FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Representation of virtual choreographies in learning dashboards of interoperable LMS analytics

Diogo Alexandre Ramos Marques Costa

U.PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Master in Software Engineering

Supervisor: Prof. Fernando José Cassola Marques

Second Supervisor: Prof. Leonel Caseiro Morgado

Second Supervisor: Prof. António Fernando Vasconcelos Cunha Castro Coelho

October 11, 2023

Representation of virtual choreographies in learning dashboards of interoperable LMS analytics

Diogo Alexandre Ramos Marques Costa

Master in Software Engineering

Approved in oral examination by the committee:

President: Prof. Ana Cristina Ramada Paiva
Referee: Prof. Ricardo Alexandre Peixoto Queirós
Referee: Prof. Fernando José Cassola Marques

October 11, 2023

Abstract

Learning management systems (LMS) collect a large amount of data from user interaction, and it isn't easy to analyze this data in a reliable and context-independent manner. This research seeks to understand how virtual choreographies can be represented in interoperable LMS analytics dashboards. In order to gain a better understanding of the problem, this objective has been divided into three sub-goals: determining which interactions can be gathered from LMS contexts, identifying virtual choreographies from LMS logs, and representing virtual choreographies in learning dashboards. To achieve these objectives, we first conducted a Systematic Literature Review to comprehend the behaviors and interactions other authors have researched in LMS contexts. Then, by applying these findings to this dissertation's case study, a methodical procedure for extracting valuable choreographies from the logs was outlined. The Design Science Research methodology was then applied to transforming logs into virtual choreographies and their representation in learning dashboards. It was implemented two services: one responsible for identifying virtual choreographies from data logs and transforming the logs into statements, recipes, and choreographies, following xAPI specification elements; and the other translates the information from the backend service into dashboard visualizations, allowing the user to view representations for statements, recipes, choreographies, and various visualizations. These artifacts provide a new flexible and cost-efficient solution for the identification of virtual choreographies, thereby facilitating the widespread adoption of their use.

Keywords: virtual choreographies, xAPI, LMS, data analytics, learning analytics dashboard

Acknowledgments

First and foremost, I would like to thank my advisors, Professor Fernando Cassola and Professor Leonel Morgado, for the opportunity to work on this research. The experience would have been more difficult without your guidance, patience, motivation, and support during the entire process.

I want to thank all my friends and colleagues for their support and the great moments we shared. From all the fun to all the hard work that we accomplished together, all of the memories that I would not trade for anything; without all of you, this ride would not have been as enjoyable and memorable.

I can't thank Sofia enough, my ride or die, for your love, unconditional support, belief in my dreams, for never letting me give up, and most importantly, for being my rock. Thank you very much!

My heartfelt gratitude goes to my grandparents, Abel and Ilda, for being my role models, inspiring me to strive for more, always encouraging me, believing in me, and always having my back. I'm sure you'd be proud of this work, avó!

Last but not least, I want to thank my family for their unwavering support and belief in me. Huge thanks!

Diogo Marques Costa

"You can never cross the ocean
until you have the courage to lose sight of the shore."

Christopher Columbus

Contents

List of Figures

List of Tables

List of Listings

List of Abbreviations

LMS	Learning Management System
LAD	Learning Analytics Dashboard
RO	Research Objectives
SCORM	Sharable Content Object Reference Model
SLR	Systematic Literature Review
SoS	Survey of Surveys
MOOCs	Massive Open Online Courses
LDS	Laboratório de Desenvolvimento de Software
DSR	Design Science Research
RE	Requirements Engineering
CRISP-DM	Cross-Industry Standard Process for Data Mining

Chapter 1

Introduction

1.1 Context

Information is more accessible than ever due to the digitization of systems across all industries. This digitization improves teaching and learning, particularly in higher education, where students use online platforms to complete their assignments [7]. Learning Management Systems (LMS) facilitate this type of interactive learning between teacher and student by providing features that can be applied in a learning setting [1]. This type of e-learning platform enables the collection and storage of raw data [35]. Educational data can originate from a variety of interactions and levels of granularity, making analysis challenging [98]. With this amount of available data, it is essential to extract useful information. Learning Analytics Dashboards (LAD) analyze and provide data visualizations to deliver insightful knowledge [104].

With the rise of big data, finding innovative ways to analyze and interpret this data is crucial. LAD are a popular topic because it provides significant data insights. For this, they can use a variety of data representations and analysis methods. The most common representations are bar charts, line graphs, and tables, but these can vary depending on the domain and the requirements of the stakeholders [104]. However, these dashboards have known challenges. Sarikaya *et al.* [102] mention the lack of end-user flexibility. One example is the storytelling capabilities because most of the dashboards only capture the data rather than the semantics of the data. Considering this, virtual choreographies are a novel method for describing interactions between actors that are also platform-independent [109], making it a potential candidate for integration into a LAD.

This research aims to understand how virtual choreographies can be represented in learning dashboards of interoperable LMS analytics. This will introduce a new method for gaining data insights, and because it is platform agnostic, it will be applicable in a variety of scenarios.

1.2 Motivation

LMS have become indispensable tools for educational institutions and e-learning platforms. The data these platforms gather are powerful since they can offer valuable insights to stakeholders, such as learners' interactions, engagement, and performance. But in order to reach the point where the data is valuable in decision-making, a long process has to be made.

Being able to extract valuable information from these platforms is difficult by itself, and adding to the fact that there is currently a lack of interoperability between them, turns this theme into an exciting data analysis and software engineering project.

Nonetheless, adding the novelty around the virtual choreographies makes this process of identifying and displaying meaningful information to aid decision-making really interesting. Because it is an expanding field, being at the forefront of the adoption and improvement of the usage of virtual choreographies is what sparks my interest in researching this topic.

1.3 Problem

The problem that this dissertation aims to solve is how virtual choreographies can be represented in learning dashboards of interoperable LMS data. By analyzing the problem deeply, we can divide it into three steps:

- Step 1: Understand which interactions can be gathered from LMS contexts;
- Step 2: Identification of virtual choreographies from LMS log data;
- Step 3: Produce learning dashboards with the identified choreographies.

By dividing the problem into three distinct problems is possible to understand all the pieces needed to achieve the end goal. To solve the first step, we will need to conduct a literature review to identify the interactions and behaviors being identified in the LMS context in the scientific literature. The second step will necessitate the most amount of effort since it divides itself into two sub-steps: 1) identify which choreographies are worth identifying in log data; and 2) how can we make this identification automatic taking into consideration that it should be a platform agnostic approach. Lastly, we have step three, which will produce learning dashboards with the identified choreographies.

Since steps one and two take a significant amount of work to solve, they will be the project's focus. However, simple visualizations will also address step three as proof of concept.

1.4 Objectives

From the dissertation problem, it is possible to extract some Research Objectives (RO):

RO1 Conduct a comprehensive analysis to ascertain the diverse spectrum of interactions and behaviors within LMS contexts, which have been investigated in the existing literature.

RO2 Formulate a methodical approach to delineate valuable choreographies to be extracted from the data logs within LMS contexts.

RO3 Design and implement an efficient solution that enables the identification of virtual choreographies from the data logs present in LMS environments.

R04 Design and develop a dashboard that showcases potential visualizations using the output generated from the identification of virtual choreographies.

The first objective `RO1` will address the first step of the problem. Then `RO2` and `RO3` will compose the step 2. Lastly, the `RO4` will address the last step. By fulfilling these objectives the problem will be tackled as well.

1.5 Document Structure

This dissertation comprises ten chapters, and the first is this introduction. Then, Chapter 2 lays the groundwork for the entire dissertation by defining critical concepts. Chapter 3 entails the conducted literature review of the state of the art. Then, in Chapter 4, a detailed understanding of the case study is presented, and a method to identify valuable choreographies in logs is devised. Chapter 5 explains the methodology followed, Design Rocket Science, and how it was applied to this work. The problem is then thoroughly analyzed in Chapter 6, which is followed by a discussion of how the SLR results apply to the problem and documentation of the Requirements Engineering process. After it, we have Chapter 7 where the non-functional prototypes for each artifact and an overview of the software architecture design are presented. Then, Chapter 8 details the decisions made during the implementation phase and the approach taken to achieve the objectives. Subsequently, Chapter 9 discusses the research findings in the context of the goals and use cases, and the dissertation is examined against the research objectives defined in the introduction. Lastly, Chapter 10 reflects on the results achieved, identifying the areas of success, potential development, and limitations.

Chapter 2

Background

This chapter will define some fundamental concepts that are essential for understanding the topics covered throughout the dissertation. The concepts were identified during the exploratory phase of the literature review, which arose from the identification of the research's concept, context, and perspective, as discussed in Section 3.1. Additionally, an introduction to Clean Architecture is given due to being the architecture followed in the design and implementation of the prototype.

2.1 Virtual Choreographies

Human behavior is a well-known research field of study. It is typically depicted as a sequence of behaviors, in this case, human, that can be designated a behavior [97]. Human behavior is a concept that incorporates various dimensions, being cognitive, psychomotor and socio-affective that are dependent between them [59]. Hutchison [47] identified three components of the concept of human behavior: 1) person, 2) environment and 3) time, making it a complex and multidimensional concept.

Considering the complexity of human behavior, it became necessary to develop models for its research. The US Department of Defense presented the Human Behavior Representation (HBR) [59]. HBR's primary objective is "to build a computational model of a human being to achieve embodied goals and predict performance" [107] by analyzing variability in behavior due to variations in the person, situation, or both.

Virtual choreographies arise as a simplified approach to the use of frameworks such as HBR [21]. A choreography depicts a set of acts (behaviors, interactions and occurrences) performed by actors, with well-defined goals and possible boundaries [109]. These actors can be controlled by humans or computers, allowing for greater versatility when employing them [110]. According to Cassola *et al.* [21], virtual choreographies must allow multi-user operations and be platform, actor and scenario independent.

Due to the rapid evolution and growth of the software market, software was quickly becoming obsolete, and the activities could not be reused since they were deeply embedded in the technological foundation [67]. With choreographies, it is feasible to apply the same choreography to a variety of contexts, as demonstrated by Lacet *et al.* [60]. These authors present a unique choreography for a comic strip platform, a three-dimensional castle, and two distinct game rooms. Apart from this, Cassola *et al.* [21] mentions that various scenarios can be choreographed such as aircraft maintenance, industry simulation, and disaster simulation.

Lacet *et al.* [61] proposed a method that permits the reuse of digital story choreographies independent of the platform for which they were developed. This proposal is based on the xAPI specification, representing virtual choreographies as statements. These statements may contain actors, verbs, objects, places, and context. The xAPI specification also includes the recipes, which consist of a list of valid xAPI elements representing the elements that can be used in the statements that follow the recipe. In Figure 2.1, we have a representation in xAPI natural language of each of the described concepts; in the recipe, we have the list of available ingredients; next, we have statements that were created with the recipe ingredients; and finally, we can join interactions or statements from John to create a choreography in which John first viewed the coliseum and then visited it.

Figure 2.1: Example of recipe, statements, and choreography in xAPI natural language

With a comprehension of the key xAPI concepts, we can concentrate on the proposed method by Lacet *et al.* [61], which this dissertation will further adapt to address the specified design issue. Figure 2.2 depicts the methodology that consists of four essential components:

- Label 1, Choreography: represents the group of choreographies comprising valid statements defined by the history recipe and which can be reproduced on multiple platforms.
- Label 2, History recipe: represents the available elements to create the history that the choreographies will represent, or narrative semantic spaces, as described by Lacet *et al.* [61]
- Label 3, Platform recipe, denotes the available platform components that can be used to reproduce the history.
- Label 4, Matcher: is the process's engine, responsible for translating choreography recipes to platform recipes and vice versa. Thanks to these mappings, the choreography can be automatically adapted to any platform.

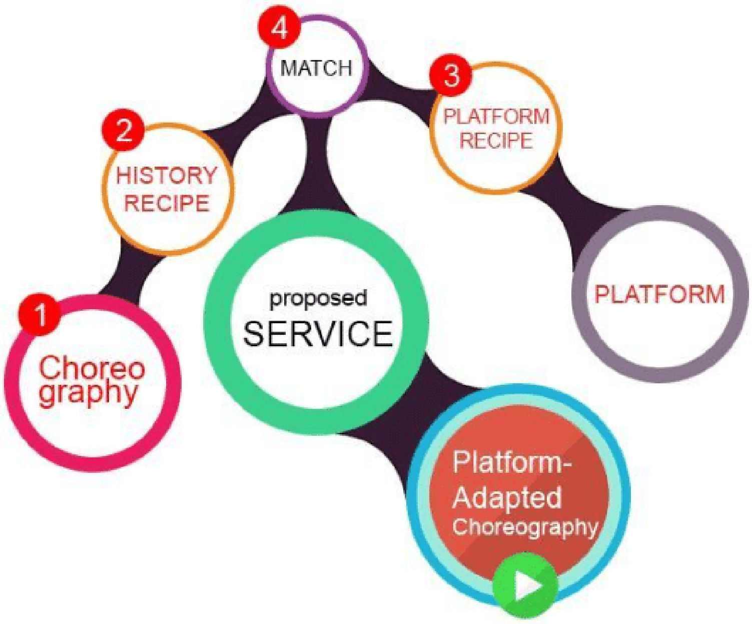

Figure 2.2: Preserving story choreographies across multiple platforms methodology [61]

This methodology provides virtual choreographies with a longer lifespan and the ability to adapt the same choreography to various contexts, as demonstrated by both studies by Lacet *et al.* [61, 60]. Nonetheless, this strategy assumes that the list of available elements and choreographies is known in advance, which is only sometimes the case. For instance, when the amount of data to analyze is enormous and unknown before the analysis, like a Moodle log file, it is difficult to define which elements are available beforehand, making it impossible to construct the recipe.

2.2 Learning Dashboard

Since the turn of the century, learning analytics has arisen as a significant topic of study in technology-enhanced learning [37]. In addition, the emergence of big data, which is typically described as a dataset that cannot be processed by conventional IT methods in a reasonable amount of time, has necessitated new approaches to information technology [23]. The widespread implementation of LMS in learning contexts leaves educational institutions with ever-growing datasets which eventually raises the question of how these data may be utilized to create value [37].

The 1st International Conference on Learning Analytics provides one of the literature's consensus definitions of learning analytics: "Learning analytics is the measurement, collection, analysis, and reporting of data about learners and their contexts, for the purposes of understanding and optimizing learning and the environments in which it occurs" [65].

Learning analytics dashboards are responsible for delivering valuable visualization systems that analyze and present data about learners in an educational context in order to provide relevant insights from learning analytics [104]. These dashboards are becoming more widespread in higher education and provide significant insights to a variety of stakeholders, including: 1) students [14],

2) teachers [6], 3) institutions [17], and 4) multiple stakeholders [44]. This demonstrates that the value delivered by LAD is subject to a number of scenarios and contexts.

According to Schwendimann *et al.* [104], a LAD is “a single display that aggregates different indicators about learner(s), learning process(es) and/or learning context(s) into one or multiple visualizations”. A single display may not necessarily represent a single visualization. Typically, dashboards offer multiple visualizations for an indicator. In their review, Schwendimann *et al.* [104] determined that bar charts, line graphs and tables were the most popular visualizations based on different target audiences (teachers, students, administrators and researchers).

Figure 2.3 depicts the Learning Analytics Cycle presented by Clow [25]. The cycle begins with the learners being directly or indirectly accountable for data production. The subsequent phase, data gathering, is typically performed by LMS. The third phase involves metrics derived from data processing and visualizations, these visualizations are commonly known as LADs. The final step is interventions, in which stakeholders take action based on the analytics offered by LAD. This cycle offers a comprehensive breakdown of a typical Learning Analytics approach.

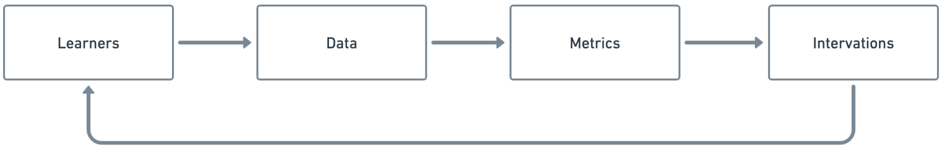

Figure 2.3: Learning Analytics Cycle, adapted from Clow [25]

2.3 LMS and SCORM

LMS enable teachers and students to communicate and collaborate remotely [79]. This communication can be synchronous or asynchronous, allowing for the use of various techniques. According to Kasim and Khalid [53], an LMS gives students platform-independent access to online learning content (e.g. web browsers, mobile). Similarly, Lonn and Teasley [66] define LMS as web-based systems that enable collaboration between students and instructors through the exchange of materials, the submission of assignments and online communication. Communication is an essential aspect of an LMS, and the available choices include forums, chat rooms and others.

Aldiab *et al.* [5] identified four relevant advantages of LMS: 1) absence of a physical location, 2) accessibility, 3) attractive environments, and 4) integration. Due to the absence of a requirement for a physical location, distant education programs are feasible, and distance learning is achievable [55]. The second aspect is that it is easily accessible by any device (e.g., tablet, phone, computer) and via browser or official applications, making it available to anybody with device and Internet access. The third advantage is an attractive environment, which makes it easier to implement methods such as gamification, which can boost student engagement [111]. The final benefit is that the majority of LMS is easily extendable, meaning that any missing feature may be developed or integrated via a marketplace (e.g. Moodle) [40].

There are numerous possibilities for LMS, ranging from open source (e.g., Moodle, Sakai) to commercial (e.g., Blackboard, SumTotal). With such a large number of solutions, having a standard that could facilitate content reuse between LMS would be beneficial. Sharable Content Object Reference Model (SCORM) is one of the *de facto* standards for portability. It brings together the concept of learning objects, which are packages of study materials and activities that may be reused across LMS that are compatible with SCORM objects [16]. This portability decreases the expenses and burdens associated with modifying content in LMS. The standard is composed of the following characteristics: accessibility, adaptability, affordability, durability, interoperability, and reusability [83].

Sharable content object (SCO) is the smallest unit in SCORM and it contains obtainable learning content, such as an HTML file, an image, or javascript, that can be used in another course [16]. Figure 2.4 depicts the composition of a sample SCORM package and its interaction with an LMS.

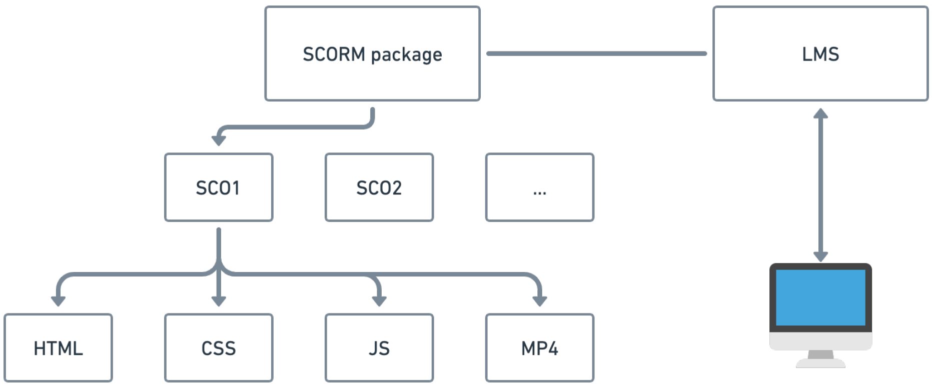

Figure 2.4: SCORM package overview

2.4 Interoperability

Interoperability is a wide and complex topic with numerous definitions. Interoperability is described in the IEEE standard glossary [48] as "The capacity of two or more systems or components to communicate information and use the information that has been exchanged". Ford *et al.* [38] researched the most frequent definition of interoperability and discovered that the US Department of Defense's (DoD) definition was the most popular: "The ability of systems, units, or forces to provide services to and accept services from other systems, units, or forces and to use the services so exchanged to enable them to operate effectively together". The DoD definition of interoperability is more generic which means it aggregates more levels of interoperability.

According to Rezaei *et al.* [96], interoperability has four levels. Technical interoperability is, first and foremost, the capacity of systems to communicate data using common technical standards and protocols, notably communication protocols and the related infrastructure. The second level of interoperability is syntactic, which focuses on transferring data in a standard format and with the same structure. The next level of interoperability is semantic interoperability, which is typically connected with the capacity to comprehend the meaning of shared data. The organizational level, which relates to effective communication between organizations and their systems, is the final level of interoperability and is dependent on the preceding levels.

Interoperability is difficult to achieve because every system has unique properties and requirements. However, if achieved, it can result in financial benefits such as reduced transaction costs, improved operational efficiency and enhanced service quality, among others identified by Carmagnola *et al.* [19].

2.5 Graphs

With the rise of big data, value extraction has become crucial. Since there is a relationship between the data in most situations, graphs can be used to potentially obtain value from it [54]. It is known that graphs are used in social networks by taking into account the links that emerge from them. By examining the graph, it is possible to classify user profiles, detect users' communities, and even make predictions about possible interactions [18].

Even though the use of graphs has many benefits, a common issue is that the graph can grow indefinitely and demands an excessive amount of computer resources, which can become unbearable [18]. By using optimizations, systems such as GraphX, developed by Gonzalez *et al.* [43], make the process of generating graphs more efficient.

According to Stigall and Tasar [112], a graph shows the presence or absence of relations between sets of items. Graphs are commonly separated between undirected and directed types. Graph or undirected graph links do not have a direction and can transverse in both directions. Alternatively, there is a directed graph in which the edges must be directed [112]. Graphs can also be weighted by assigning a numeric value to each edge [108]. On Figure 2.5 are examples of undirected, directed and weighted graphs respectively.

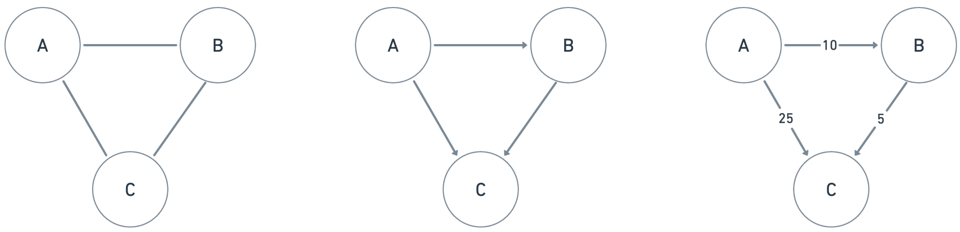

Figure 2.5: Types of graph: undirected, directed, and weighted

2.6 Clustering

Grouping patterns or objects is a common task, whether it is in daily life or in a specific science area. Clustering is a type of unsupervised classification, which means that there is no label assigned to a pattern [103]. According to Jain *et al.* [50] clustering is useful in a set of cases such as exploratory pattern analysis, grouping, decision-making and machine learning situations. On the other hand, it is considered more difficult than supervised classification since the data is not labeled [50].

Choosing a suitable clustering algorithm requires deliberation, as each has its own benefits and drawbacks. Usually, clustering techniques are separated into hierarchical and partitioning-based methods [50].

According to Rai and Shubha [95], hierarchical clustering algorithms aim to create a clustering hierarchy. More specifically, Chatziafratis *et al.* [22] states that it displays a dataset as a binary tree. Clustering is an iterative technique that divides patterns using a top-down or bottom-up strategy [103]. The top-down technique is referred to as divisive, whilst the bottom-up strategy is known as agglomerative. The algorithms ROCK, COBWEB and QROCK are examples.

According to Saemi *et al.* [99], partitioning-based clustering separates the data into k categories and then clusters the data using an iterative process. Based on the strategies employed to generate clusters, this approach can be classified as hard/crisp clustering, fuzzy clustering, or mixture resolving clustering [2]. Among the clustering methods based on partitioning are k-means, CLARA and fuzzy c-means.

2.7 Clean Architecture

The Clean Architecture, as proposed by Robert C. Martin [74], is a synthesis of various system architecture concepts, incorporating elements from Hexagonal Architecture [27], DCI [30], and BCE [49]. Despite their differences, these architectures share a common goal: the separation of concerns [74]. They accomplish this by dedicating at least one layer to business rules and a second to user and system interfaces.

As depicted in Figure 2.6, Martin [74] suggests combining all architectures into a single, unified construct. The purpose of defining distinct layers is to ensure a separation of concerns, facilitate testing at multiple levels of abstraction, and maintain a clear demarcation between business rules and specific implementation details. Martin [74] states that systems adhering to clean architecture will exhibit the following characteristics:

- Framework Independence: The idea is that the system is not entwined with the framework. Instead, with the separation of concerns, isolating business rules and employing frameworks or libraries as auxiliary tools is feasible.
- Testability: Business rules can be tested without reliance on external elements such as User Interfaces (UI) or databases.
- UI Independence: User interfaces, which evolve rapidly and possess distinct requirements, can be modified from a web to a CLI, without necessitating alterations in the business layer.
- Database Independence: Changing databases becomes viable at any juncture, as the business rules are not tied to specific databases.

- External Agency Independence: External interfaces do not directly interact with the business layer, making them independent of any library or tool.

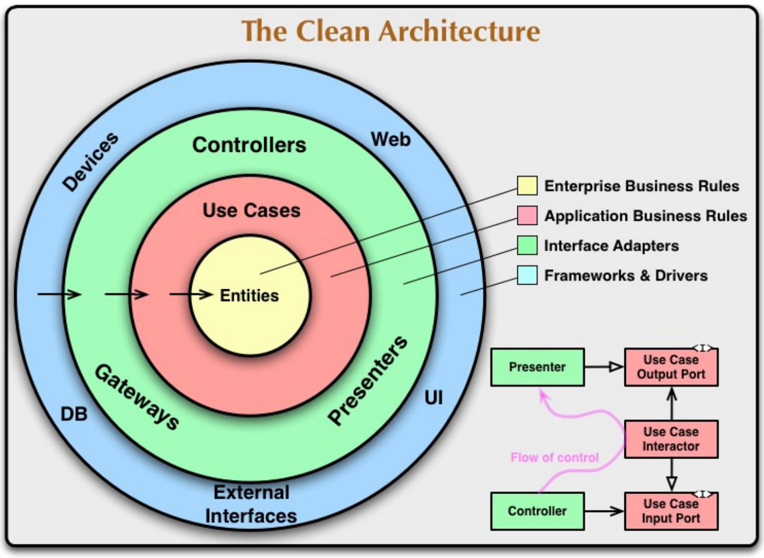

Figure 2.6: Clean architecture by Martin [74]

Given these benefits, this architecture will be used to build the backend layer. This decision will improve the backend's robustness and adaptability to emergent requirements without requiring major changes to the business rules.

Chapter 3

State of the Art

Learning Analytics is a broad topic of study. If a Systematic Literature Review (SLR) is undertaken without a well-defined scope, the results would be limitless and the literature review will lack scientific value [77]. Consequently, a protocol outlined by Morgado and Beck [77] was used to carry out this literature review. This protocol synthesizes the two primary protocols [62, 89] used for systematic scoping reviews, resulting in a more robust protocol designed to maximize the potential for influence on research literature [77]. The timelines for the development of this literature review would not be satisfied by doing such a process, so in order to still define a meaningful scope, only the exploratory phase of the protocol was undertaken. With this phase's artifacts, the keywords, a systematic literature research was conducted according to the guidelines of Kitchenham and Charters [57].

Then, from a global perspective, this literature review consists of a survey of surveys, a step from the exploratory phase, in order to establish the scope. Then, a literature review with a well-defined scope is conducted to answer the research questions. Figure 3.1 illustrates an overview of the process.

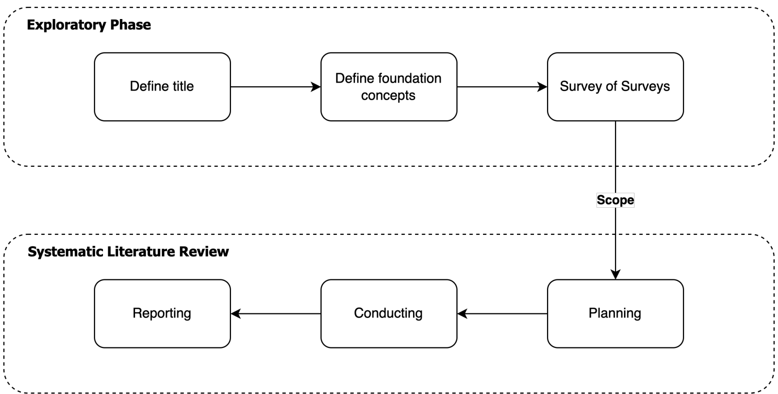

Figure 3.1: Methodology overview including exploratory phase and SLR

3.1 Exploratory Phase

Defining the scope of a research project in a wide field like Learning Analytics might be difficult. Using an exploratory strategy such as scoping reviews, it is possible to understand the current state of learning analytics and to discover significant subjects and potential gaps in the literature [29]. According to Arksey and O'Malley [8], there are two different ways to use scoping reviews: 1) as a part of the review process that results in a systematic literature review, and 2) as a comprehensive approach that results in research findings on its own.

Considering the time-frame available to conduct this research and the need for a well-defined scope, the exploratory phase of Morgado and Beck's [77] suggested protocol provides that. By employing only this phase, we will be able to reduce the scope of the task without doing a whole rigorous review.

The exploratory phase consists of three phases. The first step is "Define title", which should reflect the concept, context and perspective of the research. The second step is "Define foundational concepts", which seeks to understand the current state of the key concepts that can be inferred from the previous step. The final step is "Define theoretical background", which involves conducting a Survey of Surveys (SoS) in order to define the path to conducting the literature review.

3.1.1 Define Title

It is essential to identify the concept, context and perspective of the review while defining a title. The results for each definition are:

- **Concept:** Represent virtual choreographies in learning dashboards;
- **Context:** Analytics from LMS including SCORM objects;
- **Perspective:** Collecting and displaying interoperable data independent of the source LMS and SCORM platforms.

Ultimately resulted in the temporary title *"Representation of virtual choreographies in learning dashboards of interoperable LMS and SCORM analytics"*.

3.1.2 Define Fundamental Concepts

This step is essential for understanding how the research community discusses the concepts, identifying potential keywords and identifying significant works. These concepts are virtual choreography, learning dashboard, learning management system, sharable content object reference model, and interoperability.

This step's findings can be found in the Background Section 2, where their definitions are provided.

3.1.3 Define Theoretical Background

This chapter is where the SoS is conducted, with the result being the current gap and related work concerning the predefined concept, context, and perspective.

3.1.3.1 Sources and Search Strings

Based on the definition of fundamental concepts, a list of keywords was identified for each concept. Each concept's keywords and search terms are included in Table 3.1.

Table 3.1: First iteration of keywords and synonyms (SoS)

Virtual choreographies	**Learning dashboard**	**LMS and SCORM**	**Interoperability**
virtual choreographies	learning dashboard	learning management systems	interoperability
virtual choreography	learning analytics	LMS	independence
virtual representations	dashboards	virtual learning environments	interoperate
choreographies	information visualizations	VLE	interoperable
choreography	educational dashboard	sharable content object reference model	interoperation
story choreographies	dashboard for learning analytics	SCORM	similarity
behavior patterns	learning analytics dashboard		integrate
choreography recipes	data dashboard		systems integration
	web dashboard		

Due to the fact that this is a SoS, these searches are constrained to secondary studies such as surveys or reviews.

The study for secondary studies was undertaken using the Google Scholar database, as it provides substantially more citations than Web of Science and Scopus for example [76]. Due to Google Scholar's 256-character search limit, it is impossible to include all keywords in the same search phrase. To discover the significant and relevant keywords to include in the search string, all keywords were subjected to review to determine their importance and relevance. Table 3.2 provides an example of virtual choreography keywords.

This table allows for the elimination of key terms without compromising search results. The reduced form of Table 3.1 is displayed in Table 3.3 after applying this method.

The search string outcomes for each concept are:

1. *"choreographies"* OR *"choreography"* OR *"behavior patterns"*

Table 3.2: Virtual choreography keywords execution

Search string	Result
allintitle: (“survey” OR “review”)(“virtual choreographies”)	0
allintitle: (“survey” OR “review”)(“virtual choreography”)	0
allintitle: (“survey” OR “review”)(“virtual representations”)	2
allintitle: (“survey” OR “review”)(“choreographies”)	10
allintitle: (“survey” OR “review”)(“choreography”)	46
allintitle: (“survey” OR “review”)(“story choreographies”)	0
allintitle: (“survey” OR “review”)(“behavior patterns”)	61
allintitle: (“survey” OR “review”)(“choreography recipes”)	0

2. *“* dashboard”* OR *“learning analytics”* OR *“information visualization”*

3. *“learning management systems”* OR *“LMS”* OR *“virtual learning environments”* OR *“VLEs”* OR *“SCORM”*

4. *“interoperability”* OR *“interoperable”* OR *“systems integration”*

To achieve the best outcomes, all possible combinations of these principles were tested. For this initial step, search only by title. The results are available in Appendix A.

Since the findings were unsatisfactory under the condition of only retrieving results that matched the title, all possible combinations were examined, while keeping one of the concepts on the title condition and rotating the others throughout the body of the paper. The results are available in Appendix B.

3.1.3.2 Conducting

Having defined the search strings, they were then performed against the Google Scholar database. Since it is difficult to retrieve metadata from the selected database, all search strings were con-

Table 3.3: Final keywords (survey of surveys)

Virtual choreographies	Learning dashboard	LMS and SCORM	Interoperability
choreographies	* dashboard	learning management systems	interoperability
choreography	learning analytics	LMS	interoperable
behavior patterns	information visualization	virtual learning environments	systems integration
	VLE		
	SCORM		

ducted via the academic citation extraction and analysis software Publish or Perish[1], a total of n=321 papers were retrieved.

Table 3.4: Inclusion and exclusion criteria (SoS)

Inclusion criteria
The paper addresses learning analytics
The paper addresses interoperability
The paper addresses patterns or interactions
Exclusion criteria
The paper does not answer at least one research question
Not peer reviewed
The title identifies a distinct area apart from the objectives
The paper is not written in English

After discarding the duplicates, n=256 papers remained. Following this, inclusion and exclusion criteria presented in Table 3.4 were applied by first examining the title and then the abstract, leaving n=62 papers. All papers that did not mention behaviors were eliminated, leaving n=12 papers. Because two of these last papers were published prior to 2017, they were eliminated, leaving n=10 papers. The process can be viewed in Figure 3.2, and the secondary studies that resulted are presented in Table 3.5.

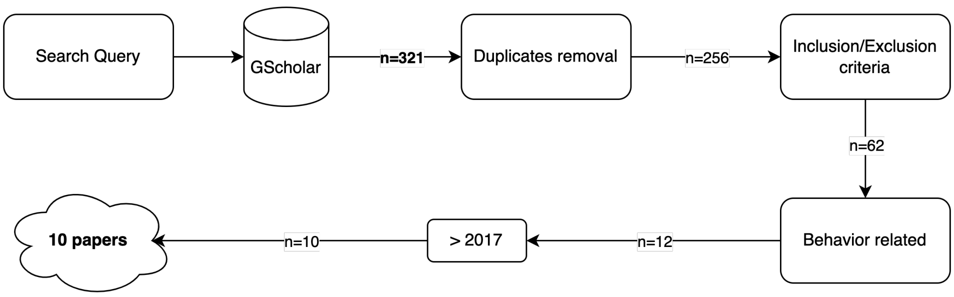

Figure 3.2: Overview of the SoS process

3.1.3.3 Thematic Analysis

The approach of thematic analysis, as presented by Vaismoradi *et al.* [115], was used to analyze the results. By breaking the knowledge of a certain issue into themes, the thematic analysis provides a systematic approach to content analysis, making it easier to understand the existing state

[1] Publish or perish: https://harzing.com/resources/publish-or-perish

Table 3.5: Survey of surveys results

Authors	Title	Year
Ramaswami *et al.*	Use of Predictive Analytics within Learning Analytics Dashboards: A Review of Case Studies	2022
Liu *et al.*	Business process collaboration using semantic interoperability: Review and framework	2021
C Romero and S Ventura	Educational data mining and learning analytics: An updated survey	2020
KC Li and BTM Wong	Personalising learning with learning analytics: a review of the literature	2020
H Aldowah *et al.*	Educational data mining and learning analytics for 21st century higher education: A review and synthesis	2019
J Moon and Z Liu	Rich representations for analyzing learning trajectories: Systematic review on sequential data analytics in game-based learning research	2019
K Mangaroska and M Giannakos	Learning analytics for learning design: A systematic literature review of analytics-driven design to enhance learning	2018
Banihashem *et al.*	Learning analytics: A systematic literature review	2018
Liu *et al.*	Learning analytics as an assessment tool in serious games: A review of literature	2017
Leitner *et al.*	Learning analytics in higher education—a literature review	2017

of knowledge and possible relationships between themes [115]. This will ultimately aid in understanding the present scope of available approaches, making it easier to determine a scope for the SLR.

First, we identified a set of questions based on the specified concept, context, and perspective. These queries are used to extract relevant units from the literature, allowing us to discover multiple responses to the same question. The identified Thematic Analysis Questions (TAQ) were:

1. *TAQ1: How are choreographies presented on a dashboard?*
2. *TAQ2: How is the data for choreographies gathered?*
3. *TAQ3: How is the data analyzed in order to identify choreographies?*

For each question, the literature was evaluated and excerpts indicating the answers were extracted. As a phase of thematic analysis, these excerpts are reduced to an atomic abstraction that translates the excerpt's meaning, so that it can fully replace the entire excerpt [115]. These abstract atomic entities are known as codes.

The codes were analyzed iteratively to identify similarities and differences, enabling the identification of code clusters. These groups are referred to as themes and serve as responses to research questions. The outcome of this procedure can be seen in Table 3.6, Table 3.7 and Table 3.8.

Table 3.6: TAQ1: How are choreographies presented on a dashboard?

Theme	**Definition**
Patterns and trends graph	Use of graphs to identify patterns and trends in different types of data
Player behavior heatmaps	Use of heatmaps to represent behavior patterns among players
Time-related multi-dimensional datasets as circular diagrams	Use of circular diagrams to visualize data that changes over time and has multiple dimensions
Spatial representations and student patterns as random walks	Use of random walks to represent patterns in data, such as student interactions
Fuzzy workflow diagrams to visualize student patterns while completing a task	Use of fuzzy workflow diagrams to visualize student patterns while completing a task
Comparisons of student patterns as bubble diagrams	Use of bubble diagrams to illustrate comparisons between patterns
Timeline visualization of choreography completion	Use timeline visualization to see choreography status

After assessing the outcomes of this analysis, a course of action and a scope were determined. To illustrate the choreographies, the topic “Use of graph for identification of patterns and trends in different data” was selected because it has the more developed literature. To analyze the data to detect patterns, the topic “Use of clustering techniques to group and classify data” was chosen since it is a trending topic for identifying patterns. Lastly, for data collection, the “Use of learning management systems (LMS) and content management systems (CMS) to record interactions and behaviors” was chosen since it is extensively utilized and there are a vast number of patterns that can be extracted from these types of platforms.

3.2 Systematic Literature Review

A systematic approach to a literature review allows for the reproducibility of the literature. An SLR is one of the most prevalent methods for this goal, and by employing it, we can ensure that our findings are consistent, robust, and transferable [57]. The guidelines from Kitchenham and Charters [57] were followed to perform the literature review.

Table 3.7: TAQ2: How is the data for choreographies gathered?

Theme	Definition
Recording interactions and behaviors with LMS and CMS	Use of LMS and CMS as tools for collecting data on student interactions and behaviors
Log files and databases to store data	Use of log files and databases to save and store data collected from LMS and CMS, as well as other sources
VLEs and e-learning platforms to collect data from multiple domains.	Use of VLEs and e-learning platforms as a way to gather data from different domains
Surveys, interviews, formal observation, and other methods to collect qualitative and quantitative data	Use of various methods, such as surveys, interviews, and formal observation, to gather both qualitative and quantitative data
Games and other interactive tools student progress and interaction	Use of data from games and other interactive tools to track student progress and interactions
Historical, biographical, and course-related data essential information	Use of various types of data, such as past student performance, demographics, or course-related information, to gather relevant information about student interactions and behaviors
Log files, eye tracking, video, and electrodermal activity data gathering	Use of various technologies and techniques to collect data about student interactions and behaviors

Table 3.8: TAQ3: How is the data analyzed in order to identify choreographies?

Theme	**Definition**
Data pre-processing techniques to clean and in usable format	Use of techniques, such as data cleaning, normalization, and feature extraction, to prepare the data for analysis
Software and tools to analyze and visualize data	Use of techniques, such as data cleaning, normalization, and feature extraction, to prepare the data for analysis
Statistical analysis to identify patterns and relationships in behavioral data	Use of various software and tools to analyze and visualize data about student interactions and behaviors
Machine learning techniques to classify and analyze patterns in data	Use of statistical techniques, such as hypothesis testing, correlation analysis, or regression analysis, to analyze behavioral data and identify patterns and relationships
Clustering techniques to group and classify data	Use of machine learning techniques, such as decision trees, neural networks, or support vector machines, to classify and analyze patterns in data
Descriptive analysis to compare different clusters	Use of clustering techniques, such as k-means, hierarchical clustering, or density-based clustering, to group and classify data
Visualized behavior-transition diagram for complex behaviors	Use of descriptive analysis, such as mean, median, or mode, to compare different clusters of data Use of visualized behavior-transition diagrams, such as state transition diagrams, to represent and analyze more complex behaviors

Since undertaking an SLR can be a complex task, we used Parsifal[2] to conduct it. Parsifal is designed to aid researchers in performing SLR within the context of Software Engineering while guiding through all the phases proposed by Kitchenham and Charters [57].

3.2.1 Planning

The planning usually confirms the necessity of a review prior to making it [57]. In this case, the necessity is already validated since a SoS was made before the scope.

Considering the scope outlined in the previous section, the primary objective of this literature review is: *Understand how graphs can be used to represent behaviors (interactions) in LMS/CMS, applying clustering techniques*.

3.2.1.1 Research Questions

To frame research questions, the PICOC (Population, Intervention, Comparison, Outcomes, Context) approach has been proposed by Petticrew and Roberts [90].

The PICOC results that arose from the objective were:

- Population: learners
- Intervention: graphs
- Comparison: typification of clustered behaviors representations
- Outcome: awareness about the learning process
- Context: LMS

With the PICOC defined, we can clearly understand the objective's essential components. By analyzing it, the following Research Questions (RQ) will be addressed:

1. *RQ1: What are the types of behaviors/interactions in LMS/CMS represented by graphs?*
2. *RQ2: Which clustering techniques are used to identify behaviors/interactions in LMS/CMS?*
3. *RQ3: How can graphs be used to represent behaviors/interactions in LMS/CMS?*
4. *RQ4: How can clustering algorithms be used to represent behaviors/interactions in LMS/CMS?*

3.2.1.2 Sources and Search Strings

Similarly to the SoS, the search strings will just be executed on Google Scholar since it provides more citations than Web of Science and Scopus for example [76].

[2]Parsifal: https://parsif.al/

It is possible to infer some keywords from the PICOC, and by performing a quick search, it is possible to extract some commonly used synonyms in the literature. The keywords and their synonyms are:

- **LMS**: CMS, content management system, e-learning platform, learning environment, learning management system, learning platform;
- **awareness**: decision-making, learning analytics;
- **clustering**: clusters, group, grouping, patterns;
- **graph**: behavior patterns, choreography, interaction pattern, interaction patterns, multimodal analytics, network visualization, node-link, rhizomatic;
- **learners**: education, educational, learning, school, student, students, trainee, trainees, training, university.

Considering Google's 256-character search limit, a combination of keywords and synonyms was utilized until the search yielded an appropriate number of results. Eight distinct search strings were generated so that the results would be more complete and there would be no risk of losing valuable results. These are the search strings:

1. (*"LMS"* OR *"CMS"*) *"learning-analytics"* (*"clustering"* OR *"patterns"*) (*"graph"* (*"interaction patterns"* OR *"behavior patterns"* OR *"choreography"*)) (*"learners"* OR *"trainees"* OR *"university"*)
2. (*"learning management system"* OR *"CMS"*) *"learning-analytics"* (*"clustering"* OR *"patterns"*) (*"graph"* (*"interaction patterns"* OR *"behavior patterns"* OR *"choreography"*)) (*"learners"* OR *"trainees"* OR *"university"*)
3. (*"learning management system"* OR *"content management system"*) *"learning-analytics"* (*"clustering"* OR *"patterns"*) (*"graph"* (*"interaction patterns"* OR *"behavior patterns"* OR *"choreography"*)) (*"learners"* OR *"trainees"* OR *"university"*)
4. (*"LMS"* OR *"CMS"*) *"learning-analytics"* (*"clustering"* OR *"patterns"*) (*"graph"* (*"interaction patterns"* OR *"behavior patterns"* OR *"choreography"*)) (*"learning"* OR *"training"* OR *"education"*)
5. (*"LMS"* OR *"CMS"*) *"learning-analytics"* (*"clustering"* OR *"patterns"* OR *"group"*) (*"graph"* (*"interaction patterns"* OR *"behavior patterns"* OR *"choreography"*)) (*"learning"* OR *"training"* OR *"education"*)
6. (*"LMS"* OR *"CMS"*) (*"learning-analytics"* OR *"awareness"*) (*"clustering"* OR *"patterns"* OR *"group"*) (*"graph"* (*"interaction patterns"* OR *"behavior patterns"* OR *"choreography"*)) (*"learning"* OR *"training"* OR *"education"*)

7. (*"LMS"* OR *"CMS"*) (*"learning-analytics"* OR *"awareness"*) (*"clustering"* OR *"patterns"* OR *"group"*) ((*"graph"* OR *"rhizomatic"*) (*"interaction patterns"* OR *"behaviour patterns"* OR *"choreography"*)) (*"learning"* OR *"training"* OR *"education"*)

8. (*"LMS"* OR *"CMS"* OR *"learning environment"*) *"learning-analytics"* (*"clustering"* OR *"patterns"*) (*"graph"* (*"interaction patterns"* OR *"behavior patterns"* OR *"choreography"*)) (*"learners"* OR *"trainees"* OR *"university"*)

3.2.1.3 Selection Criteria

As criteria for selection, a list of inclusion and exclusion criteria was created and is presented in Table 3.9.

Table 3.9: Inclusion and exclusion criteria (SLR)

Inclusion criteria
Studies published after 2017
Studies written in English
The paper answers at least one of the research questions
Exclusion criteria
Book/chapter or unpublished papers
Duplicate articles
Not peer reviewed
The focus of the paper is on the semantic analysis of messages
The paper does not answer at least one research question
The paper focuses on interaction patterns that are irrelevant to the objectives of the study
The title identifies a distinct area apart from the objectives

3.2.2 Conducting

After the protocol is defined, the review can be executed. The search strings were performed against the Google Scholar database. Since it is challenging to retrieve metadata from the selected database, all search strings were conducted via the academic citation extraction and analysis software Publish or Perish[3], a total of n=5321 papers were retrieved.

Following the elimination of duplicates, n=1308 papers remained. After applying the inclusion and exclusion criteria stated in Table 3.9, n=56 papers remained. With these 56 publications, a source analysis revealed that 32 were from journals and 24 were from conferences. Priority was given to journals, but due to the nature of journal publication, a sample of conferences was

[3]Publish or perish: https://harzing.com/resources/publish-or-perish

included in the final sample. By eliminating conferences held prior to 2020, n=44 papers remained. The process can be viewed in Figure 3.3 and the results are presented in Appendix C.

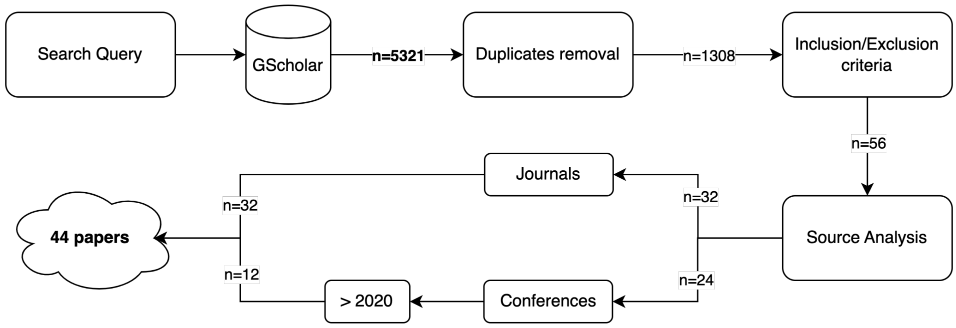

Figure 3.3: Overview of the SLR process

3.2.3 Thematic Analysis

The thematic analysis of this chapter follows the same guidelines by Vaismoradi *et al.* [115] conducted on SoS. In order to get value from the results, simplifying the data into codes and themes will make it easier to answer the research questions identified in Section 3.2.1.1.

The questions used to extract information will be similar to the research questions, making it easy to answer them later. These are the Thematic Analysis Questions (TAQ) used for extraction:

1. *TAQ1: Behaviors/Interactions in LMS/CMS represented by graph*
2. *TAQ2: Clustering techniques are used to identify behaviors/interactions in LMS/CMS*
3. *TAQ3: Ways to use graphs to represent behaviors/interactions in LMS/CMS*
4. *TAQ4: Ways to use clustering techniques to represent behaviors/interactions in LMS/CMS*

After extracting excerpts from the papers, each question was addressed. The next step was to convert these excerpts into codes, which are small informational units. In addition to providing all the information necessary to answer the question, these codes are clean of distractions. The next stage was an iterative process of identifying clusters of codes for each question, culminating in identifying themes. Tables 3.10, 3.11, 3.12 and 3.13 display the results of the identification of themes and their corresponding definitions.

3.3 Results

This chapter will use the SLR results to answer the Thematic Analysis Questions previously mentioned in Section 3.2.3.

Table 3.10: TAQ1: Behaviors/Interactions in LMS/CMS represented by graph

Theme	**Definition**
Social interactions between peers	Interactions and patterns of collaboration among students in online learning environments
Interactions with course pedagogical assessments	Interactions with course assessments such as quizzes and exercises
Course activity interactions	Student behavior and interactions with course activities in online learning environments
Learning object interactions	Learner behavior patterns in online environments while interacting with learning objects
General student interactions with LMS	Student behavior and learning activity in online learning environments using a variety of this environment's components.
Student trace data	Trace data collected from students in online learning environments, such as clickstream behavior
Teacher interactions in LMS	Teacher interactions with online learning environments

Table 3.11: TAQ2: Clustering techniques are used to identify behaviors/interactions in LMS/CMS

Theme	**Definition**
Partitioning-based Clustering	Clustering technique that divides a dataset into a fixed number of clusters, as specified by the user or algorithm (e.g. k-means)
Hierarchical-based Clustering	Clustering technique that creates a hierarchical structure of clusters, where each cluster is a subset of the previous
Density-based Clustering	Based on the density of the data points, a clustering technique that groups together data points that are densely packed
Time-series Clustering	Method of clustering used to group similar time-series data elements

Table 3.12: TAQ3: Ways to use graphs to represent behaviors/interactions in LMS/CMS

Theme	**Definition**
Directed graphs to represent social interactions	Use of directed graphs to represent the flow of communication between students
Social-cognitive network to represent social interactions	Represent how learners interact and influence each other's learning behaviors and outcome
Social-cognitive network to represent knowledge contribution patterns	Represent how learners contribute to the collective knowledge of an online learning community
Course assessment behaviors	Represent sequence of actions during course assessment
Process models to represent interaction sequences	Represent the steps that learners take when engaging with course materials and with each other
Learners learning behaviors	Represent learners' learning behaviors with the LMS (e.g., logs, actions)
Teacher interactions	Overall teacher interactions with the LMS
Learning situation of cohorts	Represent learning situation of cohorts using spatiotemporal digital profiles

Table 3.13: TAQ4: Ways to use clustering techniques to represent behaviors/interactions in LMS/CMS

Theme	**Definition**
Group similar students	Group students based on similarity
Distinguish different learning patterns	Group learning patterns which aid in distinguishing them
Cluster different types of students	Group students according to their type differences

3.3.1 TAQ1: What are the types of behaviors/interactions in LMS/CMS represented by graphs?

This question's objective was to identify the types of behaviors and interactions from LMS/CMS that can be represented by graphs that are currently being studied in the research community. With the thematic analysis, we were able to identify groups of behaviors and interactions that answer this question.

The first type of interaction identified comes from the social interactions between peers. These interactions are usually collaborative interactions between students [92, 11] and can happen in different contexts, being the most common interactions occurring in an LMS [122, 88, 101, 3, 34]. However, alternative sources can be utilized to collect these interactions. For instance, Ouyang *et al.* [80] conducted a study on student engagement during discussions in Massive Open Online Courses (MOOCs), while on the other hand, Sedrakyan *et al.* [105] investigated interaction patterns in social media contexts. Finally, it is noteworthy that interactions can be specific, such as interactions between two students, interactions between two students commenting on the initial posts of others, or interactions between students commenting on a thread [119].

The second theme relates to the course component of an LMS and its associated pedagogical assessments. Specifically, the focus is on the interactions that occur within this context. Zhang *et al.* [122] and Quispe *et al.* [71] examined how these interactions can provide valuable insight into students' learning behavior. Similarly, Waheed *et al.* [118] and Jang *et al.* [51] analyzed as one of their indicators how the interactions with assessments can be used to predict students' academic performance. More specifically, the interaction with quizzes can also provide valuable insights into identifying students' behavior patterns [52] and examine the engagement level and performance [4]. Quiz interactions can be helpful to aid in decision-making by providing a set of visualizations that enable the stakeholder to get a high view of how the students are behaving [120]. Finally, the study conducted by Mandlazi *et al.* [70] investigated how students interact with exercises to create a predictive model for determining their likelihood of answering subsequent questions correctly.

Still, in the context of courses, the third theme concerns the interactions related to course activity. This specific type of interaction exhibits a diverse range of applications, including but not limited to the creation of visual representations [101], recognition of pedagogical approaches [69], analysis of learning behavior [82, 58, 123], and evaluation of a learner's current situation [32]. Some examples of interactions analyzed are interactions with course videos [4, 120], and course material [93]. Lastly, The course activity interactions are usually analyzed using the logs from LMS or MOOCs. Studies from Marticorena-Sanchéz *et al.* [73], Manzanares *et al.* [114], Cobos and Garcia [26] and Bessadok *et al.* [13] demonstrate the possibility of using log-based methodologies to improve comprehension of online course student engagement and learning behaviors.

Learning object interactions represent another category of interactions. In their study, Frost and McCalla [39] used the previously mentioned interactions to devise a pedagogical planner, which can suggest learning trajectories to learners based on their prior interactions. At the same time, it is possible to utilize them to comprehend the correlation between the recognized interaction patterns and educational achievements [20].

LMS/CMS can provide numerous interactions, and grouping them can provide valuable insights. In most studies [13, 94, 63, 33, 93], logs were used to analyze and identify general student interactions. Examples of indicators of online behavior include access information, rating, accessibility metrics, and time spent on site [12].

Trace data representing student interactions on educational platforms [56, 64] is an additional valuable interaction that can yield significant insights and outcomes. Specifically, Waheed *et al.* [118] analyzed the clickstream behavior of students.

Lastly, most research focuses on student interactions, but comprehending and optimizing the processes from a teacher's perspective can benefit from teacher interactions [36].

3.3.2 TAQ2: Which clustering techniques are used to identify behaviors/interactions in LMS/CMS?

This question's objective was to identify which clustering techniques can be used to identify behaviors/interactions in LMS/CMS currently being discussed in the research community.

Multiple clustering techniques can be used to identify patterns/interactions in data. There is one type of clustering technique that is mentioned a lot in literature, and that is partitioning-based clustering. Inside this type, the most used is k-means [26, 13, 64, 75, 4, 20, 58, 33]. In their work, Manzares *et al.* [114] mention two other variants, k-means++, and fuzzy k-means.

Another type of clustering technique is the agglomerative hierarchical clustering technique [69], which uses this technique to identify groups of learners based on the identified interaction sequence patterns.

On the other hand, when the applications may have a lot of noise, using density-based clustering techniques such as DBSCAN [114] can be beneficial.

For the last type of clustering technique, Zhang *et al.* [122] used time-series clustering to classify students based on the similarity of their encoded series.

3.3.3 TAQ3: How can graphs be used to represent behaviors/interactions in LMS/CMS?

The third objective of the thematic analysis was to determine how graphs can be used to create visualizations of LMS/CMS context-specific behavior and interactions.

Using directed graphs to represent social interactions is an effective method for representing behaviors and interactions. Various approaches to using directed graphs to represent various forms of social interactions in online learning environments have been researched [101, 11, 31]. For instance, Ye and Pennisi [119] described student interactions in online discussions using directed graphs. Similarly, Sedrakyan *et al.* [105] used force-directed graphs to represent social interaction patterns in conversations and forums. Another instance was Desai *et al.*'s [34] use of graphs to describe social relationships.

According to Ouyang *et al.* [81], social-cognitive networks can represent social interactions, for example, their participatory roles or network positions. Another example is knowledge contribution patterns, such as individual knowledge inquiry and group knowledge construction are used to calculate the cognitive engagement of the participants.

Graphs can be used to represent course assessment behaviors, such as the completion of assignments, by using spatiotemporal digital profiles [63]. Another approach involves utilizing process maps to depict behaviors related to taking quizzes [52].

An alternative approach to illustrating interactions through graphs involves using process models that represent sequences of interactions, such as learners' interactions with an LMS [69].

Finally, it is possible to depict various behaviors, including student engagement with technology [20], through Moodle logs [58], actions related to learner learning behavior [33], teacher interactions with MOOCs [36], and the learning context of groups of learners [63].

3.3.4 TAQ4: How can clustering algorithms be used to represent behaviors/interactions in LMS/CMS?

Clustering techniques provide valuable methods to represent behaviors and interactions in LMS/CMS by grouping similar students, differentiating between distinct learning patterns, and clustering various student types.

Clustering techniques can combine similar students in an LMS/CMS, allowing educators to gain insight into student behavior and adjust instructional strategies. Using time-series clustering, Zhang *et al.* [122] grouped students based on their similarities. Similarly, Manzares *et al.* [114] used clustering to rapidly group students in different monitoring periods, enabling teachers to analyze groups of students in various monitoring periods. Finally, Cobos and Garcia [26] utilized clustering to organize students into cohorts.

Clustering techniques in LMS/CMS provide a valuable method for identifying students' distinct learning patterns, enabling educators to provide more effective support. Bessadok *et al.* [13] employed clustering to obtain different student profile groups based on student activities. Similarly, Loginova and Benoit [64] used clustering to differentiate between low- and high-achiever learning strategies. Lastly, Mahauad *et al.* [69] and Denden *et al.* [33] grouped students according to their learning behavior patterns.

Lastly, clustering can be used to identify different types of students [58].

Chapter 4

Case Under Analysis

In this Chapter, the case under analysis is presented with an in-depth introduction to the context of the problem, followed by a dive into the Moodle data that comes from the case, where a methodical approach will be constructed to determine which choreographies are valuable to identify in our case study context, and finally an examination of some problems that arose during this analysis.

4.1 Case Description

Real-world case studies are essential because they provide valuable insights into real-world circumstances. Unlike hypothetical processes, real-world data enables more realistic conclusions. Ultimately, results and contributions are more precise and comparable to those of other case studies. It does not imply that hypothetical scenarios have no value. These will be used throughout the early phases of this research to validate concepts before moving on to the real-world scenario.

The nature of this study entails that the case study data originate from an LMS. In light of this, the bachelor's degree in Informatics Engineering course Software Development Laboratory (LDS, Portuguese acronym) conducted using Moodle at Universidade Aberta (UAb) was selected. This course implements a pedagogic method proposed by the project SCReLProg (Self and Co-regulation in e-Learning of Computer Programming), which aims to improve the transition from introductory to advanced computer programming courses by focusing on the e-learning context [85]. Since it was designed with the e-learning environment and the asynchronous teaching method in mind and because it was the project's pilot course [86], it collected a vast amount of anonymized data, including cognitive challenges, programming exercises, task submissions, and raw Moodle logs. The raw Moodle logs is the artifact that will receive the full attention going forward.

To comprehend how the LDS course was conducted, it is necessary to understand the Sim-Programming project's foundations, which was the project that preceded the SCReLProg. The first is that the class should be run simulating a business environment, with distinct roles and responsibilities for each student [85]. Thus, students will experience a context similar to future work experiences. The second foundation is self-regulated learning, which encourages student engagement and ultimately leads to developing work routines [85]. Following, we have the third foundation, which is co-regulated learning. The objective is to encourage students to participate in communities and to learn how to solve problems as a group rather than individually [85]. Lastly, formative assessment is given by the tutors or professors, which is a crucial stage in providing and receiving feedback, mentoring, and managing expectations [85]. From these fundamental principles, the e-learning version of SimProgramming, was developed, e-SimProgramming, which clarified the business-like environment principle as adopting more ambitious immersive learning principles [84]. The LDS course follows this approach, with students portrayed as newly hired interns at a software development company.

The log data will comprise all the actors' actions in the various Moodle course page functionalities. These include interactions with forums and discussions, assignments, assessments, and learning content in the form of books, pdfs, and others.

4.2 Moodle Data Analysis

Even though our study is not considered a data mining problem, the data generated by the Moodle interactions in the LDS course is a considerable amount to analyze and discover valuable information, in this case, choreographies. It is essential to use a structured process when analyzing data, therefore, we adopted the initial phases of the CRISP-DM model to help us comprehend and prepare the data to extract valuable information. The first three phases are business understanding, data understanding, and data preparation [106].

With the information from the Case Description section, we now have a solid comprehension from a business standpoint. We know the context of the course, where the data was collected, why it was collected, and most importantly, what we can expect from it.

We had to delve deeply into the raw data records during the data understanding phase. In this phase, it was essential to know what we were searching for in the spreadsheet, so we began by identifying the columns and the information they contained. Contextually, we are analyzing the anonymized data from 2019-2020 from the LDS course, which had been anonymized by the SCReLProg project team prior to this analysis. This analysis enabled the following column descriptions to be determined:

1. Timestamp: indicates the date and time when an event or action was captured.

2. Username: represents the username of the user who initiated or was involved in the logged event.

3. AffectedUser: contains the username of the user or system directly affected by the logged event.

4. Context: specifies the environment or setting where the event occurred within Moodle. The column's contents may vary but may include the course in which an event occurred or the page affected by an event.

5. Component: contains information regarding the particular functionality or feature associated with an event. This column enables the place of origin of an event within Moodle to be determined. For instance, if the event occurred in the main system or a Fórum.

6. EventName: describes the standardized name or type of the logged event.

7. Description: represent additional details or context about the logged event. It may include event details such as a description, the activity that triggered the event, or user actions that describe what the user did in the event.

8. Origin: indicates the source or location from which the event originated. Some examples of values in this column include web, mobile, and cron jobs, among others.

9. IP: captures the IP address associated with the user who initiated the event.

Focusing on the component column determining the events' place of origin, we identified nine main components in the data. They are:

1. Choice: where teachers can create polls or surveys, and students can select one or more responses.

2. Forum: allows communication and discussion between course participants and the teacher.

3. Assignment: enables instructors to collect and evaluate student work through file uploads or text submissions.

4. Book: is typically used to present course materials in a book-like format and allows instructors to create structured content with sections.

5. Page: the resource that allows teachers to construct static web pages to present information.

6. Folder: where instructors can organize and exhibit course-related files and documents.

7. Quiz: where teachers can assess student knowledge and provide automatic grading through multiple-choice or short-answer questions.

8. Course: a component that encompasses the full learning experience for a particular subject or topic. It combines various components such as assessments, forums, and others.

9. URL: instructors can use this resource to link to external web content.

Now that we comprehend the data type in each column, we must examine each row. Considering the time and effort required to analyze all the data in nine distinct components, we restricted our analysis to forums and assignments. By narrowing the scope to these two, it becomes clearer to comprehend how choreographies can be identified and possible challenges. These components were chosen since they play a significant role inside an LMS, and because of that, more choreographies and edge cases can be found.

In the context that the course was undertaken, these two components will provide a substantial number of interactions. To identify possible choreographies is essential to have a good amount of interactions to start identifying patterns between them. However, analyzing it raw would require a great deal of time, so to simplify things, we began the data preparation phase. A script was developed to quicken this procedure, and Figure 4.1 depicts the applied rules.

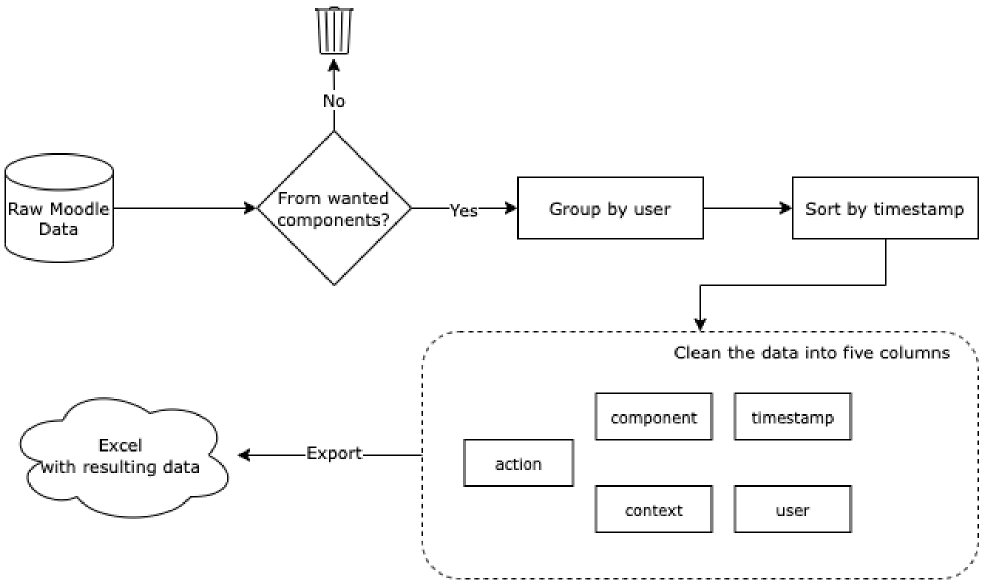

Figure 4.1: Flow chart of the script decision rules

As shown in Figure 4.2, after running this script, all unnecessary data for the analysis was removed, leaving a more focused view of the interactions and potential patterns.

	A	B	C	D	E
1	timestamp	user	component	context	action
3047	5/04/20, 23:30	S54	Trabalho	Trabalho: Sala de desenvolvimento C16	Estado da submissão visualizado
3048	6/04/20, 00:18	S54	Trabalho	Trabalho: Sala de desenvolvimento C16	Formulário de submissão visualizado
3049	6/04/20, 00:18	S54	Trabalho	Trabalho: Sala de desenvolvimento C16	Estado da submissão visualizado
3050	6/04/20, 00:19	S54	Trabalho	Trabalho: Sala de desenvolvimento C16	Estado da submissão visualizado
3051	6/04/20, 00:19	S54	Trabalho	Trabalho: Sala de desenvolvimento C16	Enviada uma submissão do trabalho
3052	6/04/20, 00:19	S54	Trabalho	Trabalho: Sala de desenvolvimento C16	Formulário de confirmação de submissão visualizado
3053	6/04/20, 00:19	S54	Trabalho	Trabalho: Sala de desenvolvimento C16	Estado da submissão visualizado
3054	6/04/20, 00:19	S54	Submissão de ficheiros	Trabalho: Sala de desenvolvimento C16	Submissão criada
3055	6/04/20, 00:19	S54	Submissão de ficheiros	Trabalho: Sala de desenvolvimento C16	Um ficheiro foi carregado
3056	8/06/20, 14:26	S54	Trabalho	Trabalho: P-fólio	Estado da submissão visualizado
3057	9/03/20, 18:05	S54	Trabalho	Trabalho: E-fólio A	Estado da submissão visualizado
3058	1/06/20, 16:56	S59	Trabalho	Trabalho: Sala de desenvolvimento C45	Estado da submissão visualizado
3059	1/06/20, 19:38	S59	Trabalho	Trabalho: Sala de desenvolvimento C48	Estado da submissão visualizado

Figure 4.2: Result of the script execution on Moodle SCReLProg project raw data, the anonymized users come up as S<number>

After running the data through the three phases, we are in a condition to identify patterns of interaction in the data.

4.3 Virtual Choreographies Identification

During the state-of-the-art research, indicators representing the different interactions observed within an LMS were identified. These indicators can be found in Section 3.3.1.

We identified the need to name the set of interactions from the platform logs as state-of-the-art statements (EdA statements). In this context, when we refer to a choreography composed of an EdA statement, it denotes that this EdA statement encapsulates a group of interactions extracted from the logs. This approach allows for a more streamlined representation of the choreography, signifying a coherent and cohesive process such as "Create topic and publish content", rather than a collection of n interactions conveying the same idea. Throughout the choreography identification process, our primary focus lay in constructing choreographies composed of a single EdA statement. Consequently, it is common for both the EdA statement and the choreography to share matching names in our findings.

The first type of component that was analyzed was the Assignments. There are 17 different assignments and more than 4500 rows in the data received. This component is responsible for creating and managing online Moodle assignments. Even though both teachers and students use this component, the scope of this analysis will be on student interactions. Students can submit any digital content (e.g., PDF, images, audio), teachers can enable just a plain text submission, and in some cases, both the text and file uploads. Identifying the choreographies within the logs required significant effort, as a typical action, such as submitting an assignment with file upload, corresponds to seven different steps within the logs. It was necessary to comprehend how Moodle website actions were reflected in the logs to identify the choreographies listed in Table 4.1. All choreographies identified include interactions with course pedagogical assessments, general student interactions with LMS, and student trace data.

The other type of component analyzed to identify choreographies is the Forum. There are 16 forum topics with almost 11500 rows in the data received. This Moodle component is powerful for creating online discussions where students and teachers can exchange comments. Table 4.2 shows the identified choreographies in the forum component. The choreographies related to the visualization, creation, and updating of topics can be classified under four indicators: social interactions between peers, interactions with course pedagogical assessments, general student interactions with LMS, and student trace data.

In conclusion, we identified seven distinct choreographies in two types of components corresponding to the indicators discovered in the state-of-the-art research. These examples illustrate the variety of choreographies that can be identified in LMS data and will serve as a foundation for the remainder of the study. Rather than simply identifying them, we now have a process that can be replicated, allowing us to identify more.

4.4 Problems

After identifying the choreographies from LMS data logs, some problems arose:

- What should be done if several log items (e.g., steps) can be interpreted as belonging to several overlapping choreographies?

Table 4.1: Choreographies of Assignment component

Choreography name	Steps
01 Submit assignment with file upload	1. Viewed submission status 2. Viewed submission form 3. A file has been uploaded 4. Created submission 5. Submitted an assignment 6. Viewed submission status 7. Comment created
02 Submit assignment with text	1. Viewed submission status 2. Viewed submission form 3. Viewed submission status 4. Submitted an assignment 5. Created submission 6. An online text has been loaded
03 Update assignment with text	1. Viewed submission form 2. Viewed submission status 3. Submitted an assignment 4. Updated submission 5. An online text has been loaded 6. Viewed submission status 7. Submitted an assignment 8. Updated submission 9. An online text has been loaded
04 Visualize submission	1. Viewed submission status

Table 4.2: Choreographies of Forum component

Choreography name	**Steps**
01 Create topic and publish content	1. Viewed course module 2. Discussion Viewed 3. Topic created 4. Some content has been published
02 Update topic and publish content	1. Viewed course module 2. Discussion Viewed 3. Topic updated 4. Some content has been published
03 Visualize discussion	1. Discussion Viewed

- Since Moodle logs record all interactions from every user concurrently, what to do if a sequence is intertwined with log entries about the actions of other users?
- What should be done if the steps needed to complete a choreography are made at different moments in time (e.g. started in the morning and completed in the afternoon)?

4.4.1 What should be done if several log items (e.g., steps) can be interpreted as belonging to several overlapping choreographies?

Having false positives while identifying choreographies can be harmful when displaying information and visualizations that aid decision-making and course management. Occasionally, the same sequence of interactions can fall into two or more choreographies, leading to false positives. A plan should be considered to be consistent and confident that the choreographies are being identified. This overlap can be seen in choreographies 02 and 04 from Table 4.1, where the step "Viewed submission status" is identical between them.

To prevent unwanted overlaps having a hierarchy of choreographies is essential. In this case, the adopted criterium was that the interactions will be matched against the most complex choreographies first and then go toward the less complex ones. This way, overlaps will be less likely, and if they occur, it means that a longer or more complex choreography was performed. Using the example of the choreographies identified for the Assignment component in Table 4.1, the order would be choreography 01, 03, 02, and 04. Having this hierarchy will prevent those widely made interactions such as "Viewed submission status" from being matched with the wrong choreography, as seen in Figure 4.3.

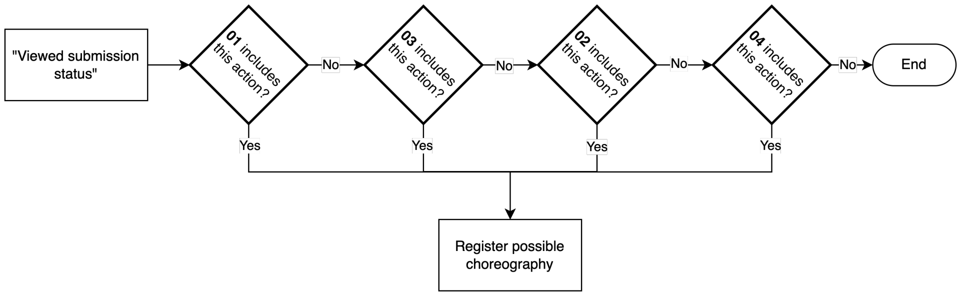

Figure 4.3: Problem 1: Hierarchy of choreographies

4.4.2 Since Moodle logs record all interactions from every user concurrently, what to do if a sequence is intertwined with log entries about the actions of other users?

Raw data usually is recorded independently of the context. Consequently, for example, in the case of Moodle, if user A is in the process of submitting an assignment and user B is making the same process at the same time, logs could be:

1. User A: Viewing course assignment
2. User A: Writing assignment
3. User B: Viewing course assignment
4. User B: Writing assignment
5. User B: Submitting the assignment
6. User A: Submitting the assignment

As can be seen, without treating the data, a mistake could be made while identifying the choreographies. Due to the presence of user B's actions between user A's actions, this error would result in the incorrect identification of user A's choreography for submitting an assignment. As a result, the raw data needs to be filtered by the user who made it and ordered by timestamp to prevent this from happening. However, this approach is unable to detect multi-user choreographies, so subsequent extensions of this work will require devising a new solution for this problem.

4.4.3 What should be done if the steps needed to complete a choreography are made at different moments in time (e.g. started in the morning and completed in the afternoon)?

With the approach from the previous question, this problem is mitigated for the cases where the difference between interactions is low. For the cases where the difference in time is too high, a maximum time could be defined for each choreography. This time could be calculated from the average time to complete the choreography plus a pre-defined control value. Other criteria, like the user session timeout configuration, could also be considered, and subsequent works should explore how to have richer criteria to determine this.

As an example, if on average choreography A takes 10 minutes, and the pre-defined control value is 5 minutes. In that case, if user A's interactions lead to choreography A but take longer than 15 minutes, it will not be identified as choreography A.

This method opens the door to additional possibilities, such as identifying these cases as prevalent errors or different approaches to help identify different types of profiles.

Chapter 5

Methodology

Due to the software engineering nature of this dissertation's objective, the Design Science Research (DSR) methodology will be applied. This chapter will provide a brief introduction to the DSR, followed by an overview of how this dissertation applies the guidelines proposed by Hevner *et al.* [45], and conclude with a review of how the DSR is used throughout the dissertation.

5.1 Design Science Research

Information Systems (IS) is a broad topic with multiple definitions considering the perspective. Boel and Cecez-Kecmanovic [15], in their review, identified the existence of four different perspectives: technology, social, socio-technical, and process. Such a big topic and no unique definition, the researchers must capture the dual nature of IS: knowledge-producing and knowledge-using. They can be captured with two distinct paradigms, behavioral science and design science [45]. The first concerns understanding reality, while the other attempts to create things to solve problems [72].

DSR is a methodology in which knowledge and comprehension of a design problem are acquired during the development of a solution. It addresses the truth, which is the objective of behavioral science, and utility, which is the objective of design science, resulting in a more curated scientific outcome. The iterative process in DSR recognizes that the development of an artifact is a continuous and dynamic journey between understanding, executing, and evaluating.

In their study, Hevner *et al.* [45] presented a conceptual framework that, in addition to the design itself, represents two key foundational pillars. The first is analyzing the environment to determine the business's requirements and, by extension, the work's relevance. On the other hand, the research rigor is derived from the knowledge base, which consists of all relevant knowledge and information. Later in their research, they formulated seven guidelines to aid the research community in conducting design-science research. Understanding these guidelines allows the entire framework to be digested easily. The guidelines are:

- **Guideline 1 - Design as an Artifact**: The outcome of the research should be an innovative and purposeful artifact that addresses a particular problem or challenge.

- **Guideline 2 - Problem Relevance**: The research problem should be meaningful and relevant by addressing real-world problems or demands. The problem should be clearly defined and supported by prior research, emphasizing the gap or improvement opportunity.

- **Guideline 3 - Design Evaluation**: The artifact should be evaluated rigorously using appropriate methodologies and standards. The evaluation should demonstrate, for example, the artifact's efficacy, usability, and impact in addressing the identified issue. In their study, Hevner *et al.* [45] provided in Figure 5.1, dimensions and methods that can be used to evaluate artifacts.

- **Guideline 4 - Research Contributions**: The artifact should provide insights and knowledge that advance the theoretical understanding of the discipline while also providing practitioners with practical implications and tangible benefits.

- **Guideline 5 - Research Rigor**: The research process must adhere to scientific principles and rigorous procedures to ensure the findings' validity and reliability.

- **Guideline 6 - Design as a Search Process**: The development of the artifact should be iterative and exploratory, with initial designs refined and enhanced based on feedback, testing, and introspection.

- **Guideline 7 - Communication of Research**: The findings should be effectively published to academics and practitioners along with the artifact's design process, findings, and implications.

1. Observational	Case Study: Study artifact in depth in business environment
	Field Study: Monitor use of artifact in multiple projects
2. Analytical	Static Analysis: Examine structure of artifact for static qualities (e.g., complexity)
	Architecture Analysis: Study fit of artifact into technical IS architecture
	Optimization: Demonstrate inherent optimal properties of artifact or provide optimality bounds on artifact behavior
	Dynamic Analysis: Study artifact in use for dynamic qualities (e.g., performance)
3. Experimental	Controlled Experiment: Study artifact in controlled environment for qualities (e.g., usability)
	Simulation – Execute artifact with artificial data
4. Testing	Functional (Black Box) Testing: Execute artifact interfaces to discover failures and identify defects
	Structural (White Box) Testing: Perform coverage testing of some metric (e.g., execution paths) in the artifact implementation
5. Descriptive	Informed Argument: Use information from the knowledge base (e.g., relevant research) to build a convincing argument for the artifact's utility
	Scenarios: Construct detailed scenarios around the artifact to demonstrate its utility

Figure 5.1: Design evaluation methods [45]

Similar to Hevner *et al.* [45], Peffers *et al.* [87] proposed a design science research methodology (DSRM) for developing and presenting design science in IS based on prior research. The authors adopt a different approach, with the most notable difference being the existence of multiple entry points into the cycle and the ability to enter various iteration phases. As depicted in Figure 5.2, if a prototype already exists, it can enter the cycle directly in the design and development phase, making it a design and development-centered initiation.

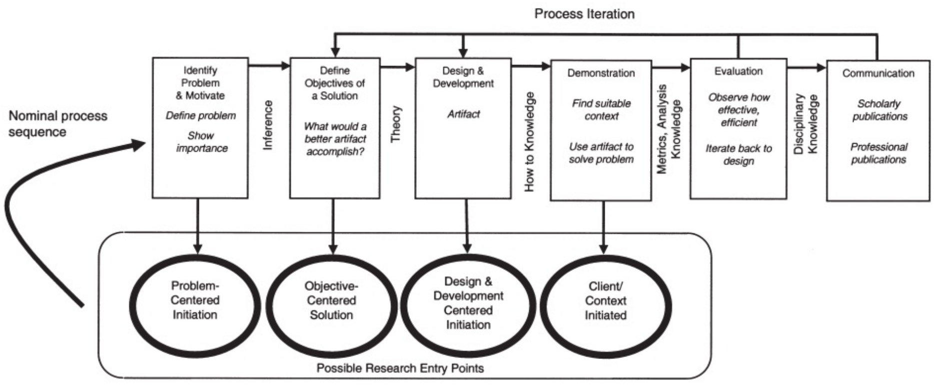

Figure 5.2: Design Science Research Methodology Process Model [87]

As can be seen in Figure 5.2, the process is divided into six different activities. The first activity aims to specify the research problem and the value of the resulting artifact; then, the objectives can be defined, which can serve as a guideline or acceptance criteria for the solution; in activity three, the artifact is designed and developed; now, with the artifact is possible to demonstrate in practice, how it can solve the problem defined in the first activity; next, the evaluation phase assesses if the researchers need to do more iterations or if they can move on to the next step; lastly, the knowledge generated from this process must be effectively communicated to the research community, ensuring comprehensive documentation to uphold the highest standards of rigor and value.

Even though the process was described sequentially, it does not mean it is in a strict order. The researchers have the opportunity to enter different phases of the cycle because the research can be in different maturity levels. However, even though the cycle can be started at any step, it must follow the iteration process once it starts [87].

5.2 Guideline Application

As mentioned in Section 7.1, Havner *et al.* [45] proposed seven guidelines explained in that section. Here, these guidelines will be matched with the research done in this study.

The first rule states that the outcome should be a purposeful artifact. This study created a service that transforms logs from platforms such as Moodle into virtual choreographies. This artifact results from a primary issue on how virtual choreographies can be represented in learning interoperable data dashboards, resulting in a second problem on transforming meaningless data from multiple sources into useful information that describes various interactions, the virtual choreographies. Therefore, this research advances the identification and application of virtual choreographies. The third principle stipulates that the artifact should be evaluated in various methods. In this study, multiple types of evaluation were conducted across multiple iterations. Unit tests were used at the first and lowest level to ensure that business rules were followed and results were accurate. Also two specialists in the research field of this study evaluated the proposed solutions. The results of this evaluation would determine whether the artifact addressed the specified problem or required additional iterations. This study will yield several contributions corresponding to the fourth guideline. The first contribution is a tool that enables the identification of virtual choreographies from raw data logs, the second is a proof-of-concept dashboard that showcases the potentialities of using this approach to improve the visualizations used in decision-making, and the third is a methodical approach to identify valuable choreographies from logs and a systematic literature review on the topic of study. Using DSR, we then ensure that the study adheres to the scientific procedures essential to ensure the validity and reliability of the findings, thereby adhering to guidelines five and six. The findings of DSR will be communicated to the research community with this dissertation, along with the artifacts' source code.

5.3 Application in this Study

From the model depicted in Figure 5.2, this dissertation enters the cycle as a problem-centered approach. In Chapter 1, we described the problem and the motivations. To comprehend the artifacts objectives, an SLR was conducted in Chapter 3, and the results, in conjunction with our Case Study in Chapter 4, led to the problem definition in Chapter 6. Design occurred in 7, followed by development in 8, and evaluation was performed at the conclusion of each iteration against the objectives and with the evaluation of two stakeholders. The evaluation feedback and results of the design and implementation phases are discussed in 9. It is important to note that, despite the iterative nature of this process, each phase has been grouped to improve readability. This dissertation describes the research methodology and findings, completing the final stage of DSR.

Chapter 6

Problem and Requirements

This chapter serves as the foundation for the subsequent chapters, in which we briefly mention the problem of this dissertation and then the findings from the SLR that apply to the problem. With the conclusions of the former chapters, the results will be materialized into actionable goals and use cases by conducting a Requirements Engineering process.

6.1 Problem

The problem of this dissertation is explained comprehensively in Section 1.3; nevertheless, as stated in that section, the problem that gives this dissertation its title was divided into three steps. At this state we have already made progress in solving the problem.

The first step, understanding which interactions can be gathered from LMS contexts, represents a literature research problem, addressed in the SLR question "TAQ1: What are the types of behaviors/interactions in LMS/CMS represented by graphs", described in Section 3.3.1. Understanding the main interactions that researchers are expecting in LMS contexts enabled the extraction of choreographies from the Case Study data described in Section 4.3.

The second step, the identification of virtual choreographies from LMS log data, as mentioned, had two components. The first was to determine which choreographies could be identified based on the Case Study, addressed in Chapter 4, where a method for identifying virtual choreographies was created. The second component, is how can these choreographies be identified automatically. This is the main problem of this project since the interactions in the log file are unknown, and the volume of logs can make manual identification unfeasible. Therefore, it is essential to understand how to simplify this process. For this, we will reconstruct the methodology proposed by Lacet *et al.* [60], where the output will be learning virtual choreographies that the dashboard will consume.

After identifying virtual choreographies from log data, we want to represent them visually. To accomplish this, we will approach step three by creating visualizations from the data generated in latest phase. These visualizations will be straightforward and solely act as proof of concept.

6.2 Systematic Literature Review Learnings

When the SLR was conducted (Chapter 3), the focus was on how to visualize choreographies using graphs and clustering techniques. However, we promptly understood that identifying the choreographies from the logs was necessary before addressing the visualization problem. However, this does not imply that the results were worthless. Conversely, through the outcomes of TAQ1, for instance, we could determine which types of interactions are analyzed and visualized in LMS.

Zhang *et al.* [122] examined course pedagogical assessments, including peer interaction, course assessment access, and grade checking. This analysis revealed students' self-regulated learning patterns. Marticonera-Sánchez *et al.* [114], Cobos and Ruiz-Garcia [26] studied course activity interactions covering forums, quizzes, surveys, wikis, and submitted assignments. Loginova and Benoit [64] traced student data from LMS interactions. Within the same subject of study, Hassan *et al.* [56] evaluated how models distinguish between the learning strategies of low- and high-achievers.

Regarding interactions data, Waheed *et al.* [118] studied student clickstream behavior and evaluation performance in pedagogical course assessments. Juhaák *et al.* [52] analyzed quiz interactions within a course. These researchers created process maps showing quiz-taking behavior variants. In a more general way, Qu *et al.* [94] evaluated LMS student interaction logs to find features that indicate final exam failure.

The authors Villagrá-Arnedo *et al.* [117], Saqr *et al.* [101], Becheru *et al.* [11] and Desai *et al.* [34] focused their researches on social interactions between peers. The last three studies' results were direct graphs representing social interactions.

Youngs *et al.* [120] constructed a sample of preliminary statistical visualizations from student interactions with a course video and its attendant questions. On the same topic, Park [82] examined online learner behavioral interactions, time-on-task, attendance, and performance throughout a semester in two online courses.

Lastly, Carroll and White [20] examined how behavioral learning patterns affect learning outcomes which resulted in the identification of student interaction with technology.

These results helped understand what was being investigated in the literature and aided in determining the choreographies on which to target our research. Some of the application of these results can be seen in the different choreographies we identified in Section 4.3.

6.3 Requirements Engineering process

The achievement of a software development project is contingent upon a comprehensive understanding of the system requirements. Requirements Engineering (RE) is a specialized domain within the field of software engineering that focuses on the practical objectives, capabilities, and constraints associated with software systems [121]. The RE process plays a crucial role in capturing, analyzing, and documenting these requirements to ensure that the outcome software solution properly meets the needs and expectations of the intended solution.

This chapter will discuss the RE process's application within this project's context. Initially, an examination will be conducted regarding the process of requirement elicitation. Next, we shall delve into requirement analysis, followed by an exploration of the requirements specification, and conclude with an examination of the methods employed to validate the requirements.

6.3.1 Requirements Elicitation

In requirements elicitation, we discover, review, document, and understand users' needs and constraints for the system. One of the main goals of this process is to identify the needs and possible boundaries of the system [78].

Throughout the literature, we have different methods to elicit requirements. Nuseibeh and Easterbrook [78] mention a few techniques:

- Traditional techniques: involve the application of questionnaires and surveys, the conduct of interviews, and the analysis of pre-existing documentation.
- Group elicitation techniques: team dynamics is employed to effectively elicit a more carefully selected and collaboratively agreed upon set of requirements. Two commonly employed methods in this particular domain include brainstorming and focus groups.
- Prototyping: used when there is a lack of certainty regarding the requirements or demands. However, the primary objective of its usage resides in its ability to function as an additional instrument for diverse activities, such as enhancing group discussions or conducting questionnaires.
- Contextual techniques: using ethnographic methodologies, such as participant observation, conversation analysis, and pattern identification, are employed to discern and ascertain the various needs within a given context.

Goguen and Linde [42] discussed the practice of introspection as a method involving the imaginative adoption of the perspectives of users or experts to understand their needs and requirements. The authors argue that in order to improve the accuracy of this technique, it is necessary to employ other techniques alongside it, as the technique itself has inherent limitations.

The requirements for this project were obtained through the utilization of three distinct techniques:

- Analyzing pre-existing documentation: analyzing the method that Lacet *et al.* [60] proposed and identifying the common requirements for both approaches.
- Introspection: given the author's[1] proficiency in the field of software engineering, employing this technique can offer significant benefits in discerning the potential requirements that a system of such scale may require.

[1] The author holds a BEng in Computer Science and Engineering. He has professional experience in building e2e solutions, contributed to the digitalization of processes in numerous city councils, including Lagos, Ovar, Vila Real, Penafiel, Évora, and Estarreja, among others.

- Brainstorm sessions: during these sessions, discussions were conducted with two professors[2][3] who actively employ LMS in their everyday teaching methodologies.

Moreover, these professors demonstrate significant proficiency in employing virtual choreographies across various contexts. So all the requirements gathered from the three sources were discussed with them. The brainstorming sessions were the main source for the majority of requirements, but by having two other sources, the accuracy is improved, which leads to a better requirements' quality.

6.3.2 Requirements Analysis

Requirement analysis is the cognitive process of critically examining and evaluating the requirements gathered [100]. During this phase, the parties involved reach a consensus regarding the requirements' validity, importance, completeness, and priority.

The elicited requirements were subject to interactive analysis, typically conducted at the beginning of the subsequent brainstorming session. This task aims to verify the accuracy of the requirements collected during the previous session and make any necessary adjustments to enhance their precision. Lastly, another session took place to prioritize the requirements regarding their importance, feasibility, and impact.

The results of this analysis have established a robust basis for making well-informed decisions in further stages of the software development lifecycle.

6.3.3 Requirements Specification

Requirements specification is the phase where the user's needs and constraints are documented, following models like the Goal Model [91]. The requirements' documentation plays a vital role in ensuring their readability, analysis, rewriting, and validation [78].

The requirements specification phase will produce two outputs. The first is the goals, a curated view of the product's vision; the second is use cases as a method to document the system's functionalities.

6.3.3.1 Goals

Van Lamsweerde [116] describes a goal as the objective that the system should attain through the collaborative efforts of agents within the software-to-be and the surrounding environment. These goals can be documented using natural language or models [91]. Models, such as AND/OR trees,

[2] Leonel Morgado is an Associate Professor with Habilitation at Universidade Aberta, a public e-learning university, lecturing on research methods, programming, and immersive learning, and a senior researcher at INESC TEC. Aside from being a specialist in e-learning and LMS use in educational contexts (having written a book and papers on the subject), he has also written or co-written publications that have advanced the virtual choreography study.

[3] Fernando Cassola is a PhD researcher at INESC TEC and also auxiliar professor at FEUP. Fernando does research in Computer Science, namely on the use of gamification to promote behavior change. Fernando has also contributed to the field of virtual choreography research with numerous articles and a PhD thesis on the subject.

offer a visual representation of goals that facilitates a hierarchical understanding of the required sub-goals for goal attainment.

In the context of this project, we have two different types of goals: 1) general goals and 2) specific goals. The second type is used when inside one general goal exists various ramifications or sub-goals that need to be accomplished to set the main goal as complete.

Traceability is one of the foundational pillars of RE since it maintains relationships between all the artifacts [100]. Every goal defined in this section will have an identifier. The meaning of the identifier is represented in Figure 6.1.

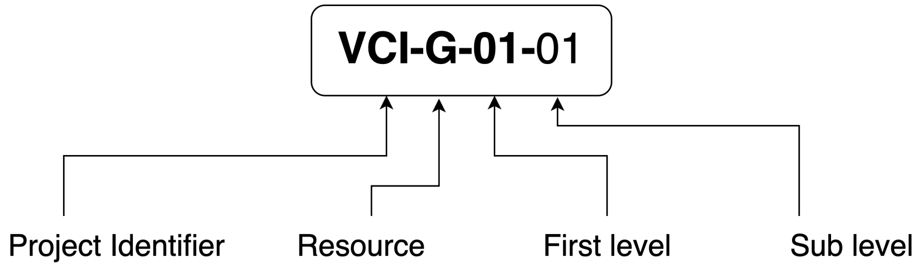

Figure 6.1: Goal identifier meaning

Figure 6.2 depicts the first level of the AND/OR table for the project's general goal: Virtual choreographies identification. As represented, the goal is divided into three other goals: 1) Transform logs into xAPI specification, 2) Identify Virtual Choreographies in xAPI elements, and 3) Represent Virtual Choreographies.

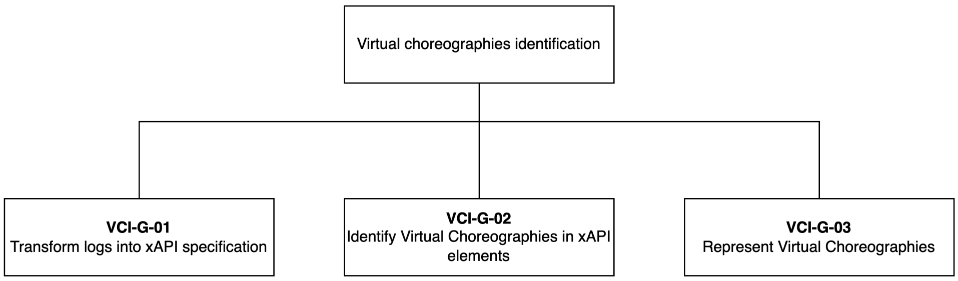

Figure 6.2: Main goals

The first and second objectives are intertwined; the VCI-G-01 goal output will be platform statements and recipes based on xAPI, and VCI-G-02 will associate these statements with an independent statements configuration and identify the choreographies through this match. A broad view of the goals' inputs and outputs can be seen in Figure 6.3.

Then we will focus on the sub-goals needed to achieve the first goal VCI-G-01, depicted in Figure 6.4. This goal's primary focus is on providing a solution in which it is possible to transform raw data into statements and recipes using xAPI specification elements, which focus on having a common "language" between all the systems. To achieve this goal, we first need to be able to convert the logs into xAPI statements (VCI-G-01-01), after this it is possible to generate recipes

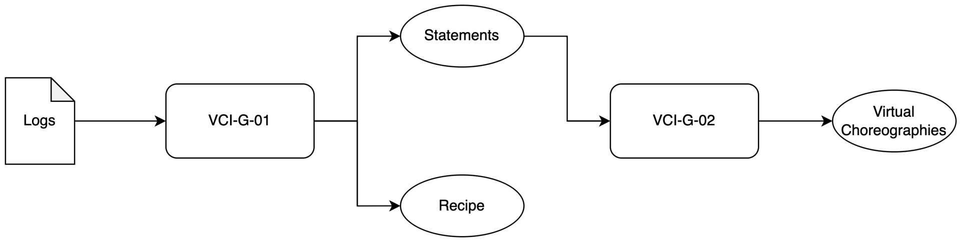

Figure 6.3: VCI-G-01 and VCI-G-02 inputs and outputs

through the xAPI statement (VCI-G-01-02) while taking into consideration the third goal (VCI-G-01-03), which aims to have this transformation process as agnostic as possible. This implies that the system can be easily adapted to new sources with minimal or no effort.

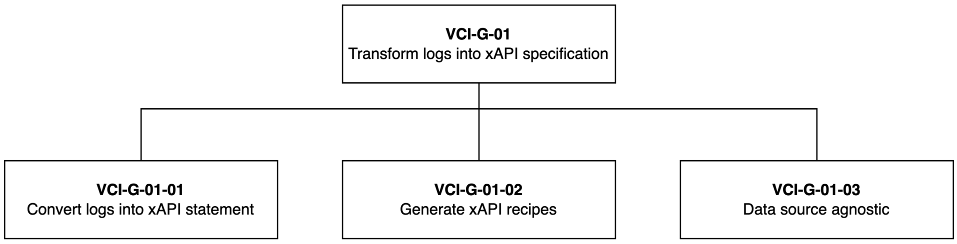

Figure 6.4: Subgoals of Goal 1

Now that the first goal is explained, we can focus on the second goal VCI-G-02 and its sub-goals, represented in Figure 6.5. The focus is on identifying the virtual choreographies from the VCI-G-01 outputs and the pre-defined interactions that we want to find in the data. For this to happen, the sub-goal VCI-G-02-01 focus on creating choreographies by doing the match between the xAPI elements with the configurable match rules from the sub-goal VCI-G-02-02.

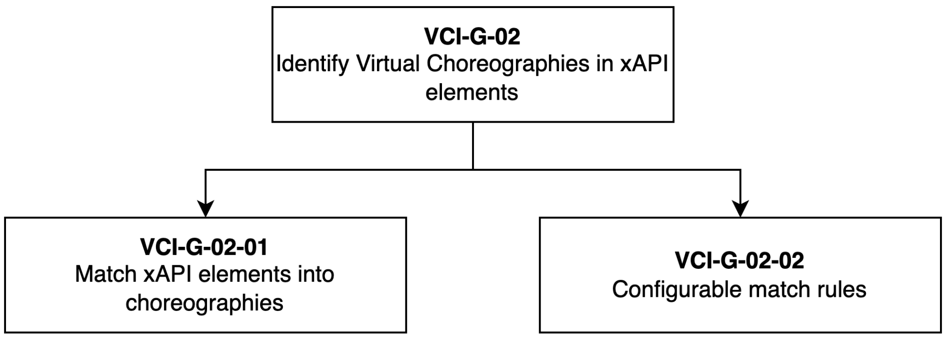

Figure 6.5: Subgoals of Goal 2

Lastly, the focus is on goal VCI-G-03, depicted in Figure 6.6, which focuses on representing the identified virtual choreographies in a visual interface. For that to happen, visualizations have to be produced (VCI-G-03-01), and it has to be possible to import a source into the system (VCI-G-03-02).

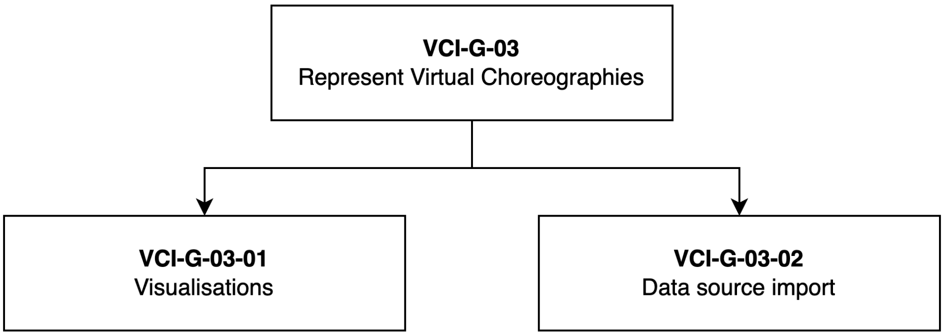

Figure 6.6: Subgoals of Goal 3

6.3.3.2 Use Cases

A use case is a compilation of potential interactions between a system and its actors, all connected to a particular goal [28]. An actor represents a class of users, specific roles, or even other systems [68]. Thus, the use case enables to describe interactions with the system without having developed the system. It captures, in a complete way, who (actor) does what (interaction) with the system and for what purpose (goal) [68].

In order to depict use cases, the method of textual representation will be employed, using an adaption of the template outlined by Pohl and Rupp [91] in their research. The use cases were prioritized using the MoSCoW technique, which analysts and stakeholders employ to prioritize use cases [46]. These techniques are divided into four categories, enumerated from highest to lowest priority: 1) Must have, 2) Should Have, 3) Could Have, and 4) Won't have.

The following use cases are related to one of the services to be implemented:

(a) **Use case** VCI-UC-01 Read local file data

Goals VCI-G-01-01 and VCI-G-01-03

Priority Must have

Pre-conditions

- Strategy defined for the file format
- File to be uploaded

Result Logs in JSON format

Main Scenario

1. User uploads a file
2. System read file content
3. System transforms content into JSON
4. System creates a source
5. System return logs and source

(b) **Use case** VCI-UC-02 Create statements from logs

Goals VCI-G-01-01 and VCI-G-01-03

Priority Must have

Pre-conditions

- Logs in JSON format
- Configuration defined for the logs transformation

Result xAPI Statements

Main Scenario

1. Actor inputs the logs
2. System converts logs into xAPI statements
3. System returns xAPI statements

(c) **Use case** VCI-UC-03 Create recipe

Goals VCI-G-01-02

Priority Must have

Pre-conditions

- VCI-UC-01
- VCI-UC-02

Result xAPI Recipe

Main Scenario

1. Actor sends the source
2. System transforms the source xAPI statements into a recipe
3. System returns xAPI recipe

(d) **Use case** VCI-UC-04 Export source

Goals VCI-G-01-01 and VCI-G-01-02

Priority Must have

Pre-conditions

- VCI-UC-01
- VCI-UC-02
- VCI-UC-03

Result Source contents

Main Scenario

1. Actor sends the source
2. System gathers the information about source
3. System gathers the information about the xAPI statements
4. System gathers the information about the xAPI recipe

5. System returns all the source contents

(e) **Use case** VCI-UC-05 Get statements

Goals VCI-G-01-01

Priority Could have

Pre-conditions

- VCI-UC-02

Result xAPI Statements

Main Scenario

1. Actor requests all statements for a source
2. System fetches the statements for the source
3. System returns the statements

(f) **Use case** VCI-UC-06 Get recipe

Goals VCI-G-01-02

Priority Could have

Pre-conditions

- VCI-UC-03

Result xAPI Recipe

Main Scenario

1. Actor requests the recipe for a source
2. System fetches the recipe of the source
3. System returns the recipe

(g) **Use case** VCI-UC-07 Write local file

Goals VCI-G-01-01 and VCI-G-01-02

Priority Should have

Pre-conditions

- JSON information

Result File

Main Scenario

1. System sends javascript JSON object
2. System converts it into a JSON file
3. System outputs a JSON file with the information

Then, the following is related to the other service:

(a) **Use case** VCI-UC-08 Get Virtual Choreographies

Goals VCI-G-02-01 and VCI-G-02-02

Priority Must have

Pre-conditions

- xAPI log statements
- EaD statements configuration

Result Virtual choreographies

Main Scenario

1. Actor sends the xAPI statements
2. System orders and sort the data
3. System identifies the EaD statements based in the configurations
4. System transforms the identified xAPI log statements into EaD statements
5. System transforms the EaD statements into choreographies
6. System returns the choreographies

Lastly, we have the use cases to be implemented by the dashboard:

(a) **Use case** VCI-UC-09 Import source

Goals VCI-G-03-02

Priority Must have

Pre-conditions

- Valid file to be uploaded

Result Feedback message

Main Scenario

1. Actor clicks on file upload
2. Actor uploads the file
3. System uploads the file
4. System shows a feedback message

(b) **Use case** VCI-UC-10 View recipe information

Goals VCI-G-03-01

Priority Should have

Pre-conditions VCI-UC-09

Result View recipe information

Main Scenario

1. Actor clicks on the Recipe page
2. System displays the statistics about the recipe
3. Actor clicks in a specific xAPI element

4. System provides the specific information in a list

(c) **Use case** VCI-UC-11 View logs statements

Goals VCI-G-03-01

Priority Should have

Pre-conditions VCI-UC-09

Result View converted log statements

Main Scenario

1. Actor clicks on the Log Statements page
2. The system displays a list of the transformed log statements

(d) **Use case** VCI-UC-12 View choreographies

Goals VCI-G-03-01

Priority Must have

Pre-conditions VCI-UC-09

Result View choreographies identified

Main Scenario

1. Actor clicks on the Choreographies page
2. The system displays a list with the identified choreography types and associated choreographies

(e) **Use case** VCI-UC-13 View dashboard visualisations

Goals VCI-G-03-01

Priority Must have

Pre-conditions VCI-UC-09

Result View dashboard visualisations

Main Scenario

1. Actor clicks on the dashboard
2. System calculates and displays a chart with the recipe information
3. System calculates and displays a chart with the types of choreography identified
4. System calculates and displays a chart with the choreographies identified throughout time

6.3.4 Requirements Validation

Requirements validation is the RE phase ensuring that the goals and use cases accurately reflect the requirements of the stakeholders [24]. This phase is often confused with the requirements analysis phase; however, it is important to note that they are distinct from one another. Requirements verification focuses on "Have we got the right requirements?" while requirements validation focuses on "Have we got the requirements right?".

Pohl and Rupp [91] present an exposition on three common review types employed in the process of requirements validation:

- Commenting: the requirements are given to a domain expert to identify possible flaws regarding the quality of the requirements.
- Inspection: a systematic process used to check the requirements for errors.The inspection demands much preparation since it is divided into several phases: planning, overview, error detection, error correction, follow-up, and reflection.
- Walk-through: a systematic approach for identifying potential quality deficiencies. It is similar to an inspection but with less formality and preparation. During the walk-through, participants engage in discussions regarding requirements that have been validated. The procedure is guided by moderators.

In the context of this project, the same stakeholders identified at the outset of the RE process participated in a walk-through conducted remotely using Zoom. The author presented each of the gathered requirements, and the feedback was registered. After analyzing and applying the recommendations, the requirements were validated.

Chapter 7

Design

In this chapter, we will present the non-functional prototypes for each artifact and an overview of software architecture design. This chapter contributes to understand what will be implemented.

7.1 Non-functional Prototypes

In software development, prototypes are early, incomplete, or streamlined variants of a software system or its components. A prototype can be low-fidelity, meaning it is basic and lacks details, or high-fidelity, meaning it appears exactly like part or all of the final user interface [9]. Prototypes can be used to test both small and large concepts. Ultimately, a prototype that can be presented can help to validate, extend, and evolve ideas [9], and because low-fidelity prototypes require less effort to develop than the proper functions of the system, it is ideal to validate it first and then implement it, thereby reducing the possibility of failing [113].

This chapter will go over two distinct prototype categories. The initial component involves converting logs into the xAPI specification, which employs JSON notation. The prototype effectively illustrated the manual implementation of the transformation. Utilizing this information will increase precision and reduce error susceptibility at the outset of the development process. The next prototype relates to the standard dashboard interface. Prior to the development process, a prototype was created to determine the visual appearance and layout of the pages.

7.1.1 Logs Transformation into xAPI Specification Prototype

The case study data presented in Chapter 4 will serve as the premise for this section's discussion.

The absence of a direct match between the columns and the xAPI elements posed the most significant difficulty when transforming log data into xAPI statements. Therefore, we examined each column comprehensively about the elements we desired to identify: the actor, verb, object, location, and context.

We studied a randomly selected row from the log spreadsheet to improve our analytical process's efficiency. This approach facilitated the identification of columns associated with xAPI elements and elements that the data does not directly provide. Table 7.1 represents this process.

Table 7.1: Moodle log analysis with xAPI elements

Columns	**Values**	**xAPI Elements**
timestamp	10/12/20, 22:27	Context
username	S56	Actor
affectedUser	O1	-
context	Fórum: Sala de desenvolvimento C47	Place
component	Fórum	Context
eventName	Módulo de disciplina visualizado	Context
description	The user with id '******' viewed the 'forum' activity with course module id '******'.	Context
origin	ws	-

Upon analysis, by joining the row with the descriptions gathered for each column in Section X, it was possible to identify that the data already includes the actor, location, and context. Nonetheless, it is imperative to delineate the verb and object of the statement. Although the `eventName` column provides some indication of the object and verb, our process did not include column parsing due to time restrictions. To maintain simplicity, we propose that xAPI elements may be manually defined when they cannot be extracted directly from the data. In the specific instance of the event labeled "Módulo de disciplina visualizado", we will interpret the verb as "visualizou", and the object as "módulo de disciplina".

Now that we have identified all the necessary elements for formulating a statement, we can express it in natural language: *"O utilizador S56 visualizou o módulo de disciplina no Fórum: Sala de desenvolvimento C47"*.

Using the knowledge obtained from our previous analysis, we intend to determine the most efficient method for generalizing this process. To accomplish this, we define transformation as "rules". These rules constitute a collection of configurations that facilitate the automatic transformation of logs into xAPI statements. Using examples from previously analyzed log data, we can classify the functionality of these principles into two distinct phases:

- Verification: Determine if the `eventName` from the current row under inspection corresponds to the type "Módulo de disciplina visualizado".
- Transformation: If the specified condition is met, the corresponding row undergoes a conversion process, transforming it into a statement according to a pre-established transformation rule.

It is crucial to recognize that this process follows an iterative approach. If available, additional rules are evaluated before a row is discarded. For illustrative purposes, the column `eventName` was utilized. In practical situations, however, this check demonstrates high adaptability, allowing matching against any column of the provided source data. Alternatively, a rule applicable to all rows can be implemented if the transformation process remains constant. This method has the benefit of not requiring unique rules for each event.

Listing 7.1 provides an illustrative example of a transformation rule, considering the objective of transforming a row with `eventName` "Módulo de disciplina visualizado". Each rule comprises two principal components: 1) match and 2) generate.

The match component delineates the specific columns that this rule will transform. For instance, in our scenario, we aim to target rows with `eventName` "Módulo de disciplina visualizado". This condition, however, could be replaced with any other valid condition or omitted entirely, in which case the rule would apply to all rows.

In contrast, the generate component contains the instructions for converting logs into xAPI statements. The yield statement's structure will consist exclusively of xAPI elements (actor, verb, object, location, and context). The variation resides in the values that these elements can take on, which can be categorized into three categories:

- Manual Value: This type applies to our demonstrated case where we manually ascertain the verb and object. These are explicitly denoted as strings.
- Generalization Values: These are used to extract values directly from the row. If we specify that the actor corresponds to the value `{{ username }}`, upon encountering a row, the actor will represent the value from the username column. In our example, this will be S56.
- Pre-defined Values: For our current requirements, the sole identified necessity is `{{ randomUUID }}`, which generates a unique identifier each time it is invoked.

```
{
    "match": {
        "event": { "en-us": "Viewed course module" }
    },
    "generate": {
        "actor": {"id": "{{ username }}", "name": "{{ username }}"},
        "verb": {
            "id": "http://.../viewed",
            "display": { "en-us": "viewed" }
        },
        "object": {
            "id": "{{ randomUuid }}",
            "definition": { "name": "{{ event_context }}" }
        },
        "place": {
            "id": "{{ randomUuid }}",
            "name": "{{ component }}"
        },
        "context": { "...": "..." }
    }
}
```

Listing 7.1: Transformation rule example in JSON

Having established the rule configuration, when encountering our example row during a transformation in the source, a JSON xAPI statement is produced, as depicted in Listing 7.2.

```
{
    "actor": { "id": "S53", "name": "S53" },
    "verb": {
        "id": "http://.../viewed",
        "display": { "en-us": "viewed" }
    },
    "object": {"id": "...", "definition": { "name": "discussion"}},
    "place": { "id": "...", "name": "Forum: Lorem" },
    "context": { "...": "..." }
}
```

Listing 7.2: JSON xAPI statement example

The ultimate objective of our solution is to generate a recipe derived from the xAPI statements, a task made feasible by assigning a unique id to each element. Listing 7.3 depicts an example recipe originating from this source, presuming that the only existing statement corresponds to the one depicted in Figure 7.2.

```
{
        "actors": ["S53"],
        "verbs": ["viewed"],
        "objects": [ "discussion"],
        "places": ["Forum: Lorem"],
        "contexts": [{}]
}
```

Listing 7.3: JSON xAPI recipe example

Although our prototype is adaptable to numerous scenarios, we have identified one significant limitation. Incorporating parsing capabilities into the architecture of generalization rules would be beneficial. Our case study data is an excellent candidate for this improvement. Given that the `eventName` column contains both the verb and object elements, it would be advantageous to be able to designate the verb and object components from specific sections of the `eventName` column. Despite being a step in the right direction, such a parsing solution would not compose a universal solution, as the same column may contain different values and attributes. Consider the `eventName` “Disciplina visualizada”, where the verb corresponds to the last segment of the event and the object to the first. Conversely, the same file could contain the `eventName` “Pré-visualização da tentativa de Teste iniciada”, which demands an opposite parsing rule. For such instances, the manual value designation would still be required.

7.1.2 Dashboard Prototype

The prototype presented in this section pertains to the proof of concept dashboard, which will be created to validate basic visualizations and data derived from the virtual choreographies model identification process.

This prototype was created to comprehensively understand the page layout and identify the essential elements required for its successful implementation. The layout consisted of four main components: a dashboard page, a recipe information page, an xAPI statements page, and a choreographies page.

As depicted in Figure 7.1, the dashboard page was designed to include various visualizations. These visualizations encompass the quantity of xAPI elements detected from the source, the type of identified choreographies, and the temporal distribution of these choreographies.

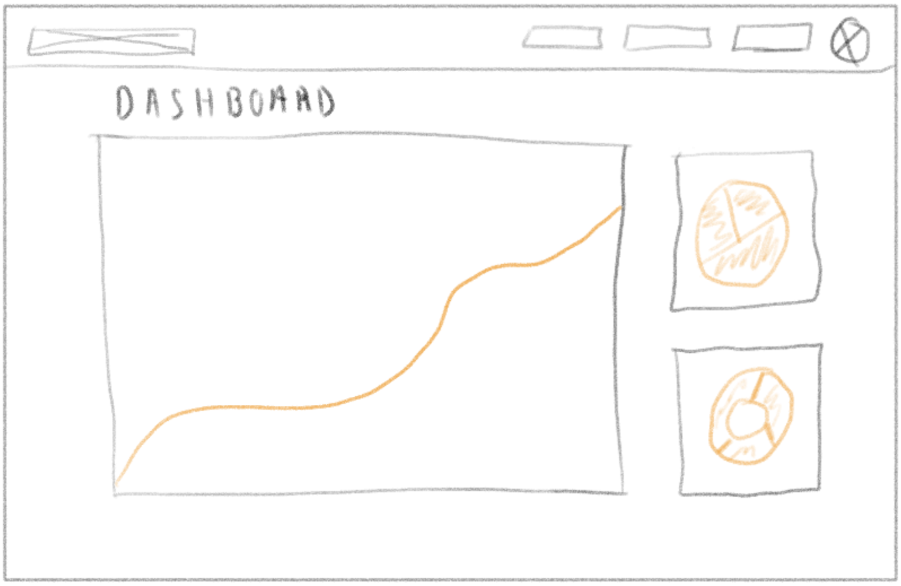

Figure 7.1: Dashboard page prototype

In the recipe information page, it was crucial to present a simplified representation of the actors, verbs, objects, and places identified from the uploaded source. The results are visually represented in Figure 7.2.

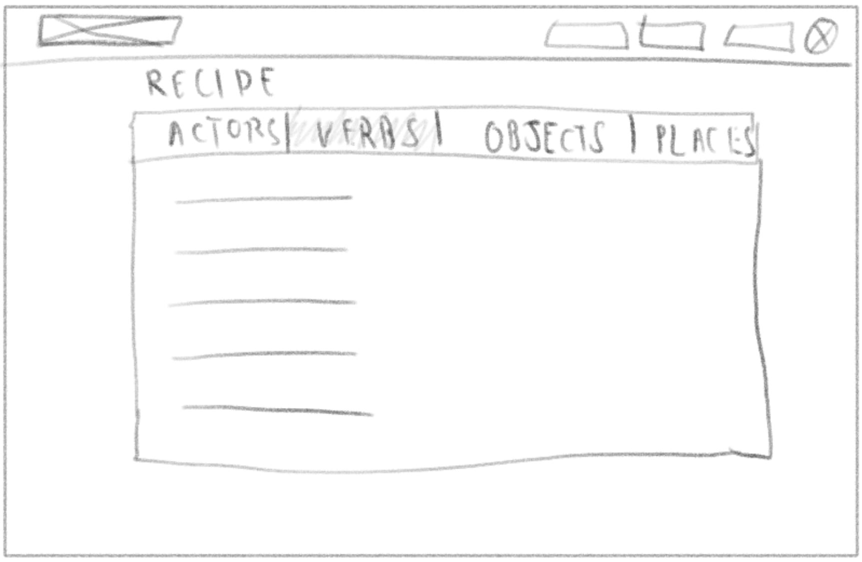

Figure 7.2: Recipe page prototype

The log statements page exhibits the xAPI statements converted from the logs. In order to accommodate the potential expansion of this page due to the conversion of numerous logs, pagination has been used within the table. The result can be seen in Figure 7.3.

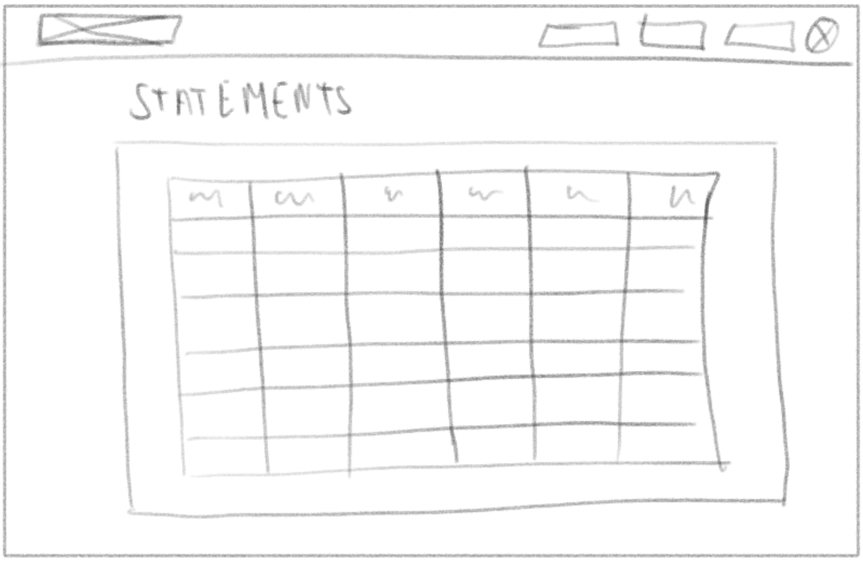

Figure 7.3: Statements page prototype

Lastly, the choreographies page will present the choreographies that have been identified based on the logs. In order to effectively present the various identified types of choreography on this page, it was necessary to establish a clear delineation between these types. A stack layout was envisioned, wherein only the displayed tab results are visible. The prototype can be seen in Figure 7.4.

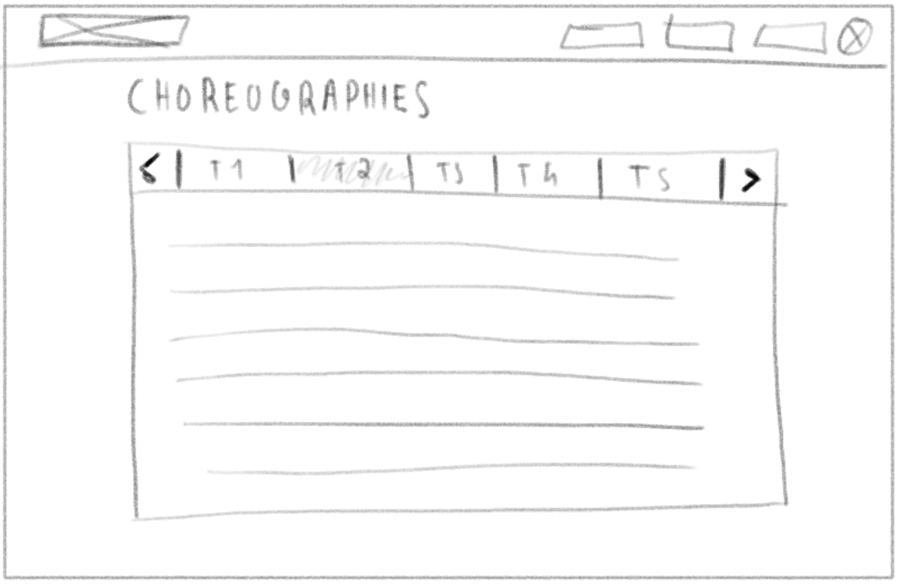

Figure 7.4: Choreographies page prototype

In summary, this prototype will prove beneficial during the planning and development phase as it provides insight into the needs for creating this dashboard.

7.2 Software Architecture

According to Bass *et al.* [10], the software architecture of a program or computing system encompasses the different software components, their externally observable attributes, and how they interact with each other. The presence of a clearly defined specification can have a beneficial influence on various aspects of software, including its comprehension, reusability, adaptability, analysis, and management [41].

This section will provide an overview of the system's architecture, followed by a detailed explanation of each component within the system. Then, a concise overview of clean architecture, that will serve as the basis for the backend services.

7.2.1 System Architecture

As depicted in Figure 7.5, the system requirements necessitated the development of three specific services. The architecture of the system is divided into two major layers: the backend and the frontend.

The backend layer, titled "Virtual Choreography Interpreter", comprises two services and was built using Clean Architecture, mentioned in Section 2.7. The initial service, denominated as "Platform Data Extractor", transforms logs from various platform sources into xAPI statements or xAPI recipes. Its ultimate responsibility is to transform the raw data into an interpretable and universally

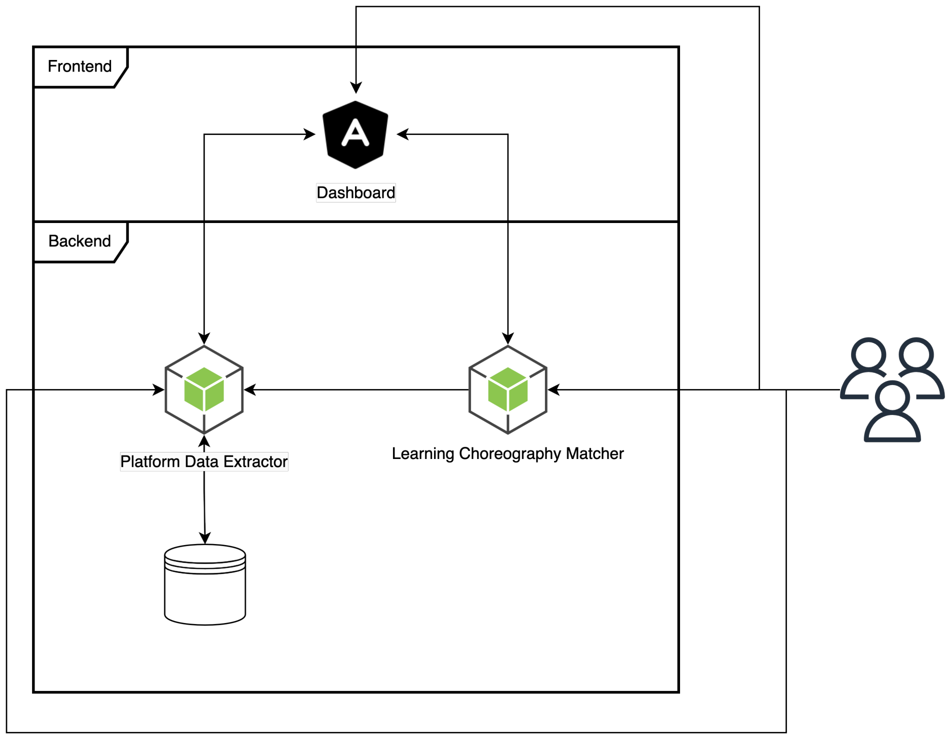

Figure 7.5: Virtual Choreography Interpreter System Architecture

comprehensible language, JSON xAPI. Consequently, the identification of choreographies is the responsibility of the second service, “Learning Choreography Matcher”. It accomplishes this by processing the output derived from the Platform Data Extractor and the definitions of choreographies saved for identification within the logs. The division of responsibilities between two distinct services offers numerous benefits:

- Scalability: Since the two services handle distinct tasks, i.e., the translation of logs and the identification of choreographies, they may encounter varying demands at different periods.
- Division of Responsibilities: This bifurcation allows an individual possessing logs complying with the xAPI specification to utilize solely the matcher service and vice versa.
- Independent Evolution: Each of the services can evolve independently, ensuring that the modification or enhancement of one does not adversely impact the other.

Complementing the backend layer is the frontend layer, Virtual Choreography Interpreter Dashboard or Dashboard, which consists of a web application developed using Angular and Echarts for data visualization. Users can interact with the frontend to integrate a data source and then observe visualizations associated with the imported data.

Chapter 8

Implementation

This chapter describes implementation decisions and the path followed to achieve certain objectives. The chapter begins with an overview of the domain entities shared by all layers. Then, a detailed explanation of the backend service, Virtual Choreography Interpreter, and the frontend service, Virtual Choreography Interpreter Dashboard, explaining each of the implemented use cases.

8.1 Domain Entities

As illustrated in Martin's book [74], entities represent enterprise-wide business rules applicable to multiple applications. They are resistant to operational changes or external factors within an application or in external environments. Entities can assume various forms, such as objects of methods, interfaces, or a collection of data structures and functions. As a result, it is possible to encapsulate all business principles within a collection of entities.

As previously introduced in Section 2.7, we will use Clean Architecture. In this context, the domain entities remain consistent across the two backend services and correspond to the statement, recipe, and choreography entities.

The entity diagram depicted in Figure 8.1 clarifies the aforementioned entities as well as the common attributes linking the statement and recipe entities. These attributes represent the available xAPI elements: actor, verb, object, place, and context. They solely depict types conceptually and do not include any associated functionality. The attributes were established to ensure that the interfaces related to the elements were uniform across all entities. The attributes always include a unique identifier (ID), assuring their exclusivity. This solution eliminates the prospect of encountering two actors or places with the same name.

The Statement entity, which forms the basis of our solution, is responsible for preserving all of the information that the other two entities will eventually utilize. It contains vital business rules, such as the JSON conversion and the exportation of a statement into natural language. In addition, it ensures that all information used to create a statement is accurate. For instance, the statement

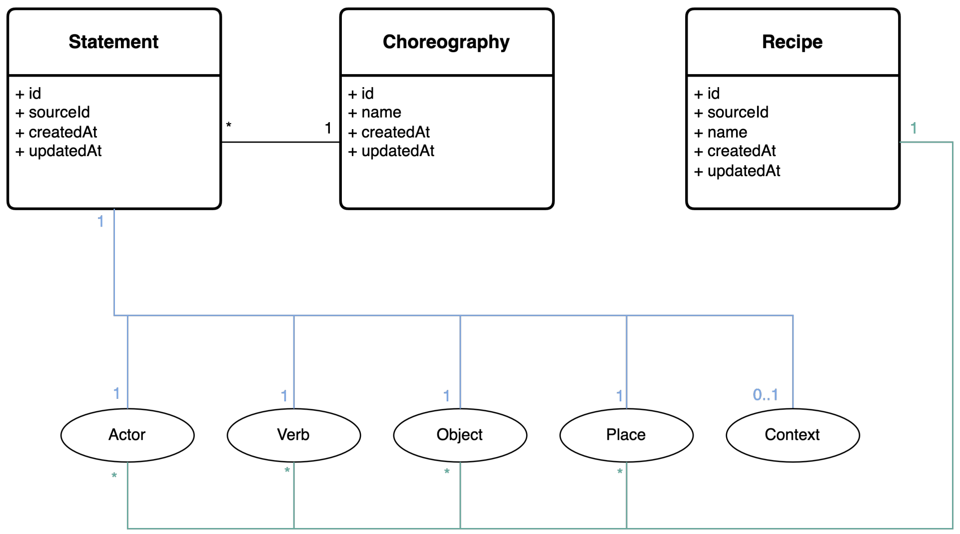

Figure 8.1: Virtual Choreography Interpreter Domain Entities

verifies whether a timestamp conforms to a valid format that is widely recognized. Without a precise timestamp, the process of identifying patterns over time is impossible.

The Recipe entity then takes statements as input from which the recipe will be generated. As specified in Section 7.1.1, this entity's capabilities range from generating the recipe with distinct elements to transforming the entity into JSON format. Despite the possibility of multiple statements being associated with a single actor, the recipe should only contain a single entry.

Lastly, the Choreography entity receives the statements constituting the identified choreography. Like the other entities, this entity can export itself into a widely supported format, JSON.

8.2 Virtual Choreography Interpreter

In this chapter, our focus is directed toward implementing the system's backend layer. As outlined in Section 7.2, the backend layer is denominated the "Virtual Choreography Interpreter" and is constituted by two distinct services known as the "Platform Data Extractor" and the "Learning Choreography Matcher". Although these services can operate independently, given their different capabilities and differing entry points, they are employed in conjunction within the context of this dissertation.

To enhance comprehension of how these services function collectively, Figure 8.2 provides a diagrammatic representation of the inputs and outputs associated with each service. The Platform Data Extractor, receives the log as input and produces xAPI statements and xAPI recipe as output. The Learning Choreography Matcher, receives the xAPI statements as input and identifies choreographies within these statements as output.

This overview offers an understanding of the inputs and outputs of the backend layer. Subsequent sub-sections delve further into the intricacies of these inputs and outputs, the functionalities they offer, and the rationale behind decisions made during the development of each service.

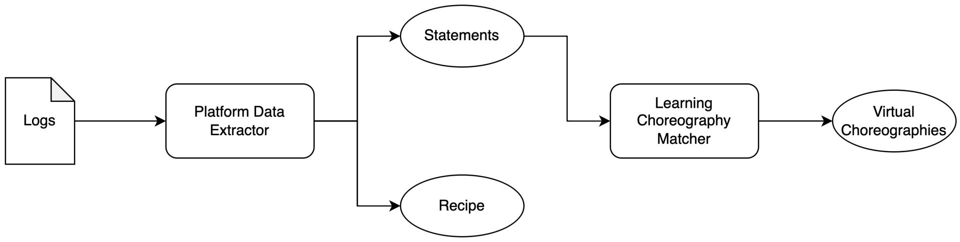

Figure 8.2: Virtual Choreography Interpreter services inputs and outputs

8.2.1 Platform Data Extractor

The Platform Data Extractor was the first service to be developed. It was in this service that the entities were developed initially and where it was implemented most of the business rules. Nevertheless, the creation of this service aimed to solve the goal `VCI-G-01 Transform logs into xAPI specification`. To fulfill it, all the sub-goals have to be addressed: `VCI-G-01-01 Convert logs into xAPI statement`, `VCI-G-01-02 Generate xAPI recipes`, and `VCI-G-01-03 Data source agnostic`. Taking this into account, all the decisions were made thinking of the sub-goals and the correspondent use cases.

This service, by following Clean Architecture, considered the division of responsibilities inherent to the architecture. To accomplish this, the service is composed of four layers:

- Domain: This layer represents the core of the service and contains the business logic and rules. It contains the entities, repository interfaces, and types. This layer has no external dependency, meaning it can be extracted to any project, and the business rules would still be applied.
- Application: This layer encapsulates the use cases or application-specific interactions. This layer acts as a middleman between the domain and the infrastructure layer and is responsible for orchestrating the data flow. In the context of this project, this layer contains the use cases, helpers, and strategy pattern implementation for different types of files uploaded.
- Adapters: This layer implements the logic that handles the communication between the application and the external systems or frameworks. In the context of this project, it converts the data incoming from HTTP requests, handled by a library called express.js, into the format that the use cases can read.
- Infra: In this layer is where all the external dependencies needed to run the application live. Here is where the code related to database access, HTTP servers, configurations, and data is stored.

With a better understanding of how the service is divided in terms of layers, we will focus more on the implementation details of the use cases. By following this approach, it is possible to understand what were the decisions and approaches that were followed to reach the objectives. Lastly, with this approach is possible to trace back to the use cases and goals defined in Chapter 6.

8.2.1.1 VCI-UC-01 - Read local file data

For this use case, one of the requirements was that it would be simple to extend to multiple file formats, since platform agnostic was an objective. Since we are using Case Study data in Excel format, the first available file extension would naturally be Excel.

The responsibility of this use case was simple: 1) read the file, 2) parse it, and 3) convert it into JSON with the associated columns. Imagine a file with two columns, name and place, and two rows, the first row with the values John and Portugal and the other Alice and Spain. This use case's output would be the JSON object represented in Listing 8.1.

```json
{
    "logs": [
        {
            "name": "John",
            "place": "Portugal"
        },
        {
            "name": "Alice",
            "place": "Spain"
        }
    ]
}
```

Listing 8.1: Output of use case Read local file data

The first stage of this use case is to receive the file's location as input. With the file's location known, it is now possible to read and parse it. However, since dealing with various file extensions was necessary, a different method was adopted than hard-coding Excel file reading. Figure 8.3 illustrates the strategy pattern that was implemented for this purpose. A strategy pattern is a behavioral design pattern that allows distinct algorithms to be encapsulated and interchangeable at runtime. It permits the selection of various algorithms based on the situation and requirements. In this case, the approach varies depending on the file type being analyzed.

Each strategy consists of two methods: `retrieveData` and `toObjectArray`. The first method is responsible for extracting data from the file provided as input and will output the columns/keys and their corresponding values in no particular order or format. The second method will convert the output into a readable JSON array format, as shown in Listing 8.1.

The logs are finally valid, and in a format that the application understands, so a Source is created and persisted so that it can be utilized throughout the service. A Source is a unique entity from this service, consisting of an id and a type of source; currently, only the file type is supported. This entity was required since a way of attributing every element to its source was needed. So, we are able to determine which statements, choreographies, and recipes correspond to which source.

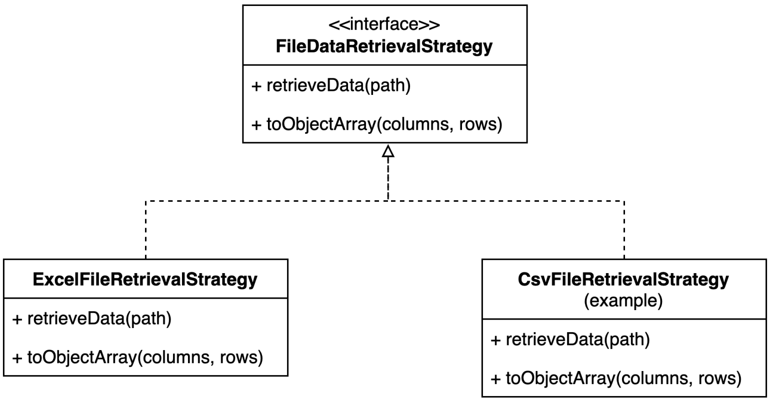

Figure 8.3: File retrieval strategy diagram

Now that a source has been developed and the log data is in a legible format, the second use case, `VCI-UC-02 Create statements from logs`, has all the necessary inputs to convert the logs into statements.

8.2.1.2 VCI-UC-02 Create statements from logs

This use case depends on the input from the previous use case (`VCI-UC-01`) and represents the core functionality of this service, which is the transformation of logs into valid xAPI statements. This use case nearly achieves two sub-goals, `VCI-G-01-01` and `VCI-G-01-03`, demonstrating its complexity.

The use case consists of two steps: converting logs into xAPI statements and then persisting them. The input for this use case consists of the `sourceId` from which the logs are from, the logs themselves, and the config key. This config key enables the use case to specify the rules by which logs must be converted into statements. Currently, we have the `MOODLE_CONFIG`, which corresponds to the rules outlined in Section 7.1.1. Still, we could have multiple configuration files for additional types of data.

Each config has its own configuration file, in this case, is a typescript file, but since the rules are JSON objects, we could transform the configs into JSON if needed. Taking the example given in Section 7.1.1, "Módulo de disciplina visualizado", the corresponding transformation rule is represented in Listing 8.2.

```
export const MOODLE_CONFIG = [
  {
    pattern: {
      eventName: {
        "en-us": "Viewed course module",
        pt: "Módulo de disciplina visualizado",
      },
    },
    generate: {
      actor: { id: "{{ randomUUID }}", name: "{{ username }}" },
      verb: { id: "http://.../verbs/viewed", display: "visualizou" },
      object: {
        id: "{{ randomUUID }}",
        definition: {
            name: "módulo de disciplina"
        }
      },
      place: { id: "{{ randomUUID }}", name: "{{ context }}" },
      context: {
        id: "{{ randomUUID }}",
        extensions: { timestamp: "{{ timestamp }}" },
      },
    },
  },
];
```

Listing 8.2: Moodle config transformation rules example

Having defined these configurations, we created a helper called `ConfigMatcherHelper`, which transforms the logs into statements based on the config key passed. This helper is responsible for transforming the generalizations, for example, `{{ randomUUID }}` into the appropriate data. This is accomplished by iterating over the logs and checking if any of the rules matches the pattern; if so, regex expressions are used to transform it into a Statement based on the defined transformation rules.

When the conversion process concludes, all the statements are persisted, making them always accessible through their associated id.

By running the use cases `VCI-UC-01` and `VCI-UC-02`, we have created and persisted all the statements that matched the defined rules. An example of a statement converted by this use case, can be observed in Listing 8.3.

```json
{
    "id": "877987d0-ca53-497c-805c-e5a928813bf5",
    "actor": { "id": "...", "name": "T3" },
    "verb": { "id": "...", "display": "visualizou" },
    "object": {
        "id": "...",
        "definition": { "name": "módulo de disciplina" }
    },
    "place": { "id": "...", "name": "Teste: Como eu vejo a ..." },
    "context": {
        "id": "...",
        "extensions": {
            "timestamp": "2021-01-14T09:54:00.000+00:00",
            "description": "...",
            "event": "Módulo de disciplina visualizado",
            "component": "Teste"
        }
    }
}
```


Listing 8.3: Generated statement example

8.2.1.3 VCI-UC-03 Create recipe

This use case intends to create an xAPI recipe that uses the statements and achieves the goal `VCI-G-01-02`. Before this use case could be executed, a source had to be uploaded and the logs had to be transformed into statements. Therefore, the required inputs are the name and the `sourceId` corresponding to the source from which we wish to generate the recipe.

All of the logic of this use case is handled directly by the Recipe entity, which handles the transformation of statements into a valid recipe. It accomplishes this by creating unique arrays for each of the xAPI elements. If the recipe is created, the use case ensures its persistence to be accessed if necessary. Listing 8.4 illustrates an example of an output recipe from this use case.

```
{
    "name": "test_recipe_name",
    "actors": ["T3", "S53"],
    "verbs": ["visualizou", "criou"],
    "objects": ["módulo de disciplina"],
    "places": ["Teste: Como eu ...", "Fórum: Sala de ..."] },
    "contexts": ["..."]
}
```

Listing 8.4: Generated recipe example

8.2.1.4 VCI-UC-04 Export source

This use case offers flexible options for exporting data. Users have the option to specify whether they wish to export the statements, the recipe, or both simultaneously. One can achieve optimal performance by utilizing this particular use case for retrieving the necessary information, as there is no necessity for remaking the statement or recipe. Through the `sourceId`, this use case locates both.

Additionally, it is possible to specify the option of exporting the result to a JSON file. This practice can prove highly advantageous for subsequent analysis and preservation of results related to a particular timeframe. Listing 8.5 shows an example of the output when we choose to export both the statements and the recipe.

```
{
    "source": {
                "id": "1b0dc0f7-2ba9-4892-a5b5-42990cbdef0d",
                "name": "logs_2019_2020.xlsx",
                "type": "file",
        },
        "statements": [{}],
    "recipe": {}
}
```

Listing 8.5: Export source with statements and recipe example

8.2.1.5 VCI-UC-05 Get statements and VCI-UC-06 Get recipe

This subsection describes two use cases, which are simple and straightforward due to their nature as list operations. Since the export source is more robust and flexible, these are rarely used.

Starting with `VCI-UC-05 Get statements`, this use case returns a list of statements. It accepts a filter object as input, and if nothing is passed, then the use case returns all available statements. Currently, it's possible to filter by `sourceId`. If this filter is passed, only the statements that correspond to the given source are returned. All of the logic from the statement is made by the repository implementation of the statements.

Then, `VCI-UC-06 Get recipe` returns the recipe information if it is found; otherwise, it returns a `null` value representing it was not found. Again, this use case's logic is centered on the recipe repository, which implements a `findById` function.

8.2.1.6 VCI-UC-07 Write local file

The necessity to write to a local file arose with the need to export the results of use case `VCI-UC-04 Export source`. To accomplish this, we save the file locally using the passed-in information as input. The received data is in object format; employing the language's functionalities enables us to transform this object into valid JSON. When this transformation is performed, we are in a position to save to a JSON file.

8.2.2 Learning Choreography Matcher

The Learning Choreography Matcher's primary responsibility is to identify choreographies in xAPI statements considering pre-defined behaviors that want to be identified. This primary responsibility matches the goal "VCI-G-02 Identify Virtual Choreographies in xAPI elements". To accomplish this, we must first concentrate on the sub-goals "VCI-G-02-01 Match xAPI elements into choreographies" and "VCI-G-02-02 Configurable match rules", which, how they were approached, will be described while explaining the use case "VCI-UC-08 Get Virtual Choreographies".

We adopted the same layering strategy as the Platform Data Extractor service, with domain, application, adapters, and infra layers.

Figure 8.4 illustrates the inputs and outputs. There are two types of inputs: local and external. Local inputs in our case are the statements identified in the Case Study 4, defined locally through configuration files. Then there are the external inputs, which in this case, are the xAPI statements generated by the antecedent service. After the match is performed, the main output of this service is the identified choreographies.

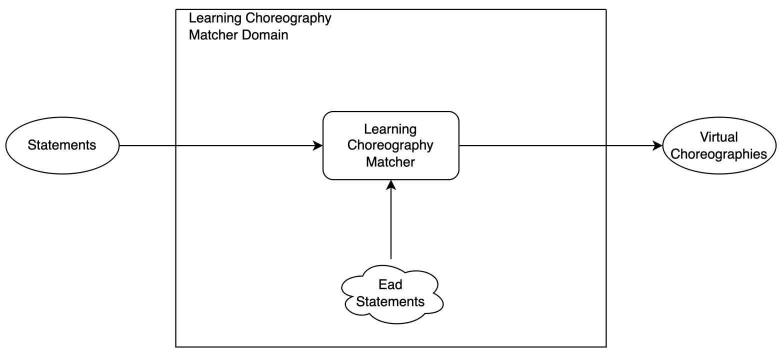

Figure 8.4: Learning Choreography Matcher diagram

In the following subsection, we will focus on explaining the decisions and the approach taken to achieve the previously mentioned goals, which are covered by this use case.

8.2.2.1 VCI-UC-08 Get Virtual Choreographies

The first issue that arose in this use case was how the service would recognize the choreographies we wish to identify in incoming statements.

To make the EdA statements configurable, configuration files were developed for them. When developing this solution, we focused just on the choreographies identified for the Forum component, as described in Section 2.1. Therefore, the examples in this section will be related to these.

Our configuration files are composed of three different configuration items:

- EdA Statement types: this holds the types of statements that we want to identify.
- Statement rules: using xAPI elements, this configuration specifies the steps to be identified in the logs.
- Generate statements: this configuration follows the same principles described in Use Case 8.2.1.2. Essentially, we create statements containing the conversion they will take if the criteria are satisfied.

Now that we understand the terminology, let's demonstrate a practical application. In particular, we will utilize the Choreography titled "Create topic and publish content". First, we define the EdA statement types, of which there is only one in this case, as shown in Listing 8.6.

```
enum EadForumStatementId {
  CREATE_TOPIC_AND_PUBLISH_CONTENT = "CREATE_TOPIC_AND_PUBLISH_CONTENT"
}
```

Listing 8.6: EdA Forum Statement Types

Then we have to define which are the rules that need to be matched for a statement to be considered from type `CREATE_TOPIC_AND_PUBLISH_CONTENT`. Those rules are an array of statement elements, that can be used to deepen or widen the search. For example in the rule from Listing 8.7, we are ensuring that the verb and object are matched, but if we wanted we could add other elements such as the place where those actions occur. The rules' order matter, if the set of statements matches the first rule, the remaining rules will not be examined. The implementation of this behavior took into account the solutions to the case study problem described in Section 4.4.1.

```
const FORUM_STATEMENT_RULES = {
  [EadForumStatementId.CREATE_TOPIC_AND_PUBLISH_CONTENT]: [
    { verb: "viewed", object: "módulo de disciplina" },
    { verb: "viewed", object: "discussão" },
    { verb: "create", object: "tópico" },
    { verb: "published", object: "conteúdo" },
  ],
  [EadForumStatementId.VIEW_DISCUSSION]: [],
};
```

Listing 8.7: Forum Statement Rules, for creating topic and publishing content

Using the same example as before, if we wanted to identify the same exact interactions but we just wanna identify the choreography if they happen in “Portugal”, we can define it. And, we can go even further and define individually each of the elements for each of the interactions, a good example would be if we wanna know if the module was viewed by user T3 and if the content was published by actor S3. That is possible with this approach, which makes it extremely flexible and robust for a lot of cases. An example of these two cases together that we mentioned can be seen on Listing 8.8. Using the same example as before, we can describe it if we want to find the same interactions but only want to identify the choreography when they happen in “Portugal”. And we can go even further and describe each part of each interaction separately. For example, if we want to know if user T3 viewed the module and if actor S3 published the content, we could do it. This method can do that, making it very flexible and valuable in many situations. Listing 8.8 is an example of these two cases together.

```
const FORUM_STATEMENT_RULES = {
  [EadForumStatementId.CREATE_TOPIC_AND_PUBLISH_CONTENT]: [
    {
        actor: "T3",
        verb: "viewed",
        object: "módulo de disciplina",
        place: "Portugal"
    },
    { verb: "viewed", object: "discussão", place: "Portugal" },
    { verb: "create", object: "tópico", place: "Portugal" },
    {
        actor: "S3",
        verb: "published",
        object: "conteúdo",
        place: "Portugal"
    },
  ],
  [EadForumStatementId.VIEW_DISCUSSION]: [],
};
```

Listing 8.8: Forum Statement Rules, with more complex rules

Now that our algorithm understands which statements we're interested in and what rules indicate that a given statement is of that type, we must define how we wish to transform the log statements found into the EdA statement. Here, we define one generate statement for each of the EdA statement types, introducing the concept of generate statements. If we wish to populate the actor name from the log statements, we need only designate the actor as `{{ username }}`, as shown in Listing 8.9. Here, the log statements are transformed into a single, well-defined EdA statement. In this particular case, we reduce four log statements to a single EdA statement.

```
const FORUM_GENERATE_STATEMENTS = {
    [EadForumStatementId.CREATE_TOPIC_AND_PUBLISH_CONTENT]:
        new Statement({
          id: "{{ randomUUID }}",
          sourceId: "ead-forum",
          actor: {
            id: "{{ randomUUID }}",
            name: "{{ actor }}",
          },
          verb: {
            id: "http://.../verb/create-and-publish",
            display: "criou e publicou",
          },
          object: {
            id: randomUUID(),
            definition: {
              name: "conteúdo",
            },
          },
          place: {
            id: "{{ randomUUID }}",
            name: "{{ place }}",
          },
        })
};
```

Listing 8.9: Forum Generate Statements, for creating topic and publishing content

Now that we have a well-defined configuration, we will provide an illustration of the various phases from raw log to EdA statement. The example in Figure 8.5 uses natural language to make the various phases easier to comprehend. Beginning with raw log data, the Platform Data Extractor converts them into xAPI statements. Then, these xAPI statements are compared to the rules specified in Listing 8.7; if a match is found, the statement is transformed according to the generate EdA Statement rule, or in code format Listing 8.9. The final output of this transformation is an EdA statement containing the actor and place specified in the generated rule.

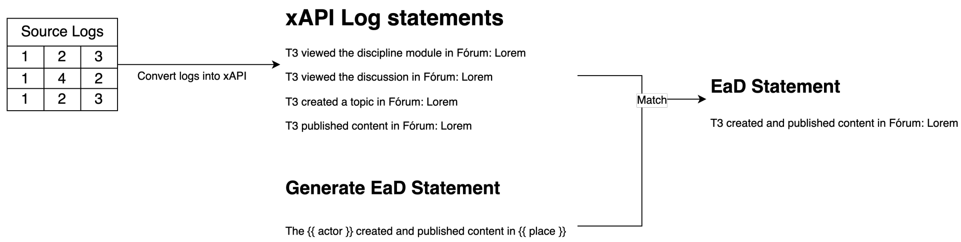

Figure 8.5: From log to EaD Statement lifecycle

After defining the configurations, it is time to describe the algorithm that utilizes them. After obtaining the input statements from the use case, it becomes necessary to process and organize them according to the specifications outlined in Section 4.4.2 in response to the second problem of the Case Study. Initial grouping of statements by actor prevents the occurrence of interactions from other users during analysis, which could result in false positives. The statements are then sorted by place to ensure that all user interactions involving a particular actor are consolidated in the same location. The statements are then arranged chronologically by timestamp to accurately reflect the order in which the interactions occurred.

After the data has been organized and sorted, the next stage involves iterating over the statements and determining if any of them adhere to the configuration-specific rules. In practice, this is done by systematically comparing each statement to the rules. If the first statement matches the first statement of a given rule, we proceed to the next statement and evaluate it in the same manner. This process of validation persists until a complete match or discrepancy is identified. If multiple rules exist, the cycle continues until no more rules or statements exist. However, suppose a group of statements corresponds to any of the rules. In this situation, we implement the solution proposed for the third problem in Section 4.4.3 in a simplified form. Notably, if the duration between the first and last interaction exceeds three hours, the match is discarded.

Lastly, when a collection of interactions matches any of the rules, we construct an EdA statement depending on the statement generating setup. The produced EdA statement is then used to build a one-statement Choreography. After analyzing all the statements, we create an in-depth output that identifies all the identified choreographies, as well as their corresponding types. We conclude the process of identifying choreographies from the provided logs by producing this output.

8.3 Virtual Choreography Interpreter Dashboard

The Virtual Choreography Interpreter Dashboard (VCID) is the frontend layer of the solution. This layer was developed to solve the `VCI-G-03 Represent Virtual Choreographies` goal. Due to time constraints, it was not possible to focus on the visualizations identified in the literature review, thus a basic dashboard was created to demonstrate the potential of the choreographies' visualizations.

This dashboard was developed using the frontend framework Angular[1] and used Apache Echarts[2] as the visualizations library.

In this section, we will analyze the use cases defined for this dashboard and provide an overview of the actions taken for each.

8.3.1 VCI-UC-09 Import source

This use case independently achieves the sub-goal `VCI-G-03-02 Data source import`, which is the platform's driving force. Without this, no visualizations or pages will contain data.

To solve this, we added a button on the header that, when clicked, opens the filesystem to choose the file to be uploaded. The button can be seen in Figure 8.6.

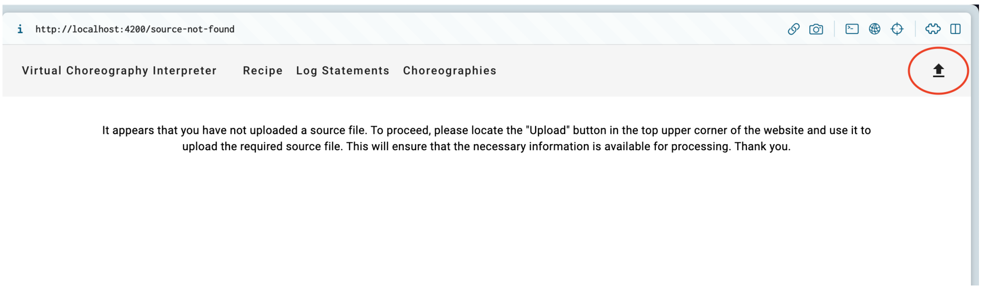

Figure 8.6: Dashboard source upload button

The VCDI only supports one source at a time, so every time a source is uploaded, all the visualizations and page data are refreshed to ensure the user is viewing the most up-to-date data.

8.3.2 VCI-UC-10 View recipe information

In the view recipe information use case, we provide the user a comprehensive view of how many and which are the constituents of the recipe generated from the uploaded source. This is particularly useful for analyzing each element in a more readable format than JSON.

This page was modified concerning the prototype to give it a more aesthetically pleasing appearance, with two sections. The first section represents the statistics of the recipe, and it is clear how many actors or places exist; they also serve as buttons for the second section. The button presently active is represented by a gray color. As shown in Figure 8.7, the second section consists of a card that displays the elements for the selected statistic in a list format.

[1] https://angular.io/
[2] https://echarts.apache.org/en/index.html

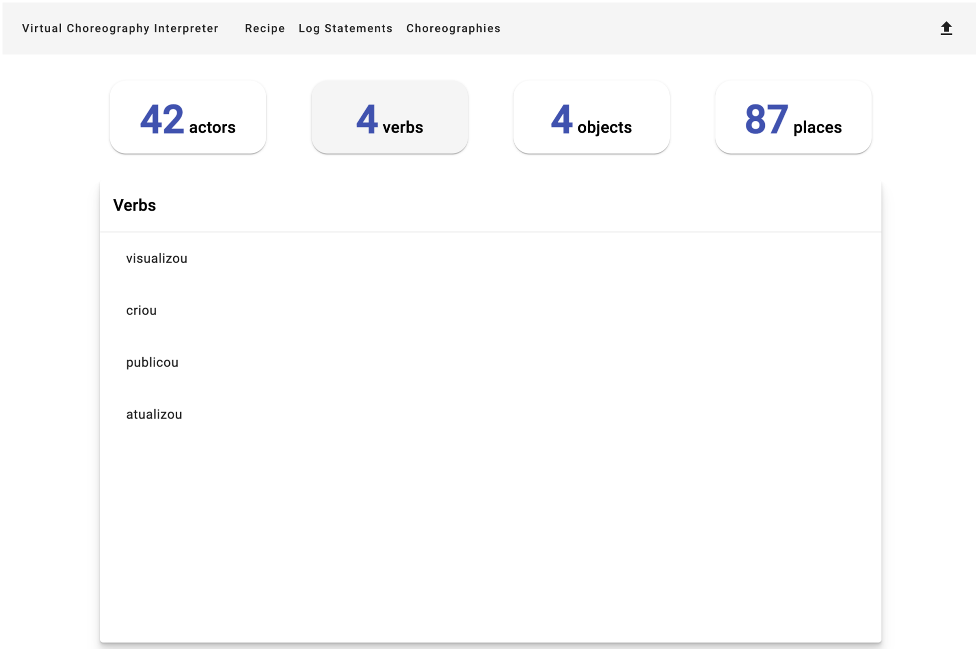

Figure 8.7: Dashboard recipe page

8.3.3 VCI-UC-11 View logs statements

This page was developed to allow dashboard users to view the logs uploaded in the format of xAPI statements. If necessary, this component can be used to trace the logs in the uploaded file.

Since we could have multiple lines, the primary element of this page is the table with pagination, which was a design decision. Figure 8.8 demonstrates the result of implementing this table.

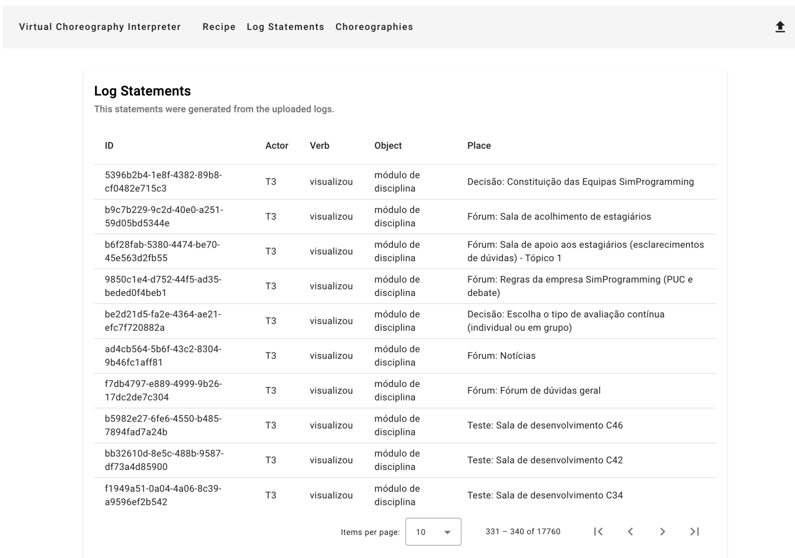

Figure 8.8: Dashboard log statements page

8.3.4 VCI-UC-12 View choreographies

During the selection process for the development of this proof-of-concept dashboard, the prioritization of virtual choreographies as a key output of this work was a crucial consideration.

It was selected as a tabbed component because this component is used when type-based content focus is desired. The user can alternate between all available choreography types using this method. Each time the tab changes, the card's content is updated to reflect the designated choreography type. The finalized page is shown in Figure 8.9.

The content is represented in the form of a list and shows the most important information for each choreography: statement in natural language, identifier of the choreography, and timestamp.

Figure 8.9: Dashboard choreographies page

8.3.5 VCI-UC-13 View dashboard visualisations

The dashboard is the final and most essential page in a visualization problem. As stated previously, since this project's focus was identifying choreographies, we showcase a simple prototype dashboard. During the design phase, three visualizations were specified, but four were developed during implementation.

Figure 8.10 displays the first two visualizations, which are the simplest. The first is a bar chart that represents the statistics of the recipe. This allows the user to quickly identify which xAPI elements are more prevalent in the recipe and facilitates comparisons. The second is a pie chart displaying the number of choreographies determined by type. This visualization makes comprehending which choreographies occur most frequently and seeing their distribution simpler.

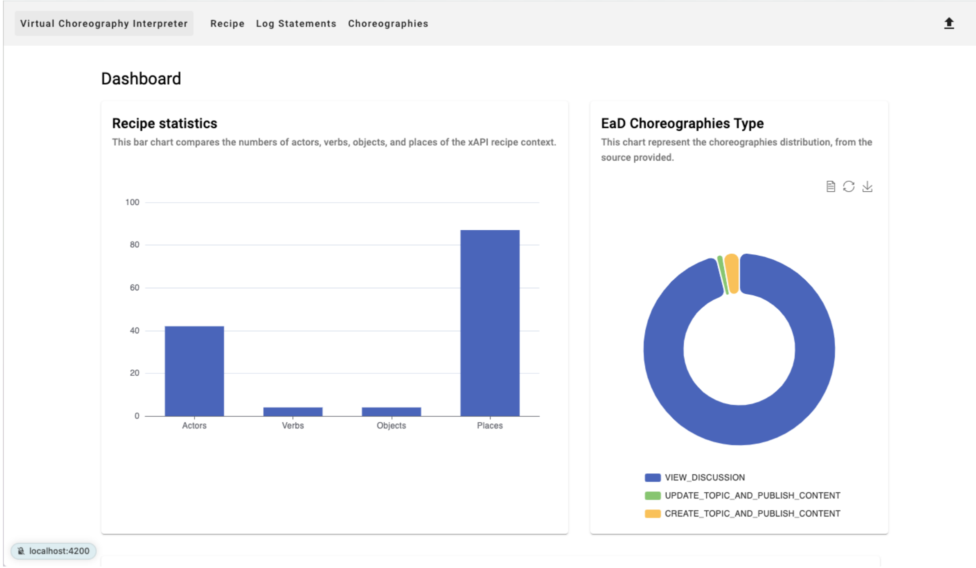

Figure 8.10: Recipe statistics and Choreographies type distribution charts

The third visualization is a bar chart representing the distribution of choreographies over time. To create this representation, the day was divided into four sections: 1) Morning: 06:00 - 11:59; 2) Afternoon: 12:00 - 17:59; 3) Evening: 18:00 - 21:59; and 4) Night: 22:00 - 05:59. By having this division and the timestamp, we can match each choreography to the appropriate group. Figure 8.11 depicts the visualization important to identify patterns.

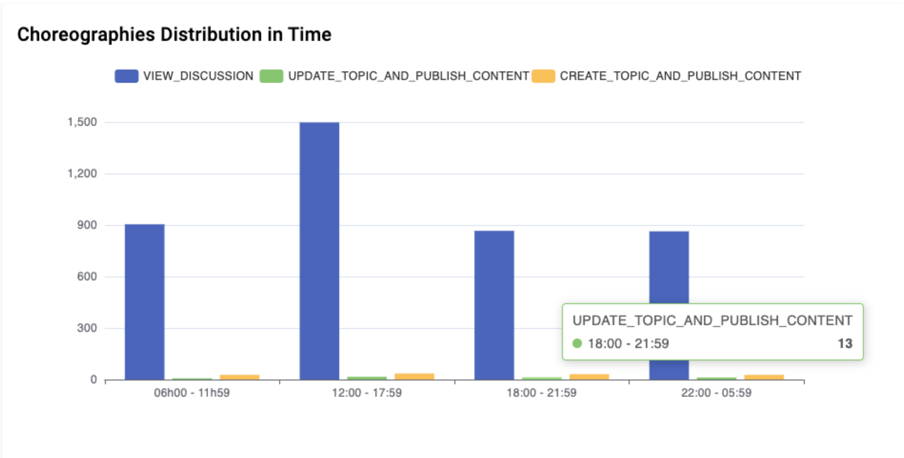

Figure 8.11: Choreographies distribution throughout time

Lastly, we have the added visualization that was not present in the original design. Considering that the SLR was centered on graph representation, we wished to demonstrate how graphs could be used to represent these elements, even if it was a simple representation. In Figure 8.12, we illustrate the relationships between recipe elements and their respective types. This visualization is beneficial for observing the relationships and element count for each xAPI element.

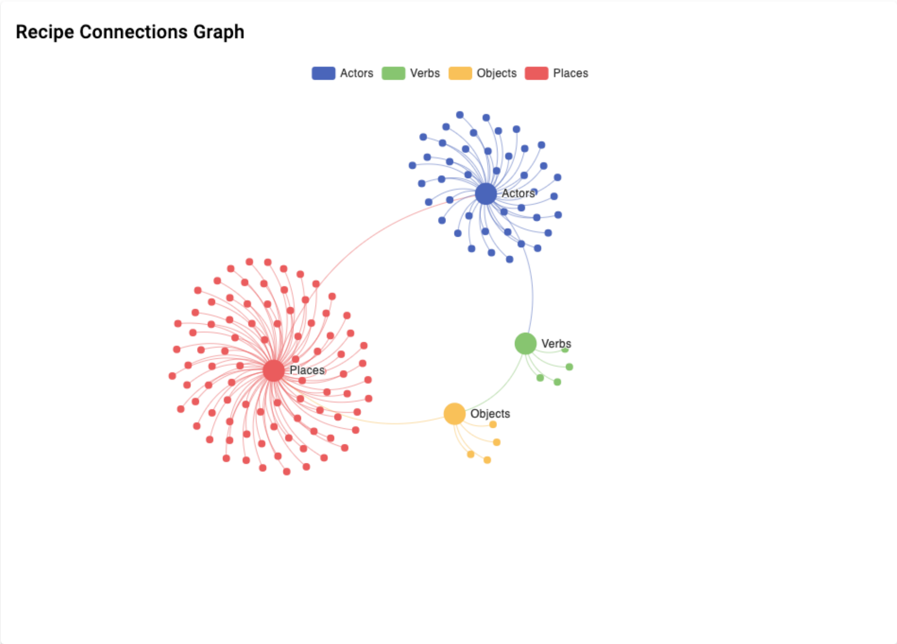

Figure 8.12: Recipe connections graph

Chapter 9

Results

This chapter will discuss the present research results by going through the final solution with the goals and use cases. Subsequently, the outcomes throughout this dissertation will be matched against the research objectives defined in Section 1.4. Both of these sections will include the results and discuss the potential enhancements and constraints that were identified during evaluations of the prototypes throughout the iterative process.

9.1 Requirements Results

This section is subdivided into the three implemented services, each with its own use cases and goals. Separating the services based on their respective use cases and goals makes it possible to conduct a comprehensive review of the outcomes, enabling a focused evaluation of the individual contributions and areas requiring further improvement.

9.1.1 Platform Data Extractor

For this service, we had one main goal, three subgoals, and nine use cases with different priorities as defined in Section 6.3.3. Figure 9.1 depicts in a tree view the connections between them.

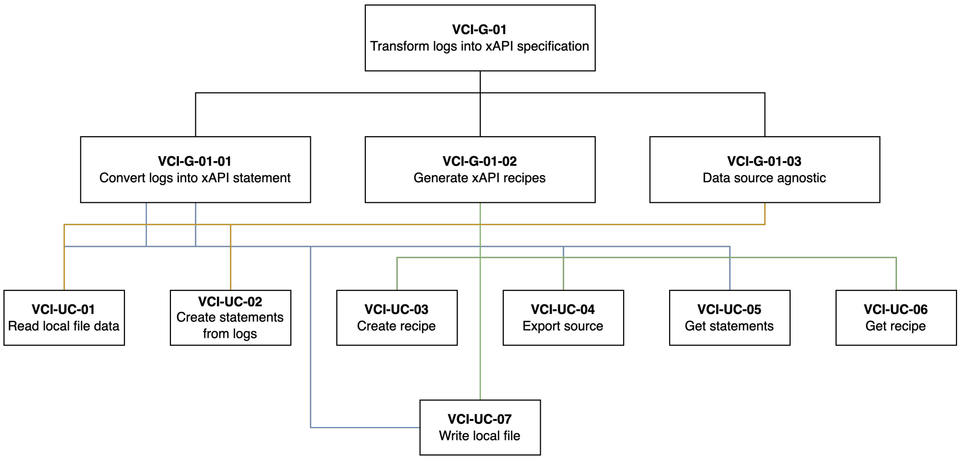

Figure 9.1: Platform Data Extractor goals and use cases connections

We will begin with use cases, the smallest unit of the requirements. The results will be described for each, as well as any enhancements or limitations:

(a) **Use case** `VCI-UC-01 Read local file data` (Completed)

Results This use case enables the user to convert the raw data from a Moodle file into a format that the service can understand: JSON objects. Since a strategy pattern was created, these files that the use case can read are highly flexible; if we need to read another file format, we simply need to construct a new strategy for that particular file format.

(b) **Use case** `VCI-UC-02 Create statements from logs` (Completed)

Results This use case allows the transformation of logs in JSON format into xAPI statements. By introducing configuration files through which you define the set of rules used to transform those, an agnostic approach was achieved. In this instance, we defined the configuration for the Case Study file. These rules are adaptable and permit both specific and general applications.

Improvements/Limitations In our approach, we developed rules for transforming statements using the statement-based method. This method was chosen due to its simplicity and adaptability. Similar to the approach adopted by Lacet *et al.* [61], an alternative procedure could aim to develop a platform recipe. Using a recipe could be advantageous because its format is more user-friendly than configuration files and it adheres to the xAPI standard. This recipe-based strategy may provide insights and enhance future studies.

(c) **Use case** `VCI-UC-03 Create recipe` (Completed)

Results The user can create recipes. To accomplish this, we analyze the uploaded source with the logs already converted to xAPI statements. While generating the recipe, we verify the unique identifiers of each element to ensure that they are all distinct.

(d) **Use case** `VCI-UC-04 Export source` (Completed)

Results In this use case, the user can request the export of a source. It was not specified during the requirements phase, but we allow the user to choose what to export. Currently, it can export either statements or recipes, or both. Finally, we allow the user to export the results to a local JSON file, which can be useful for saving and analyzing the results later.

Improvements/Limitations This use case can be expanded to include the ability to export additional source-related data, such as raw JSON logs.

(e) **Use case** `VCI-UC-05 Get statements` (Completed)

Results The service allows the user to request the statements. We currently enable the user to retrieve all available statements and filter them by source.

Improvements/Limitations It is possible to investigate additional filters that may be useful when retrieving statements. For instance, filtering by the statements made by a specific user or in a specific place.

(f) **Use case** `VCI-UC-06 Get recipe` (Completed)

Results The user is able to get information about a specific recipe from a given source.

(g) **Use case** `VCI-UC-07 Write local file` (Completed)

Results Using this use case, the system can export a javascript JSON object to a JSON file. Currently employed by the use case `VCI-UC-04 Export source` when the user requests to save the results to a file.

Improvements/Limitations It can export additional files within the service, such as the uploaded source's transformed JSON logs.

Now focusing on the goals, as represented in Figure 9.1, we have three subgoals: `VCI-G-01-01 Convert logs into xAPI statement`, `VCI-G-01-02 Generate xAPI recipes`, and `VCI-G-01-03 Data source agnostic`. Considering that all use cases have been completed, it is possible to assert that all objectives have been attained. Even though this service may have some additional features, it performs effectively its primary function of converting logs into xAPI specification (`VCI-G-01`).

9.1.2 Learning Choreography Matcher

The Learning Choreography Matcher had one main goal, two subgoals, and one use case. The connections between them are represented in Figure 9.2.

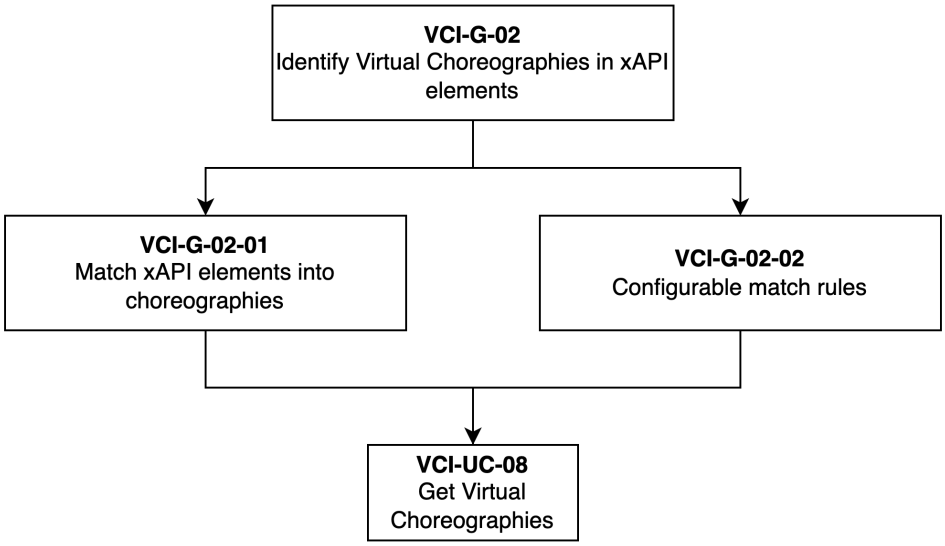

Figure 9.2: Learning Choreography Matcher goals and use cases connections

Even though this service only has one use case, it is highly complex and solves the goal defined for this service.

Use case `VCI-UC-08 Get Virtual Choreographies` (Completed)

Results The user submits xAPI statements, and if the service possesses the choreographies that the user intends to identify, the service will systematically analyze each statement to determine the presence of indicated choreographies. If choreographies are determined effectively, the user will receive them.

Improvements/Limitations Potential advancements in this use case present a wide range of opportunities for future research. The first approach to be investigated is using xAPI recipes, which could be employed instead of statements to compare the results produced by our current method. As configurations are defined using JSON notation, an enhancement could include the ability to add new configurations at runtime. In addition, according to our current methodology, all interactions must occur within a three-hour window for a choreography to be considered valid. This rigorous approach may not provide the most accurate depiction of the duration required for various forms of choreography. This value could be calculated by analyzing the average time needed to execute the specific choreography type. This adaptive strategy could provide more precise and adaptable timeframes for choreography validation, thereby enhancing the system's overall efficacy.

By analyzing the obtained results, it is possible to conclude that all stated objectives have been met. Enhancements that have the potential to advance the solution have been identified.

9.1.3 Virtual Choreography Interpreter Dashboard

In the dashboard solution, we have one main goal, two subgoals, and five use cases, as depicted in Figure 9.3. The main focus of this dashboard is the visualizations.

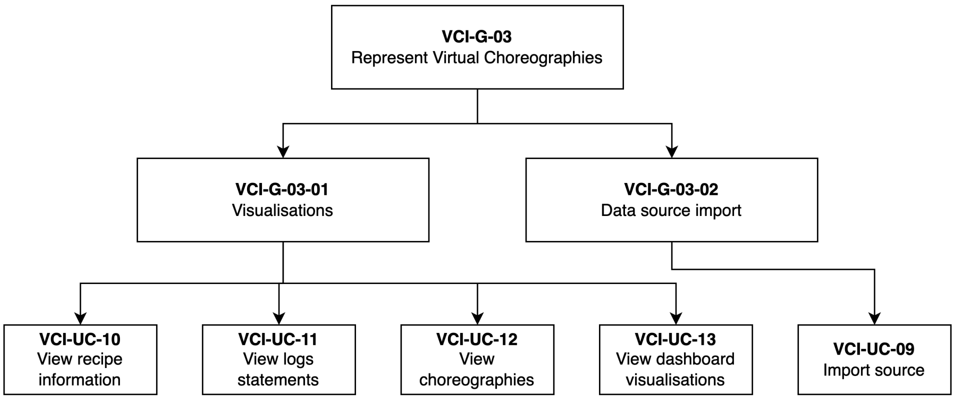

Figure 9.3: Platform Data Extractor goals and use cases connections

Beginning with the use cases, we will examine each to determine the outcomes and possible improvements.

(a) **Use case** `VCI-UC-09 Import source` (Completed)

Results The user can upload a source directly from the page's header. This upload will give all the other pages information, making it a high-priority task. When a source is uploaded with success, the page will refresh, and a successful alert message will appear.

Improvements/Limitations We currently accept one source at a time. This layout could be modified in the future, and the addition of multiple source uploads could be extremely useful for visualizing various classes of courses without the need to constantly submit a new source.

(b) **Use case** `VCI-UC-10 View recipe information` (Completed)

Results With this use case implemented, the user can access all recipe-related information. This page contains more functionality than when the use case was defined, as we wanted to display all recipe statistics at a glimpse, such as the number of actors or verbs. These statistics are interactive, and when clicked, the page's focus shifts to a list of the element that was selected.

Improvements/Limitations Having increased functionality and interconnections between all elements. An example would be to make all list elements interactive, so that when a user clicks on an actor, the associated statements are displayed.

(c) **Use case** `VCI-UC-11 View logs statements` (Completed)

Results By implementing this use case, the user can access all information regarding xAPI log statements. This page contains a table that collect all the data generated by the uploaded logs. This table includes pagination because it may comprise thousands of entries.

Improvements/Limitations Custom filters would provide the user with greater flexibility. Establishing connections between the pages would also be significant; for instance, if a statement is chosen and it is part of a choreography, we could be redirected to that choreography.

(d) **Use case** `VCI-UC-12 View choreographies` (Completed)

Results This use case allows the user to view all choreographies that the Learning Choreography Matcher identified. The information is presented to him as a stack element, with the selected choreography type taking center stage. For each of the identified choreographies, the generated EdA statement, identifier, and timestamp are displayed.

Improvements/Limitations The functionality of the provided information would be a significant enhancement. Alternatively to a simple list, you could divide the document into distinct sections and graphically depict the choreographies and statements that comprise it.

(e) **Use case** `VCI-UC-13 View dashboard visualisations` (Completed)

Results This dashboard contains visualizations that assist the user in recognizing specific patterns and data. The user can view a bar chart containing the recipe's statistics. Afterwards, a pie chart displaying the number of choreographies identified for each type. The distribution of choreographies across time is another visualization that divides the day into four groups and shows the types of choreographies that occur during each group. Finally, we have a graph containing all the connections between the recipe elements, allowing us to represent xAPI elements using this type of representation.

Improvements/Limitations This interface consists in a proof-of-concept; it opens up the possibilities for applying the Virtual Choreography Interpreter's elements. Numerous visualizations are possible. Exploring the results of the SLR centered on graphs and clustering would be a suitable starting point for the evolution of this visualization.

This dashboard meets all the defined objectives and use cases and demonstrates the possibilities that arise from employing this solution's backend.

9.2 Research Objectives Results

9.2.1 RO1: Conduct a comprehensive analysis to ascertain the diverse spectrum of interactions and behaviors within LMS/CMS contexts, which have been investigated in the existing literature.

As stated in Section 6.2 of the SLR Learnings, we reviewed the behaviors and interactions that can be represented using graphs and clustering techniques. Due to our emphasis on the identification of choreographies from the part of the log, we did not make full use of the results that we identified with the four SLR questions. Nonetheless, this objective was met with the response to the first question: "TAQ1: What types of LMS/CMS behaviors/interactions are represented by graphs?"

This question, answered in Section 3.3.1, identified seven distinct interactions/behaviors observed in LMS or CMS contexts. With these definitions and practical applications of each paper, identified in Section 6.2, we were able to focus on some of these interactions while analyzing the Case Study Moodle data in quest of valuable choreographies within it. The process was made more valuable by analyzing the research literature prior to analyzing the data, and focusing on the patterns identified by other researchers can make comparisons of results more feasible.

Other than the initial SLR question, none of the others were explored, despite the fact that they have empirical value and serve as an excellent starting point for further research. The second SLR question centered on which clustering techniques can be used to identify behaviors/interactions, which is an extension of this dissertation's solution to identify which students exhibit similar behavior while using the LMS. The third question identified in the literature how graphs can be used to represent the behaviors/interactions, there were multiple approaches and different types of graphs used, which can be extremely useful for exploring the findings and creating a valuable dashboard with the interactions identified by the service developed in this dissertation. The fourth and final question addressed how clustering techniques can be used to represent behavior/interactions. One example would be differentiating learning patterns based on the results of virtual choreography.

This objective was achieved, and in the process, we conducted a systematic review of additional topics which can lead to additional improvements and accomplishments in employing virtual choreographies in more contexts.

9.2.2 RO2: Formulate a methodical approach to delineate valuable choreographies to be extracted from the data logs within LMS contexts.

Identifying which choreographies have value to identify in the context of our Case Study was one of the integrating parts of our problem. This is crucial because we cannot automatically identify something on the logs if we do not know what we need to visualize. To comprehend the possibilities, we conducted the SLR to determine what the research community had identified.

With this information, we could begin our analysis of the Case Study, described in Chapter 4. The initial stage was to comprehend the business and the data's context. After that, we began by understanding the provided data, and then we were able to prepare the data. To prepare the data, a script was developed to simplify the data cleansing procedure. We were then able to analyze the resulting and cleansed data to determine which interactions we desired to identify. As a result, we limited our search to the Assignment and Forum components, as they contained the majority of interactions identified in the SLR. Lastly, we could identify seven choreographies. However, we only focused on the three identified for the Forum component during implementation. This means that the choreographies already identified for the Assignment component can be used in future research initiatives that yield a larger quantity of data results.

Ultimately, our method can be applied to other business contexts and data types, thereby accomplishing our goal of formulating a methodical approach to identifying choreographies.

9.2.3 RO3: Design and implement an efficient solution that enables the identification of virtual choreographies from the data logs present in LMS environments.

Throughout the development of the dissertation, an extensive amount of focus was placed on this objective, gathering more results and potential for improvement. To achieve this objective, a solution to transform logs into the xAPI specification, which will then be analyzed to identify virtual choreographies, has been developed. This is accomplished by two previously mentioned services: Platform Data Extractor and Learning Choreography Matcher.

Platform Data Extractor was designed to convert all platform log data into xAPI specification notation. To make it platform-independent, transformation rules were introduced, allowing the user to define the principles for transforming particular types of files. It is powerful to enable the construction of configurations to enable the transformation of specific rows; for instance, it is possible to transform all rows from a specific user according to *X* rule. Important to note that if the transformation rules are already defined for the most commonly exported file types, such as when the file is exported directly from Moodle, there is no need to define new rules unless the user wishes to construct their own unique rules. In addition, the service offers additional functionalities, such as producing a recipe from the statements, exporting all the elements from a specified source, and considering multiple filters, among others.

Although this service is robust and able to convert logs into statements, the evaluation of the prototypes revealed some improvements and limitations. The primary one is that the statement approach with configuration files isn't directly supported by the xAPI specification, which can be a drawback. Therefore, future research could investigate an alternative approach, as demonstrated by Lacet et al. (2019) [61], which involves creating a platform recipe, which could offer advantages due to its user-friendly format and adherence to the xAPI standard. In future investigations, examining the recipe-based method and comparing its results to the current statement-based method may yield valuable insights and further improve the methodology. The possibility of adding additional filters and eliminating a source after the transformation has been completed could be other improvements.

The Learning Choreography Matcher identifies the choreographies from the statement logs. The identification of choreographies is accomplished through a configuration, as detailed in Section 8.2.2. This adaptable configuration allows the user to tailor the virtual choreography identification procedure to his particular requirements. By selecting this approach, the service is nearly plug-and-play; we only need to define the choreographies we wish to identify, and it is set to go.

In the validation of the prototypes, some points for future analysis and improvements arose: 1) explore the use of xAPI recipes, as mentioned before, which could be used instead of relying directly on statements and compare the results in both approaches; 2) Enable configurations to be added during run-time, since at this moment the configurations are hard-coded and can be only added by the owners of the code; 3) improve our method to consider if a choreography is valid or it was a mismatch. We currently use a fixed amount of time (three hours) which can still lead to valid choreographies to be excluded. For more accurate results, this value could be calculated by analyzing the average time required to complete the specific choreography type and adding a control value.

The development of our backend service Virtual Choreography Interpreter allowed us to achieve this objective. Even though it is a prototype, it has produced valuable results that can accelerate the use of virtual choreographies in additional contexts.

9.2.4 RO4: Design and develop a dashboard that showcases potential visualizations using the output generated from the identification of virtual choreographies.

Lastly, in the domain of visualization, we created a dashboard to display the Virtual Choreography Interpreter's output. As a proof-of-concept dashboard, the primary objective of this dashboard was to construct some visualizations with the service's outputs.

The user can import a source directly through the website's interface, which provides a great deal of convenience. The interface then initiates the display of pertinent information. Despite this development, we have identified a limitation in our system, as it only allows the upload of a single source at a time. The implementation of asynchronous uploads could be a prospective system enhancement. In addition, modifying the layout to accommodate a list of sources rather than the current single-source layout could provide a more comprehensive view and user-friendly experience when managing multiple data sources.

The user is then presented with a page holding the recipe's information with statistics about it. When a user clicks on one of the elements, the focus and information are shifted to the corresponding information. Since the information is only accessible in view mode, it was identified as an area for future development to establish relationships with other pages. For instance, if an actor is selected, the user should be redirected to a page containing all his statements.

Additionally, the user can view his uploaded logs in xAPI statements format. It was presented as a table because the number of statements generated by this type of transformation may exceed the user's ability to analyze them without pagination. Moreover, additional enhancements were identified: 1) Add filters to the table to facilitate the search for specific elements, and 2) establish connections between other pages, for example, if a statement is selected and it is associated with a choreography, the page should redirect to the choreography.

In addition, we display to the user the choreographies identified by the service. This page displays the identified choreography types, and when the user selects one of the types, all choreographies identified for that type are displayed. Similar to the other pages, we identified enhancements such as redesigning the layout and incorporating distinct sections that present choreographies and statements in a more representative manner than a simple list.

Lastly, the interface allows users to examine visualizations that can aid in decision-making. We provide visualizations, including recipe statistics, a comparison between the number of choreographies identified for each type, the distribution of choreographies over time, and a graph illustrating all the connections between the recipe elements. We attempted to provide the user with various visualizations, but there are still many visualizations that can be investigated, such as using graphs in a more in-depth manner, as was explored during the literature review.

Chapter 10

Conclusions and Future Work

The purpose of this chapter is to present on the achieved results and identify what was accomplished, areas for development, and potential limitations. Finally, the Future Work section will consolidate all identified future work.

10.1 Conclusion

This dissertation aimed to transform the data from LMS/CMS logs into virtual choreographies and demonstrate how they can be visualized in dashboards to assist decision-making. In this chapter, we will check against the research objectives identified in Section 1.4 and see what was accomplished and what needs more work.

RO1 Conduct a comprehensive analysis to ascertain the diverse spectrum of interactions and behaviors within LMS/CMS contexts, which have been investigated in the existing literature: An SLR was conducted to first determine the behaviors and interactions that may be represented using graphs and clustering techniques. This investigation identified seven themes concerning interactions and behaviors in the context of LMS/CMS. These themes were investigated further using Moodle-derived case study data. This analysis of the currently available literature made identifying patterns and comparing results easier. The secondary question sought to determine the value of clustering techniques for identifying behaviors and interactions. Simultaneously, the third question analyzed the capability of graph representations to depict behaviors and interactions. The fourth question investigated how clustering techniques can depict behaviors and interactions, such as identifying learning patterns through virtual choreographies. While the primary focus of this dissertation is on examining the results pertinent to the first question, the remaining discoveries can potentially guide future endeavors in visualizing virtual choreographies.

RO2 Formulate a methodical approach to delineate valuable choreographies to be extracted from the data logs within LMS contexts: This work presents a methodology for analyzing a log file to identify valuable choreographies. This identification procedure follows the initial steps from CRISP-DM model's. As stated in this work, the interactions between these elements can take various forms. The results will likely be excessively general and insignificant without an explicit definition of what constitutes identification. We identified seven choreographies in the Case Study data by aligning the data with what we inferred from the SLR to be significant. Several of these were utilized during the implementation phase, further demonstrating the applicability and effectiveness of our proposed methodology.

RO3 Design and implement an efficient solution that enables the identification of virtual choreographies from the data logs present in LMS environments: Based on the insights gained from the literature and the case study analysis, a Requirements Engineering process was initiated. This resulted in developing specific goals and use cases for the provided objective. As part of this procedure, the Virtual Choreography Interpreter backend system was developed. This system comprises two services: the Platform Data Extractor and the Learning Choreography Matcher. The first service, the Platform Data Extractor, is responsible for the platform-agnostic conversion of data from logs into xAPI statements. These statements are then used to generate an xAPI recipe representing all the logs' accessible elements. The second service, the Learning Choreography Matcher, is then prepared to recognize the choreographies from the generated statements. The choreographies desired for recognition within the statements can be configured, allowing for high adaptability to accommodate various user requirements. These two services enable the identification of choreographies from LMS logs. This adaptable approach allows for the extension of our system to a variety of additional use cases, thereby increasing its overall utility.

RO4 Design and develop a dashboard that showcases potential visualizations using the output generated from the identification of virtual choreographies: After identifying choreographies, a proof-of-concept dashboard was constructed to illustrate a variety of potential visualization strategies that could be applied to the data provided by the Virtual Choreography Interpreter. Users can submit a source file and see the elements of the uploaded logs (recipe), the logs transformed to xAPI statements, and the virtual choreographies identified in the logs. Lastly, the dashboard displays visualizations including a comparison of the number of recipe elements, the number of choreographies designated by type, the distribution of choreographies completed throughout the day's intervals, and a graph view of the recipe elements. Even though the interface provides valuable information, its proof-of-concept nature makes it unsuitable for deployment in real-world scenarios.

All established goals were ultimately achieved. This dissertation makes a significant contribution by presenting the results of an SLR, describing a procedure for identifying valuable choreographies, proposing a backend service that automatically determines virtual choreographies from logs, and introducing a dashboard for displaying this data. This work paves the way for the identification and visualization of virtual choreographies at a higher level of complexity.

10.2 Limitations and Future Work

No system is perfect, and neither is this dissertation delivery. Throughout the evaluation phases, some limitations and future work were identified.

The most significant limitations identified throughout the development of this work were:

- The current approach can't detect multi-user choreographies since the interactions are grouped by user.
- To be considered as a choreography, interactions must occur within a predetermined time frame. This current method cannot detect a valid choreography that the student began in the morning and resumed in the evening.
- The current rules must be pre-defined and can't be added at runtime.
- The dashboard only supports uploading one source at a time, which means that users with multiple classes can't analyze simultaneously.

Throughout the development of this work and in consideration of its limitations, the following future work was identified:

- The solution must be further validated with additional users to determine the intuitiveness and simplicity of the configuration creation process.
- Use the Systematic Literature Review findings to develop a valuable dashboard by exploring graphs and clustering in the visualization process.
- Another variation of virtual choreographies is the identification of common errors. The algorithm developed for the identification can be extended to identify common errors automatically while interacting with the system. For example, identify users who start creating a topic but never finish.
- Expand the use cases of the Virtual Choreography Interpreter backend to enable more operations with the resulting data.
- Explore the use of asynchronous processes to analyze the incoming data. With multiple users uploading massive files, the concept of queues needs to be raised.
- Explore other criteria to understand if a choreography is valid.

References

[1] Timothy Yudi Adinugroho, Reina, and Josef Bernadi Gautama. Review of Multi-platform Mobile Application Development Using WebView: Learning Management System on Mobile Platform. *Procedia Computer Science*, 59:291–297, January 2015.

[2] Amir Ahmad and Lipika Dey. A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering*, 63(2):503–527, November 2007.

[3] Yahya Al-Ashmoery, Hisham Haider, Adnan Haider, and Najran Nasser. Learning analytics toolset for evaluating students' performance in an E-learning Platform. In *2021 International Conference of Modern Trends in Information and Communication Technology Industry (MTICTI)*, pages 1–10, December 2021.

[4] Raghad Al-Shabandar, Abir Jaafar Hussain, Panos Liatsis, and Robert Keight. Analyzing Learners Behavior in MOOCs: An Examination of Performance and Motivation Using a Data-Driven Approach. *IEEE Access*, 6:73669–73685, 2018. Conference Name: IEEE Access.

[5] Abdulaziz Aldiab, Harun Chowdhury, Alex Kootsookos, Firoz Alam, and Hamed Allhibi. Utilization of Learning Management Systems (LMSs) in higher education system: A case review for Saudi Arabia. *Energy Procedia*, 160:731–737, February 2019.

[6] Liaqat Ali, Marek Hatala, Dragan Gašević, and Jelena Jovanović. A qualitative evaluation of evolution of a learning analytics tool. *Computers & Education*, 58(1):470–489, January 2012.

[7] Solomon Arason. The impact of digitisation on teaching and learning. *Educor Multidisciplinary Journal*, 3(1):102–115, December 2019. Publisher: Educor Holdings (EH).

[8] Hilary Arksey and Lisa O'Malley. Scoping studies: towards a methodological framework. *International Journal of Social Research Methodology*, 8(1):19–32, February 2005. Publisher: Routledge _eprint: https://doi.org/10.1080/1364557032000119616.

[9] Jonathan Arnowitz, Michael Arent, and Nevin Berger. *Effective Prototyping for Software Makers (Interactive Technologies)*. December 2006.

[10] Len Bass, Paul Clements, and Rick Kazman. Software Architecture In Practice. January 2003.

[11] Alex Becheru, Andreea Calota, and Elvira Popescu. Analyzing students' collaboration patterns in a social learning environment using StudentViz platform. *Smart Learning Environments*, 5(1):18, September 2018.

[12] Maraza-Quispe Benjamín, Ricardo Quispe-Figueroa, and Manuel Valderrama-Solis. Dashboard proposal implemented according to an analysis developed in the KNIME platform. *World Journal on Educational Technology: Current Issues*, 13:617–634, October 2021.

[13] Adel Bessadok, Ehab Abouzinadah, and Osama Rabie. Exploring students digital activities and performances through their activities logged in learning management system using educational data mining approach. *Interactive Technology and Smart Education*, ahead-of-print(ahead-of-print), January 2021.

[14] Robert Bodily and Katrien Verbert. Review of Research on Student-Facing Learning Analytics Dashboards and Educational Recommender Systems. *IEEE Transactions on Learning Technologies*, 10(4):405–418, October 2017. Conference Name: IEEE Transactions on Learning Technologies.

[15] Sebastian K. Boell and Dubravka Cecez-Kecmanovic. What is an Information System? In *2015 48th Hawaii International Conference on System Sciences*, pages 4959–4968, January 2015. ISSN: 1530-1605.

[16] O. Bohl, J. Scheuhase, R. Sengler, and U. Winand. The sharable content object reference model (SCORM) - a critical review. In *International Conference on Computers in Education, 2002. Proceedings.*, pages 950–951 vol.2, December 2002.

[17] Michael Brown. Seeing students at scale: how faculty in large lecture courses act upon learning analytics dashboard data. *Teaching in Higher Education*, 25(4):384–400, May 2020. Publisher: Routledge _eprint: https://doi.org/10.1080/13562517.2019.1698540.

[18] HongYun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637, September 2018. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

[19] Francesca Carmagnola, Federica Cena, and Cristina Gena. User model interoperability: a survey. *User Modeling and User-Adapted Interaction*, 21(3):285–331, August 2011.

[20] Paula Carroll and Arthur White. Identifying Patterns of Learner Behaviour: What Business Statistics Students Do with Learning Resources. *INFORMS Transactions on Education*, 18(1):1–13, September 2017. Publisher: INFORMS.

[21] Fernando Cassola, Leonel Morgado, António Coelho, Hugo Paredes, António Barbosa, Helga Tavares, and Filipe Soares. Using Virtual Choreographies to Identify Office Users' Behaviors to Target Behavior Change Based on Their Potential to Impact Energy Consumption. *Energies*, 15(12):4354, January 2022. Number: 12 Publisher: Multidisciplinary Digital Publishing Institute.

[22] Vaggos Chatziafratis, Rad Niazadeh, and Moses Charikar. Hierarchical Clustering with Structural Constraints. In *Proceedings of the 35th International Conference on Machine Learning*, pages 774–783. PMLR, July 2018. ISSN: 2640-3498.

[23] Min Chen, Shiwen Mao, and Yunhao Liu. Big Data: A Survey. *Mobile Networks and Applications*, 19(2):171–209, April 2014.

[24] Betty H.C. Cheng and Joanne M. Atlee. Research Directions in Requirements Engineering. In *Future of Software Engineering (FOSE '07)*, pages 285–303, May 2007.

[25] Doug Clow. The learning analytics cycle: closing the loop effectively. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, LAK '12, pages 134–138, New York, NY, USA, April 2012. Association for Computing Machinery.

[26] Ruth Cobos and Juan Carlos Ruiz-Garcia. Improving learner engagement in MOOCs using a learning intervention system: A research study in engineering education. *Computer Applications in Engineering Education*, 29(4):733–749, 2021. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cae.22316.

[27] Alistair Cockburn. Hexagonal architecture.

[28] Alistair Cockburn and Norges Bank. Structuring Use cases with goals. December 1997.

[29] Heather L. Colquhoun, Danielle Levac, Kelly K. O'Brien, Sharon Straus, Andrea C. Tricco, Laure Perrier, Monika Kastner, and David Moher. Scoping reviews: time for clarity in definition, methods, and reporting. *Journal of Clinical Epidemiology*, 67(12):1291–1294, December 2014.

[30] James O. Coplien and Trygve Reenskaug. *Agile Software Architecture: Chapter 2. The DCI Paradigm: Taking Object Orientation into the Architecture World*. Morgan Kaufmann, November 2013.

[31] Marielle Dado and Daniel Bodemer. A review of methodological applications of social network analysis in computer-supported collaborative learning. *Educational Research Review*, 22:159–180, November 2017.

[32] Giuseppe D'Aniello, Massimo De Falco, Matteo Gaeta, and Mario Lepore. A Situation-aware Learning System based on Fuzzy Cognitive Maps to increase Learner Motivation and Engagement. In *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8, July 2020. ISSN: 1558-4739.

[33] Mouna Denden, Ahmed Tlili, Maiga Chang, Ted Krahn, Rita Kuo, Mourad Abed, and Mohamed Jemni. Can we predict learners' personalities through their behavioural patterns? A pilot study using Behaviour Analytics-Moodle plugin. In *2021 8th International Conference on ICT & Accessibility (ICTA)*, pages 1–5, December 2021. ISSN: 2379-4402.

[34] Urvashi Desai, Vijayalakshmi Ramasamy, and James D. Kiper. A Study on Student Performance Evaluation using Discussion Board Networks. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, SIGCSE '20, pages 500–506, New York, NY, USA, February 2020. Association for Computing Machinery.

[35] Ann Hill Duin and Jason Tham. The Current State of Analytics: Implications for Learning Management System (LMS) Use in Writing Pedagogy. *Computers and Composition*, 55:102544, March 2020.

[36] Oscar Esteban Bajo, Ishari Amarasinghe, Nicolás Felipe Gutiérrez-Páez, and Davinia Hernández-Leo. Using Process Mining Techniques to Discover the Collective Behaviour of Educators in a Learning Community Platform. In Lung-Hsiang Wong, Yugo Hayashi, Cesar A. Collazos, Claudio Alvarez, Gustavo Zurita, and Nelson Baloian, editors, *Collaboration Technologies and Social Computing*, Lecture Notes in Computer Science, pages 175–189, Cham, 2022. Springer International Publishing.

[37] Rebecca Ferguson. Learning analytics: drivers, developments and challenges. *International Journal of Technology Enhanced Learning*, 4(5/6):304–317, January 2012.

[38] Thomas C. Ford, John M. Colombi, Scott R. Graham, and David R. Jacques. Survey on Interoperability Measurement. 2007. Section: Technical Reports.

[39] Stephanie Frost and Gord McCalla. A Planning Algorithm to Support Learning in Open-ended, Unstructured Environments. *International Journal of Artificial Intelligence in Education*, 31(4):847–877, December 2021.

[40] Sithara H. P. W. Gamage, Jennifer R. Ayres, and Monica B. Behrend. A systematic review on trends in using Moodle for teaching and learning. *International Journal of STEM Education*, 9(1):9, January 2022.

[41] D. Garlan and D. Perry. Introduction to the Special Issue on Software Architecture. *IEEE Trans. Software Eng.*, 1995.

[42] J.A. Goguen and C. Linde. Techniques for requirements elicitation. In *[1993] Proceedings of the IEEE International Symposium on Requirements Engineering*, pages 152–164, January 1993.

[43] Joseph E Gonzalez, Reynold S Xin, Ankur Dave, Daniel Crankshaw, Michael J Franklin, and Ion Stoica. GraphX: Graph Processing in a Distributed Dataflow Framework. 2014.

[44] Sten Govaerts, Katrien Verbert, Erik Duval, and Abelardo Pardo. The student activity meter for awareness and self-reflection. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pages 869–884, New York, NY, USA, May 2012. Association for Computing Machinery.

[45] Alan Hevner, Alan R, Salvatore March, Salvatore T, Park, Jinsoo Park, Ram, and Sudha. Design Science in Information Systems Research. *Management Information Systems Quarterly*, 28:75, March 2004.

[46] Amjad Hudaib, Raja Masadeh, Mais Haj Qasem, and Abdullah Alzaqebah. Requirements Prioritization Techniques Comparison. *Modern Applied Science*, 12, January 2018.

[47] Elizabeth D Hutchison. Dimensions of Human Behavior: Person and Environment. 2018.

[48] IEEE. IEEE Standard Glossary of Software Engineering Terminology. *IEEE Std 610.12-1990*, pages 1–84, December 1990. Conference Name: IEEE Std 610.12-1990.

[49] Ivar Jacobson. *Object-Oriented Software Engineering: A Use Case Driven Approach*, volume 1. Addison-Wesley, 1992.

[50] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, September 1999.

[51] Yeonju Jang, Seongyune Choi, Heeseok Jung, and Hyeoncheol Kim. Practical early prediction of students' performance using machine learning and eXplainable AI. *Education and Information Technologies*, 27(9):12855–12889, November 2022.

[52] Libor Juhaňák, Jiří Zounek, and Lucie Rohlíková. Using process mining to analyze students' quiz-taking behavior patterns in a learning management system. *Computers in Human Behavior*, 92:496–506, March 2019.

[53] Nurul Kasim and Fariza Khalid. Choosing the Right Learning Management System (LMS) for the Higher Education Institution Context: A Systematic Review. *International Journal of Emerging Technologies in Learning (iJET)*, 11:55, June 2016.

[54] Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. Representation Learning for Dynamic Graphs: A Survey, April 2020. arXiv:1905.11485 [cs, stat].

[55] Mümine Kaya Keleş, Selma Ayşe Özel, Mümine Kaya Keleş, and Selma Ayşe Özel. *A Review of Distance Learning and Learning Management Systems*. IntechOpen, December 2016. Publication Title: Virtual Learning.

[56] Hassan Khosravi, Shiva Shabaninejad, Aneesha Bakharia, Shazia Sadiq, Marta Indulska, and Dragan Gasevic. Intelligent Learning Analytics Dashboards: Automated Drill-Down Recommendations to Support Teacher Data Exploration. *Journal of Learning Analytics*, 8(3):133–154, 2021. Publisher: Society for Learning Analytics Research ERIC Number: EJ1325376.

[57] Barbara Kitchenham and Stuart Charters. Guidelines for performing Systematic Literature Reviews in Software Engineering. 2, January 2007.

[58] Rita Kuo, Ted Krahn, and Maiga Chang. Behaviour Analytics - A Moodle Plug-in to Visualize Students' Learning Patterns. In Alexandra I. Cristea and Christos Troussas, editors, *Intelligent Tutoring Systems*, Lecture Notes in Computer Science, pages 232–238, Cham, 2021. Springer International Publishing.

[59] T. F. Kusumasari, K. Surendro, H. Sastramihardja, and I. Supriana. Human behavior conceptual model in collaborative software development product quality. In *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pages 89–94, September 2013.

[60] Demetrius Lacet, Filipe Penicheiro, and Leonel Morgado. Magical Board Theatre: interactive stories that can be played on multiple boards: two educational prototypes. *Videojogos 2020 - 12th International Conference on Videogame Sciences and Arts*, pages 164–176, 2020. Accepted: 2022-01-31T12:19:16Z ISBN: 9789895460625 Publisher: Sociedade Portuguesa de Ciências dos Videojogos.

[61] Demetrius Lacet, Filipe Penicheiro, Leonel Morgado, and António Coelho. Preserving story choreographies across multiple platforms: an approach to platform-independent reuse of characters' behaviors for games, simulations, animations and interactive videos. *Proceedings of the 9th International Conference on Digital and Interactive Arts*, pages 1–4, 2019. Accepted: 2020-02-17T14:29:12Z ISBN: 9781450372503 Publisher: ACM.

[62] Danielle Levac, Heather Colquhoun, and Kelly K. O'Brien. Scoping studies: advancing the methodology. *Implementation Science*, 5(1):69, September 2010.

[63] Fang Liu, Yifan Zhang, Liang Zhao, Qin Dai, Xiaona Liu, and Xiangbin Shi. A Metaverse-based Student's Spatiotemporal Digital Profile for Representing Learning Situation. pages 1–8, May 2022.

[64] Ekaterina Loginova and Dries F Benoit. Embedding navigation patterns for student performance prediction. 2021.

[65] Phillip Long, George Siemens, Gráinne Conole, and Dragan Gašević. Proceedings of the 1st International Conference on Learning Analytics and Knowledge. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, Banff Alberta Canada, February 2011. ACM.

[66] Steven Lonn and Stephanie D. Teasley. Saving time or innovating practice: Investigating perceptions and uses of Learning Management Systems. *Computers & Education*, 53(3):686–694, November 2009.

[67] Antonio Lopes, Bruno Pires, Marcio Cardoso, Arnaldo Santos, Filipe Peixinho, Pedro Sequeira, Leonel Morgado, Hugo Paredes, and Oleguer Camerino Foguet. Use of a virtual world system in sports coach education for reproducing team handball movements. *Journal For Virtual Worlds Research*, 2(1), March 2009. Number: 1.

[68] Ruth Malan, Hewlett-Packard Company, Dana Bredemeyer, and Bredemeyer Consulting. Functional Requirements and Use Cases. 2001.

[69] Jorge Maldonado-Mahauad, Mar Pérez-Sanagustín, René F. Kizilcec, Nicolás Morales, and Jorge Munoz-Gama. Mining theory-based patterns from Big data: Identifying self-regulated learning strategies in Massive Open Online Courses. *Computers in Human Behavior*, 80:179–196, March 2018.

[70] Joel Mandlazi, Ashwini Jadhav, and Ritesh Ajoodha. Educational data mining: using knowledge tracing as a tool for student success. In *2021 3rd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, pages 1–7, November 2021.

[71] Benjamin Maraza Quispe, Jhon Edwar Ninasivincha Apfata, Ricardo Carlos Qusipe Figueroa, and Manuel Alejandro Valderrama Solis. Design proposal of a personalized Dashboard to optimize teaching-learning in Virtual Learning Environments. In *2021 13th International Conference on Education Technology and Computers*, ICETC 2021, pages 77–84, New York, NY, USA, February 2022. Association for Computing Machinery.

[72] Salvatore March and Gerald Smith. Design and Natural Science Research on Information Technology. *Decision Support Systems*, 15:251–266, December 1995.

[73] Raúl Marticorena-Sánchez, Carlos López-Nozal, Yi Peng Ji, Carlos Pardo-Aguilar, and Álvar Arnaiz-González. UBUMonitor: An Open-Source Desktop Application for Visual E-Learning Analysis with Moodle. *Electronics*, 11(6):954, January 2022. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.

[74] Robert Martin. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Pearson, London, England, 1st edition edition, September 2017.

[75] Roberto Martinez-Maldonado, Judy Kay, Simon Buckingham Shum, and Kalina Yacef. Collocated Collaboration Analytics: Principles and Dilemmas for Mining Multimodal Interaction Data. *Human–Computer Interaction*, 34(1):1–50, January 2019. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/07370024.2017.1338956.

[76] Alberto Martín-Martín, Enrique Orduna-Malea, Mike Thelwall, and Emilio Delgado López-Cózar. Google Scholar, Web of Science, and Scopus: A systematic comparison of

citations in 252 subject categories. *Journal of Informetrics*, 12(4):1160–1177, November 2018.

[77] Leonel Morgado and Dennis Beck. Unifying Protocols for Conducting Systematic Scoping Reviews with Application to Immersive Learning Research. In *2020 6th International Conference of the Immersive Learning Research Network (iLRN)*, pages 155–162, San Luis Obispo, CA, USA, June 2020. IEEE.

[78] Bashar Nuseibeh and Steve Easterbrook. Requirements engineering: a roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, pages 35–46, New York, NY, USA, May 2000. Association for Computing Machinery.

[79] Paulo Cristiano de Oliveira, Cristiano Jose Castro de Almeida Cunha, and Marina Keiko Nakayama. LEARNING MANAGEMENT SYSTEMS (LMS) AND E-LEARNING MANAGEMENT: AN INTEGRATIVE REVIEW AND RESEARCH AGENDA. *JISTEM - Journal of Information Systems and Technology Management*, 13:157–180, August 2016. Publisher: TECSI Laboratório de Tecnologia e Sistemas de Informação - FEA/USP.

[80] Fan Ouyang, Xu Li, Dan Sun, Pengcheng Jiao, and Jiajia Yao. Learners' discussion patterns, perceptions, and preferences in a chinese massive open online course (mooc). *The International Review of Research in Open and Distributed Learning*, 21(3):264–284, Jun. 2020.

[81] Fan Ouyang, Xu Li, Dan Sun, Pengcheng Jiao, and Jiajia Yao. Learners' Discussion Patterns, Perceptions, and Preferences in a Chinese Massive Open Online Course (MOOC). *International Review of Research in Open and Distributed Learning*, 21(3):264–284, 2020. Publisher: Athabasca University Press (AU Press).

[82] Sanghoon Park. Analysis of Time-on-Task, Behavior Experiences, and Performance in Two Online Courses with Different Authentic Learning Tasks. *International Review of Research in Open and Distributed Learning*, 18(2):213–233, 2017. Publisher: Athabasca University Press (AU Press).

[83] Akanksha Parmar. Paper Review on Sharable Content Object Reference Model (SCORM): Framework for E-learning Standard. In *2012 Second International Conference on Advanced Computing & Communication Technologies*, pages 409–411, January 2012. ISSN: 2327-0659.

[84] Daniela Pedrosa, José Cravino, and Leonel Morgado. *e-SimProgramming: planificar, conceber e acompanhar atividades didáticas online de engenharia de software*. Universidade Aberta, 2022.

[85] Daniela Pedrosa, José Cravino, Leonel Morgado, Carlos Barreira, Ricardo Rodrigues Nunes, Paulo Martins, and Hugo Paredes. SIMPROGRAMMING: THE DEVELOPMENT OF AN INTEGRATED TEACHING APPROACH FOR COMPUTER PROGRAMMING IN HIGHER EDUCATION. pages 7162–7172, Valencia, Spain, March 2016.

[86] Daniela Pedrosa, Leonel Morgado, José Cravino, Mario Madureira Fontes, Maria Castelhano, Claudia Machado, and Eliana Curado. Challenges Implementing the SimProgramming Approach in Online Software Engineering Education for Promoting Self and Co-regulation of Learning. In *2020 6th International Conference of the Immersive Learning Research Network (iLRN)*, pages 236–242, June 2020.

[87] Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3):45–77, December 2007.

[88] Jesús Peral, Antonio Ferrández, Higinio Mora, David Gil, and Erick Kauffmann. A Review of the Analytics Techniques for an Efficient Management of Online Forums: An Architecture Proposal. *IEEE Access*, 7:12220–12240, 2019. Conference Name: IEEE Access.

[89] Micah D. J. Peters, Christina M. Godfrey, Hanan Khalil, Patricia McInerney, Deborah Parker, and Cassia Baldini Soares. Guidance for conducting systematic scoping reviews. *JBI Evidence Implementation*, 13(3):141, September 2015.

[90] Mark Petticrew and Helen Roberts. *Systematic reviews in the social sciences: A practical guide*. Systematic reviews in the social sciences: A practical guide. Blackwell Publishing, Malden, 2006. Pages: xv, 336.

[91] Klaus Pohl and Chris Rupp. *Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam, foundation level, IREB compliant*. Rocky Nook, Santa Barbara, CA, second edition edition, 2015.

[92] Elvira Popescu and Florin Leon. Predicting Academic Performance Based on Learner Traces in a Social Learning Environment. *IEEE Access*, 6:72774–72785, 2018. Conference Name: IEEE Access.

[93] Tuti Purwoningsih, Harry B. Santoso, and Zainal A. Hasibuan. Data Analytics of Students' Profiles and Activities in a Full Online Learning Context. In *2020 Fifth International Conference on Informatics and Computing (ICIC)*, pages 1–8, November 2020.

[94] Shaojie Qu, Kan Li, Bo Wu, Shuhui Zhang, and Yongchao Wang. Predicting Student Achievement Based on Temporal Learning Behavior in MOOCs. *Applied Sciences*, 9(24):5539, January 2019. Number: 24 Publisher: Multidisciplinary Digital Publishing Institute.

[95] Pradeep Rai and Singh Shubha. A Survey of Clustering Techniques. *International Journal of Computer Applications*, 7, October 2010.

[96] Reza Rezaei, Thiam Kian Chiew, Sai Peck Lee, and Zeinab Shams Aliee. Interoperability evaluation models: A systematic review. *Computers in Industry*, 65(1):1–23, January 2014.

[97] Natalia Díaz Rodríguez, M. P. Cuéllar, Johan Lilius, and Miguel Delgado Calvo-Flores. A survey on ontologies for human behavior recognition. *ACM Computing Surveys*, 46(4):1–33, April 2014.

[98] Cristobal Romero and Sebastian Ventura. Educational data mining and learning analytics: An updated survey. *WIREs Data Mining and Knowledge Discovery*, 10(3):e1355, 2020. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1355.

[99] Behzad Saemi, Ali Asghar Rahmani Hosseinabadi, Maryam Kardgar, Valentina Emilia Balas, and Hamed Ebadi. Nature Inspired Partitioning Clustering Algorithms: A Review and Analysis. In Valentina Emilia Balas, Lakhmi C. Jain, and Marius Mircea Balas, editors, *Soft Computing Applications*, Advances in Intelligent Systems and Computing, pages 96–116, Cham, 2018. Springer International Publishing.

[100] Shweta Sankhwar, Virendra Singh, and Dhirendra Pandey. REQUIREMENT ENGINEERING PARADIGM. *Global Journal of Multidisciplinary Studies*, March 2014.

[101] Mohammed Saqr, Uno Fors, Matti Tedre, and Jalal Nouri. How social network analysis can be used to monitor online collaborative learning and guide an informed intervention. *PLOS ONE*, 13(3):e0194777, March 2018. Publisher: Public Library of Science.

[102] Alper Sarikaya, Michael Correll, Lyn Bartram, Melanie Tory, and Danyel Fisher. What do we talk about when we talk about dashboards? *IEEE Transactions on Visualization and Computer Graphics*, 25(1):682–692, 2019.

[103] Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding, and Chin-Teng Lin. A review of clustering techniques and developments. *Neurocomputing*, 267:664–681, December 2017.

[104] Beat A. Schwendimann, María Jesús Rodríguez-Triana, Andrii Vozniuk, Luis P. Prieto, Mina Shirvani Boroujeni, Adrian Holzer, Denis Gillet, and Pierre Dillenbourg. Perceiving Learning at a Glance: A Systematic Literature Review of Learning Dashboard Research. *IEEE Transactions on Learning Technologies*, 10(1):30–41, January 2017. Conference Name: IEEE Transactions on Learning Technologies.

[105] Gayane Sedrakyan, Erik Mannens, and Katrien Verbert. Guiding the choice of learning dashboard visualizations: Linking dashboard design and data visualization concepts. *Journal of Computer Languages*, 50:19–38, February 2019.

[106] Colin Shearer. The CRISP-DM Model: The New Blueprint for Data Mining. 5(4), 2000.

[107] Shaoyong Shi, Zaijiang Tang, Quanfeng Ma, and Jingye Wang. Human Behavior Representation Theory Progress in Constructive Simulation. In Tianyuan Xiao, Lin Zhang, and Shiwei Ma, editors, *System Simulation and Scientific Computing*, Communications in Computer and Information Science, pages 336–344, Berlin, Heidelberg, 2012. Springer.

[108] Andysah Putera Utama Siahaan and Mhd Furqan. A Review of Prim and Genetic Algorithms in Finding and Determining Routes on Connected Weighted Graphs. preprint, INA-Rxiv, October 2018.

[109] Emanuel Silva, Nuno Silva, and Leonel Morgado. Model-Driven Generation of Multi-user and Multi-domain Choreographies for Staging in Multiple Virtual World Platforms. In Yamine Ait Ameur, Ladjel Bellatreche, and George A. Papadopoulos, editors, *Model and Data Engineering*, Lecture Notes in Computer Science, pages 77–91, Cham, 2014. Springer International Publishing.

[110] Emanuel Silva, Nuno Silva, and Leonel Morgado. Staging Choreographies for Team Training in Multiple Virtual Worlds Based on Ontologies and Alignments. In Randall Shumaker and Stephanie Lackey, editors, *Virtual, Augmented and Mixed Reality. Applications of Virtual and Augmented Reality*, Lecture Notes in Computer Science, pages 105–115, Cham, 2014. Springer International Publishing.

[111] Mark Sinnott and Ling Angela Xia. A Review of the Moodle Gamification Plugin "Level Up": Using a Moodle Plugin to Gamify Learning of Academic Vocabulary. *International Journal of Computer-Assisted Language Learning and Teaching (IJCALLT)*, 10(3):89–95, July 2020. Publisher: IGI Global.

[112] Paul D. Stigall and Omur Tasar. A review of directed graphs as applied to computors. *Computer*, 7(10):39–47, October 1974. Conference Name: Computer.

[113] Beni Suranto. Software prototypes: Enhancing the quality of requirements engineering process. In *2015 International Symposium on Technology Management and Emerging Technologies (ISTMET)*, pages 148–153, August 2015.

[114] María Consuelo Sáiz-Manzanares, Juan José Rodríguez-Díez, José Francisco Díez-Pastor, Sandra Rodríguez-Arribas, Raúl Marticorena-Sánchez, and Yi Peng Ji. Monitoring of Student Learning in Learning Management Systems: An Application of Educational Data Mining Techniques. *Applied Sciences*, 11(6):2677, January 2021. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.

[115] Mojtaba Vaismoradi, Jacqueline Jones, Hannele Turunen, and Sherrill Snelgrove. Theme development in qualitative content analysis and thematic analysis. *Journal of Nursing Education and Practice*, 6(5):p100, January 2016.

[116] A. van Lamsweerde. Requirements engineering in the year 00: a research perspective. In *Proceedings of the 2000 International Conference on Software Engineering. ICSE 2000 the New Millennium*, pages 5–19, June 2000. ISSN: 0270-5257.

[117] Carlos J. Villagrá-Arnedo, Francisco J. Gallego-Durán, Faraón Llorens-Largo, Patricia Compañ-Rosique, Rosana Satorre-Cuerda, and Rafael Molina-Carmona. Improving the expressiveness of black-box models for predicting student performance. *Computers in Human Behavior*, 72:621–631, July 2017.

[118] Hajra Waheed, Saeed-Ul Hassan, Naif Radi Aljohani, Julie Hardman, Salem Alelyani, and Raheel Nawaz. Predicting academic performance of students from VLE big data using deep learning models. *Computers in Human Behavior*, 104:106189, March 2020.

[119] Dan Ye and Svoboda Pennisi. Analysing interactions in online discussions through social network analysis. *Journal of Computer Assisted Learning*, 38(3):784–796, 2022. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jcal.12648.

[120] Bonnie L. Youngs, Akhil Prakash, and Rebecca Nugent. Statistically-driven visualizations of student interactions with a French online course video. *Computer Assisted Language Learning*, 31(3):206–225, March 2018. Publisher: Routledge _eprint: https://doi.org/10.1080/09588221.2017.1367311.

[121] P. Zave. Classification of research efforts in requirements engineering. In *Proceedings of 1995 IEEE International Symposium on Requirements Engineering (RE'95)*, pages 214–216, March 1995.

[122] Mingyan Zhang, Xu Du, Jui-Long Hung, Hao Li, Mengfan Liu, and Hengtao Tang. Analyzing and Interpreting Students' Self-regulated Learning Patterns Combining Time-series Feature Extraction, Segmentation, and Clustering, 2022.

[123] Muhittin Şahin, Marc Egloffstein, Max Bothe, Tobias Rohloff, Nathanael Schenk, Florian Schwerer, and Dirk Ifenthaler. Behavioral Patterns in Enterprise MOOCs at openSAP. pages 281–288, June 2021.

Appendix A

Keyword execution of survey of surveys with the title only

Combination	Search string	Result
VC + LD + L&S + I	N/A	0
VC + LD + L&S	allintitle:("survey" OR"review")("choreographies" OR "choreography" OR "behavior patterns")("* dashboard" OR "learning analytics" OR "information visualization")("learning management systems" OR "LMS" OR "virtual learning environments" OR "VLEs" OR "SCORM")	0
VC + LD + I	allintitle:("survey" OR "review")("choreographies" OR "choreography" OR "behavior patterns")("* dashboard" OR "learning analytics" OR "information visualization")("interoperability" OR "interoperable" OR "systems integration")	0
VC + L&S + I	allintitle:("survey" OR "review")("choreographies" OR "choreography" OR "behavior patterns")("learning management systems" OR "LMS" OR "virtual learning environments" OR "VLEs" OR "SCORM")("interoperability" OR "interoperable" OR "systems integration")	0
LD + L&S + I	allintitle:("survey" OR "review")("* dashboard" OR "learning analytics" OR "information visualization")("learning management systems" OR "LMS" OR "virtual learning environments" OR "VLEs" OR "SCORM")("interoperability" OR "interoperable" OR "systems integration")	0
VC + LD	allintitle:("survey" OR "review")("choreographies" OR "choreography" OR "behavior patterns")("* dashboard" OR "learning analytics" OR "information visualization")	0
VC + L&S	allintitle:("survey" OR "review")("choreographies" OR "choreography" OR "behavior patterns")("learning management systems" OR "LMS" OR "virtual learning environments" OR "VLEs" OR "SCORM")	0

Combination	Search string	Result
VC + I	allintitle:("survey" OR "review")("choreographies" OR "choreography" OR "behavior patterns")("interoperability" OR "interoperable" OR "systems integration")	0
LD + L&S	allintitle:("survey" OR "review")("* dashboard" OR "learning analytics" OR "information visualization")("learning management systems" OR "LMS" OR "virtual learning environments" OR "VLEs" OR "SCORM")	1
LD + I	allintitle:("survey" OR "review")("* dashboard" OR "learning analytics" OR "information visualization")("interoperability" OR "interoperable" OR "systems integration")	2
L&S + I	allintitle:("survey"|"review")("learning management systems" OR "LMS" OR "virtual learning environments" OR "VLEs" OR "SCORM")("interoperability" OR "interoperable" OR "systems integration")	0

Appendix B

Keyword execution of survey of surveys with a mix of title and body

Combination	Search string	Result
Title: VC Keywords: LD	(intitle:“survey” OR intitle:“review”)(intitle:“choreographies” OR intitle:“choreography” OR intitle:“behavior patterns”)(“* dashboard” OR “learning analytics” OR “information visualization”)	1
Title: LD Keywords: VC	(intitle:“survey” OR intitle:“review”)(intitle:“* dashboard” OR intitle:“learning analytics” OR intitle:“information visualization”)(“choreographies” OR “choreography” OR “behavior patterns”)	26
Title: VC Keywords: L&S	(intitle:“survey” OR intitle:“review”)(intitle:“choreographies” OR intitle:“choreography” OR intitle:“behavior patterns”)(“learning management systems” OR “LMS” OR “virtual learning environments” OR “VLEs” OR “SCORM”)	0
Title: L&S Keywords: VC	(intitle:“survey” OR intitle:“review”)(intitle:“learning management systems” OR intitle:“LMS” OR intitle:“virtual learning environments” OR intitle:“VLEs” OR intitle:“SCORM”)(“choreographies” OR “choreography” OR “behavior patterns”)	3
Title: VC Keywords: I	(intitle:“survey” OR intitle:“review”)(intitle:“choreographies” OR intitle:“choreography” OR intitle:“behavior patterns”)(“interoperability” OR “interoperable” OR “systems integration”)	10
Title: I Keywords: VC	(intitle:“survey” OR intitle:“review”)(intitle:“interoperability” OR intitle:“interoperable” OR intitle:“systems integration”)(“choreographies” OR “choreography” OR “behavior patterns”)	13
Title: LD Keywords: L&S	(intitle:“survey”OR intitle:“review”)(intitle:“* dashboard”OR intitle:“learning analytics”OR intitle:“information visualization”)(“learning management systems” OR “LMS” OR “virtual learning environments” OR “VLEs” OR “SCORM”)	157

Combination	Search string	Result
Title: L&S Keywords: LD	(intitle:“survey”OR intitle:“review”)(intitle:“learning management systems” OR intitle:“LMS” OR intitle:“virtual learning environments” OR intitle:“VLEs” OR intitle:“SCORM”)(“* dashboard” OR “learning analytics” OR “information visualization”)	17
Title: LD Keywords: I	(intitle:“survey” OR intitle:“review”)(intitle:“* dashboard” OR intitle:“learning analytics” OR intitle:“information visualization”)(“interoperability” OR “interoperable” OR “systems integration”)	38
Title: I Keywords: LD	(intitle:“survey” OR intitle:“review”)(intitle:“interoperability” OR intitle:“interoperable” OR intitle:“systems integration”)(“* dashboard” OR “learning analytics” OR “information visualization”)	19
Title: L&S Keywords: I	(intitle:“survey” OR intitle:“review”)(intitle:“learning management systems” OR intitle:“LMS” OR intitle:“virtual learning environments” OR intitle:“VLEs” OR intitle:“SCORM”)(“interoperability” OR “interoperable” OR “systems integration”)	29
Title: I Keywords: L&S	(intitle:“survey” OR intitle:“review”)(intitle:“interoperability” OR intitle:“interoperable” OR intitle:“systems integration”)(“learning management systems” OR “LMS” OR “virtual learning environments” OR “VLEs” OR “SCORM”)	6

Appendix C

Systematic literature results

Authors	**Title**	**Year**
D Ye and S Pennisi	Analysing interactions in online discussions through social network analysis	2022
Zhang *et al.*	Analyzing and Interpreting Students' Self-regulated Learning Patterns Combining Time-series Feature Extraction, Segmentation, and Clustering	2022
Marticorena-Sánchez*et al.*	UBUMonitor: An Open-Source Desktop Application for Visual E-Learning Analysis with Moodle	2022
Jang *et al.*	Practical early prediction of students' performance using machine learning and eXplainable AI	2022
Sun *et al.*	Big Educational Data Analytics, Prediction and Recommendation: A Survey	2022
Kaliisa *et al.*	'My Point of Departure for Analytics is Extreme Skepticism': Implications Derived from An Investigation of University Teachers' Learning Analytics Perspectives and Design Practices	2022
Tan *et al.*	A systematic review of artificial intelligence techniques for collaborative learning over the past two decades	2022
Sáiz-Manzanares *et al.*	Monitoring of student learning in learning management systems: An application of educational data mining techniques	2021
S Frost and G McCalla	A planning algorithm to support learning in open-ended, unstructured environments	2021
R Cobos and JC Ruiz-Garcia	Improving learner engagement in MOOCs using a learning intervention system: A research study in engineering education	2021
Bessadok *et al.*	Exploring students digital activities and performances through their activities logged in learning management system using educational data mining approach	2021
Khosravi *et al.*	Intelligent Learning Analytics Dashboards: Automated Drill-Down Recommendations to Support Teacher Data Exploration	2021
Ninasivincha-Apfata *et al.*	Dashboard Proposal Implemented According to an Analysis Developed on the KNIME Platform	2021
E Loginova and DF Benoit	Embedding Navigation Patterns for Student Performance Prediction	2021
Waheed *et al.*	Predicting academic performance of students from VLE big data using deep learning models	2020
Ouyang *et al.*	Learners' discussion patterns, perceptions, and preferences in a Chinese massive open online course (MOOC)	2020
Juhaňák *et al.*	Using process mining to analyze students' quiz-taking behavior patterns in a learning management system	2019

Authors	Title	Year
Peral *et al.*	A review of the analytics techniques for efficient management of online forums: An architecture proposal	2019
Qu *et al.*	Predicting student achievement based on temporal learning behavior in MOOCs	2019
Martinez-Maldonado *et al.*	Collocated collaboration analytics: Principles and dilemmas for mining multimodal interaction data	2019
Sedrakyan *et al.*	Guiding the choice of learning dashboard visualizations: Linking dashboard design and data visualization concepts	2019
Saqr *et al.*	How social network analysis can be used to monitor online collaborative learning and guide an informed intervention	2018
Maldonado-Mahauad *et al.*	Mining theory-based patterns from Big data: Identifying self-regulated learning strategies in Massive Open Online Courses	2018
Al-Shabandar *et al.*	Analyzing learners behavior in MOOCs: An examination of performance and motivation using a data-driven approach	2018
E Popescu and F Leon	Predicting academic performance based on learner traces in a social learning environment	2018
Youngs *et al.*	Statistically-driven visualizations of student interactions with a French online course video	2018
Becheru *et al.*	Analyzing students' collaboration patterns in a social learning environment using studentviz platform	2018
Villagrá-Arnedo *et al.*	Improving the expressiveness of black-box models for predicting student performance	2017
Dutt *et al.*	A systematic review on educational data mining	2017
Dado and Bodemer	A review of methodological applications of social network analysis in computer-supported collaborative learning	2017
S Park	Analysis of time-on-task, behavior experiences, and performance in two online courses with different authentic learning tasks	2017
Carroll and White	Identifying patterns of learner behavior: what business statistics students do with learning resources	2017
Egloffstein *et al.*	Behavioral Patterns and Learner Interactions in Enterprise MOOCs	2023
Liu *et al.*	A metaverse-based student's spatiotemporal digital profile for representing learning situation	2022
Bajo *et al.*	Using Process Mining Techniques to Discover the Collective Behaviour of Educators in a Learning Community Platform	2022
Quispe *et al.*	Design proposal of a personalized Dashboard to optimize teaching-learning in Virtual Learning Environments	2021
Kuo *et al.*	Behaviour Analytics-A Moodle Plug-in to Visualize Students' Learning Patterns	2021
Sahin *et al.*	Behavioral patterns in enterprise MOOCs at openSAP	2021

Authors	Title	Year
Al-Ashmoery *et al.*	Learning analytics toolset for evaluating students' performance in an E-learning Platform	2021
Denden *et al.*	Can we predict learners' personalities through their behavioral patterns? A pilot study using Behaviour Analytics-Moodle plugin	2021
Mandlazi *et al.*	Educational data mining: using knowledge tracing as a tool for student success	2021
D'Aniello *et al.*	A situation-aware learning system based on fuzzy cognitive maps to increase learner motivation and engagement	2020
Desai *et al.*	A study on student performance evaluation using discussion board networks	2020
Purwoningsih *et al.*	Data Analytics of Students' Profiles and Activities in a Full Online Learning Context	2020