completed the course, and the high drop-out rate is in line with findings from other MOOC experiments. Interestingly, in the first week of the course only 21.60% of the learners completed the learning tasks, while 33.01% actively participated in the forums. The second week showed a decrease with only 13.80% of learners completed the tasks while 26.02% actively participated in the forums. The third week showed similar behavior with decreased participation of 10.24% learners completing the task while 18.05%

participated in the forums. It is not surprising that the people who continued to participate in the forums had created a strong online learning community by participating several times during the week.

The remainder of the findings reported in this paper is based on data collected from the 143 students who had completed and passed the course. In light of this, the students' perceptions of the usability, emotional and motivational aspects of the findings relate to those students who stayed on to complete the MOOC program. Given this, it is interesting to note that the response rate was 100%. The participation was almost equally distributed with 44% of female and 56% of male participants, and the average age of M=39 (σ=11). Sixty seven percent (67%) of the participants reported holding a degree qualification.

Registered participants	1680
Participants completed pre/post-questionnaire	690/143
Age	M=39 (σ=11)
Gender	Female: 739 (43.99%) Male: 941 (56.01)%
Country	Guatemala (76.60 %) Spain (5.11 %) U.S (3.63 %) Honduras (3.09 %) Mexico (2.20 %) and others (9.04 %).
Students Participation: a) Did not start the course b) With at least one login c) Delivered the first task d) That finished and pass degree of the course	 728 (43.33%) 952 (56.66%) 363 (21.60%) 143 (8.50%)
Final grades for pass degree students (over 100)	M=88.61 (σ=8.36)
Forum activities	773 questions/3511 answers
	273 people active in forums
Peer-assessment	5 learning activities for peer-assessment
Video resources	46 Videos

Table 8.3 Demographics data MOOC Learning experience taken from (Hernández, Gütl, Chang, & Morales, 2013a)

A set of questions using a 5-point Likert scale (from *totally disagree* to *totally agree*) was used to determine the overall perception of the MOOC experience using the cloud-based LA. Table 8.4 shows some of the results.

Focusing on Kay and Loverock's emotional aspects (Hyman, 2012), the 12 items of the Computer Emotion Scale describes four different emotions: happiness, sadness, anxiety, and anger as shown in Table 8.5. The findings revealed that MOOC participants perceived low anger and sadness as well as significantly higher happiness while performing LA using the CBT.

Identified perceptions	M	σ
I didn't have any problems with planning the LA.	4.06	1.15
It was difficult for me solving the LA.	2.41	1.34
I would have needed more information to solve the LA	2.59	1.30
It was fun doing the LA.	4.37	0.99
I liked the idea of doing these LA to represent knowledge acquisition.	4.67	0.74
The time I spent in the LA was appropriate for my learning progress.	4.01	1.12

Table 8.4 MOOC Learning Activities student's perception

Emotion	Explanation	Value
Happiness	When I used the tool, I felt satisfied/excited/curious?	2.27
Sadness	When I used the tool, I felt disheartened/dispirited?	0.52
Anxiety	When I used the tool, I felt anxious/insecure/helpless/nervous?	0.83
Anger	When I used the tool, I felt irritable/frustrated/angry?	0.53

Table 8.5 MOOC Computer Emotions Scale with 4-point Likert scale

Focusing on the motivational aspects, are applied the intrinsic motivation measures according to Tseng & Tsai (2010) to assess the learners' perception using CBT (refer also to Table 8.5) for the MOOC learning experience. Table 8.6 shows the motivational attitude with learning a new set of tools, utilizing the tools to finish the learning tasks and reflecting the knowledge gained from completing the LA. Some of the positive comments from participants included: *"I liked it because they are easy to use and free app"* and *"The tools used are very interactive and easy to learn. They are very friendly."*

Intrinsic Motivation	Completing LA using CBT	Learn to use new tools (which are cloud-based)	Reflect knowledge using the CBT
Absolutely Unmotivated	0.70%	0.00%	0.00%
Unmotivated	2.10%	0.00%	3.50%
Motivated	29.37%	18.18%	25.87%
Very Motivated	67.83%	81.82%	70.63%

Table 8.6 Intrinsic motivation regarding aspects of CBT

With respect to the usability aspects, System Usability Scale (SUS) shows a good results with M=77.46 (σ=16.28), but there is also a broad range of opinion from 30 to 100. On the negative side, participants emphasized that *"Not all tools are free, and many includes a lot of ads"* and on the positive side, there were comments such as *"I really liked that* [the CBTs] *are dynamic and allow to better attract the attention of the students of the course we implement", "I liked what make learning innovative, easy to use them and what the most interesting part of a great legacy of free applications."*

Finally, having a look at the learning outcome of the MOOCs the overall performance of students who had completed the course was very high. The average grade was M= 87.30 (σ=9.31), with a high quality assessment expressed by using the CBT.

8.3 Evaluation of Usability and Emotions using the CIS in a PLE

The following Section mainly republishes parts of the work published by Hernández, Linares, Mikroyannidis and Schmitz (2013a).

This evaluation was set up at the Institute Von Neumann (IVN) of the Galileo University, Guatemala. IVN is an online Higher Education Institute (HEI) that delivers educational programmes across Guatemala. These programmes are also available for other Spanish speaking countries around their hinterland. The CIS and PLE presented in Section 5.2 are used for this evaluation.

The research interest was to evaluate the learning experience of standard online courses (not massive courses) considering emotional and usability aspects when using *CBTs* for LA. Furthermore, LA are deployed in a *Personal Learning Environment (PLE)*, and the corresponding outcomes of using the PLE with CBTs are presented. The aim is to demonstrate the use of the CIS first-generation on this educational setting (See Section 5.2).

8.3.1 Course Description and Setup

Learners at IVN are mostly adult learners who also are in employment at the same time. The IVN courses are similar to any other University course, although the most significant difference is that IVN learners do most of their learning during the evening or at weekends. IVN offers fully online learning programmes, which generally do not contain any synchronous sessions. Learners are expected to spend around ten hours per week for studying the supplied materials in the courses. This also includes carrying out any *Learning Activities (LA)* as well as interacting online with other learners. All courses are organized in weekly units, based on a variety of online materials (e.g. multimedia, interactive animations, etc.), downloadable material in addition to the LA.

Two courses are presented in this study, the first one being *"Building Online Activities"*, and the second "*Introduction to Instructional Design"*. Both courses are part of the e-Learning certification programme of the university. This programme is particularly targeted to meet the needs of practitioners, i.e. university teachers, and instructors who want to create and deploy their experiences using e-learning delivery methods. The learners participating in this case study originated from four different countries: for the first course 15 participants came from Guatemala, six from El Salvador, and nine from Honduras, for the second course 35 came from Guatemala and two from Spain. All learners had previously used CBTs LA in other courses, thus they were quite familiar with online services and tools.

The teachers introduced the learners to new concepts, including PLEs, *Self Regulated Learning (SRL)* and ROLE, with the purpose of raising awareness about their benefits with a premise of potentially engendering mindset change amongst them. The learners were guided to engage in an interactive learning process that was presented as having benefits for long term knowledge acquisition. It was also relevant for their forthcoming assessment regarding the assigned LA (Friedrich et al., 2011). This helped to encourage them to use the ROLE system. Observation of the learners' usage of the PLE and collected feedback from both the teacher and the learners, through interviews and questionnaires, took place. It is important to note that the "learners" in the group were mostly active HEI teachers at their home universities rather than conventional undergraduate learners. The first course lasted for four weeks and the second for five weeks but only the first course had assignments that strictly required use of the PLE.

TECHNOLOGICAL ASPECTS

The course material is delivered to IVN learners using a customized version of the .LRN *Learning Management System (LMS)* (Hernández, Pardo, & Delgado, 2007). Student-to-student communication is also supported through dedicated online forums. Teachers and instructional designers create and upload all teaching and learning material into the LMS.

In this test-bed, a series of experiments were deployed with respect to ROLE Environment (see Section 3.3.1) and used specially developed widget bundles, which were designed to support the LA for two courses.

In order to facilitate the adoption and usage of the PLE, is decided to allow learners to use Google accounts to register to the PLE and to the MindMeister mind-mapping tool. The Google accounts were provided by the teacher, and were created only with the purpose to be used on this specific course. Additionally is used Facebook for further collaboration, and the authentication for Facebook was done with the learners' personal accounts.

Technical implementation details for this study include also that the CBTs used are integrated within the PLE as described in Section 5.2. This represents that is used the CIS first-generation, corresponding to the first iteration of the software development process for building the CIS. This CIS included a first approach for the *Analytics Layer* with its corresponding usage and results. The CBTs used are ObjectSpot, Binocs Media Search, MediaList, EtherdPad, MindMeister Mind Map, Facebook, Google Drive (Hernández et al., 2013a).

SCENARIO

The following scenario was designed to test the ROLE CBT LA that had been defined. The teacher assembled a widget bundle that was the basis for individual PLEs of the learners for the first course. The first row shows the search widget *"Binocs Media Search"* and also the *"ObjectSpot"* widget. The third widget is the media list. The second row had the MindMeister CBT and the *"EtherPad"* widget, and the third row contained the social network widget for discussion. It had been decided beforehand to use a social networking site for discussion, based on previous experiences (Hernández et al., 2011). No further ROLE collaboration features were used in this part of the case study.

During the first LA assigned to learners, the PLE and related concepts were introduced to the learners, with supportive material such as step by step instruction, video-tutorials and user manuals custom made for this experience. In the *First*

Course all LA required a research part first, therefore, the learners were asked to search using the previously mentioned search widgets, then collect relevant resources in the list widget. They were then asked to create a report using the EtherPad widget, select relevant terms and their relations and represent them in the mind-mapping widget. Finally, the learners published their mind-maps in the dedicated course LMS space and discussed their use of them using the social networking feature that had been provided.

The *Second Course* used a different widget bundle. The Google Drive widget was used for collaboration, some activities had to use mind-map, discussions were held via the Facebook widget or the built-in collaboration features of Google Drive. Both courses' activities have a summative evaluation for the course grading. A participant's grade is calculated from the evaluations of her course activities, participation and collaboration in the course and online discussion.

THE LEARNING ACTIVITIES

FIRST COURSE

Four LA were assigned to the learners (see Table 8.7). The first one (activity #1) was searching for web services that enable the creation of learning material or use of tools for LA. This task was followed by summarizing the characteristic and potential educational benefits and classifying them using an initial taxonomy given by the teacher in a shared mind-map. In addition all the *"learners"* (who were teachers themselves) were also given the opportunity to discuss their contributions and how they, as individuals, might apply the pedagogical approach in their own classrooms.

The second LA (activity #2) contributed to the overall research about how to measure course quality through online surveys with a target group of learners. In this case study, it was decided that each student would search, list, summarize and reflect knowledge by recording it in a mind-map. This included to create the online survey using Google forms, based on a design previously proposed for the actual course survey. In this instance the mind-map to be created would be individual, and could be shared with the rest of the learners. At this stage the learners were asked to discuss two or three of their published mind-maps using the social network widget.

The third and fourth LA (activity #3 and #4) were similar in process to the second. The objective of the third LA was to summarize a proven process for the creation of storytelling educational activities and then to present one set of learning materials based on that process by using one of the following online tools: goanimate.com, pixton.com, xtranormal.com. The fourth LA focused on modeling a process for creating visually attractive digital posters with educational themes using glogster.com. The learners had to present their work and discuss aspects of it with each other. The results of the activities were graded. The arithmetic means of the grades achieved are given in Subsection 8.3.2.

Second Course

Within the second course the following six learning activities (see Table 8.8) were carried out: in the first week (activity #1) learners were required to create a mind map within ROLE that represents the principles, models and theories behind the *Instructional Design* concepts presented during that week. Additionally, they had to choose a peer's mind map, analyze and comment on it over Facebook and thus start a discussion with the original author.

Learning Topic	Instructional Objectives	Activity and CBTs
Introduction to e-tivities Overview of main concepts of e-tivities, understanding them as "frameworks for enhancing active and participative online learning by individuals or groups" (Salmon, 2002,p.3)	Content acquisition Determine the necessary elements to create virtual activities. Encourage individual and group interaction through e-tivities Strengthen the learning process	Animations, documents, links , video tutorials and online content access by .LRN Summarize in a Word processor Mind-map creation using Mindmeister (Activity #1)
Building e-tivities- key principles Description of the key elements for the design and development of eLearning activities. A review of Bloom´s Taxonomy, collaborative and individual activities How to measure online quality through surveys Rubrics	Content acquisition Knowing the characteristics of individual and collaborative activities Build activities based on Bloom's taxonomy	Animations, documents, links , video tutorials and online content access by .LRN Summarize in a Word processor Mind-map creation using Mindmeister Creation of an online survey using Google Forms. (Activity # 2) Discussion of two or three published mind-maps using the social network widget.
Review of a number of tools in the cloud Mind Maps Glogster Wikis Facebook Xtranormal Pixton goanimate	Content acquisition Practice with some of the tools Understanding how the tools works Glimpse the possibilities of application	Creation of storytelling educational activities and present one set of learning material based on that process by using one of the studied tools (Activity #3 and #4) Discussion of two or three published activities using the social network widget.
Designing an eLearning activity Detailed guidance and structure for design and develop an activity, applying one of the cloud tool.	Use web 2.0 tools as a mean to design activities Designing an activity with the elements required in the template	

Table 8.7 Description for the four Learning Activities assigned to the learner

Learning Topic	Instructional Objectives	Activity and CBTs
Instructional Design principles Overview of main concepts, answering questions like what are the goals of instruction? What is an instructional strategy, and how will we evaluate the achievement of the students.	Content acquisition Explain the differences between traditional and online learning instruction Demonstrate an understanding of the unit content	Animations, documents and online content access by .LRN Summarize in a Word processor Mind-map creation using Mindmeister. (Activity #1) Analyze a peer´s mind map and comment on it over Facebook
Learning Theories in instructional design A quick review of the commonly known learning theories, which are the based for the eLearning modality. Behaviorism, cognitive and constructive.	Content acquisition Understanding learning theories and their influence in the eLearning design. Structure knowledge representation of the theories	
Instructional Design Models A quick review and compare of some of the common ID models and methods, advantages and disadvantages, paying special attention on the ADDIE Instructional Design Models	Content acquisition Understanding the differences between models Structure knowledge representation of the models.	
ADDIE Model The ADDIE Model is an approach used by instructional designers and content developers to create instructional course materials. It uses a systems approach with 4 stages: Analyze, design, development, implement, and evaluation.	Content Acquisition Simplify instructional Design by applying a systematic approach based on the ADDIE Model Understanding and applying the Addie model stages. Determining the needs of the learners and examining the learning context and environment Determining the outcomes of the learning program or course and formulating the learning objectives	Animations, documents and online content access by .LRN Summarize in a Word processor Mind-map creation using Mindmeister Creation of the analysis of the course filling templates via Google Docs. (Activity #2) Discussion between group members through Google Docs comments, chat tools and Facebook Definition of the course using a Google Docs template (Activity #3)
Designing an e-Learning course Detailed guidance and structure for design and develop an online course, applying the four stages of the ADDIE model.	Analysis of the Learner, the Environment, and the Course Content Exhibit knowledge, skills, and creativity related to e-learning instructional design in the practicum course	Animations, tutorials, documents, and online content access by .LRN Creation of a mind-map that reflects the structure of the course under development. (Activity #4) Creation of two lessons filling Google Docs templates. (Activity #5 and #6) Creation of learning activities with different cloud tools as educaplay. Discussion between group members through Google Docs comments, chat tools and Facebook

Table 8.8 Description for the six Learning Activities assigned to the learner

The second week had two LA. In the first LA (activity #2), the learners had to create an analysis of one of their own courses (general objectives, intended audience, timeframe, academic requisites, etc.). They had to use a template that was created and published via Google Drive for that purpose (additional templates were made available for the following activities). The work had to be done in groups of three persons. Interaction and communication between the group members and the teacher took place through Google Drive comments, via chat tools or through Facebook. The second LA (activity #3) was to give a general definition of the course (specific objectives, topics, activities, resources, indicators). Again, a template was provided in Google Drive.

The third week had three LA. The first LA (activity #4) was about creating a mind map that reflects the structure of the course under development – it had to contain the course units, all topics and objectives, LA and additional resources to be used. The second LA (activity #5) was to create an introductory unit for the course that contains the course's methodology, timeframe, objectives, topic list, and assessment structure – again based on a shared template. The third LA (activity #6) was to create the first learning unit of the course, including the introduction to the course topic, unit content and LA in detail, once more using a template. All LA included collaboration and communication between group members and the teacher, through the collaboration tools already mentioned. The results of the activities were graded. The arithmetic means of the grades achieved are given in Subsection 8.3.2.

8.3.2 Evaluation

Three different evaluations of the course environments are performed, namely *Perceived Usefulness and Ease of Use (PUEU)*, emotional aspects, and finally learning analytics through automatically logged usage data. First is presented the overall learners performance in both courses, in addition see Table 8.9 for courses' demographics.

LEARNERS PERFORMANCE

For each one of the LA a summative evaluation was created by the teacher, assessing the tasks and objectives of each activity, including the educational and meaningful use of the CBT for representing the desired learning outcomes. The courses have a summative value of 100 points, some of the activities and its values are presented in the context of the *Cloud Education Environment* within the PLE, the rest of the activities are performed outside of this environment, such as quizzes, or complementary activities or assignments using the traditional LMS. The teacher's evaluation of the student's LA is shown in Table 8.10. For the first course, activities grades were *Arithmetic Means (AM)* 79 and the final grade for this course was AM 79 (standard deviation of 32.26), which interestingly is the same value that is observed the previous year's edition of this course, where no ROLE-based PLE was used. That is, the application of the learning environment did not change the course outcomes as far as these are reflected in the course grades. For the second course the results are similar, the activities grades were AM 85 and previous course edition had equal value, and for final grade AM 89 (standard deviation of 24.3) while in previous edition was AM 79, in general for both courses learners performed well. Only those learners with a low activity (for personal reasons) also had low scores. This indicates that although the ROLE PLE and CBTs were new for most of the learners, they still completed the LA successfully.

	Course 1	**Course 2**
Registered participants	31	37
Age	Between 25 -50	Between 25 -50
Gender	Female = 13 Male = 18	Female =28 Male =9
Country	Guatemala, El Salvador, Honduras	Guatemala, Spain
Students Participation: Did not start the course With at least one login Delivered the first task That finished and pass degree students (over 100)	3 (9.6%) 1 (3.22%) 27 (87.096%) 26 (83.87%)	0 (0%) 3 (8.1%) 100 (100%) 34 (91%)
Final grades for pass degree students (over 100)	AM = 78.71 SD = 32.26	AM = 89 SD = 24.3
Forum activities	93 Questions / 251 answers 26 People active in forums	128 Questions / 487 answers 34 People active in forums
Peer assessment	2 learning activities for peer-assessment	2 learning activities for peer-assessment
Video resources	5	5

Table 8.9 Demographics data for courses 1 and 2, presented in section 8.3.1

Activity	**First Course**	**Second Course**
#1	7.09/10	4.29/5
#2	7.38/10	9.18/10
#3	8.38/10	8.43/10
#4	8.70/10	2.75/3
#5	N/A	6.43/7
#6	N/A	13.37/15

Table 8.10 Summary of scores obtained by learners, notation is current student's average out of the total possible value for the activity, course is evaluated from a total of 100 points taken from (Hernández et al., 2013a)

PERCEIVED USEFULNESS AND EASE OF USE

The participants of the evaluation of the ROLE-enabled PLE were asked to answer a short online survey. The purpose of this survey was to gather user feedback both specifically about the ROLE widgets and technological issues, as well as more generally about the perceived usefulness and ease of use of PLEs, via questions based on the *Technology Acceptance Model (TAM)* (Venkatesh, 2008). With the questionnaire both quantitative data – the participants were asked to state their disagreement and agreement to statements on the usefulness and ease of use of the system on a scale from 1 to 5 – and qualitative data – via text questions on the strengths and weaknesses of PLEs and ideas of 'perfect' PLEs – were collected. Since all of the participants were also teachers, the survey contained questions about the perceived usefulness and ease of use of PLEs both from the perspectives of the learner and the teacher. A total of 19 participants for the *First Course* and 36 for the *Second Course* responded to the survey.

The participants stated their disagreement and agreement to the following statements, among other statements, on a scale from 1 to 5, with ‘1’ meaning ‘strongly disagree’, ‘2’ ‘disagree’; ‘3’ ‘neutral’, ‘4’ ‘agree’ and ‘5’ ‘strongly agree’. That is, a value > 3 signalizes a tendency towards agreement with the respective statement, a value < 3 signalizes disagreement. The participants of the second course gave (only) slightly better evaluations compared to the participants of the first course. In the following Table 8.11, the evaluations from both courses have been pooled together, below their AM and *Standard Deviations (SD)*.

Technology Acceptance Model (TAM) Questions	Arithmetic Mean (AM)	Standard Deviation (SD)
I would find a PLE useful for my work.	3.53	1.11
I would expect a ROLE-based PLE to be useful for my learners.	3.08	1.19
I would accomplish my work more effectively with a PLE than with the learning technology I am currently using.	3.31	1.17
I would expect that my learners would accomplish their work more effectively with a ROLE-based PLE than with the learning technology they are currently using.	2.91	1.25
I think the system was easy to use.	3.18	1.24
I expect that it would be easy for my learners to use a ROLE-based PLE.	3.45	1.11
Using a PLE would improve my motivation for learning.	3.33	1.18
I expect that using a ROLE-based PLE would improve my learners' motivation for learning.	3.22	1.02
Using a PLE would enable me to learn in an independent manner.	3.45	1.16
I expect that using a ROLE-based PLE would enable my learners to learn in an independent manner.	3.44	1.07
I predict that I would frequently use a PLE if I had access to it./ I think I would use the PLE frequently.	3.35	1.27
I predict that my learners would frequently use a ROLE-based PLE if they had access to it.	3.31	1.09

Table 8.11 TAM results combined for both courses.

These results proof a positive tendency towards the usefulness and ease of use of ROLE PLEs, as well as their potential to support motivation and independence of learning. The text answers of the questionnaires give additional hints to interpret the data: the participants see the advantage of tailoring a learning environment from tools that are partly already known (Google Drive, social networks, others) to both learners and teachers. Still, using a PLE requires learning effort, that is, both teachers and learners must get used to the environment. This might be a lower burden for the learners than for the teachers: *“Learners are ready to begin using a PLE for their daily studies to a wide extend, because the new generation learns very quickly and is able to use that knowledge”.* However, although the idea of such an environment is to empower learners to learn independently – and the participants agree that such an environments supports independent learning as the data above show – one still has to be careful not to restrict the learners’ possibilities: *“I teach computer science classes and they [my learners] are capable of doing great things and they understand the technology. But if I would put them to use ROLE they would say I am limiting them.”*

The participants see the potential of integrating learning analytics into a ROLE PLE and thereby support both self-regulated learning and supervision by a teacher: *"[For my learners I would like to] have a system to visualize the progress within a week, in what areas a student has worked more or less." "In a collaborative environment I could measure the interactivity of the learners."*

Finally, the participants remind us that Internet access and connectivity are limited in some regions – quite a lot regions in the world, actually – which restricts the applicability of cloud-based environments in general and an environment like ROLE in particular.

EMOTIONAL ASPECTS EVALUATION

For both courses, are measured emotional aspects, in a pre-test and a post-test. The instrument was based on the *Computer Emotion Scale* (4pt. scale, in a range where 0 is 'none of the time' to 3 is 'all the time') developed by Kay and Loverock (2008) to measure emotions related to learning new computer software, in this case related to CBTs. With the pre-test, the emotions before using the PLEs have been measured, with the post-test the respective emotions after the course activities have been assessed. The pre-test gives an impression on the bias of the participants, before they have used the environment. The results are shown in the following tables 8.12 and 8.13, and they are summarized in table 8.14 and Figure 8.1.

Emotion	**Pre-test results**		**Post-test results**	
	AM	SD	AM	SD
Satisfied	2.79	0.42	2.26	0.81
Anxious	1.61	0.96	0.94	0.85
Irritable	0.26	0.45	0.47	0.84
Excited	2.63	0.60	1.95	0.91
Disheartened	0.26	0.45	0.42	0.61
Dispirited	0.53	0.61	0.58	0.51
Insecure	0.53	0.61	0.47	0.61
Frustrated	0.26	0.45	0.32	0.58
Curious	2.47	0.61	2.05	0.85
Nervous	0.42	0.61	0.37	0.60
Angry	0.16	0.37	0.32	0.48

Table 8.12 For the first course, results from pre and post tests, the arithmetic mean (AM) result per emotion measured and the standard deaviation (SD) taken from (Hernández et al., 2013a)

The summary with the four emotions of the CES (Kay & Loverock, 2008) scale is presented in Table 8.14, organized as *Happiness* (satisfied, excited, curious), *Sadness* (disheartened, dispirited), *Anxiety* (anxious, insecure, helpless, nervous) and *Anger* (irritable, frustrated, angry).

Emotion	Pre-test results		Post-test results	
	AM	SD	AM	SD
Satisfied	2.21	0.51	1.79	0.66
Anxious	1.39	0.77	1.30	0.69
Irritable	0.54	0.51	0.67	0.56
Excited	2.08	0.72	1.92	0.83
Disheartened	0.50	0.59	0.46	0.59
Dispirited	1.04	0.55	0.88	0.54
Insecure	0.92	0.65	0.79	0.59
Frustrated	0.29	0.46	0.46	0.59
Curious	2.33	0.70	2.33	0.64
Nervous	1.04	0.69	0.71	0.62
Angry	0.21	0.41	0.46	0.59

Table 8.13 For the second course, results from pre and post tests, the arithmetic mean (AM) result per emotion measured and the standard deviation (SD) taken from (Hernández et al., 2013a)

	First Course		Second Course	
Emotion(4pt. scale)	Pre-test results (AM)	Post-test results (AM)	Pre-test results (AM)	Post-test results (AM)
Happiness	2.208	2.014	2.632	2.088
Sadness	0.771	0.667	0.395	0.500
Anxiety	1.117	0.935	0.853	0.596
Anger	0.347	0.528	0.228	0.368

Table 8.14 For the both courses, main emotions summarized (arithmetic means) taken from (Hernández et al., 2013a)

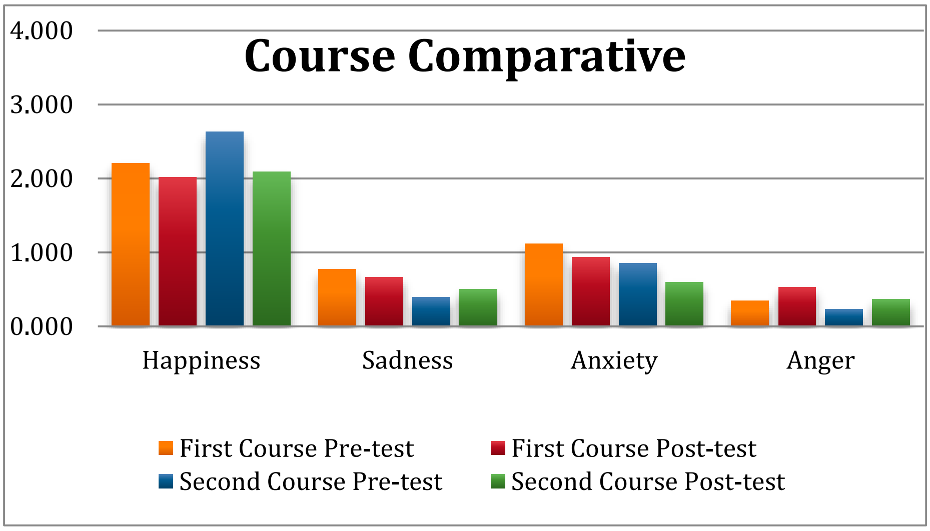


Figure 8.1 Bars-chart of the results for both courses taken from (Hernández et al., 2013a)

Figure 8.1 illustrate the emotional aspects of the participants for both courses regarding the usage of CBT that were required in the LA. The results show little difference between the pre-test and post-test. Results with a 4-point scale show a neutral reaction to “happiness” although it slightly decreases after having the experience. Regarding the perception of “anger”, the difference between the pre- and post-test is minimal. Is also observed low levels of "sadness" and a small decrease after having the experience. The perception of "anxiety" has also been low, although it was measured somewhat higher than "anger" and "sadness".

Negative emotions such as *sadness* and *anxiety* show a slight decrease, which is positive, while Happiness has a small decrease and *anger* has a mild increase as well. This corresponds to the impact that using this PLE may have into the learning process. No remarkable changes in the measured emotional aspects were recorded between the pre and post-test. It is important to mention that the CBT used in this study were not new for the learners, since they had already used them in previous courses but not within an environment such as the ROLE PLE. Therefore, is concluded that the recorded usability and emotional results reflect the new ROLE PLE experience as a whole, rather than only the usage of the individual CBT.

CAM USER INTERACTION REGISTRY AND CIS LEARNING ANALYTICS RESULTS EVALUATION

The results presented in this Section are only for the second course, since the CAM *User Interaction Registry* was not ready when the first course took place. As described, there were six activities, two of them involving the mind-mapping tool, four using mainly the Google Drive, and all with the possibility to use a discussion tool. The results reported in this Section mainly concern the usage of the MindMeister mind-mapping tool, that is, the course activities #1 and #4.

The first notable behavior identified is that none of the learners used the ROLE PLE to do their LA that were assigned using the Google Drive CBT. After interviewing both teacher and learners, it was concluded that the learners had substantial previous experience and knowledge of Google Drive, which led them to use the Documents in the environment provided by Google, instead of the ROLE PLE.

The CAM Registry tracked the sessions in the system by having continuous activity in the ROLE PLE in time no longer than one hour. In case the interval was more than one hour it was taken as another session (see Table 8.15).

	LA #1	**LA #4**
0 - 2 Sessions	40%	69%
3 - 6 Sessions	40%	17%
> 7 sessions	20%	14%
Widget Actions	up to 213 actions: 74%	up to 18 actions: 69%
Avg. Time in Mind-Map	27 mins.	22 mins.
Avg. Time in Facebook	3.5 mins.	N/A
Learners' total time in ROLE-PLE	66% between 0-22 mins.	71% between 0-16 mins
Mean of focus change between widgets	34	17

Table 8.15 Summary of data gathered by the CAM User Interaction Registry taken from (Hernández et al., 2013a)

For the activity #1, 40% of the learners had between 0 and 2 sessions, while other 40% of learners had between 3 and 6 sessions, while for activity #4, there was a decrease of the need for sessions, which corresponds with the widget actions indicator, that measures how many actions between the widget controls (but not the embeddable editor) were performed, indicating that in the beginning (activity #1) the learners needed a great deal of interaction to understand and feel comfortable with the ROLE-PLE and widgets. The mean usage of the mind-mapping tool also decreased per student from 27 minutes in activity #1 to 22 minutes for activity #4. Their interaction through Facebook for activity #1 was on average of 3 minutes only, confirming that for the activity the learners mostly limited to comment to other's mind-map but not to engage in deeper discussions. Total time in the ROLE PLE also decreased from 66% of the learners spent in a range between 0 and 22 minutes, to a 71% between 0 and 16 minutes for activity #4. These time contrast to mind-maps that are large in the number of nodes that were presented by the learners, as are detailed in the next Section. This indicates that a considerable amount of work was performed by the learners directly within the cloud-based tool and not by using the tool as widget inside the ROLE-PLE. Also mouse position movements from one widget to other widget or simply taking out the focus from the present widget were tracked, indicating that 77% of the learners had changed the point between the mind-map and Facebook widget up to 54 times.

For activity #1 the CIS Analytics results were obtained for the MindMeister mind map CBT, which the learners used in order to create a map with a mean of 30 concepts. The learners completed the task in a mean of 1.5 days. The analytics data show that only 33% of the learners fully completed the task on time, which indicates that this very first task took them more time possibly due to the time required for getting familiar with the ROLE PLE. This activity also required to review and comment on the map of other learners. The CIS Analytics shows that there is a mean of three comments per map and that each student has a mean of commenting in three maps. The recorded discussions were not particularly long or elaborated, mainly due to time restrictions for completing the activities of the course.

In activity #4 there is a great difference between the numbers of concepts added into the maps. The most active group contributed 273 concepts, three other groups in the middle created a mean of 108 concepts, and the rest of nine groups had a mean of 34 concepts. In general there are more elaborated mind maps in this group activity compared to activity #1, and some groups had created more detailed mind maps.

Throughout all these data analysis is clear that learners worked on these activities quite fast within the ROLE-PLE, although they also spend considerable time within the tool's website. Additionally, having more detailed requirements about the activity and the tool (such as the minimum number of ideas to put in a mind map) leads to similar workload for all learners. To take stock: the handling of a ROLE environment has to be learned which, however, goes rather quickly and does not take very much time. Moreover, the learners do not use the environment exclusively. Even if a tool is offered as a widget within the environment, they might switch to its original version. Possible explanations might be that they are better acquainted with the original version or that this version offers further useful functionalities.

LESSONS LEARNED

Observations of the prescribed activities and the use of the PLE with the CBTs indicated that the participants were somewhat overtaxed with this new learning scenario. The reason being that this was a totally new setting for the participants - they had not previously used such an environment. Additionally, the type and style of the LA were also new to the participants. Unfortunately no time was made available to them to become acquainted with the ROLE technologies before executing the LA either. Consequently this was reflected in the participants' negative responses in the survey. However, their emotional reactions to the tools used in the activities do not indicate that learners have negative emotions against those tools. In retrospect, it would have been better if the participants were introduced to PLEs and the CBTs ahead of the activities and be provided with sufficient documentation and guidance before attempting to complete the LA. Getting acquainted to a PLE takes some time that has to be budgeted.

In addition, it has been observed that once learners get used to a given CBT they prefer to use it on its own website, rather than as a widget within the PLE. According to interviews with learners, the main reason for this behavior is primarily due to the space restrictions posed by a widget. Another less prominent reason is the ability in some cases to have access to additional functionalities within the website, which are not available in a widget. The latter is a quite interesting and new result. ROLE gives learners the freedom to define their own learning environments. This freedom, however, can also be a burden, because defining one's own environment might be a hard task: you *must* reflect on your goals and the means you want to choose to reach these goals; you must find these means (data, contents, tools, partners) and get acquainted to them; you cannot just consume the service of a teacher. In previous ROLE evaluations in test beds both in Europe and China, learners were reluctant in spending much effort on designing their PLEs themselves. They asked for teacher support at least in the beginning of a course, that is, they wanted pre-defined PLEs (that they might possibly adjust to their own needs or preferences later). However, a student has to learn how to use a pre-defined PLE, which might include widgets that are probably new to him or her. The learners in previous evaluation did not complain about limited functionality – they rather asked for simplicity and a good justification why they should spend effort in learning how to use the environment. The Galileo test-bed shows that once a student got acquainted to the PLE and its widgets, she might feel that it does not only empower but also restrict her learning. She asks for additional functionality that she finds outside the environment. As an act of self-regulation, she breaks out of the learning environment that she started with and that was explicitly dedicated to self-regulated learning.

Finally, in one side the usability and emotional results of doing LA with CBTs within a PLE have positive outcomes. In the other side the whole presented experience proved that the underlying CIS first-generation technology supported the Web interoperability and orchestration of multiple CBTs as presented in Section 5.2.

8.4 Evaluation of Motivations and Cognitive Learning Strategies using the CLAO in MOOC

This Section mainly republishes work by Morales, Hernández, Barchino and Amelio (2014).

This study investigates the learners' motivational and cognitive learning strategies by using the *Motivated Strategies for Learning Questionnaire (MSLQ)* by Pintrich et al. (1993), which is a widely known instrument with reliable results. The MSLQ is used within a MOOC named "Cloud-Based Tools for Learning". The MOOC is given in the Telescope platform (Telescopio-UGAL, 2014). The aim is to implement *Learning Activities (LA)* that use CBTs within the CLAO environment, which in this specific course uses the CIS second-generation. Therefore, is demonstrated the effectivity of the CLAO and CIS second-generation, that both correspond to the second iteration of the software development process.

8.4.1 Course Description and Setup

Many MOOC formats exist (Fini, 2009), but most courses exhibit common defining characteristics that include massive participation, online and open access, lessons formatted as short videos combined with formative quizzes, automated assessment and (or) peer and self-assessment, and online forums. Is selected to use the xMOOC format, which promotes a teaching model emphasizing "cognitive-behavioral" learning, which follows a more traditional approach to online learning. The xMOOCs replicate the traditional model of an expert tutor and learners as knowledge consumers online, with saved video tutorials and graded assignments (Daradoumis et al., 2013). The main objective of this course is to present the opportunities provided by "the cloud" to create effective learning experiences and to innovate through tools that offer many possibilities to backup data, share information and create multimedia content.

COURSE STRUCTURE

The course was designed with 4 learning units; for each unit, an introduction described the objectives and activities, Google presentations displayed the content, and a podcast and short videos representing the main resources of the learning content were recorded for the learners. Complementary readings of pre-selected documents and hyperlinks were made available to the participants. Given that the course required the use of software or learning tools in the cloud, a set of tutorial videos and written instructions was created to support learners to complete their assignments. An overview of the main aspects of the MOOC is provided in Table 8.16.

Special focus was given to online collaboration through discussion forums and peer assessment. To overcome the problems of fragmented communication channels, the communication facility was restricted to only one tool to ensure a simple method of communication and had two types of main online discussion forums: 1) Forum of the Week: At the beginning of each week a forum was opened where the tutor started the week with a motivational message, provided the week's agenda, and presented a discussion topic. In this forum the learners were able to publish reflections and comment contributions following the thread started by the tutor. 2) Technical Forum: From the beginning of the course, this forum was open for questions and problems arising in the use of the platform.

MOOC "Cloud-Based Tools for Learning" Learning Experience	
Course offered	August 2013
MOOC pedagogical approach	xMOOC (cognitive behavioral teaching model)
Learning and instructional objectives	Acquire knowledge and skills of use to *Cloud-Based Tools*
Number of learning units	4 units (1 unit per week, 4 weeks in total)
Number of LA	8 activities (2 activities per week)
Video resources	12 Video tutorials
Collaboration type	Non-guided discussions. Question-and-answer (Q/A) forums.
Teachers	2 teachers and 2 tutors
Assessment type	Peer assessment and self-grading

Table 8.16. General Description of MOOC

In addition to the main forums, the application allowed participants to post additional questions that others could respond to and help answer, contribute new topics, or present ideas. Throughout the course, participants could propose topics for discussion, answer questions posed by teammates, vote, comment, and exchange views and information with the rest of the participants. The online collaborative forums followed a gamification (Hernández et al., 2013b) approach. Badges were used as electronic rewards for learners based on their contributions to the course learning community. This approach increases learners' positive emotions by the mere fact of their overcoming challenges (Daradoumis et al., 2013). For our case is used badges differently, to represent recognition within the community. Among the most awarded were "Teacher" for first response with at least one positive vote, "Collaborator" for the first positive vote, and "Student" for the first question with at least one positive feedback.

Participation in the forums had a value of 10% of the final grade of the course on the basis of accumulated points, known as *Karma*. Once the course was completed, each participant was rated for their participation in forums by measuring their karma, which was accumulated by responding, generating questions, voting, and being active in the forums.

PEER ASSESSMENT

Peer assessment consisted of each participant grading LA assignments. A rubric was created for each LA and learners used the rubric to assess their peers. Learners first had to complete their own assignment before randomly doing blind peer assessments. The participation and the level of quality contributions in the peer assessments were counted towards their course grade.

LEARNING OBJECTIVES AND LEARNING ACTIVITIES

Every learning unit had a set of instructional objectives and *Learning Activities (LA)*, and learners were expected to complete the set of assignments. The learning objectives of the MOOC can be summarized as to acquire knowledge and skills to use to CBT for learning, all summarized in Table 8.17.

TECHNOLOGICAL ASPECTS

The central access point for the MOOC was the Telescope project infrastructure (Telescopio-UGAL, 2014). It also enabled users to create accounts and log onto the system. For convenience, participants could also register and login from Facebook. The learning management system (LMS) is extended and enhanced at Galileo University and is based on .LRN LMS (Hernández, 2013a). New and customized course presentation templates were required for the proposed structure of the MOOCs. As stated at the beginning of this Section 8.4, it is used the CLAO (presented in Section 5.3), which uses the CIS second-generation. From the LMS the learner can use the CLAO (without further logins), and from the CLAO are enacted the LA using the corresponding CBTs.

Technical implementation details also include that is used the CLAO (see Section 5.3) to enact the LA using CBTs. The CLAO uses the CIS second-generation, corresponding to the second iteration of the software development process for building the CIS. The CBTs used are Google Drive, (Google, 2014), Cacoo (Cacoo, 2014), Mindmister (MindMeister, 2014), Milaulas (Milaulas, 2014), Slideshare (Slideshare, 2014), Educaplay (Educaplay, 2014), Cacoo (Cacoo, 2014).

For the online discussion forum, OSQA was used. This system is free and is a great solution to connect people to information and to get some elements to help engage more deeply with topics and questions of personal relevance, allowing everyone to collaborate, answer queries, and manage learning. This integration enabled learners to go back and forth between the LMS and OSQA. Also, a portlet was developed to inform learners of recent and highly relevant contributions. For the peer assessment activities, a new tool was created and integrated into our LMS. This assessment module included a rubric-based feature whereby instructors could create rubrics for the assessment activities. Learning assignments from peers were assigned randomly and anonymously for the peer assessment activities. The LMS calculated the average results to grade the LA or to scale the grades, and learners could view the peer assessment results; the only condition was that learners had to qualify at least two tasks.

STUDY SETUP AND METHODOLOGY

This study aims to identify the cognitive learning strategies and motivations that underpin the learning process within a MOOC, more specifically the MOOC "*Cloud-Based Tools* for Learning." Particular attention is given to motivational scales, which are closely related to enrolling in a MOOC. By obtaining an understanding of learners' motivations and learning strategies in a MOOC that heavily uses CBT for LA, is possible to enrich future courses and improve the overall student experience.

Learning Topic	Instructional Objectives	Activities and CBT	Assessment type
Unit # 1 Cloud-based Learning Concept, characteristics and opportunities of cloud-based learning	Identify the benefits of creating cloud-based learning experiences. Determine how the cloud can be used in learning environments. Collaborate in the recognition of cloud-based learning tools that can be used in learning environments.	Creating a PLE's diagram and the integration of a personal avatar Faceyourmanga[6]	Auto-grading
		Developing an essay about cloud-based learning in Google Drive[1]	Peer assessment
Unit # 2 Presentation and Documentation of Cloud-based Learning Tools Characteristics, use, and application of the tools	Create educative resources through presentation and documentation of cloud-based learning tools and apply them within learning environments appropriate to their educational needs.	Designing a Prezi[12] presentation	Peer assessment
		Development of a personal biography through a timeline and integration of a business card Dipity[8] and Cacoo[5]	Peer assessment
Unit # 3 Communication and Collaborative Cloud-based Learning tools Characteristics, use and application of the tools	Create educative resources through communication and collaborative cloud-based learning tools and apply them within learning environments appropriate to their educational needs.	Design an interactive wall that integrates multimedia resources such as images, articles, and a podcast. Padlet[10] and Soundcloud[11]	Peer assessment
		Multimedia presentation to show a project and multimedia resources such as mental map, images, and more. Google Viewer[1], Mindmeister[2]	Peer assessment
Unit # 4 Interactive and Multimedia Cloud-based Learning Tools Characteristics, use and application of the tools	Create educative resources through interactive and multimedia cloud-based learning tools and apply them within learning environments appropriate to their educational needs.	Create a learning game like a crossword puzzle or a quiz on all topics of the course. Educaplay[4]	Peer assessment
		Develop an animated online video to present a topic Goanimate[7]	Peer assessment

[1] Google Drive, http://drive.google.com
[2] Mindmeister http://www.mindmeister.com
[3] Slideshare http://www.slideshare.net
[4] Educaplay http://www.educaplay.com
[5] Cacoo https://cacoo.com
[6] Faceyourmanga http://www.faceyourmanga.com
[7] Go animate http://goanimate.com
[8] Dipity http://www.dipity.com
[9] Bubble.us https://bubble.us
[10] Padlet https://es.padlet.com
[11] Soundcloud https://soundcloud.com
[12] Prezi http://www.prezi.com

Table 8.17 learning objectives of the MOOC summarized by knowledge acquired and skills used by CBT for learning taken from Morales, Hernández, Barchino and Amelio (2014)

It was used the motivated strategies learning questionnaire (MSLQ) by Pintrich et al. (1991), which is a student self-report questionnaire that assesses the use of different cognitive learning strategies and motivational orientations in a specific course (Pintrich & García, 1993). The MSLQ consists of two Sections, the motivation Section and the learning strategies Section. The motivation Section has 6 subscales that assess learners' goals and value beliefs for the course, i.e., their beliefs about their skills to succeed in the course. The learning strategy Section has 5 subscales about learners' cognitive and metacognitive strategies. There are four subscales of resource management.

Questions use a 7-point Likert scale, from 1 (not true) to 7 (very true). Hence, from a cognitive social learning perspective it considers aspects that are determined by the context and are dynamic (Glance, Forsey, & Riley, 2012).

MSLQ was sent as an online survey to all MOOC participants, and it was optional and confidential. A sample of 230 learners answered. The survey was sent in the second week of the course and left open to answer for a week. Of those who answered the survey, 121 approved of the course. All data processing and statistical analyses were performed using SPSS statistical package software version 20.0.

8.4.2 Evaluation and Lessons Learned

This study is based on a survey of 230 learners who answered an intermediate questionnaire between the second and third week of course, all summarized in Table 8.18. A first questionnaire (before beginning the course) representing 60% of the learners enrolled revealed that for 76.71% of the learners it was their first MOOC, 54.52% indicated that they had enrolled in the course because it was related to their work. The 91.52% indicated they had never used the cloud tools that are introduced in the course, although they have used: Skype (75%), Google Drive (55.42%) and Dropbox (54.12%).

RELIABILITY

Cronbach's alpha coefficient was used to measure the internal consistency and reliability of the questionnaire. Once compared with the original publication of the MSLQ by Pintrich et al., (1991), is noted that this study has similar reliability to the original one. The coefficient is considered acceptable in $0.6 \leq \alpha < 0.7$ and good in $0.7 \leq \alpha < 0.9$.

MSLQ FOR THE MOOC

Motivation and Cognitive Learning Strategies are described in this study, using each one of the sub-scales, based in the proposed by MSLQ authors (Pintrich, Smith, Garcia, & McKeachie, 1991).

Also relevant aspects for each sub-scale are presented. Additionally, three intervals to locate groups were used for this study: low, medium and high ranks. As noted in Table 8.19 and 8.20, the learners scored high in the motivation scale, while in learning strategies, they scored the cognitive and metacognitive strategies highly, and in the resource management strategies they scored medium.

Registered participants	2045
Participants who completed intermediate questionnaire	230 (11.2%)
Age	M=38 (S=9.76) (Min=17 years/ Max=68)
Gender	Female: 35.50%
	Male: 64.50%
Country	Guatemala (57.82%)
	Peru (5.61%)
	Spain (4.78%)
	Mexico (4.78%)
	El Salvador (4.35%)
	All others (22.66%).
Learners who passed the course	121(59%)
Final grades for passing learners (over 100)	M=81.11
Forum activities	1068 questions / 3511 answers
	407 people with at least one participation
Academic level of the participants	Pre-university:16.45%
	Professional: 52.38%
	Master's degree: 29.00%
	Doctoral degrdegree:2.16%

Table 8.18 Demographic Data

Motivation scales		
Subscale	**Reliability original application** α	**Reliability this study application** α
Intrinsic goal orientation	0.74	0.73
Extrinsic goal orientation	0.62	0.74
Task value	0.90	0.87
Control beliefs	0.68	0.68
Self-efficacy for learning and performance	0.93	0.88
Test for anxiety	0.80	0.87
Learning strategies scale		
Rehearsal	0.69	0.82
Elaboration	0.76	0.86
Organization	0.64	0.77
Critical thinking	0.80	0.75
Meta-cognitive self-regulation	0.79	0.89
Time and study environment	0.76	0.73
Effort regulation	0.69	0.61
Peer learning	0.76	0.89
Help seeking	0.52	0.73

Table 8.19 Reliability of the MSLQ questionnaire, by subscales taken from (Morales, Hernández, Barchino, & Amelio, 2014)

Descriptive statistics of the motivational component					
Component	**Subscale**	**Mean**	**SD**	**Variance**	**Range**
Value component	Intrinsic goal orientation (IGO)	5.87	0.98	0.96	High
	Extrinsic goal orientation (EGO)	4.99	1.48	2.19	High
	Task value (TV)	6.34	0.8	0.7	High
Expectancy component	Control beliefs (CB)	5.63	1.1	1.2	High
	Self-efficacy for learning and performance (SELP)	5.89	1	1	High
	Test anxiety (TA)	3.41	1.6	2.7	Medium

Table 8.20 Descriptive statistics of the motivational component taken from (Morales, Hernández, Barchino, & Amelio, 2014)

MOTIVATION SECTION

The learners from the presented course had a very characteristic motivational profile (Table 8.19), the highest mean values were for task value (M=6.34, SD=0.80) and then, almost at the same value were self-efficacy for learning and performance (M=5.89, SD=1.00) and the intrinsic goal orientation components (M=5.87, SD=0.98). Thereby, it is possible to suggest that learners found the course material and contents interesting, useful, and important (task value). Learners showed a high confidence to accomplish and master the tasks and had their own intrinsic motivations (challenge, curiosity, mastery) and beliefs that their learning efforts would have a positive outcome, probably in the current profession and work.

Although grades and other goals seemed less important (extrinsic), meaning the learning task is not an end to itself, they were still important for learners. In the following Subsections the findings are presented in detail.

VALUE COMPONENT

For task value, 80% of the subscale got over 83% answers close to “very true.” In particular, 89% of the learners expressed liking the course (M=6.48, SD=1.00) and that they found it very useful to learn the course material (M=6.39, SD=0.87). For the reason of doing the tasks, it appears clear that intrinsic goals (M=5.87, SD=0.98) are slightly more relevant than extrinsic ones (M=4.99, SD=1.48). Learners indicated satisfaction in “understanding the course as thoroughly as possible” (M=6.23, SD=1.20). The learners indicated that when having th opportunity, they choose tasks where they can learn even if it does not guarantee good grades (M=5.82, SD=1.40). The “good grades” motivation was indicated with less satisfaction (M=4.65, SD=1.87), the lowest in the value component, although most of them graded with good capabilities to get better grades than their peers (M=5.50, SD=1.59). To demonstrate their newly learned abilities to friends and employers was of medium value (M=4.81, SD=1.99), although still highly relevant to 44% of the learners.

Expectancy Component

The self-efficacy for learning and performance (M=5.89, SD=1.00) and the control of learning beliefs (M=5.63, SD=1.10) subscales both scored highly, meaning that learners' beliefs are that their efforts in the MOOC bring them positive outcomes, that they study more strategically and effectively, and that this leads them to success and mastery in the course. Learners found themselves certain they could understand the basic concepts (M=6.39, SD=0.99) and that they could master the skills taught (M=6.10, SD=1.13). The control of learning beliefs subscale shows a great variation in responses. For example, learners believe that if they study appropriately they learn (M=6.29, SD=1.01) but interestingly show a lower agreement with the following two statements that got the same mean value, first the one that says, "If I don't understand the course material, it is because I didn't try hard enough" (M=5.05, SD=1.87), and second, "It is my own fault if I don't learn the material in this course" (M=5.05, SD=1.86). It is relevant to mention that 47% of the learners would not get an excellent grade in the class.

Affective Component

This is the lowest mean value for the whole questionnaire (M=3.41, SD=1.60), which indicates mid-level anxiety, with its components of concern, preoccupation, and worry that negatively affect academic performance. They were concerned about taking tests and the consequences of failing them as the highest concern (M=4.01, SD=2.13). In contrast, they show confidence while taking a test not thinking negatively compared with other learners (M=2.93, SD=1.98).

Is also assessed if there was any correlation between the subscales of the instrument. There were significant positive correlations between the motivation subscales. The task value correlates with the intrinsic goal orientation subscale (r (230)=0.667, p=0.01) and with the self-efficacy of learning and performance subscale (r (230)=0.770, p=0.01).

LEARNING STRATEGIES SECTION

To represent the learners' learning strategies, is clear that an important cognitive mean value at elaboration, building internal connections to prior knowledge with what has been learned (M=5.22, SD=1.30), organization (M=5.12, SD=1.30), and metacognitive self-regulation (planning, monitoring, and regulating) (M=5.03, SD=1.10) were the highest-degree motives. Resource management and effort regulation had significant values (M=5.02, SD=1.10), and interestingly enough is found lower perceived relevance of peer learning (M=3.38, SD=1.90) and help seeking (M=3.45, SD=1.40), see Table 8.21.

Descriptive statistics of cognitive strategies					
Component	**Subscale**	**Mean**	**SD**	**Variance**	**Range**
Cognitive and meta-cognitive strategies component	Rehearsal (R)	4.4	1.5	2.4	Medium
	Elaboration (E)	5.22	1.3	1.6	High
	Organization (OR)	5.12	1.3	1.7	High
	Critical thinking (CT)	4.86	1.2	1.4	High
	Meta-cognitive self-regulation (MSR)	5.03	1.1	1.2	High
Resource management strategies	Time and study environment (TSE)	4.53	0.95	0.9	High
	Effort regulation (ER)	5.02	1.1	1.2	High
	Peer learning (PL)	3.38	1.9	3.7	Medium
	Help seeking (HS)	3.45	1.4	2	Medium

Table 8.21 Descriptive statistics of cognitive strategies taken from (Morales, Hernández, Barchino, & Amelio, 2014)

Cognitive and Metacognitive Strategies Component

Elaboration strategies scored consistently high, such as relating ideas to what they already know (M=5.53, SD=1.49) and to other courses (M=5.47, SD=1.56), making connections between the concepts and online course material (M=5.39, SD=1.61), and incorporating ideas from course readings into discussion (M=5.40, SD=1.64). For metacognitive self-regulation, when learners are confused while reading they go back and try to Figure it out (M=5.85, SD=1.33), in contrast 39% of the learners reported they challenge themselves with questions while reading, and only 38% solve their own questions once confused. It is relevant to mention that 70% of the learners review the content before they actually study thoroughly. In critical thinking is found a low level of questioning before being convinced (M=3.63, SD=2.02), but in contrast learners use the course as a base to further develop their own ideas (M=5.45, SD=1.51).

Rehearsal learning strategies scored mid-range (M=4.40, SD=1.50). Organization strategies got high scores, such as for finding the most important ideas, which 70% of the learners reported as true or very true (M=5.86, SD=1.27). Furthermore, is seen a different opinion when asked about outlining important concepts, only 39% agreed with that.

Resource Management Component

The time and study management subscale scored mid-range (M=4.53, SD=0.95) while the effort regulation subscale scores higher (M=5.02, SD=1.10). Learners responded close to “very true” to the following: 47% found themselves in other activities rather investing time in the course, and 41% found difficulty having a study schedule, as expected. 50% reported set aside a regular place for studying, and 44% said they make good use of their study time. The group presented good effort even if feeling lazy or bored (M=5.18, SD=1.98) and worked hard even if they didn’t like the LA (M=5.13, SD=1.69). In help-seeking most other learners did not want any help when facing problems (M=3.82,SD=2.12). Only 21% of the learners were close to "very true" in their response to trying to work with classmates to complete course assignments, and a low 22% responded that way to setting aside time for study group.

STUDY FACTOR ANALYSIS AND CLUSTER

Factorial analysis was used to reduce the number of measures to a number of factors in order to try to provide a clearer interpretation of data. Therefore is conducted the analysis with all variables, making no a priori assumptions about the associations among the variables. The KMO (Kaiser-Myer-Olkin) measure of sampling adequacy returned .912 and Bartlett's test of sphericity returned 13,477,138 ($p < .001$). Therefore, the data were adequate for performing the analysis. Then is extracted 13 factors that accounted for 68.5 of the overall variance. The commonality is greater than 0.5. The data are between 0.568 and 0.818.

The first factor, accounting for 31.71% of the variance, included 18 variables. These variables have as main features management of the course documentation, practical application of the course, and positive attitude and constancy in work. The second component represented 11.27% of the overall variance and included 18 variables. These variables are characterized by relation of ideas to documentation and self-assessment.

The third component accounted for a 5.074% share of the variance, and it included 8 variables. These variables are characterized by difficulty with and negative attitude toward studying. The fourth component accounted for a 3.91% of the variance and included 6 variables. These variables have as main features study based on word memorization, teamwork, and consultation questions for teachers and other learners.

The fifth component accounted for 2.49% of the variance and included 8 variables. These variables are characterized by extension of the course document, regular time, and fixed place of study. The sixth component accounted for 2.16% of the variance and included 4 variables. These variables have as main features individual study and difficulty with time.

The seventh component accounted for 2.12% of the variance and included 3 variables. These variables are characterized by a main goal of highlighting and getting good grades. The eighth component accounted for a 1.85% of the variance and included 3 variables. These variables have as main features discussing the material, learning by memorizing course documentation, and use of graphic material.

The ninth component accounted for a 1.70% of the variance and included 2 variables. These variables are characterized by preparation before class. The tenth component accounted for 1.65% of the variance and included 2 variables. These variables have as main features approaches to training and improving concentration.

The eleventh component accounted for 1.60% of the variance and included 1 variable. This variable is characterized by challenging material. The twelfth component accounted for 1.52% of the variance and included 2 variables. These variables have as main features guilt compression and learning. The thirteenth component accounted for 1.42% of the variance and included 1 variable, the final score, and the clustering coefficient. This variable is characterized by compression of the material. The component plot (Figure 8.2) provided a visual representation of the variables outlining the three factors.

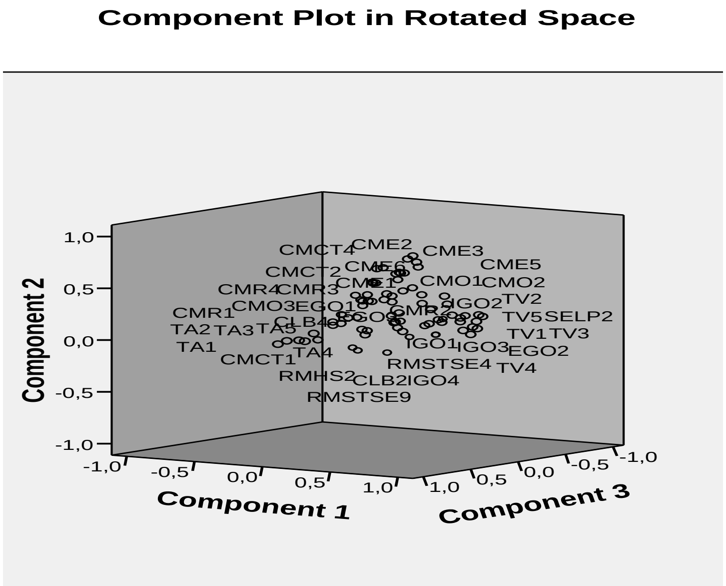


Figure 8.2 Component plot in rotated space of the factorial analysis taken from (Morales, Hernández, Barchino, & Amelio, 2014)

Our study has 3 clusters. The first clusters has 32, the second has 87, and the third has 111. The first cluster consists of components 5, 6, 8, 10, and 13, the second clusters consists of components 2 and 9, and finally the third cluster consists of 1, 3, 4, 7, 11, and12.

DISCUSSION AND LESSONS LEARNED

This study of the MSLQ for the MOOC titled “Cloud-Based Tools for Learning” sought to describe and identify the motivational factors and learning strategies that the learners followed. There is a group of learners (76.71%) for whom this was their first online learning experience, and the lack of an adaptation phase may have negatively affected them (Hill P. , 2013). Based on the results provided in this study, the learners had higher scores in the motivational scales compared to the learning strategies scales presented. As expected in the MOOC, learners begin with high motivation levels, but appropriate levels of commitment shown in the learning strategies prove them successful within a course in terms of grading. This study does not address the other types of learners as seen in the MOOCs overview Section 2.4.

It was found that learners enroll in a MOOC course for intrinsic motivations and they stay in the course because of the value of the learning tasks and because they feel they can become proficient in them. Learners' learning beliefs are that they are capable of learning, but they are less comfortable with recognizing their own responsibility in learning the material. This can be related to cultural factors (Hill, 2013), but it is not yet determined why this is the current outlook within the student population.

Only 53% of the learners chose answers close to "very true" about whether they believed they would get an excellent grade. This might be related to the dropout rates observed in MOOCs and soft commitment to course completion or that at the time of the questionnaire learners were in the middle of the course. There were 230 responses and only 121 completed the course successfully.

High correlation is seen between the task value learners give to the learning assignments, the intrinsic motivations for taking the course, and the efficacy and good course performance they expect of themselves.

Concerns regarding exams highly affect a group of the learners (31% of them). Although it is outside of the scope of the current study, it would be very interesting to link how this anxiety subscale relates to actual MOOC performance and completion for each individual.

The learning strategies that were scored highly indicated that metacognitive self-regulation learning strategies have very important roles, such as planning the activities to learn, monitoring one's own learning during the process, and continually tuning and adjusting those activities. Furthermore, metacognitive strategies present an interesting perspective with many contrasts. Our findings highlight that learners agree with making sure to understand what they are reading, but only some of them (below 45%) actually confirmed specific strategies such as questioning themselves, answering unsolved questions, or reviewing the MOOC content again.

Moreover, elaboration strategies, which consist of building connections between learned topics, got the highest score. It is relevant to mention that learners also experienced elaboration during discussion, but it was observed that only 19.90% participated in forum discussion. The value of participating in discussion was understood, but this did not actually occur.

The findings observed in the study demonstrate that a great majority of the learners prefer to go through a *scanning* phase with the content before studying it thoroughly. Critical thinking got an above-mid-range value. This may also be related to cultural behaviors (Hill, 2013), although learners tended to agree with using the course as a starting point in their learning, which is consistent with the defined course scope, which is to serve as an introduction to the use of CBT for learning. It was well-known that learners would face problems using new tools (Hill, 2013)—about 91.52% reported never having used CBT before—and considering the degree to which critical thinking is related to the problem-solving process, it may have been that learners' correct and extensive use of CBT was rather limited, therefore further analysis is required but is out of the scope of the present publication. Because rehearsal strategies ended up having less relevance, is possible to relate this to the nature of the course, which was practical rather than theoretical.

It is important to point out that the resource management strategies component offers us very interesting information about learners' views of the learning process within a MOOC, where they give moderate importance to peer learning and help-seeking, both closely related. Within the MOOC environment both activities occur in the online discussion forums. It is presumable that the large amount of learners, which is well known by the learners, and the perceived difficulty to organize communications correctly may inhibit learners' finding peers with whom they can share experiences and ask questions. Furthermore, the same behavior is observed in the student-to-tutor interaction. The learners confirmed that about 80% do not try to study with peers, and is important to mention that no peer interaction has been incentivized besides participation in online discussion forums and through peer assessment, which is actually done isolated and blind.

The time and study environment component points out a key issue in online learning, and as is reported here about half of learners struggle with time commitment, making good use of that time, and having a good place to learn. All of this hinders course performance and motivation. Effort regulation is the learners' ability to control their effort and attention; it is goal commitment. The learners responded with a relatively high value (see Table 8.22), which is especially significant considering that the course is free, massive, and online. Despite the limitations and challenges faced by this study, it has relevant results to further improve MOOC experiences and to make special considerations in developing new courses, with the presented results, is possible to focus on motivations and learning strategies that affect learners taking part in a MOOC. Although other important areas have to be considered, such as usability of the MOOC and used tools.

Regarding the technology used, the CBTs used within the MOOC allowed learners to perform the required LA, which were orchestrated using the CLAO (presented in Section 5.3). The overall experience of using CLAO was successful for learners, and in the next Section other examples will be introduced. The CIS second-generation enabled Web interoperability between the CLAO and the CBTs.

8.5 Behavior Analytics when Using CLAO

The following Section is partially republishes the work by Hernández, Gütl and Amado-Salvatierra (2014c).

In this study is investigated the use of CLAO from a learning analytics point of view, more specifically, the learners' behavior when doing the *Learning Activities (LA)*. Three MOOCs are presented using the CLAO (introduced in Section 5.3), with the corresponding results of orchestrating LA that use CBTs. The presented LA are in line with the use case presented in Section 4.2. Subsequently it was analyzed the learners' usage of the proposed CBTs, and by some measures have been identified how elaborated and complex is the work the learner has done with the CBTs. Finally, this study proves the CIS third-generation and CPH (see Chapter 7) enables the flexible CEE, which uses Semantic interoperability technology. The CIS and CPH correspond to the third iteration of the software development process established for this research.

8.5.1 Course Description and Setup

The Telescope's MOOCs' 2013 experiences presented in this study cover the following courses: (a) Cloud-based (Web 2.0) Tools for Education; (b) Introduction to E-Learning, and (c) Medical Urgencies.

The three courses take an xMOOC pedagogical approach (cognitive-behavioral teaching model): the MOOC courses have been prepared with a peer assessment type of evaluation and include a collaboration approach of question and answer (Q/A) forums with a support of four tutors. The courses have been designed with four learning units (one week per learning unit). Each unit includes an introduction describing the main objectives and LA, presentations displaying the content, and short videos representing the main resources of the learning content. Each of the learning units has between two to four activities that are part of an evaluation rubric divided between peer assessment and tutor review of the work. Table 8.22, summarizes the three MOOCs' experiences, including the final product that learners are expected to achieve.

	(a) Cloud-Based Tools for Education	**(b) Introduction to e-Learning**	**(c) Medical Urgencies**
Learning and instructional objectives	Understand how to include innovative LA using *Web 2.0 Tools* and apply this knowledge in an online course	Understand the e-learning fundamentals, the related concepts and tools; apply knowledge by designing an online course	Understand basic skills in first aid techniques and apply knowledge to design a basic handbook for kids
Final product	Define LA using CBT for online courses	Create an Online Course	Production of a short tutorial for kids
Course offered	September 2013	October 2013	November 2013
MOOC pedagogical	XMOOC (cognitive-behavioral teaching model		
Collaboration type	Non-guided discussions. Question and Answers (Q/A) forums		
Assessment type	4 units (1 unit per week, 4 weeks in total)		
Social Support	Facebook	Facebook	-
Tutors	2	2	1
Video resources	15	43	29
Number of activities	8	7	9

Table 8.22Instructional activities and selected CBT taken from (Hernández, Gütl, & Amado-Salvatierra, 2014c)

LA presented in MOOC courses are detailed in Table 8.23. This table presents the instructional objective, the activity, and the CBT (Google Drive document editor GD, MindMeister mind maps editor: MM) used within the proposed CLAO architecture to complete the activity.

Learning Topic	**Instructional Objective**	**Learning Activity Title (Google Drive (GD) or MindMeister (MM)) (Total Student Assignments)**
Cloud Based Tools for Education	Demonstrate an understanding of unit contents	(a.1) Benefits of learning in the cloud (GD) (219)
Cloud Based Tools for Education	Acquire content	(a.2) Presenting a project with three innovative tools (MM) (111)
Introduction to e-Learning	Demonstrate an understanding of unit contents	(b.1) Preparing a presentation on basic LMS functionality (GD) (161)
Introduction to e-Learning	Learn-by-doing activity: create, produce the course based on content templates; design and build an introductory welcome video.	(b.2) Producing my first virtual course (GD) (72)
Medical Urgencies	Structure for knowledge representation	(c.1) Creating my first mental map about medical emergencies (GD, MM) (83,71)
Medical Urgencies	Structure for knowledge representation	(c.2) Accident prevention measures (GD,MM) (43,47)
Medical Urgencies	Learn-by-doing activity	(c.3) Handbook for kids: "What to do in an emergency"(GD,MM) (56,58)

Table 8.23 Description of MOOC experiences taken from (Hernández, Gütl, & Amado-Salvatierra, 2014c)

TECHNOLOGICAL ASPECTS

From the technical implementation point of view is used the CLAO (see Section 5.3) to perform the LA using CBTs. The CLAO in the MOOC titled “Cloud-Based Tools for Education” uses the CIS second-generation, corresponding to the second iteration of the software development process for building the CIS. And for the other two MOOCs titled “Introduction to e-Learning” and “Medical Urgencies”, is used the CIS third-generation (presented in Chapter 7), corresponding to the third iteration of the software development process for building the CIS, which includes the Semantic Web technologies. The CBTs presented for this study are Google Drive, (Google, 2014) and MindMeister (MindMeister, 2014). Using the CIS third-generation requires to use the CPH presented in Section 7.2.

In addition, is used the same LMS, forums and peer-assessment technology presented in the technology part of the Subsection 8.2.1 and 8.4.1.

8.5.2 Evaluation and Lessons Learned

These courses had more than 6,000 enrolled learners and drew learners from more than 15 countries, including Spain, Mexico, Guatemala, Colombia, and Peru. Participants were equally distributed in gender, with an age average of 30 years (SD = 10.63) for courses a, b and c. Participants who visited the contents of the course at least once represented 60% of the total enrolled users. On average across the three courses, less than 10% of those active participants completed and approved the course. Interestingly, more than 70% of all the participants indicated that this was their first MOOC experience (see Table 8.24 for complete demographics).

	(a) Cloud-Based Tools for Education		**(b) Introduction to e-Learning**		**(c) Medical Urgencies**	
Registered participants	2037		2128		2114	
Female	36%		40%		54%	
Male	64%		60%		46%	
Age	M= 38, (σ=14)		38, (σ=13)		31, (σ=14)	
Did not start the course	32.93%		37.82%		30.86%	
With at least one login	67.07%		62.18%		68.81%	
Delivered the first assignment	23.77%		8.55%		33.88%	
Finished and successfully pass course degree	5.94%		4.18%		2.37%	
Country	Guatemala	60.53%	Guatemala	27.12%	Guatemala	24.69%
	Spain	4.03%	Spain	19.25%	Spain	42.72%
	Mexico	3.04%	Mexico	9.09%	Mexico	5.20%
	Colombia	3.58%	Colombia	8.76%	Colombia	5.30%
	Peru	4.42%	Peru	7.03%	Peru	4.35%
Levels of educational	Pre-	15.37%	Pre-	11.31%	Pre-	21.19%
	Student	33.92%	Student	27.38%	Student	47.02%
	Bachelor's Degree	50.52%	Bachelor's degree	61.17%	Bachelor's Degree	31.60%

Table 8.24 Demographics data for the three MOOC courses

The overall goal of the experience was to gain insights in how learners used the CBT enabled in CLAO for the 3 MOOCs, identifying usage and failure patterns in the *Learning Activity* and how effectively these CBTs were used. This identification leads us to further improvement of the CLAO architecture. In LMS, the modified version of the .LRN LMS (Hernández et al., 2007) was used and integrated using the LEC. Learning analytics is moving forward outside closed learning management systems (Pardo & Kloos, 2001), towards such environments as CLAO. The CIS analytics layer (described in Section 7.1) was used to explore data about the user experience, based on previous conclusions in (Chatti, Dyckhoff, Schroeder, & Thüs, 2012). In terms of learning analytics, the authors in (Verbat, Duval, Klerkx, Govaerts, & Santos, 2013) describe the four stages involved in the analysis: awareness, reflection, sense-making, and impact. In this sense, experiences prepared with the CIS analytics layer focus on the reflection stage, and our aim is to provide teachers and researchers with valuable information on how learners interact with CBT when performing a *Learning Activity*, and examine at the same time the effectiveness of the proposed cloud-based tool for learning and the CLAO unified workspace environment. This experience is also intended to provide experiences with MOOC extending the work in (Clow, 2013).

The procedure conducted for this study was: following the basic enrollment and introduction steps presented in similar experiences (Hernández et al., 2013a), seven innovative LA (see Table 8.24) using two CBT (Google Drive editor, GD, and the mind-mapping tool MindMeister, MM) were prepared in the CLAO architecture. Information presented was extracted from the CIS analytics layer dashboard. The three courses, described in Table 8.23, were executed in sequential order, providing feedback to the next course, thus improving the learner experience and data gathering (each group has a different group of learners).

For online document editor (GD), data generated from the CIS analytics layer is presented in Table 8.26. Some of the relevant identified variables for analysis are: number of words; number of paragraphs; revisions (number of editions in document); change rate between revisions; time used to complete the activity; number of sessions (NS) in the CLAO, and average time per session in minutes (TPS) where a session stands for use of GD in the CLAO with inactivity less than 30 minutes. The most important results from data generated, according to number of words, revisions, and TPS are the following: In MOOC activity (a.1) it was identified that learners were not using the cloud-based tool directly, and the ones who used it through CLAO presented a small number of revisions, indicating possibly that they basically copied their final work from an offline document editor, and did not at all use the GD/CLAO interaction layers. For MOOC activity (b.1) results from Table 8.26 present a slightly better use of the CLAO, with an increased TPS. In (b.2), CLAO usage is improved in all observed variables. Interestingly, the same pattern is identified for MOOC (c), beginning with learners with fewer revisions in (c.1), for (c.2), with the use of GD increased significantly in (c.3); the results are in Table 8.25. These first results reflect a behavior in which the learners, as they grow more "used to" the CLAO and cloud-based tool, make a more enhanced and meaningful use of the tools.

Activity	Revisions	Total words	TPS
a.1	M=1.08 (SD 0.37)	M=525 (SD 648)	M=5.44 (SD 3.56)
b.1	M=2.51 (SD 1.58)	M=736 (SD 599)	M=25.27 (SD 47.53)
b.2	M=3.60 (SD 3.17)	M=2237 (SD 1505)	M=66.09 (SD 170.66)
c.1	M=1.94 (SD 1.25)	M=916 (SD 1011)	M=41.39 (SD 57.22)
c.2	M=1.33 (SD 0.71)	M=547 (SD 249)	M=58.66 (SD 67.20)
c.3	M= 3.2 (SD 1.58)	M=1438 (SD 542)	M=54.76 (SD 72.71)

Table 8.25 Results for Google Drive–based activities generated by CIS analytics layer taken from (Hernández, Gütl, & Amado-Salvatierra, 2014c)

For the online mind map editor tool (MM) the following metrics were evaluated: total number of changes; number of first ideas (NFI); number of ideas in further depth level, (NFD; from 2nd level and further levels); time used to complete the activity; number of sessions (NS) in the CLAO, and average time per session in minutes (TPS). An exploratory activity with mind maps was prepared in activity (a.2), in which learners were using the tool but the final work was not linked with the CLAO; results in number of ideas were (M = 16.73, SD = 13.81). In MOOC (c), a new implementation was made, activating the learning resource (mind map) directly in the CLAO; learners started the mind map in the CLAO, and all related information was stored by the CLAO analytics layer. Activities are presented in Table 8.26, where (c.2) and (c.3) show an interesting increase in mind maps metrics, exposing how learners learned to use the tool via LAO's UI. Once learners have an experience with MMs, they use it to create more elaborate work.

Act.	Ideas	NFI	NFD	TPS
c.1	M=15.77 (SD 21.32)	M=8.72 (SD 2.46)	M=9.97 (SD 19.89)	M=55.30 (SD 69.78)
c.2	M=67.57 (SD 70.18)	M=5.07 (SD 0.88)	M=61.62 (SD 70.87)	M=40.07 (SD 63.98)
c.3	M=65.71 (SD 40.03)	M= 5.44 (SD 1.89)	M=59.27 (SD 38.90)	M=50.18 (SD 59.20)

Table 8.26 Results for mind-map-based activities generated by CIS analytics layer taken from (Hernández, Gütl, & Amado-Salvatierra, 2014c)

The CIS analytics layer made clear how learners used the CLAO and related CBTs, identifying failure/success patterns. A general pattern is evident: as time passes in the course, and learners grow used to the CLAO and related CBT, the assignments the learners present using these tools become more relevant, more detailed (with more complex structures in the case of MM), and more extensive.

Related to assignment submission, in the first two MOOCs (a) and part of (b), it was identified that learners had the opportunity to send the assignment through the LMS, consisting in a static link of the work they had prepared in MM or GD as a standalone tool; but this approach gave rise to several problems.

First, learners used the tools GD/MM and avoided using the proposed interface in the CLAO (reflected in results in MOOC (a)). Interestingly, a correction was introduced in MOOC (b) which clearly emerged in results presented in the *Learning Activity* (b.2) increasing CLAO usage.

Second, the MM or GD resource belonged to the learner, revealing the following issues: (1) a learner could modify an assignment even after submitting it, (2) a learner could erase the resource and all its history and so keep it from further analysis, and (3) a learner had to *manually share resource editing rights* with the CLAO in order for allow it to be able to perform automatic analytics.

The solution to these issues was done by automatically create the GD o MM instance resource through the CLAO, and share them to the learner, and therefore the assignment delivery is only through the CLAO itself. The final experience, MOOC (c), provided the best experience, as clearly shown in the data collection and CLAO usage. One factor that was key to improving overall data gathering was to clearly tell learners that they had to use the shared MM or GD at all times, and that they should not work in other MM or GD and then copy and paste to the final resource. Because that would not keep the history of use within the CBT, something explicitly mentioned to the learners is assessed within the course.

The CIS, both second and third generation proved to perform well under the MOOC environment, which as already mentioned, handles thousands of learners. Both CIS generations bring the same results from the learners and teachers perspective, for the required Web interoperability that enables to enact and orchestrate LA, although the CIS third-generation brings all the Semantic Web advantages elaborated in more detail in Section 8.1.

Finally, the results of this study are complemented by the previously presented usability, motivation and emotional evaluations introduced in Section 8.2. As indicated in that Section 8.2, the learners results to those evaluations are highly positive and complement the results presented in the present study.

8.6 Discussion

The technical implementation was demonstrated to be quite straightforward (Section 8.1), although it requires a semantic proxy, the CPH, that handles the entire mapping between the CBT and the corresponding API Doc built for it. Furthermore, new CBTs may require support extensions in the CPH. The semantic third-generation CIS requires only the identification of objects, properties, and operations in a structured form, which allows us to express them semantically. The developer only has to create the high-level Web API description using Hydra vocabulary and then map it to the current Web API (if it does not use Hydra). The scalability tests along with the MOOC experiments confirm the architecture scalability. Moreover, in accordance with the targeted use case (Section 4.2), the current architecture can be used for any type of educational scenario, such as face-to-face, blended, or distance learning.

The study in Section 8.2 evaluated the MOOC experience considering emotional, motivational, and usability aspects and reviewed the use of CBTs for LA. The participants' attitudes toward motivational and emotional aspects were highly ranked, and participants also indicated positive learning outcomes. At the same time, the study provided results indicating a very positive evaluation of the usability of CBTs.

For the ROLE PLE, a complete CEE was enabled (see Section 8.3) at both the infrastructure and the application levels. The results indicate that the technologies provided by the ROLE project enable the development of a truly cloud-based PLE. Initial results from evaluating this PLE with learners from three different Latin-American countries have shown that it is generally perceived as a useful learning platform. However, given the novelty of this approach, the need to provide guidance and scaffolding to new users was clearly outlined. In general, more time is required for learners to perform these types of activities, especially activities that are group based.

From an educational orchestration point of view, the ability to include and manage several CBTs, and setting learning paths with them opens a large set of opportunities for enhancing and expanding the construction of learning experiences. Challenges still remain to be addressed, such as how to ensure that the learning path is followed and that the student does not get lost, confused, or distracted in the CBT. The first-generation CIS was proven to be a key component to allow interoperability between CBTs and learning environments such as the ROLE PLE.

Regarding the MSLQ study (Section 8.4) on motivations and cognitive learning strategies, the learner's present high motivations in the MOOC. Through the use of the CLAO, learners see each *Learning Activity* as relevant to their own contexts, and they see themselves as intrinsically motivated and as having capabilities to perform well in the course. Having a solid learning strategy in place for MOOCs will probably increase commitment to the learning experience and decrease the high dropout rates that are very common among MOOCs, where organization, elaboration, and metacognitive strategies are fundamental to success. Despite the current low peer interaction and help seeking indicated by learners, it is of great interest to create communities that to do not *reset or restart themselves* with every course ending. Instead, the learning community must be enabled to continue, reinforce itself, and grow across time independently of course schedules and editions, as with educational resources such as Khan Academy (Morales, et al. 2014). Hence, the frontier between xMOOCs and educational resources might blur in the future.

Learners have demonstrated not only that these types of LA are motivational but also that the system is usable and evokes positive emotions. However, there are computer literacy traps—for example, the very first time using the CBTs and (or) the CLAO requires an introduction to its usage (see Section 8.5). Otherwise, the cognitive load may be too high, transforming the learning experience into a disappointing one. When using CBTs, the user needs to be conducted and guided through the system with the corresponding instructions on the usage of the CBT. If a CBT is somewhat detached from the learning environment and learning overall grading, even if the CBT use is required, is possible that it is not be used as expected, or even not used at all.

Although previous studies have shown (Sections 8.2, and 8.3) that learners are willing to use and enjoy using CBTs, some sort of summative evaluations and grades have to be embedded into the *Learning Activity* to ensure full exploitation of the learning experience as it was conceived by the teacher. If a *Learning Activity* uses more than one CBT, the system must require the use of all of them. If it does not, the learner tends to use just the CBT that is required to present the final work.

The behavior analytics results (Section 8.5) reinforce the idea that a new CBT inclusion in the CLAO must be carefully crafted to achieve the correct set of automation, where it is important in the LA, as in the use case, that the VLE owner can perform CBT instance creation, have instance ownership, and perform permission management.

9 CONCLUSIONS AND FUTURE WORK

This doctoral dissertation aims to provide solution for creating flexible educational environment with a focus on CBT *Learning Activities (LA)* orchestration and to realize interoperability through innovative semantic Web technologies. This includes fine-grained controls over the resources in the CBTs' ecosystem to allow control of every resource (or object) within the CBT. Realizing such controls requires access to the CBT Web API, which is currently available using common static Web contracts. Thus, using semantic Web technologies, it is possible to enable the automatic processing of such Web APIs. It is no longer required to program custom contracts to enable interoperability. Instead, the Web API is defined at a higher level, and the CIS constructed in this thesis is capable of automatically recognizing and processing the Web API.

This Chapter aims to present and discuss the research results and contributions concerning this doctoral dissertation. First (Section 9.1), a summary of the whole thesis is given, and then Section 9.2 discusses the contributions by focusing on each of the general research questions. Finally, Section 9.3 identifies open issues and further work concerning the CIS, the CBTs and education, and related specifications.

9.1 Summary

The theoretical background from Chapter 2 and 3 provides a solid framework to work on the idea of flexible educational environments, first with an overview of education and IC, addressing several learning theories and keeping in mind the use CBTs. By applying different learning paradigms, frameworks, and models when creating and orchestrating LAs, it is possible to create enhanced learning experiences. A comprehensive overview of open education, with a special emphasis on massive environments, prepares the way for the evaluation of such experiences. CBTs were identified for their capabilities, besides their innovative features, and because they provide programmatic access to such features, opening possibilities to create enhanced and tailored educational scenarios that can overcome many of the identified pedagogical issues to create a flexible education environment. Furthermore, the current state of the art in educational interoperability toward providing a flexible environment is reviewed. The results of the review indicated that are still major gaps to fill to guarantee a flexible yet tailored experience by using the CBTs' ecosystem for education. An improved interoperability approach is required, so advanced semantic Web technologies are introduced to shape the landscape of possibilities.

Problems and challenges are identified based on the preview Chapters and then described in Chapter 4. The challenges lie with the need to enable a *Cloud Education Environment* and to achieve enhanced orchestration through open Web APIs and sematic interoperability for automatic machine recognition of CBTs. Several objectives are set accordingly, and a main use case is developed that serves to further develop the rest of the research. Then the core foundation for this dissertation is presented, a conceptual model for flexible interoperability that creates a *Cloud Education Environment*. First, the semantics necessary to enable such environment are defined, and then the interoperability framework architecture and its various components are introduced. A key semantic proxy is also developed because it is foreseen as necessary while the semantic technology is adopted. The last part of the model is the orchestration system, which addresses the deployment of the LAs using CBTs.

Chapters 5, 6, and 7 describe the development of the framework architecture model, the CIS and CPH with the CLAO. It is a three-stage iterative development process. First, the interoperability (CIS) is built within a PLE. Second, the CBTs' LA orchestration system, the CLAO, with an improved interoperability framework, the CIS, is created. Based on these first two iterations, interoperability ontologies are developed for two *Application Domain Types (ADTs)*, and then, using the semantic model and the chosen technologies (JSON-LD and Hydra), corresponding *Generic Vocabularies (GVs)* are created. In the next step, specific API Documentation for each CBT Web API is built. All these pieces enable semantic interoperability, although the layer for processing such semantics requires an interoperability service that is capable of processing. Therefore, this layer is the focus of the third development iteration for the CIS architecture, which encompasses a multi-layered framework that addresses semantics. In addition, a *Communication Process Handler (CPH)* to support semantic interoperability is built. This enables semantic communication with non-semantic CBTs' Web APIs.

The evaluation phase first presents the lessons learned while building the framework architecture, the technical objectives set are accomplished, and scalability tests demonstrate the viability of such an environment, which suits open and massive education quite well, although more extensive scalability tests needs to be performed. A cost-effective comparison of the semantic approach compared to traditional Web contracts reveals that the semantics are not difficult to implement for average developers and demonstrates that the total development effort is lower compared to custom contract programming and handling. In contrast, the technology presented is still very new, and Semantic technologies are not widely adopted by the software developer community, thus is required more dissemination of the work addressed by this thesis. Moreover, Chapter 8 contains a comprehensive set of four evaluation studies that address different perspectives related to flexible educational environments. First, a study focuses on evaluating emotions, motivation, and usability in standalone CBTs in a MOOC environment. Another evaluation perspective is added in the second study by evaluating emotions and the usability of CBTs embedded in a PLE, this time in two standard online courses. In those two experiences, it is clear that is highly required to have an introductory phase in a course, for the learner to gain experience with the CBTs and the deployment environment, before trying to perform a learning activity with CBTs. In addition, the use of the CIS framework architecture for interoperability is introduced. Analytics are gathered as well. The third study evaluates motivations and cognitive learning strategies using MSLQ, it has reveal the relevant strategies regarding successful learners, although longitudinal evaluations are required to identify learning patterns across multiple courses and disciplines in a MOOC setting. The scenario is set in a MOOC, now using the CLAO. Several reliability instruments are employed and demonstrate support of the results. The fourth study addresses the evaluation of users' behavior in three different MOOCs, supporting the dissertation's main use case with over 6,000 learners. Again, this fourth study reveals that is not only necessary to have an introductory phase for CBT usage, furthermore is required more precise processes for following a learning activity using multiple CBTs. This means that instructions have to be clear, but enforced (e.g. automatically orchestrated), if not the learner is likely to lose the correct learning path. The outcomes for each study are highly positive, elaborate different perspectives, use widely accepted measurement instruments, provide insights on the use and impact of CBTs, and demonstrate that a flexible educational environment has been successfully created.

9.2 Results and Contributions

In this Subsection, the general research questions presented in Chapter 1 are addressed. The research conducted investigated a flexible educational environment with a focus on the creation of LAs that use innovative and external tools such as CBTs and orchestrate the learning experience by interoperating with the CBT Web API. The interoperability is achieved through a semantic Web approach. The main research goals are described along with a summary of the results and contributions. Furthermore, over 20 research publications has been done for this dissertation, many of them experimenting with the flexible CEE created, which is the CLAO, the CIS and the CPH.

How can the Cloud-Based Tools be used to fulfill the current and future needs of e-Education?

A holistic literature review has been performed pertaining to the inclusion of CBTs in education, this review covers the potential use of CBTs, current experiences, learning theories, cultural, technical and societal changes, and other factors. Chapter 2 presents the learning theories and perspectives that provide the foundation to support the inclusion of innovative CBTs in education. The literature review conducted the research to emphasize in *Computer-Supported Collaborative Learning (CSCL)* (Dillenbourg et al., 2009), although not necessarily all the learning experiences enabled through CBTs must be collaborative. CSCL has already shown that the design of LAs is not enough and that orchestration is a key foundation to deploy LAs, addressing issues such as monitoring, adaption, workload management, intervention coordination, regulation, social interactions, and changes in the teacher's role (Prieto et al., 2011).

Moreover, orchestration requires the use of models and frameworks (Conole & Alevizou, 2010), which can be applied and (or) adapted to the LAs that are created using CBTs. At Galileo University, the five-stage e-moderation model (Salmon, 2000) has been used for several years, so it was a natural choice to review it for this research. This type of model or framework can be used as a mediating artifact when designing LAs. Properly structured LAs with their components include (Conole & Alevizou, 2010) 1) the context, the intended learning outcomes and the environment where the LA takes place (aims, pre-requisites, skills, subject, environment, time, and difficulty) 2) associative, cognitive, or situational educational approaches and 3) the task undertaken, including the type, technique, interaction, roles, resources, tools, assessment, and sequence. LAs with the use of *Mediating Artifacts* such as narratives and case studies, lesson plans, templates and wizards, toolkits, models, and patterns result in a simple yet robust use case for CEE (see Section 4.2).

Therefore, when addressing the LAs that were presented and evaluated in Sections 8.2, 8.3, 8.4, and 8.5, the aforementioned LAs approaches and models were taken into account. A comprehensive evaluation in relevant areas was performed, including evaluating emotions, motivation, usability, and cognitive learning strategies. This included evaluations in the CEE created for this dissertation, as well as the use of a very advanced PLE. The evaluations were performed in several scenarios, including MOOCs and traditional online courses. All these scenarios confirmed, through the positive results exposed in Chapter 8, that the CBTs could fulfill the needs of e-education for both learners and teachers. In total, this doctoral dissertation reports results from six courses and over 30 LAs. Over 6,000 learners were enrolled in the courses in over 20 countries, with statistically significant responses to surveys. The results of the presented evaluations (see Chapter 8) indicated that learners are avid to have such CBTs embedded in their learning experiences.

How can we achieve a highly flexible education environment in terms of the inclusion and orchestration of Learning Activities that use Cloud-Based Tools? Thereby, simplifying interoperability mechanisms in terms of authentication, communication, and processing of *Cloud-Based Tools*' Web APIs and providing interfaces for process automation, enabling what is known as a *Cloud Education Environments*.

Chapter 3 gives a comprehensive overview of flexibility in education environments, beginning with the pedagogical aspects. Building on the literature review provided in Chapter 2, several of the current problems and challenges from a pedagogical point of view are identified. One key challenge is the capability to achieve *granular* management of the resources in the CBTs, including authority, control, and role management within the CBTs, to enable process automation, such as assigning roles, creating groups, assigning tools and resources, etc. This mentioned challenge is aligned with other identified adoption barriers such as teachers' full control over the educational experience, along with simplifying the learner's overall experience. Challenges also extend to institutional-wide adoption in terms of interoperability with current legacy systems. The literature review in this regard focused on current systems, frameworks, toolkits, standards, and specifications that could bring flexibility to an educational environment. It was found that, although many technologies to support the inclusion of third-party tools exist, many of them fall short of CBTs' resource management and automation, so the aforementioned challenges were still open issues.

Chapter 4 provides a comprehensive model for flexible interoperability, enabling the inclusion of any type of CBT by means of its own Web API. A system for the orchestration of LAs is presented. Furthermore, a single interoperability framework architecture that solves the CBTs' issues involving authentication, processing, communication, and visualization is presented. These topics, modeled in Chapter 4, are consequently realized in Chapter 5 by building the aforementioned architecture. First by enabling the core components of the architecture in a PLE (see Section 5.2), and the evolving that architecture into a solid foundation for further development (Section 5.3.1) and the introduction of the *Cloud Learning Activity Orchestration* (CLAO) system. The architecture was evaluated with learners, as described in the evaluation experiences of Section 8.3 and 8.4. Learners' motivations and emotions were assessed, and systems usability studies were performed. Moreover, a relevant cognitive learning strategies study was conducted using the CLAO. Positive outcomes from the evaluation phase indicated that the built educational environment is highly flexible and complies with the objectives of this research.

What interoperability mechanisms design can enable machine-to-machine processable Cloud-Based Tools' Web APIs? Such mechanisms do not require human intervention to write custom standard contracts between systems. Instead, based on interoperability ontologies, create a pragmatic semantic description of the Web API, both per tool and per application domain. With such description be able to automatically recognize the tool's capabilities that are available at a given time.

Another relevant part of the literature survey was a review of the state of the art on Web interoperability (Chapter 3). The outcomes of the review indicated that traditional Web contracts limit interoperability to become machine processable. The research led toward the semantic Web and corresponding current approaches. Despite the many promises and relatively mature research community in the field of the semantic Web, the predominant approaches are perceived as complex and impractical. In contrast, the use of JSON has become a major trend in communication between Web services, so a recent specification, JSON-LD, includes support for the semantic annotation of resources. Still, a vocabulary capable of describing a Web API is required. Hydra does that and is built atop JSON-LD, thus making a perfect combination to create and leverage the presented semantic definition of CBTs' Web APIs (introduced in Section 4.3.1).

With this insight in mind, an interoperability ontology for CBTs' specialized ADT was created. Two domain types were selected, namely mind maps and an online document editor. From the ontology was derived the *Generic Vocabulary* that uses Hydra to describe the Web API generics for each domain. That *Generic Vocabulary* was then used as a starting point to describe the specifics of a given CBT Web API through creating the API Documentation (see Chapter 6).

To realize this new interoperability method that uses Hydra, the *Cloud Interoperability Service (CIS)* framework, was extended to support it (Chapter 7). An enhanced *business logic layer* was built to support Hydra to process the Web API automatically by discovering its features. This did not require custom programming or any other type of human intervention to write contracts between systems. As the evaluation in Chapter 8 indicated, the system proved to be fully functional, scalable, and cost-effective for developers, successfully meeting the various challenges, problems, and objectives set for this research. Finally, to enable such advanced technology with current CBTs' Web APIs, it was necessary to implement an intermediary layer that communicates between the Hydra-based system and the current CBT Web API (Section 7.2)

9.3 Open Issues and Further Work

Although several CBTs were used in the evaluation phase, a more comprehensive review of more CBT is still needed. The richness and possibilities that current and new CBTs bring to education is ample, so more deep evaluations in several application domains need to be performed (this dissertation focuses on two application domains, nevertheless there are hundreds that can be used for education). Still missing is a comprehensive taxonomy that relates CBTs with pedagogically sound models, frameworks, etc. This includes comprehensive compendium of CBTs in different educational dimensions, such as learning theory, thinking skill order that it promotes, application into pedagogical approaches like *Problem Based Learning* (and many others). Furthermore development LAs examples for using CBTs are required, along with a better classification of the types of *mediating artifacts* that could be used. Many models like the one presented in this thesis can be used, evaluated, and adapted. The evaluations with learners presented are still limited, more extensive, and if possible longitudinal studies are needed to support a better approaches for the inclusion of CBTs in education along with a CEE.

The best assessment methods for CBTs are yet to be identified. In the MOOC experiences presented throughout the current research, the capability to have semi- or fully automatic assessments of CBT LAs is of high relevance because manually assessing thousands of assignment in such a massive environment does not scale. Furthermore, with a robust CBT Web API, it is possible to think about extracting all interaction data and all that has been done within the CBT, and with that information and data, it is possible to create generic proximity models that later can be assessed by the teacher to assess automatically each individual resource created by learners. For example, in a mind map, learners create their own mind map for a given LA. Then, the proximity model system creates a few mind maps based on common similarities that later is evaluated by the teacher. Thus, instead of assessing thousands of mind maps in a MOOC, the teacher needs to assess only a few. Then, the system uses those already assessed mind maps as models and automatically assess the real mind maps created by the learners. In contrast to this automatic assessment example, some types of tools such as an online document editor, proves highly difficult to assess automatically.

More work is required to create generic and per-application domain-type vocabularies to represent *Learning Analytics* models. In the experience evaluated in this thesis, the more access and information the CBTs' Web APIs provide, the better the information gathering is. Future work shall include the creation of a statistical prediction dashboard to guide learning interventions, as opposed to just summarizing behavior after course completion. The *Learning Analytics* presented are introductory cases that needs to be deeply developed and evaluated to maximize the potential of this CEE approach.

How the CIS and the corresponding semantic interoperability methods can be used with LTIv.2 remains an open question. Both benefit from strong features and may be a solution for each other's blind spots. They can be combined to create a new version of the specification or serve as complementary approaches. For instance, with IMS-LTI v.2, the use of the context concept to identify a class, course, etc., is of high value to the CIS architecture.

As TP adopts Hydra, the CPH plays a major role, so it is foreseen as a possible enhancement to create an API Documentation builder to simplify the developer's work to create and maintain tools or services that they want to convert to Hydra. This means that there is no further need to write JSON-LD and Hydra of existing Web APIs. Instead, developers build the Hydra-based Web API using an automated process of recognition discovery and manual mapping of the Web APIs to an API Doc that uses a *Generic Vocabulary*. Another further enhancements is in the *Template Layer* to easily created management tools in the CLAO for each CBT to be included.

A major obstacle for adoption of the outcomes of this research is to enable the current prototype technology to many more VLE and CBTs, to gain a real adoption momentum in the educational community, and also with the technical developer community. The *Communication Process Handler (CPH)* serves to interface with CBTs that do not respond with JSON-LD and Hydra. Although the presented cases are well supported by the CPH, the long-term maintenance of it is complex and cumbersome in terms of technical maintainability, due the changes that the CBT Web APIs experiment through the time. In addition, the CBTs providers need to open up the Web APIs, allowing and continually increasing controls over every single resource and feature of their CBT, at this moment many of the CBTs have limited features available in their Web API.

Finally, one important issue that has been not addressed in the CLAO and in general in this research is the accessibility of CBTs, so a general review of accessibility and the associated challenges to overcome is required.

10 REFERENCES

Adamopoulos, A. (2013). What Makes a Great MOOC? An Interdisciplinary Analysis of Student Retention in Online Courses. Proceedings of the 34th International Conference on Information Systems, ICIS Band.

Alario-Hoyos, C., & Wilson, S. (2010). Comparison of the main alternatives to the integration of external tools in different platforms. In Proceedings of the International Conference of Education, Research and Innovation, ICERI, (pp. 3466-3476).

Alario-Hoyos, C., Bote-Lorenzo, M., Gómez-Sánchez, E., Asensio-Perez, J., & Vega-Gorgojo, G. (2013). GLUE!: An architecture for the integration of external tools in Virtual Learning Environments. Computers & Education, 60(1), 122-137.

Albert, D., & Lukas, J. (Eds.). (1999). Knowledge spaces: Theories, empirical research, and applications. Psychology Press.

AL-Smadi, M. (2012). Flexible and Integrated E-Assessment in complex learning resources. Dissertation in fulfillment of the requirements for the academic degree Doctor of Technical Sciences (Dr. techn.) in Computer Science.

Amazon. (1996). Retrieved from http://www.amazon.com/

Apache (2014). Apache Shindig. Retrieved 2014, from https://shindig.apache.org

AppBrain. (2014). App Brain Stats. Retrieved 9 2014, from http://www.appbrain.com/stats/number-of-android-apps

AppleApp. (2014). AppleApp Store. Retrieved 9 2014, from https://itunes.apple.com/es/genre/ios/id36?mt=8

Armbrust, M. F. (2010). A view of cloud computing. Communications of the ACM, 53(4), 50-58.

Aroyo, L., Dolog, P., Houben, G. J., Kravcik, M., Naeve, A., Nilsson, M., & Wild, F. (2006). Interoperability in personalized adaptive learning. Educational Technology & Society, 9(2), 4-18.

Azebedo, R. (2002). Beyond intelligent tutoring systems: using computers as METAcognitive tools to enhance learning? Instructional Science, 44-45.

Balakrishnan, G. (2013). Predicting Student Retention in Massive Open Online Courses using Hidden Markov Models. EECS.

Barr, J., Varia, J., & Wood, M. (2006). Amazon EC2 Beta. Retrieved 10 2013, from http://aws.ty- pepad.com/aws/2006/08/amazon_ec2_beta.html

Beckett, D. (2004). RDF/XML Syntax Specification (Revised). W3C Recommendation.

Beetham, H. (2013). Designing for active Learning in technology-rich Contexts. Rethinking Pedagogy for a Digital Age: Designing for 21st Century Learning, page. 31. In Beetham, H., & Sharpe, R. (Eds.). (2013). Rethinking pedagogy for a digital age: Designing for 21st century learning. routledge.

Berenfeld, B., & Yazijian, H. (2010, March). Global Lab: Harnessing the cloud & social networking for K-12 science learning… since 1991. In Society for Information Technology & Teacher Education International Conference (Vol. 2010, No. 1, pp. 3544-3547).

Berners-Lee, T.. (2006). Linked Data. Design Issues for the World Wide Web.

Berners-Lee, T. (1989). Information Managment . A Proposal, CERN.
Biggs, J. (1999). Teaching for Quality Learning at University. Society for Research into Higher Education (SRHE) and Open University Press, Buckingham.
Biron, P., & Malhotra, A. (2004). XML schema part 2: Datatypes. World Wide Web Consortium Recommendation REC-xmlschema-2-20041028.
Blackboard (2011). Edugarage. Retrieved from http://www.edugarage.com/display/BBDN/Technical+Building+Blocks+API+overview
Bloom, B. (1956). Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain. New York: Longman.
Brooke, J. (1996). SUS-A quick and dirty usability scale. Usability evaluation in industry, 189(194), 4-7.
Brown, S., Race, P., & Bull, J. (1999). Computer-Assisted Assessment in Higher Education. Staff and Educational Development Series. Stylus Publishers Inc., PO Box 605, Herndon, VA 20172-0605.
Burgess, J. (2006). Blogging to learn, learning to blog. . In: A. Bruns and J. Jacobs (Eds.), Uses of blogs. New York: Peter Lang, , 105 ‐ 114.
Buyya, R., Yeo, C. S., & Venugopal, S. (2008). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on (pp. 5-13).
Buzan, T., & Buzan, B. (1993). The Mind Map Book How to Use Radiant Thinking to Maximise Your Brain's Untapped Potential. New York: Plume.
Cáceres, M. (2010). Widget Packaging and Configuration. W3C Working Draft.
Cacoo. (2014). Cacoo website, Online Diagram Software for Flow Chart & UML. Retrieved 6 2014, from https://cacoo.com
Cardon, D., & Aguiton, C. (2007). The Strength of Weak Cooperation: an Attempt to Understand the Meaning of Web 2.0. Communications and Strategies, Vol 65: 51 ‐ 65.
Carrington. (2013). The Padagogy Wheel V2.0: It's all about transformation and integration. Retrieved from http://www.unity.net.au/allansportfolio/edublog/?p=836
Chamberlin, L., & Parish, T. (2011). MOOCs: Massive Open Online Courses or Massive and Often Obtuse Courses?. eLearn, 2011(8), 1.
Chang, V., & Guetl, C. (2007). E-Learning Ecosystem (ELES)-A Holistic Approach for the Development of more Effective Learning Environment for Small-and-Medium Sized Enterprises (SMEs). In Digital EcoSystems and Technologies Conference, 2007 (pp. 420-425). IEEE.
Chang, V., & Guetl, C. (2010). Generation Y learning in the 21st century: Integration of virtual worlds and cloud computing services. Abas, Z. (Eds.), Proceedings of Global Learn Asia Pacific, pp. 1888–1897.
Chao, L. (2012). Overview of Cloud Computing and its Application in E-Learning. Cloud Computing for Teaching and Learning: Strategies for Design and Implementation: Strategies for Design and Implementation, 1.
Chatti, M., Dyckhoff, A., Schroeder, U., & Thüs, H. (2012). A reference model for learning an-alytics. International Journal of Technology Enhanced Learning, 4(5). pp. 318-331.
Chi, E. H., Pirolli, P., & Pitkow, J. (2000). The scent of a site: A system for analyzing and predicting information scent, usage, and usability of a web site. In Proceedings

of the SIGCHI conference on Human factors in computing systems. ACM, p. 161-1.
Chinnici, R., Gudgin, M., Moreau, J. J., Schlimmer, J., & Weerawarana, S. (2004). Web services description language (WSDL) version 2.0 part 1: Core language. W3C working draft, 26.
Chittaro, L., & Ranon, R. (2007). Web3D technologies in learning, education and training: Motivations, issues, opportunities. Computers and Education., 49(1): 3 ‑ 18.
Churches, A. (2008). Bloom's taxonomy blooms digitally. Tech & Learning, 1.
Chyung, S. Y. (2004). Hoping to Reduce Attrition? Follow the SIEME Model and Investigate Motivation-Hygiene Factors. In Proceedings of the 20th Annual Conference on Distance Teaching and Learning (pp. 1-6).
Clement, L., Hately, A., Riegen, C., & Rogers, T. (2004). UDDI Version 3.0.2. Oasis Standard.
Clow, D. (2013). MOOCs and the funnel of participation. Proceedings of the Third International Conference on Learning Analytics and Knowledge, 185-189.
Cochrame, T. (2012). Mobile Cloud Services as Catalysts for Pedagogical Change. Cloud Computing for Teaching and Learning: Strategies for Design and Implementation. Ed. Lee Chao, Chapter 11.
Conole, G. (2007). Describing learning activities: tools and resources to guide practice. In H. B. (Eds.), Rethinking Pedagogy for a Digital Age: Designing and Delivering E-Learning (pp. 81-91). London: RoutledgeFalmer.
Conole, G., & Fill, K. (2005). A learning design toolkit to create pedagogically effective learning activities. Journal of Interactive Media in Education, 2005(1), Art-9.
Conole, G., Brasher, A., Cross, S., Weller, M., Clark, P., & Culver, J. (2008). Visualising learning design to foster and support good practice and creativity. Educational Media International, vol. 45, no 3, p. 177-194.
Conole, G., & Alevizou, P. (2010). A literature review of the use of Web 2.0 tools in Higher Education. A report commissioned by the Higher Education Academy.
Costello, S. (2014). How Many Apps Are in the iPhone App Store? Retrieved 09 2014, from http://ipod.about.com/od/iphonesoftwareterms/qt/apps-in-app-store.htm
Crockford, D. (2006). RFC4627: The application/json Media Type for JavaScript Object Notation (JSON). Internet engineering Task Force (IETF) Request for comments.
Crook, C. (2008). Web 2.0 technologies for learning: The current landscape–opportunities, challenges and tensions.
D2L. (2014). Desire2Learn. Retrieved from http://www.d2l.com/
Daniel, J. (2012). Making Sense of MOOCs: Musings in a Maze of Myth, Paradox and Possibility. Journal of Interactive Media in Education.
Daradoumis, T., Bassi, R., Xhafa, F., & Caballé, S. (2013). A review on massive e-learning (MOOC) design, delivery and assessment. In P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2013 Eighth International Conference on (pp. 208-213). IEEE.
Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly, 13(3), 319-340.
Dagger, D., O'Connor, A., Lawless, S., Walsh, E., & Wade, V. P. (2007). Service-oriented e-learning platforms: From monolithic systems to flexible services. Internet Computing, IEEE, 11(3), 28-35.

Desmet, N. (2008). Sakai. Retrieved from https://confluence.sakaiproject.org/display/MYSAK/Widget+Development+Manual+-+How+to+write+a+widget+successfully
Dev, M. (2014). Mashery Dev. Retrieved 9 2014, from http://developer.mashery.com/apis#all
Dicheva, D., Mizoguchi, R., & Greer, J. (2009). Semantic Web Technologies for e-Learning. The Netherlands: IOS Press.
Dillenbourg, P. (2002). The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed), Three worlds of CSCL: Can we support CSCL, pp. 61-91.
Dillenbourg, P., Järvelä, S., & Fischer, F. (2009). The evolution of research on computer-supported collaborative learning. In Technology-enhanced learning (pp. 3-19). Springer Netherlands.
Dipity. (2011). Retrieved from http://www.dipity.com/
DiSessa, A. A. (2001). Changing minds: Computers, learning, and literacy. Mit Press.
Donaldson-Pressman, S., Jackson, R., & Pressman, R. (2014). The Learning Habit: A Groundbreaking Approach to Homework and Parenting that Helps Our Children Succeed in School and Life. Penguin.
Dong, Q. (2008). Explaining why young adults use MySpace and Facebook through uses & gratification theory. Paper presented at the Annual Meeting of the NCA 94th Annual Convention, Nov 20, TBA, San Diego, California.
Downes, S. (2005). E-learning 2.0. eLearning Magazine.
Drive (2014). Retrieved from http://drive.google.com
Duncan, C. (2003). Granularization. in A. Littlejohn (ed.) Reusing Online Resources – A Sustainable Approach to e-Learning, London: KoganPage, , pp. 12–19.
Educaplay (2014). Retrieved from http://www.educaplay.com/
Education (2014). Web 2.0 Tools. Retrieved from http://web2014.discoveryeducation.com/web20tools.cfm
Edudemic (2011). The 100 Best Web 2.0 Classroom Tools Chosen By You. Retrieved from http://www.edudemic.com/best-web-tools/
edX (2014). Retrieved from www.edx.org
Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication, 13(1), 210-230.
Facebook (2012). Facebook Social Plugin for Comments. Retrieved from http://developers.facebook.com/docs/reference/plugins/comments/
Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures (Doctoral dissertation, University of California, Irvine).
Fini, A. (2009). The technological dimension of a massive open online course: The case of the CCK08 course tools. The International Review of Research in Open and Distributed Learning, 10(5).
Fishbein, M., & Ajzen, I. (1975). Belief, attitude, intention and behavior: An introduction to theory and research. Reading MA: Addison-Wesley.
Flanagan, D. (2002). JavaScript: the definitive guide. O'Reilly Media, Inc.
Fox, A. (2009). Cloud computing in education. Retrieved July 26, 2010.
Friedrich, M., Wolpers, M., Shen, R., Ullrich, C., Klamma, R., Renzel, D., & Von Der Heiden, B. (2011). Early Results of Experiments with Responsive Open Learning Environments. J. UCS, 17(3), 451-471.
FusionTables (2012). Retrieved from Tool available online http://www.google.com/fusiontables/Home/

Garrett, J. J. (2005). Ajax: A new approach to web applications.
GIT. (2014). GIT. Retrieved 2014, from http://en.wikipedia.org/wiki/Git_(software)
GitHub (2014). ROLE SDK. Retrieved from https://github.com/ROLE/ROLE/wiki/ROLE-SDK
Glance, D. G., Forsey, M., & Riley, M. (2012). The pedagogical foundations of massive open online courses. First Monday, 18(5).
GLOBE (2014). Global Learning Objects Brokered Exchange. Retrieved from http://www.globe-info.org/
Goanimate (2014). Retrieved from http://www.goaimate.com
Gonick, L. (2013). The Year Ahead in IT, 2013. Inside Higher Ed.
Goodyear, P., Jones, C., Asensio, M., Hodgson, V., & Steeples, C. (2004). Undergraduate students' experiences of networked learning in UK higher education: A survey-based study. In Advances in research on networked learning (pp. 91-121). Springer Netherlands.
Google (2014). Google Analytics. Retrieved 2014, from https://developers.google.com/analytics/
OWL (2009). OWL 2 Web Ontology Language. W3C Recommendation.
Brickley, D., & Guha, R. V. (2004). RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, February 2004.
Gütl, C., Hernández, R., Chang, V., & Morales, M. (2014a). Attrition in MOOC: Lessons Learned from Drop-Out Students. In Learning Technology for Education in Cloud. MOOC and Big Data (pp. 37-48). Springer International Publishing.
Gütl, C., & Chang, V. (2008). Ecosystem-based Theoretical Models for Learning in Environments of the 21st Century. International. Journal of Emerging Technologies in Learning (iJET), Vol 3 (2008), 50-60.
Gütl, C., Hernández, R., Chang, V., & Morales, M. (2014b). MOOC in Latin America: Lessons Learned from Drop-out Learners. In the 3rd International Workshop on Learning Technology for Education in Cloud (LTEC), Knowledge Management in Organizations.
Haake, J. (2006). A concept for seamless integration of chat conversations and documents in shared workspaces. 7th International Conference on the Design of Cooperative Systems.
Halawi, Pires, & McCarthy. (2009). An Evaluation of E-Learning on the Basis of Bloom's Taxonomy: An Exploratory Study. Journal of Education for Business.
Hammond, M. (1999). Issues associated with participation in on line forums - the case of the communicative learner. Education and Information Technologies.
Hernández, R., Pardo, A., & Delgado, C. (2007). Creating and Deploying Effective eLearning Experiences Using .LRN. IEEE Trans. on Education. vol 50, no. 4, pp 345-351.
Hernández, R., Guetl, C., & Amado-Salvatierra, H. (2011). Facebook for e-moderation: a Latin-American experience. Proceedings of the 11th I-Know Conference, (p. doi: 10.1145/2024288.2024332). Graz, Austria.
Hernández, R., Gütl, C., Amado-Salvatierra, H., & Al-Smadi, M. (2012). Facebook for CSCL, Latin-American experience for professors. Proceedings for the 12th IEEE International Conference on Advanced Learning Technologies iCalt2012, pages 327-328.
Hernández, R., Linares, B. H., Mikroyannidis, A., & Schmitz, H. C. (2013a). Cloud services, interoperability and analytics within a ROLE-enabled personal

learning environment. Journal of Universal Computer Science, 19(14), 2054-2074.

Hernández, R., Gütl, C., Chang, V., & Morales, M. (2013b). MOOCs Concept and Design using cloud-based Tools: Spanish MOOCs Learning Experiences. Foley A. (ed), Proceedings of The Sixth Conference of MIT´s Learning International Networks Consortium (LINC).

Hernández, R., Amado-Salvatierra, H., & Guetl, C. (2013c). Cloud-based Learning Environments: Investigating learning activities experiences from Motivation, Usability and Emotional Perspective. CSEDU 2013 Proceedings. Aachen, Germany: ScitePress.

Hernández, R., Gütl, C., Chang, V., & Morales, M. (2013d). MOOC in Latin America: Implementation and Lessons Learned. n the 2nd International Workshop on Learning Technology for Education in Cloud (LTEC), Knowledge Management in Organizations (pp. 147-158). Netherlands: Springer.

Hernández, R., Gütl, C., & Amado-Salvatierra, H. (2014a). Interoperability for Cloud-based Applications for Education Settings Based on JSON-LD and Hydra: Ontology and a Generic Vocabulary for Mind Map Tools. In Proceedings of the 14th i-Know Conference. ACM.

Hernández, R., Gütl, C., & Amado-Salvatierra, H. (2014b). Using JSON-LD and Hydra for Cloud-based tool Interoperability: A working prototype based on a Vocabulary and Communication Process Handler for Mind Map Tools. In Proceedings of the Ninth European Conference on Technology Enhanced Learning EC-TEL 2014. Springer.

Hernández, R., Gütl, C., & Amado-Salvatierra, H. (2014c). Cloud Learning Activities Orchestration for MOOC Environments. In the 3rd International Workshop on Learning Technology for Education in Cloud (LTEC), Knowledge Management in Organizations. Netherlands: Springer.

Hernández, R., Gütl, C., & Chang, V. (2014d). Must we be concerned with the Massive Drop-outs in MOOC? - An Attrition Analysis of Open Courses. International Conference on Interactive Collaborative Learning ICL.

Hernández, R (2015a) API DOC FOR MINDMEISTER. Available online from: http://research.ges.galileo.edu/cis/spec/latest/core/#api-doc-mindmeister

Hernández, R (2015b) API DOC FOR CACOO. Available online from: http://research.ges.galileo.edu/cis/spec/latest/core/#api-doc-cacoo

Hernández, R (2015c) API DOC FOR GOOGLE DRIVE ONLINE DOCUMENT EDITOR. Available online from: http://research.ges.galileo.edu/cis/spec/latest/core/#api-doc-google-drive

Hernández, R (2015d) API DOC FOR DRAFTIN. Available online from: http://research.ges.galileo.edu/cis/spec/latest/core/#api-doc-draftin

Hernández, R (2015f) GENERIC VOCABULARY FOR MIND MAPS. Available online from: http://research.ges.galileo.edu/cis/spec/latest/core/#generic-vocabulary-mind-maps

Hernández, R (2015g) GENERIC VOCABULARY FOR ONLINE DOCUMENT EDITOR. Available online from: http://research.ges.galileo.edu/cis/spec/latest/core/#generic-vocabulary-document-editor

Hernández, R. (2015h). Cloud Interoperability Service Architecture for Education Environments. special issue on Massive Open Online Courses: Combining

Methodologies and Architectures for a Success Learning. Journal of Universal Computer Science. (in print)

Hernández, R., & Gütl, C. (2015a). Towards a Flexible Cloud Education Environment - A Framework for E-learning. Fonseca D., Redondo E. (Eds.). Handbook of Research on Applied E-Learning in Engineering and Architecture Education. Hershey: Information Science Reference. (chapter submitted in review).

Hernández, R., & Gütl, C. (2015b) A Cloud-Based Learning Platform: STEM Learning Experiences with New Tools. In: Cloud-Based STEM Education for Improved Learning Outcomes. IGI Global. (chapter submitted in review)

Hernández, R., Morales, M., Medina, A., & Barchino, R. (2015a). Modeling educational usage of cloud-based tools. Computers & Education journal. (Submitted in review)

Hernández, R., Guetl, C., & Amado-Salvatierra H. (2015b). Using Cloud-Based Applications for education, a technical interoperability exploration for Online Document Editors. Proceedings for the 4th International Workshop on Learning Technology for Education in Cloud (LTEC2015). (Submitted in review)

Hew, K. F. (2011). Students' and teachers' use of Facebook. Computers in Human Behavior, 27(2), 662-676.

Hill, P. (2013a). The most thorough summary (to date) of MOOC completion rates. e-Literate, February 26th.

Hill, P. (2013b). Emerging student patterns in MOOCs: A (revised) graphical view. 2013-04-31]. http://mfeldstein. com.

Hydra-W3C (2014). Hydra W3C community group project. Retrieved 03 2014, from http://www.w3.org/community/hydra/

Hydra-Project (2013). Hydra Project. Retrieved 07 2014, from https://github.com/lanthaler/HydraClient

Hylén, J., & Schuller, T. (2007). Giving knowledge for free. OECD Observer, 263, 21-22.

Hyman, P. (2012). In the Year of Disruptive Education. Communications of the ACM, 55, 12, 20-22.

IETF (2012). The OAuth 2.0 Authorization Framework RFC 5849. Internet Engineering Task Force (IETF) http://tools.ietf.org/html/rfc6749.

IMS. (2013). IMS Global Learning Consortium. Retrieved 05 2014, from www.ims.org

IMS. (2014). IMS GLC Learning Tools Interoperability Basic LTI Implementation Guide. Retrieved 05 2014, from http://www.imsglobal.org/lti/ltiv2p0/ltiIMGv2p0.html

Isaksson, E., & Palmér, M. (2010, September). Usability and inter-widget communication in PLEs. In Fifth European Conference on Technology Enhanced Learning (EC-TEL10), The 3rd Workshop on Mash-Up Personal Learning Environments (MUPPLE10).

ISO (2006). Topic Maps: Information Technology. Retrieved 02 2014, from http://www.topicmaps.org/xtm/1.0/

ISSUU (2014). Retrieved from http://www.issuu.com

Jacobs, I., & Walsh, N. (2004). Architecture of the World Wide Web. W3C Recommendation, Volume One.

James, C. (2009). Young people, ethics, and the new digital media: A synthesis from the GoodPlay Project. MIT Press.

JavaScript. (2014). Retrieved 2014, from http://es.wikipedia.org/wiki/JavaScript

Jenkins, H. (2009). Confronting the challenges of participatory culture: Media education for the 21st century. Mit Press.

Jonassen, D., Davidson, M., Collins, M., Campbell, J., & Haag, B. B. (1995). Constructivism and computer-mediated communication in distance education. American journal of distance education, 9(2), 7-26.
Jordan, K. (2014). Initial trends in enrolment and completion of massive open online courses. International Review of Research in, 15(1) pp. 133–160.
Jovanovic, J., & Gasevic, D. (2011). Learning Context ontology of the IntelLEO Ontology Framework. Retrieved 01 2014, from http://www.intelleo.eu/ontologies/learning-context/spec/
Jovanovic, J., & Gasevic, D. (2012). Activities ontology of the IntelLEO Ontology Framework. Retrieved 03 2014, from http://www.intelleo.eu/ontologies/activities/spec/
JSON-LD (2014). JSON-LD Project code. Retrieved 04 2014, from www.json-ld.org
Kay, R., & Loverock, S. (2008). Assessing emotions related to learning new software: The computer emotion scale. Computers in Human Behavior, 24, 1605-1623.
Keen, A. (2007). The Cult of the Amateur: How Today's Internet is Killing Our Culture. Doubleday.
Kirschenmann, U., Scheffel, M., Friedrich, M., Niemann, K., & Wolpers, M. (2010). Demands of modern PLEs and the ROLE approach. In Sustaining TEL: From Innovation to Learning and Practice (pp. 167-182). Springer Berlin Heidelberg.
Kizilcec, R. F., Piech, C., & Schneider, E. (2013, April). Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In Proceedings of the third international conference on learning analytics and knowledge (pp. 170-179). ACM.
Kolas, L. (2006). Topic maps in e-learning: An ontology ensuring an active student role as producer. In World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, (pp. Vol. 2006, No. 1, pp. 2107-2113).
Kolb, D. (1984). Experiential Learning; Experience as the Source of Learning and Development. Englewood Cliffs, NJ: Prentice Hall.
Kraan, W., & Mason, J. (2005). Issues in Federeting Repositories. A Report on the First International CORDRA Workshop. D-Lib Magazine, 11(3).
Krasner, G., & Pope, S. (1988). A description of the model-view-controller user interface paradigm in the smalltalk-80 system. Journal of object oriented programming, 1(3), 26-49.
Kuutti, K., & Kaptelinin, V. (1997). Rethinking cognitive tools: From augmentation to mediation. In Cognitive Technology, 1997. Humanizing the Information Age. Proceedings., Second International Conference on (pp. 31-32). IEEE.
LAMS (2009). Retrieved from http://lamsfoundation.org/
Lane, L. (2012). Three kinds of MOOCs. Lisa's (Online) Teaching Blog.
Lanthaler (2014). Third Generation Web APIs Bridging the Gap between REST and Linked Data. Doctoral Dissertation.
Lanthaler, M. (2013). Creating 3rd generation web APIs with hydra. In Proceedings of the 22nd international conference on World Wide Web companion, (pp. 35-38).
Lanthaler, M., & Gütl, C. (2013). Hydra: A Vocabulary for Hypermedia-Driven Web APIs. In Proceedings of the 6th Workshop on Linked Data on the Web (LDOW2013) at the 22nd International World Wide Web Conference (WWW2013).
Lanthaler, M. (2014a). Hydra Core Vocabulary. Retrieved from A Vocabulary for Hypermedia-Driven Web APIs: http://www.hydra-cg.com/spec/latest/core/

Lanthaler, M. (2014b). JSON-HydraClient project. Retrieved 06 2014, from http://www.markus-lanthaler.com/
Lanthaler, M., & Gütl, C. (2013). Model your application domain, not your JSON structures. In Proceedings of the 22nd international conference on World Wide Web companion, (pp. 1415-1420).
Lave, J., & Wenger, E. (1991). Situated learning: Legitimate peripheral participation. Cambridge university press.
Leavitt, N. (2009). Is cloud computing really ready for prime time? (Vol. 42). Computer.
Lewis, G. (2012). The Role of Standards in Cloud-Computing Interoperability. Software Engineering Institute, 682.
Life (2014). Retrieved from http://secondlife.com/
Linden, G., Smith, B., & York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. Internet Computing, IEEE, 7(1), 76-80.
Liu, M., Kang, J., Cao, M., Lim, M., Ko, Y., Myers, R., & Schmitz Weiss, A. (2014). Understanding MOOCs as an emerging online learning tool: Perspectives from the students. American Journal of Distance Education, 28(3), 147-159.
LONCAPA. (2013). The LearningOnline Network with CAPA. Retrieved from http://www.lon-capa.org/
LRN. (2014). .LRN Website. Retrieved 2014, from www.dotlrn.org
Mackness, J., Mak, S., & Williams, R. (2010). The ideals and reality of participating in a MOOC. . Proceedings of the 7th International Conference on Networked Learning , 266-274.
Malik, M. (2009). Cloud Learning Environment - What it is? EduBlend.
Manga (2011). Retrieved from www.faceyourmanga.com
Martin, F. (2012). Will Massive Open Online Courses Change How We Teach? Communications of the ACM, pp 26-28.
Mayes, T., & De Freitas, S. (2004). Review of e-learning theories, frameworks and models. JISC e-learning models desk study, no 1.
Mazer, J. P., Murphy, R. E., & Simonds, C. J. (2007). I'll see you on “Facebook”: The effects of computer-mediated teacher self-disclosure on student motivation, affective learning, and classroom climate. Communication Education, 56(1), 1-17.
Mazman, S., & Usluel, Y. (2010). Modeling educational usage of Facebook. Computers & Education 55, 444-453.
McAuley, A., Stewart, B., Siemens, G., & Cormier, D. (2010). The MOOC model for digital practice.
McNaught, C. (2003). Identifying the complexity of factors in the sharing and reuse of resources, in Reusing Online Resources. Kogan Page, London.
McPherson, T. (2008). Digital Youth, Innovation, and the Unexpected. The John D. and Catherine T. MacArthur Foundation Series on Digital Media and Learning. Cambridge, MA: The MIT Press.
MetroStore (2014). MetroStore Scanner. Retrieved 9 2014, from http://metrostorescanner.com/
Mikroyannidis, A. (2012). A semantic framework for cloud learning environments. In L. Chao, Cloud Computing for Teaching and Learning: Strategies for Design and Implementation (pp. 17-31). Hershey, PA: IGI Global.

Mikroyannidis, A., Schmitz, H.-C., & Hernández, R. (2012). Proceedings of the 1st International Workshop on Cloud Education Environments, Vol-945, CEUR Workshop Proceedings ISSN 1613-0073.
MindMeister (2014). MindMeister Tool. Retrieved 04 2014, from http://www.mindmeister.com
Mitra, N., & Lafon, Y. (2003). Soap version 1.2 part 0: Primer. W3C recommendation, 24, 12.
MVC (2014). Retrieved 2014, from http://en.wikipedia.org/wiki/Model–view–controller
Moen, W. E., & McClure, C. R. (1994). The Government Information Locator Service (GILS): Expanding Research and Development of the ANSI/NISO Z39. 50 Information Retrieval Standard.
Morales, M., Hernández, R., Barchino, R., & Amelio, J. (2014). MOOC Using Cloud-based Tools: a Study of Motivation and Learning Strategies in Latin America. International Journal of Engineering Education. In review process.
Mott, J. (2010). Envisioning the post-LMS era: the Open Learning Network. Educause Quarterly, 33(1), 1-9.
Nejdl, W., Wolf, B., & Qu, C. (2002). EDUTELLA: a P2P networking infrastructure based on RDF. Proceedings of the 11th international conference on World Wide Web, pp. 604-615.
OECD. (2007). New millennium learners: initial findings on the effects of digital technologies on school age learners.
O'Reilly, T. (2005). Spreading the Knowledge of Innovators. What is web, vol. 2.
OOP. (2014). Retrieved 2014, from http://en.wikipedia.org/wiki/Object-oriented_programming
Olmedilla, D., Saito, N., & Simon, B. (2006). Interoperability of Educational Systems. Editorial of Special Issue. Educational Technology & Society, 9(2), 1-3.
One Drive. (2014). Retrieved from https://onedrive.live.com/
OpenApp. (2012). OpenApp Specification. Retrieved from http://sourceforge.net/apps/mediawiki/role-project/index.php?title=Library:Open_Application
OpenLearn. (2014). The Open University. Retrieved from http://www.open.edu/openlearn/
OpenScout. (2012). OpenScout Consortium. Retrieved from http://www.open.edu/openlearn/
OpenSocial. (2014). OpenSocial. Retrieved from A common Web API for social applications: https://developers.google.com/opensocial/
Openwonderland. (2014). Retrieved from http://openwonderland.org/
OSQA. (2014). OSQA. Retrieved from www.osqa.net
Padlet. (2014). Retrieved from http://padlet.com/
Papaioannou, J. (n.d.). Blocks. Retrieved from Moodle: https://docs.moodle.org/dev/Blocks
Pardo, A., & Kloos, C. (2009). Combining Web 2.0 technology and problem-based learning in a blended learning environment. Engineering Education, vol. 19, pp. 222-231.
Pardo, A., & Kloos, C. (2011). Stepping out of the box: towards analytics outside the learning management system. Proceedings of the 1st International Conference on Learning Analytics and Knowledge, 163-167.
Pavlov, I. (1927). Conditioned reflexes. London: Oxford University, Press.

PearlTrees. (2014). PearlTrees MindMap. Retrieved 9 2014, from http://www.pearltrees.com/

Pedró, F. (2009). New Millennium Learners in Higher Education: Evidence and Policy Implications. Paris: Centre for Educational Research and Innovation (CERI) Directorate for Education, OECD.

PHP. (2014). Retrieved 2014, from PHP: http://php.net

Pintrich, P., & García, T. (1993). Intraindividual differences in students' motivation and self regulated learning. German Journal of Educational Psychology,, 7, 99-107.

Pintrich, Smith, Garcia, & McKeachie. (1991). A manual for the use of the motivated strategies for learning questionnaire (MSLQ). Ann Arbor, MI: University of Michigan.

Pisutova, K. (2012). Open Education. Proc. of the 10th IEEE International Conference on Emerging eLearning Technologies and Applications (ICETA), November 8-9 2012, Stará Lesná, Slovakia, , 297-300.

PostgreSQL. (2014). PostgreSQL. Retrieved 2014, from http://www.postgresql.org

Prezi. (2014). Retrieved from http://prezi.com/

Prieto, L. P., Dlab, M. H., Gutiérrez, I., Abdulwahed, M., & Balid, W. (2011). Orchestrating technology enhanced learning: a literature review and a conceptual framework. International Journal of Technology Enhanced Learning, 3(6), 583-598.

Rentsch, T. (1982). Object oriented programming. ACM Sigplan Notices, 17(9), 51-57.

Roblyer, M. D., McDaniel, M., Webb, M., Herman, J., & Witty, J. V. (2010). Findings on Facebook in higher education: A comparison of college faculty and student uses and perceptions of social networking sites. The Internet and Higher Education, 13(3), 134-140.

Rodriguez, O. (2013). The concept of openness behind c and x-MOOCs (Massive Open Online Courses). Open Praxis, 5, 1, pp. 67–73.

Rogers, E. (2003). Diffusion of innovation. New York: Free Press.

ROLE (2013). Retrieved from http://role-widgetstore.eu

Roschelle, J., & Teasley, S. D. (1995). The construction of shared knowledge in collaborative problem solving. In Computer supported collaborative learning (pp. 69-97). Springer Berlin Heidelberg.

Rowntree, D. (1995). Teaching and learning online: a correspondence education for the 21st century? British Journal of Educational Technology, 26(3), pp 205-215.

Ryberg, T., & Larsen, M. C. (2008). Networked identities: understanding relationships between strong and weak ties in networked environments. Journal of Computer Assisted Learning, 24(2), 103-115.

Saint-Andre, P. (2004). End-to-End signing and object encryption for the extensible messaging and presence protocol (XMPP).

Saint-Andre, P., Smith, K., & Tronçon, R. (2009). XMPP: the definitive guide. " O'Reilly Media, Inc.".

Salmon, G. (2000). E-moderating: the key to teaching and learning online. London: Kogan Page.

Sarker, B. K., Wallace, P., & Gill, W. (2008). Some observations on mind map and ontology building tools for knowledge management. Ubiquity, March (2).

Schellens, T., & Valcke, M. (2005). Collaborative learning in asynchronous discussion groups: What about the impact on cognitive processing?. Computers in Human behavior, 21(6), 957-975.

Schmitz, H. C., Wolpers, M., Kirschenmann, U., & Niemann, K. (2011). Contextualized attention metadata. Human attention in digital environments, 186-209.
Schwartz, H. (2009). Facebook: The new classroom commons? The Chronicle of Higher Education . Online Article: http://chronicle.com/article/Facebook-The-New-Classroom/48575/.
Severance, C., Hanss, T., & Hardin, J. (2010). IMS Learning Tools Interoperability: Enabling a Mash-up Approach to Teaching and Learning Tools. IMS.
Shadbolt, N., Hall, W., & Berners-Lee, T. (2006). The Semantic Web Revisted. Intell. Syst. IEEE, vol. 21, no.3, pp. 96-101.
Shepherd, E. (2006). Definitions, Uses, and Benefits of Standards, Questionmark Corporation, USA. Online Assessment and Measurement Foundations and Challenges Information Science, USA.
Siemens, G., & Downes, S. (2008). Connectivism & connective knowledge. Universidad de Manitoba.
Siemens, G. (2012). MOOCs are really a platform, Elearnspace. Retrieved 06 2013, from http://www.elearnspace.org/blog/2012/07/25/moocs-are-really-a-platform/
Simon, B., Massart, D., & Duval, E. (2004). Learning Object Repositories Interoperability Framework.
Siochos, V., & Papatheodorou, C. (2011). Developing a Formal Model for Mind Maps. In First Workshop on Digital Information Management.
Soundcloud. (2014). Retrieved from soundcloud.com
Soylu, A., Mödritscher, F., Wild, F., De Causmaecker, P., & Desmet, P. (2012). Mashups by Orchestration and Widget-based Personal Environments: Key Challenges, Solution Strategies and an Application. Emerald.
Sporny, M., Kellogg, G., & Lanthaler, M. (2013). JSON-LD 1.0-A JSON-based Serialization for Linked Data. W3C.
Spurny, M., Kellogg, G., Lanthaler, M., & Lindström, N. (2014). W3C. Retrieved from JSON-LD 1.0: http://www.w3.org/TR/json-ld/
STEM. (2012). Retrieved from Technology Outlook for STEM+ Education 2012-2017. Retrieved from http://www.nmc.org/pdf/2012-technology-outlook-for-stem-education.pdf
Surowiecki, J. (2004). The wisdom of crowds: why the many are smarter than the few and how collective wisdom shapes business, economies, societies and nations. New York: Little Brown.
Lassila, O., & Swick, R. R. (1998). Resource Description Framework (RDF) model and syntax specification. Available: http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/.
Symfony (2014). Symfony: High Performance PHP Framework for Web App. Retrieved 03 2014, from http:// symfony.com
Telescopio (2014). Telescope, MOOC's. Website Galileo University. Retrieved 06 2014, from http://telescopio.galileo.edu
Tseng, S. C., & Tsai, C. C. (2010). Taiwan college students' self-efficacy and motivation of learning in online peer assessment environments. The Internet and Higher Education, 13(3), 164-169.
Vardi, M. (2012). Will MOOCs destroy academia? Commun. ACM 55, 11, 5-5.
Venkatesh, Morris, & Davis. (2003). User acceptance of information technology: toward a unified view. MIS Quarterly,, 27(3), 425e478.

Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. Management science, 46(2), 186-204.
Venkatesh, V., & Bala, H. (2008). Technology acceptance model 3 and a research agenda on interventions. Decision sciences, 39(2), 273-315.
Verbat, K., Duval, E., Klerkx, J., Govaerts, S., & Santos, J. (2013). Learning analytics dashboard applications. Americal Behavioral Scientist, 57(10), pp. 1500-1509.
Vitvar, T., & Musser, J. (2010). ProgrammableWeb. com: Statistics, trends, and best practices. In Keynote of the Web APIs and Service Mashups Workshop at the European Conference on Web Services.
Vogten, H., Martens, H., Nadolski, R., Tattersall, C., Van Rosmalen, P., & Koper, R. (2006). CopperCore Service Integration-Integrating IMS Learning Design and IMS Question and Test Interoperability. In Advanced Learning Technologies, 2006. Sixth International Conference on (pp. 378-382). IEEE.
Vygotsky, L. S. (1978). Mind and society: The development of higher mental processes.
W3C_SOP. (2014). W3C Same Origin Policy. Retrieved 03 2014, from http://www.w3.org/Security/wiki/Same_Origin_Policy
Wegerif, R. (2007). Dialogic education and technology: Expanding the space of learning. Computer-Supported Collaborative Learning Series.
Wenger. (1998). Communities of practice: Learning, meaning, and identity. Cambridge university press.
Wenger, E., McDermott, R. A., & Snyder, W. (2002). Cultivating communities of practice: A guide to managing knowledge. Harvard Business Press.
XBlock (2014). XBlock. Retrieved from https://github.com/edx/XBlock
XML-RPC. (2014). Remote Procedure Call. http://xmlrpc.scripting.com/default.html.
Xtranormal. (2014). Retrieved from http://www.xtranormal.com
Yang, D., Sinha, T., Adamson, D., & Rose, C. P. (2013, December). Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In Proceedings of the 2013 NIPS Data-Driven Education Workshop (Vol. 10, p. 13).