People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Oum El Bouaghi

Faculty of Exact Sciences and Science of Nature and Life

# Thesis

Presented to obtain

## 3rd Cycle Doctorate

**Branch: Computer Science**

**Specialty: Distributed Computing and Networks**

**Title:**

# Multimedia Content Access in Context-aware Pervasive Systems: Application to Ubiquitous Learning

Presented by:

**Aziz Smaala**

Publicly defended on 20/03/2025 in front of the following committee members:

| N° | first and last name | Grade | University | Quality |
|---|---|---|---|---|
| 01 | **Tahar Gueram** | **MCA** | University of Oum El Bouaghi | President |
| 02 | **Zakaria Laboudi** | **Prof** | University of Oum El Bouaghi | Supervisor |
| 03 | **Asma Saighi** | **MCA** | University of Oum El Bouaghi | Co-reporter |
| 04 | **Hicham Houassi** | **Prof** | University of Khenchela | Examiner |
| 05 | **Sofia Koah** | **MCA** | University of Oum El Bouaghi | Examiner |
| 06 | **Abdelkader Moudjari** | **MCA** | University of Constantine 3 | Invited |

People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Oum El Bouaghi

Faculty of Exact Sciences and Science of Nature and Life

# Thesis

Presented to obtain

## 3rd Cycle Doctorate

**Branch: Computer Science**

**Specialty: Distributed Computing and Networks**

**Title:**

---

# Multimedia Content Access in Context-aware Pervasive Systems: Application to Ubiquitous Learning

---

Presented by:

**Aziz Smaala**

Publicly defended on 20/03/2025 in front of the following committee members:

| N° | first and last name | Grade | University | Quality |
|---|---|---|---|---|
| 01 | **Tahar Gueram** | **MCA** | University of Oum El Bouaghi | President |
| 02 | **Zakaria Laboudi** | **Prof** | University of Oum El Bouaghi | Supervisor |
| 03 | **Asma Saighi** | **MCA** | University of Oum El Bouaghi | Co-reporter |
| 04 | **Hicham Houassi** | **Prof** | University of Khenchela | Examiner |
| 05 | **Sofia Koah** | **MCA** | University of Oum El Bouaghi | Examiner |
| 06 | **Abdelkader Moudjari** | **MCA** | University of Constantine 3 | Invited |

*This Thesis is dedicated to my dear mother,*

*may God envelop her in his vast mercy*

*and grant her eternal rest in his paradise.,*

*to my father, my brother and sisters,*

*and to all those who believed in me,*

*without whose support during the hardest of times,*

*this work would not have been possible.*

# Acknowledgments

First and foremost, my sincere gratitude goes to the Most Merciful ALLAH for all the blessings He has bestowed upon me throughout my life. Without His countless blessings, I would not be in this position today.

During my doctoral studies, I have been accompanied and supported by many people. It is a great pleasure to take this opportunity to express my heartfelt appreciation to all of them.

I owe my deepest gratitude to my supervisor, Pr. LABOUDI Zakaria, for his kindness, enthusiasm, patience, and unwavering support. His countless hours of reading, insightful comments, and invaluable advice reflect not only his professionalism but also his generous and compassionate heart. From the depths of my heart, thank you, dear professor, for being more than just a supervisor—for being a true source of inspiration.

I also feel honored and grateful to thank my co-supervisor, Dr. SAIGHI Asma, who has always closely followed the progress of my work and was readily available whenever I needed her guidance. Her encouragement and support have been truly invaluable.

I extend my sincere gratitude to the members of the jury for accepting to evaluate my work, which represents a significant commitment of their time and effort.

I would also like to express my appreciation to every member of my faculty and the ReLa(CS)2 Laboratory for providing the assistance I needed throughout my research journey.

Words cannot express how grateful I am to my parents for their sacrifices and unwavering prayers since the day I was born. My deepest appreciation also goes to my brother, Saber, and my sisters for their continuous support and constant care.

*Thank you all*

**ملخص**

أدى النمو السريع للحوسبة المتنقلة والذكاء المحيطي إلى تغيير تفاعلات المستخدمين مع الأجهزة والوصول إلى المحتوى الرقمي، مما ساهم في ظهور أنظمة محيطية حساسة للسياق. تهدف هذه الأنظمة إلى تقديم تجارب مخصصة، خاصة من خلال تكييف المحتوى المتعدد الوسائط مع السياقات الديناميكية للمستخدمين مثل الموقع وخصائص الجهاز والبيئة. توجد العديد من مقاربات تكييف المحتوى المتعدد الوسائط الحساس للسياق، وتنقسم إلى أربع فئات رئيسية: التكييف من جانب العميل، التكييف من جانب الخادم، التكييف القائم على الوكيل، والتكييف من نظير إلى نظير. ومع ذلك، تعتمد معظم هذه المقاربات بشكل أساسي على البيانات السياقية في الوقت الفعلي، وغالباً ما تهمل إمكانية دمج البيانات التاريخية للمستخدمين في عملية التكييف. في الواقع، يمكن للبيانات التاريخية أن تعزز بشكل كبير تكييف المحتوى المتعدد الوسائط من خلال تمكين اتخاذ قرارات مستنيرة ودعم التخصيص. تسعى هذه الأطروحة إلى سد هذه الفجوة من خلال اقتراح إطار عمل يدمج البيانات التاريخية للمستخدمين في عمليات تكييف الوثائق المتعددة الوسائط في الأنظمة الحساسة للسياق. تتمثل المساهمات الرئيسية لهذا العمل في ثلاث نقاط رئيسية. أولاً، يقترح إطار عمل فعال لتكييف المحتوى يستخدم البيانات التاريخية للمستخدمين لتحسين الوصول إلى المحتوى المتعدد الوسائط، وسهولة استخدامه، وتخصيصه في الأنظمة الحساسة للسياق. يتضمن الإطار قواعد بيانات SQL و NoSQL لتخزين البيانات بكفاءة، ويطور وظائف معالجة لاسترجاع وتحليل البيانات التاريخية، ويستخدم هذه الرؤى لتحسين تقديم المحتوى. ثانياً، يساهم هذا العمل في تطوير آلية تعلم القواعد التي تعتمد على خوارزميات التعلم الآلي مثل: ,ECLAT، sequential covering ، FP-Growth، و decision trees لتخصيص قواعد التكييف بناءً على البيانات التاريخية للمستخدمين. ثالثاً، يتناول استخدام التحليل الدلالي الكامن لتصنيف الحالات السياقية وسلوكيات المستخدمين، من خلال معاملة البيانات التاريخية للمستخدمين كوثائق نصية. تمت المصادقة على هذه المساهمات من خلال نماذج تطبيقات حقيقية مع التركيز بشكل خاص على تكييف المواد التعليمية في البيئات المحيطية الحساسة للسياق.

**الكلمات المفتاحية:**

الذكاء المحيطي، الأنظمة المحيطية الحساسة للسياق، تكييف الوثائق المتعددة الوسائط، التعلم اللامكاني، بيانات المستخدمين التاريخية، التخصيص، آلية تعلم القواعد، التحليل الدلالي الكامن.

## Abstract

The rapid growth of mobile computing and ambient intelligence has transformed users' interactions with devices and access digital content, fostering the emergence of context-aware pervasive systems. These systems aim to deliver personalized experiences, particularly by adapting multimedia content to dynamic user contexts, such as location, device features, and environment. Several context-aware multimedia adaptation approaches already exist; they belong to four main categories: client side adaptation, server side adaptation, proxy-based adaptation and peer-to-peer adaptation. However, most of these approaches primarily rely on real-time context data, often neglecting the potential of incorporating historical user data into the adaptation process. Indeed, historical data can significantly enhance multimedia adaptation by enabling informed decision-making and supporting personalization. This thesis addresses this gap by proposing a framework that incorporates historical user data into multimedia documents adaptation processes in context-aware systems. The main contributions of this work are threefold. The first contribution proposes an efficient adaptation framework that employs historical user data to enhance multimedia content accessibility, usability, and personalization in context-aware systems. The framework incorporates SQL and NoSQL databases for efficient data storage, develops processing functions for retrieving and analyzing historical data, and uses these insights to optimize content delivery. These functions support client-side, proxy-based, and server-side management options. This aim is to help enhance adaptation processes through machine learning and recommendation tasks. The second contribution of this work is a rule-learning mechanism that leverages machine learning algorithms – such as ECLAT, sequential covering, FP-Growth, and decision trees – to personalize adaptation rules, based on historical users data. Using unsupervised machine learning techniques, the system predicts adaptation actions based on past user behaviors with respect to context changes, without needing pre-trained datasets. The third contribution involves the application of latent semantic analysis to classify context situations and user behaviors. By treating historical user data as textual documents with context elements and adaptation actions serving as words, the system uncovers hidden semantic structures, improving content delivery and enhancing user experience personalization through deeper context understanding. These contributions are validated through real prototypes and applications, with a particular focus on the adaptation of learning materials in context-aware pervasive environments. The results demonstrate the effectiveness and potential of the proposed system to enhance multimedia document accessibility and personalization, offering promising improvements in user experience and adaptation efficiency.

**Keywords:**

Ambient intelligence, Context-aware pervasive systems, multimedia document adaptation, ubiquitous learning, historical users data, personalization, rule-learning mechanism, latent semantic analysis.

**Résumé**

La croissance rapide de l'informatique mobile et de l'intelligence ambiante a transformé les interactions des utilisateurs avec les dispositifs et l'accès aux contenus numériques, favorisant l'émergence de systèmes pervasifs sensibles au contexte. Ces systèmes visent à offrir des expériences personnalisées, notamment en adaptant les contenus multimédias aux contextes dynamiques des utilisateurs, tels que la localisation, les caractéristiques des dispositifs et l'environnement. Plusieurs approches d'adaptation de contenus multimédias sensibles au contexte existent déjà; elles se répartissent en quatre grandes catégories : l'adaptation côté client, l'adaptation côté serveur, l'adaptation basée sur un proxy et l'adaptation de pair à pair. Cependant, la plupart de ces approches s'appuient principalement sur des données contextuelles en temps réel, négligeant souvent le potentiel d'intégration des données historiques des utilisateurs dans le processus d'adaptation. En effet, les données historiques peuvent considérablement améliorer l'adaptation multimédia en permettant une prise de décision éclairée et en soutenant la personnalisation. Cette thèse répond à ce besoin en proposant un cadre d'adaptation des documents multimédias dans les systèmes sensibles au contexte, intégrant les données historiques des utilisateurs. Les principales contributions de ce travail sont triples. La première contribution propose un modèle d'adaptation efficace qui utilise les données historiques des utilisateurs pour améliorer l'accessibilité, l'utilisabilité et la personnalisation des contenus multimédias dans les systèmes sensibles au contexte. Le modèle incorpore des bases de données SQL et NoSQL pour un stockage efficace des données, développe des fonctions de traitement pour la récupération et l'analyse des données historiques, et utilise ces informations pour optimiser la diffusion de contenus. Ces fonctions prennent en charge les options de gestion côté client, basé-proxy et côté serveur. L'objectif est d'améliorer les processus d'adaptation grâce à des tâches d'apprentissage automatique et de recommandation. La deuxième contribution de ce travail est un mécanisme d'apprentissage des règles qui exploite des algorithmes d'apprentissage automatique — tels que ECLAT, le recouvrement séquentiel, FP-Growth et les arbres de décision — pour personnaliser les règles d'adaptation à partir des données historiques des utilisateurs. En utilisant des techniques d'apprentissage non supervisé, le système prédit les actions d'adaptation en fonction des comportements passés des utilisateurs face aux changements de contexte, sans nécessiter de jeux de données pré-entraînés. La troisième contribution concerne l'application de l'analyse sémantique latente pour classifier les situations contextuelles et les comportements des utilisateurs. En traitant les données historiques des utilisateurs comme des documents textuels, avec les éléments contextuels et les actions d'adaptation servant de mots, le système découvre des structures sémantiques cachées, améliorant la diffusion des contenus et la personnalisation de l'expérience utilisateur grâce à une compréhension approfondie du contexte. Ces contributions sont validées par des prototypes réels et des applications, en mettant particulièrement l'accent sur l'adaptation des supports d'apprentissage dans des environnements pervasifs sensibles au contexte. Les résultats démontrent l'efficacité et le potentiel du système proposé pour améliorer l'accessibilité et la personnalisation des documents multimédias, offrant des améliorations prometteuses de l'expérience utilisateur et de l'efficacité des processus d'adaptation.

**Mots-clés :**

Intelligence ambiante, systèmes pervasifs sensibles au contexte, adaptation de documents multimédias, apprentissage ubiquitaire, données historiques des utilisateurs, personnalisation, mécanisme d'apprentissage des règles, analyse sémantique latente.

# Table of Contents

**Chapter III: Multimedia Document Adaptation in Context-Aware Pervasive Systems**



**Chapter IV: Management of historical data in context-aware MDA processes**

**Chapter V: Rule-learning for the personalization of adaptation actions**



**Chapter VI: Latent Semantic Analysis for the Classification of Similar Contextual Situations and User Behaviors**

# Figure Index

# List of Tables

# Glossary of Abbreviations

| | |
|---|---|
| AI | Artificial Intelligence |
| AmT | Ambient Intelligence |
| APIs | Application Programming Interfaces |
| AVI | Audio Video Interleave |
| CAMEL | Context-Awareness Modeling Language |
| CPU | Central Processing Unit |
| COVID-19 | Coronavirus Disease 2019 |
| CSS | Cascading Style Sheets |
| DAO | Data Access Object |
| DB | Data Base |
| DITA | Darwin Information Typing Architecture |
| FP-Growth | Frequent Pattern Growth |
| FLV | Flash Video |
| FTP | File Transfer Protocol |
| GIF | Graphics Interchange Format |
| GPS | Global Positioning System |
| GSM | Global System for Mobile Communications |
| GUI | Graphical User Interface |
| HCI | Human computer interaction |
| HUD | Historical Users Data |
| HTML | HyperText Markup Language |
| ID | Identifier |
| IoT | Internet of Things |
| JPEG | Joint Photographic Experts Group |
| JSON | JavaScript Object Notation |
| LSA | Latent semantic analysis |
| MDA | Multimedia document adaptation |
| MMSA | Metamodel Multimedia Software Architecture |
| NCL | Nested Context Language |
| NLP | Natural Language Processing |
| OCSP | Over-constrained Constraint Satisfaction Problem |
| PDF | Portable Document Format |
| PNG | Portable Network Graphics |
| RAM | Random Access Memory |

| | |
|---|---|
| SMIL | Synchronized Multimedia Integration Language |
| SQL | Structured Query Language |
| SVD | Singular Value Decomposition |
| SVG | Scalable Vector Graphics |
| UML | Unified Modeling Language |
| VR | Virtual Reality |
| WMA | Windows Media Audio |
| WMV | Windows Media Video |
| Wi-Fi | Wireless Fidelity |
| X3D | Extensible 3D Graphics |
| XML | Extensible Markup Language |
| XHTML | Extensible Hypertext Markup Language |

# Chapter I

# Introduction

## Summary

# CHAPTER I

## Introduction

In this chapter, we outline the overall framework that shapes this research. We begin by introducing the context of this thesis, focusing *on the adaptation of multimedia documents within context-aware pervasive systems, with particular emphasis on the adaptation of learning materials*. Next, we define the problem to be addressed and explore the motivations driving this choice. Additionally, we provide an overview of the methodology that will guide the research work throughout this thesis, ensuring a clear path to its completion. Finally, we conclude this chapter by presenting the structure of this document, which represents the culmination of years of hard work.

### 1. Research Context

Recent advances in mobile computing and related technologies have significantly transformed the use of devices and their integration into daily life. This collection of devices includes laptops, tablets, smartphones, smartwatches, and other portable gadgets such as foldable devices, smart glasses, wearable fitness trackers, and portable gaming systems. These advancements have not only changed information communication and accessibility, but also paved the way for the emergence of a new computing paradigm: *ambient intelligence*. This paradigm focuses on seamlessly embedding communication, computation, and decision-making capabilities into everyday environments. By leveraging real-time or near-instantaneous information and historical data, ambient intelligence enhances user interactions with their surroundings, delivering personalized and context-aware experiences, and expanding the potential of mobile technology and its role in modern life.

Ambient intelligence builds upon pervasive systems by introducing a critical element: *the explicit requirement of intelligence*. This enables mobile applications to sense their environment, interpret the context of collected data, and adapt their behavior accordingly. This capability, known as *context-awareness*, allows applications to perceive contextual information and respond appropriately. Unlike traditional computing paradigms, pervasive systems paradigms, pervasive systems make information accessible anytime, in any way and from anywhere. This capability allows access to information in various contextual situations, including the user's environment (e.g., ambient noise levels, climate conditions), user preferences (e.g., preferred language), and the features of the device's capabilities (e.g., battery level, screen size). Pervasive systems thus contribute to making smart devices effective tools for understanding the user's context through their ability to collect, send, store, process, and communicate data. In such cases, context is often associated with three key dimensions: *where you are* (e.g., location and environment), *who you are with* (e.g., social context and relationships), and *what resources are nearby* (e.g., smart objects, networks, auxiliary devices).

In this PhD thesis, we focus on efficient access to multimedia documents in context-aware pervasive systems. These documents include a wide range of multimedia objects, such as text, images, audio, and video; HTML pages are a good example. In practice, multimedia documents play an important role in many fields including, among others, news, e-learning, healthcare, and tourism. The increasing diversity of devices (e.g., laptops, smart TVs, smartphones, tablets, and smartwatches) has largely contributed to making these documents omnipresent and accessible anytime, any way and from anywhere. Moreover, pervasive computing can improve their presentation as it ensures multi-device compatibility.

However, as users' contexts continually evolve and change over time (e.g., variations in noise level, location, and battery level), constraints may arise, influencing the proper execution or visualization of multimedia documents. One possible solution is to adapt their content to align as closely as possible to the current context. For instance, if the user is in a public place (e.g., bus or laboratory), the auditory content should not be played. In this case, it would be useful to convert this content into text. Similarly, if a user accesses multimedia documents on a tablet at home, the system may suggest displaying video and audio contents on the smart TV, as its screen is larger. The adaptation of multimedia documents is therefore an effective and interesting solution to this type of problem. It involves transforming their content so that it becomes compatible with the constraints of the current context. This is the focus of our research work.

## 2. Problem Statement

The primary goal of this doctoral thesis is to improve access to multimedia documents in context-aware pervasive systems by considering historical user data. Specifically, we focus on two aspects:

1) Managing historical user data with respect to content, format, and storage location.
2) Utilizing these data to perform useful tasks that enhance adaptation processes.

Before outlining our research plan, we briefly review the different categories of approaches for adapting multimedia documents in context-aware pervasive systems. This helps identify research gaps related to the stated objectives and address them accordingly.

## 2.1 Multimedia Document Adaptation in Context-aware Pervasive Systems

In the literature, several approaches for adapting multimedia document content in context-aware systems have been proposed. These proposals primarily differ in how they perform the adaptation process and the types of documents they address. Generally, adaptation processes begin by sensing users' contextual information. Next, this information is analyzed to identify constraints that prevent the context from aligning with the document features. Finally, adaptation actions are inferred and executed using dedicated services to deliver adapted documents. Depending on where decisions are made and adaptation actions are performed, multimedia adaptation approaches can be classified into four main categories, each with its own advantages and limitations:

**(1) Server-side adaptation:** The server handles the entire adaptation process.

- **Advantage:** Reduces the load on client devices.
- **Drawback:** Can cause server overload, affecting request processing times.

**(2) Client-side adaptation:** The device playing the multimedia documents executes the adaptation process.

- **Advantage:** Enables greater reactivity for the devices.
- **Drawback:** May not be feasible due to limited resources on the devices.

**(3) Proxy-based adaptation:** A proxy mediates between a client and the server to execute the adaptation process.

- **Advantage:** Reduces the load on both the client and the server.
- **Drawback:** Involves significant communication overhead.

**(4) Peer to peer (P2P) adaptation:** Adaptation tasks are distributed across multiple peers in a decentralized manner.

- **Advantage:** Enhances scalability and reduces dependency on central servers.
- **Drawback:** Complex network handling and the need for many connected nodes.

## 2.2 Research Gaps in Existing Multimedia Document Adaptation Approaches

By synthesizing the results already obtained, we observed that the approaches cited above primarily process real-time or near-instantaneous data collected to adapt multimedia documents based on context changes. In other words, they focus mainly on context collection, representation, and interpretation rather than considering its storage and analysis to integrate historical user data into the adaptation process. Logging historical data is supported by many context-aware pervasive applications, as it can help in making future decisions and provide prior knowledge. We cite as examples the application of machine-learning approaches to context-aware IoT cultural data [1], smartphone data analytics [2], activity recognition [3], and healthcare support [4].

Likewise, historical user data can play a crucial role in enhancing multimedia document adaptation processes by offering several key advantages:

**(1) Informed decision-making:** By offering insights into past user behaviors and preferences, historical data helps guide more effective and relevant adaptations for future use.

**(2) Enabling the use of advanced technologies:** Historical data serve as a foundational resource for implementing machine learning and data analysis models, which can identify patterns, predict user needs, and optimize content delivery.

**(3) Supporting personalization:** Historical data aids in developing recommendation systems and tailoring content to individual user preferences, creating personalized and engaging user experiences.

Two primary types of historical user data are relevant for multimedia adaptation processes, which can be used either jointly or separately:

1) General information about multimedia documents accessed, including user and resource characteristics.
2) Data derived from the specific features of context-aware pervasive systems, such as sensor readings and human-computer interactions.

While there is substantial research on the first type of historical user data (e.g., web log mining [5]), adaptation strategies leveraging the second type still remain underexplored. This gap highlights the potential to improve adaptation processes by integrating sensor-based contextual information and advanced analytics. Generally speaking, log data are extremely diverse and processing them is quite complex. This is due to the lack of standards and agreements defining the granularity, structure, content, format, and level of details provided by log events [6]. This introduces several management challenges.

## 2.3 Challenges in Managing Historical Data for Multimedia Adaptation Processes

Managing historical data for multimedia adaptation in context-aware pervasive systems involves several key challenges:

(1) **Data selection:** Determining which data to store is critical and depends on the intended purposes of the storage, such as adaptation, analysis, or optimization.

(2) **Data format:** The format in which data is stored directly impacts the efficiency of storage, retrieval, and further processing. Choosing the appropriate format is essential for facilitating seamless data handling.

(3) **Data location:** Deciding where the data will be stored is another challenge, particularly given the complexities of managing large volumes of data in big data contexts. Effective strategies are needed to address scalability and accessibility concerns.

(4) **Data utilization:** Leveraging stored data for useful tasks, such as user behavior analysis, predictive modeling, or optimizing adaptation rules, requires robust processing methods to extract meaningful insights and ensure practical applications.

By addressing these challenges, this research aims to bridge the gap between adaptation techniques and the underutilized potential of historical user data to enhance multimedia document accessibility and usability in context-aware pervasive systems.

## 3. Research Methodology

In light of the concerns and challenges stated above, we have drawn up a research methodology to achieve the determined objectives through the following steps:

(1) **Storing historical data with SQL and NoSQL databases:** The historical user data generated from multimedia document adaptation are stored in a database. The stored data are restricted to context element values and their corresponding adaptation actions resulting from context changes over time. The storage solution integrates both **SQL (relational)** and **NoSQL (non-relational)** databases as they operate independently of any specific data analysis framework, enabling a comparative analysis of these two models. On the one hand, relational databases handle large volumes of structured data, offering efficient storage and retrieval functionalities. On the other hand, NoSQL databases handle large data volumes at high speeds with scalable architectures, accommodating varying levels of data structuredness. Additionally, significant efforts have been made to bridge existing context modeling approaches, such as object-oriented, ontology-based, and key-value, with relational and NoSQL databases (e.g., [7], [8]).

(2) **Processing historical user data:** A set of functions is designed to process the historical user data, focusing mainly on data logging, data retrieving and data analyzing. These functions are intended to provide various options for designing efficient adaptation processes, addressing two key aspects: *issues related to the category of the adaptation process* and *those related to user requirements*. The issues related to the category of the adaptation process – namely *client-side*, *sever-side*, *proxy-based* and *peer-to-peer* – concern the system's ability to handle elements such as data volume,

analysis, processing performance, and scalability. The issues related to user requirements involve considering factors like data sharing (privacy) and data protection (security). Depending on where the historical user data are stored and processed, three methods, along with their hybridization, are proposed: *client-side management*, *server-side management*, *proxy-based management* and *hybrid management*. These methods help address these issues by efficiently placing functions for processing historical user data in accordance with application and user needs.

**(3) Leveraging historical user data for beneficial tasks:** To demonstrate the usefulness of historical user data in building intelligent context-aware applications, we explore case studies that use the proposed historical data management approaches: *client-side* and *server-side management*. These approaches incorporate a bag of data analysis algorithms designed for various purposes, primarily leveraging machine learning, data mining, and natural language processing techniques. In the case of client-side data management, we focus on how historical user data contribute to the personalization of adaptation rules through rule-learning mechanisms. This task involves predicting adaptation actions using data-driven approaches (i.e., based on data analysis and interpretation) rather than explicitly defining them. For server-side data management, our focus is on extracting meaningful and hidden patterns from historical user data, to provide a comprehensive classification of similar clusters for both contextual situations and user behaviors.

**(4) Validation via a real prototype and a specific application domain:** To demonstrate the feasibility of our proposal, we developed a real prototype that adapts multimedia documents based on contextual situations while storing the corresponding historical user data. From a functional perspective, most adaptation processes perform effectively. However, when validating adaptation approaches, prototypes are often based on simulation to collect context elements, particularly those related to connected objects (see for example [9], [10], [11]). While simulations are useful for many scenarios, the implementation of a real prototype offers the most practical, realistic, and viable solution for addressing technical and implementation challenges.

Furthermore, the validation of the outcomes should be as clear as possible and directly applicable to a widespread real-world application domain. To this end, we focus on the delivery of adapted multimedia contents for distance learning in context-aware pervasive systems, without loss of generality. This type of learning refers to an environment supported by computers, mobile devices, connected objects and wireless networks. It provides learners with content and interaction anytime, anyhow and anywhere. The starting point is the global shift driven by the COVID-19 pandemic, which significantly encouraged remote learning (online and distance learning) due to restrictions on physical interactions. It is important to note that the adaptation process is this research focuses exclusively on non-instructional purposes, as the primary emphasis is on multimedia content adaptation. Adaptation based on instructional context typically produces multimedia documents tailored to users' learning styles rather than addressing environmental requirements, which lies outside the scope of this study.

## 4. Contributions

This section provides an overview of the primary contributions of this thesis.

**(1) Designing a software component for managing historical user data:** The first contribution is the design of a software component for managing historical user data generated from the execution of multimedia document adaptation processes, enabling further processing. The stored data includes the context values and the corresponding adaptation actions. The proposed component can be integrated into multimedia document adaptation processes as it allows for the storage, retrieval and analysis of historical data. It is built around a well-devised, flexible and agile architecture, offering the ability to manage historical user data through three variants of data management, as well as their hybridization: ***client-side management***, ***proxy-based management***, and ***server-side management***. To achieve this, the context and adaptation actions are first modeled using an object-oriented approach. The resulting model is then converted into relational and NoSQL schemas through conversion procedures. Finally, algorithms for storing, retrieving and analyzing data are designed. Our proposal is validated through a set of experiments in which the historical user data are processed in different ways. The proposed software serves as a complementary element to existing adaptation approaches, aiming to enhance their overall performance by leveraging historical data to perform several beneficial tasks such as machine-learning tasks, and recommender systems.

**(2) A rule-learning approach for personalizing adaptation actions:** Our second contribution is a rule-learning approach that leverages historical user data generated from the execution of multimedia document adaptation processes to personalize context-aware adaptation rules. This intelligent system operates as an integrated component of the historical user data manager proposed in our first contribution, ensuring client-side management of historical data. The analysis of historical user data provides prior knowledge of past decisions, which can help predict future actions users may take. To achieve this, we employ rule-based classifiers – a well-known technique in machine learning and data mining – using *Eclat*, *sequential covering*, *FP-Growth*, and *decision tree algorithms*. This task involves predicting adaptation actions using data-driven approaches rather than explicitly defining them. Specifically, historical data are analyzed to uncover user behaviors, leading to the personalization of context-aware adaptation rules. This approach for building personalized adaptation rules is an unsupervised machine learning mechanism that does not require any pre-trained data or datasets.

(3) **Latent semantic analysis for classifying similar context situations and user behaviors:** Our final contribution involves extracting user behaviors using *latent semantic analysis* to provide a comprehensive classification of similar contextual situations and user clusters. This technique enhances the understanding of context, improves content delivery, and helps personalize user experiences. The rationale behind this is that latent semantic analysis is a well-established technique in text mining and natural language processing, particularly for applications that uncover relationships between terms and documents. Similarly, our approach treats each user's historical data as a textual document, with context elements and corresponding actions serving as words. Thus, our proposal offers a robust method for uncovering hidden semantic structures and extracting meaningful patterns from historical user data.

## 5. Thesis outline

This dissertation, besides the general introduction and conclusion, consists of five chapters, organized into two main parts: *theoretical background* and *contributions*. The first part focuses on basic concepts related to context-aware pervasive systems, with an emphasis on multimedia document adaptation in ubiquitous learning systems. It also provides a state-of-the-art review on the adaptation of multimedia documents in pervasive systems. The second part, which constitutes the core of the thesis, is dedicated to the proposed contributions. Together, these two parts offer a comprehensive exploration of key concepts, research gaps, contributions, and experimental results, ensuring a logical structure throughout the dissertation.

**Part I: Theoretical Background**

This part consists of two chapters, as follows:

- **Chapter 2,** entitled "Context-aware Pervasive Systems in Ambient Intelligence", explores the core principles of context-aware pervasive computing, including its fundamental elements, characteristics, operating principles, and applications. It also emphasizes the architectural components essential for designing context-aware pervasive systems.
- **Chapter 3,** entitled "Multimedia Document Adaptation in Context-Aware Pervasive Systems", explores the fundamental concepts of multimedia documents. It also reviews current multimedia adaptation approaches and includes a case study on adapting multimedia content for ubiquitous learning scenarios.

**Part II: Contributions**

The second part contains three chapters, each detailing one of the thesis's primary contributions.

- **Chapter 4,** entitled “Management of historical data in context-aware MDA processes”, presents our first contribution: the integration of historical user data into multimedia adaptation processes. It details the modeling, storage, retrieval, and analysis strategies. The chapter also includes validation through prototyping and experimentation across diverse scenarios.
- **Chapter 5,** entitled “Rule-learning for the personalization of adaptation actions”, presents our second contribution, which explores the application of rule-learning techniques to historical user data using client-side algorithms. It leverages algorithms such as Eclat, sequential covering, FP-Growth, and decision trees. The chapter also includes validation through experimentation, performance measurements, and comparisons across various test cases.
- **Chapter 6,** entitled “Latent Semantic Analysis for the Classification of Similar Contextual Situations and User Behaviors”, presents our final contribution, which employs latent semantic to analyze historical user data. It focuses on server-side algorithms for classifying context situations and user behaviors clusters. This chapter also provides validation through tests and performance evaluations.

**Conclusion and Future Work:** This dissertation concludes with Chapter 7, which provides a summary of the work completed and the results achieved. Additionally, it reflects on ongoing ideas and unfinished tasks, highlighting the potential for further exploration in the future.

# Chapter II

# Context-aware Pervasive Systems in Ambient Intelligence

## Summary

# CHAPTER II

## Context-aware Pervasive Systems in Ambient Intelligence

### 1. Introduction

Recent advances in mobile computing have revolutionized device usage and integration into daily life. From laptops and smartphones to smartwatches and wearable trackers, these innovations have broadened the scope of computing and connectivity. These developments have also paved the way for the emergence of the ambient intelligence paradigm, which integrates communication, computation, and decision-making capabilities into everyday environments. Ambient intelligence enhances user experiences by delivering personalized and context-aware interactions, adapting dynamically to the user's needs and surroundings.

Building on the foundation of pervasive systems, ambient intelligence introduces a critical layer of intelligence, enabling devices to sense, interpret, and adapt to their environment. At the heart of this capability lies context-awareness – the ability to perceive and respond to contextual information, such as a user's location, preferences, activities, and device features. Pervasive systems ensure ubiquitous access to information, transforming smart devices into powerful tools that dynamically respond to various contextual dimensions, including spatial environments, social interactions, and nearby resources. By integrating these context-aware capabilities, devices evolve into adaptive systems that enhance both usability and functionality in a wide range of scenarios.

In this chapter, we explore the principles and core elements of ambient intelligence, with a particular focus on context-aware pervasive systems. We begin by defining key concepts of context and context-awareness, emphasizing their significance and various modeling approaches. Furthermore, we examine the design elements that underpin context-aware systems and highlight their applications across different fields, demonstrating their transformative potential in modern computing environments.

# 2. Ambient Intelligence

In the rapidly advancing technological era, ambient intelligence (AmI) stands out as a groundbreaking concept focused on embedding intelligence into everyday environments. The following sections provide a comprehensive overview of the principles that underpin AmI, the key elements that make these systems functional, and a comparison of AmI with other closely related paradigms, mainly pervasive and ubiquitous systems.

## 2.1 Principle of Ambient Intelligence

AmI is a computing paradigm designed to create smart and responsive environments that adapt to human needs and preferences. The core idea behind AmI is to enrich the environment with technology, enabling systems to make decisions that benefit users based on real-time or near-instantaneous information and accumulated historical data. This is achieved by incorporating various technologies such as sensors, actuators, connected objects, high-bandwidth networks, advanced devices, and artificial intelligence (AI) algorithms. Therefore, AmI is based on the vision of seamlessly embedding computing power and intelligence into everyday objects and infrastructures, ensuring that computing devices and systems become unobtrusive, context-aware, and capable of proactively adapting to human needs and preferences. AmI-based systems are characterized by several key features, the most important of which are as follows [12]:

- **Proactivity:** AmI systems are proactive, anticipating and fulfilling user needs without explicit input. They can autonomously provide relevant information and initiate actions based on users' experiences, preferences, and other factors.
- **Context-awareness:** AmI systems are designed to be aware of the context in which they operate, including several aspects such as user location, time, environmental conditions, and preferences. This enables them to adjust their behavior according to changes in circumstances.
- **Adaptability:** AmI systems can adapt their behavior and functionality to meet users' preferences and requirements, offering personalized responses and services based on past experiences.
- **Seamlessness:** AmI systems aim for seamless and unobtrusive integration of technology into daily life. The technology should be invisible, blending naturally into the environment and activities without disrupting or confusing users.

## 2.2 Fundamental Elements of Ambient Intelligence

AmI is a multidisciplinary approach that has a strong relationship with several areas of computer science, the most relevant of which are presented in Figure 2.1 [13].

- **Ubiquitous and pervasive computing:** Ubiquitous and pervasive computing form the foundation of AmI. They involve the integration of computing devices into everyday objects and environments, making technology pervasive and seamlessly accessible. This omnipresent computing power enables AmI systems to observe, understand, and respond to human activities, needs, preferences, etc.
- **Sensor and actuators:** AmI relies on sensors and actuators that capture, collect and act on data from the surrounding environment, such as motion sensors, temperature sensors, and light sensors. The goal is to gather various types of information that help determine the system's behavior.
- **Networks:** The integration of networks within AmI is essential for ensuring functionality and responsiveness in smart spaces. When various objects (e.g., sensors, actuators) are interconnected through multiple communication technologies (e.g., Wi-Fi, ZigBee, Bluetooth), they must be capable of sharing, processing, and transmitting data both among themselves and with users. This connectivity enhances the efficiency and accessibility of AmI applications, enabling a more personalized user experience.
- **Artificial intelligence:** AmI systems use AI to process and analyze collected data. This enables these systems to perform several beneficial tasks, such as context understanding and interpretation, and predicting user actions.
- **Human-computer interaction:** AmI aims to provide seamless and natural interactions between humans and technology. Human-computer interaction (HCI) plays a key role in creating intuitive and user-friendly interfaces through various methods such as voice recognition, gesture control, augmented reality, and tangible interfaces.

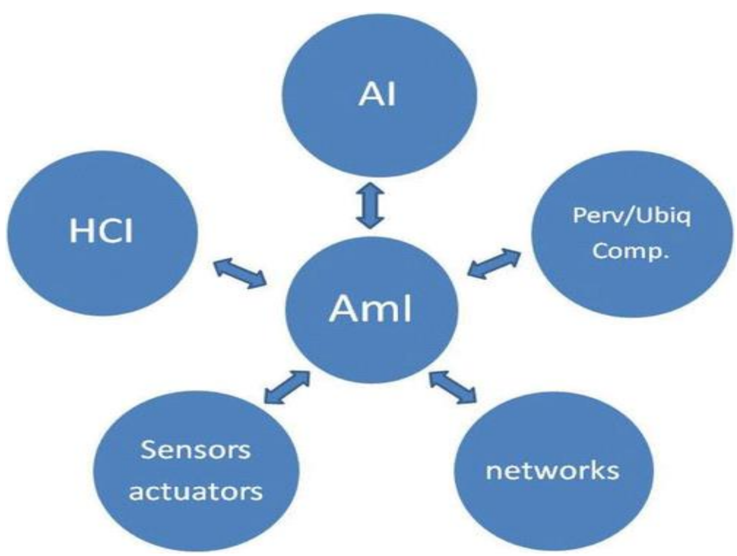



**Figure 2.1.** Relationship between AmI and other research areas [12].

It is important to note that although AmI incorporates all of the above research areas, it should not be confused with any one of them individually. Networks, sensors and actuators, HCI, ubiquitous and pervasive computing, and AI are all relevant, but none of them independently define the concept of AmI. Rather, it is the concept of AmI itself that brings together these disciplines to deliver flexible and intelligent services to users within their environments.

## 2.3 Nuances Differentiating Pervasive Systems, Ubiquitous Systems and AmI-based Systems

This section explains the concepts of pervasive systems, ubiquitous systems, and AmI-based systems, highlighting the subtle differences between them [14].

- ***Pervasive systems*** are computing systems designed to seamlessly integrate into the environment, becoming effectively invisible to users while ensuring widespread, unobtrusive, and constant availability of resources. They focus on embedding technology into everyday environments, enabling connectivity and interaction with minimal user intervention through robust infrastructure and middleware. These systems are infrastructure-driven, often associated with mobile computing, IoT, and smart devices, aiming to make computing accessible anytime and anywhere without disrupting daily life. Examples include smart homes, healthcare monitoring systems, and pervasive learning environments [14].
- ***Ubiquitous systems***, a concept introduced by Mark Weiser, focus on integrating computing seamlessly into daily life, making it a natural and omnipresent part of the environment. The vision emphasizes user-centric, context-aware applications that enable intuitive, unconscious interactions with technology, akin to using electricity. These systems prioritize user experience and aim to "disappear" into the background, leveraging smart devices, wearable technology, and context-awareness to enhance daily life. Examples include smart assistants like Alexa, location-based services, and augmented reality applications [14].
- ***Ambient Intelligence (AmI)*** is a subfield of ubiquitous and pervasive computing that focuses on creating environments sensitive, adaptive, and responsive to human presence and actions. It emphasizes intelligent spaces that proactively support user needs by learning and adapting over time, leveraging artificial intelligence (AI) to analyze context and predict behaviors. With a strong human-centric approach, AmI incorporates social interaction, emotional intelligence, and personalization, seamlessly integrating sensors, devices, and context-aware technologies. Examples include adaptive smart homes, personalized healthcare systems, and intelligent public spaces that enhance quality of life by anticipating user needs [14].

In summary:

- **Pervasive systems** emphasize infrastructure and technology's omnipresence.
- **Ubiquitous systems** focus on making computing invisible and intuitive.
- **Ambient intelligence** builds on both, adding AI-driven adaptability and proactivity.

Table 2.1 summarizes the subtle differences between pervasive systems, ubiquitous systems and AmI-based systems.

| Feature | Pervasive systems | Ubiquitous systems | Ambient intelligence |
|---|---|---|---|
| **Primary Goal** | Seamless integration into the environment. | Invisible and natural computing presence. | Intelligent, adaptive environments. |
| **Focus** | Infrastructure and connectivity. | User experience and natural interaction. | AI-based adaptation and intelligence. |
| **Key Technologies** | IoT, middleware, networking. | IoT, context-awareness, wearables. | AI, machine learning, sensors. |
| **Application Examples** | Smart environments, healthcare systems. | Location-aware apps, smart assistants. | Personalized homes, predictive healthcare. |

**Table 2.1.** Subtle differences between pervasive, ubiquitous, and AmI-based systems.

## 3. Context-aware Pervasive Systems: Overview

Context-aware pervasive systems refer to a class of systems that dynamically adapt their behavior based on the contextual information obtained from the environment in which they operate [14]. In what fellows, we present the fundamental elements of these systems.

### 3.1 What is Context?

According to the Oxford Dictionary, context defined as:

> "*The circumstances that establish the setting for an event, statement, or idea, and in relation to which it can be comprehensively understood.*"

Similarly, the Cambridge Dictionary defines context as :

> "*The situation within which something exists or occurs, and that can aid in its explanation.*"

As reported in [14], a review of the scientific literature across several disciplines shows that there are many different interpretations of the word "context". Researchers from various fields attempted to define it based on their own needs and areas of interest. Table 2.2 presents a historical compilation of the various interpretations of context.

| Reference | Context essence |
|---|---|
| Dey et al. [15] | *any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity and state of people, groups and computational and physical objects.* |
| Schilit and Theimer [16] | *places, identities of proximate individuals, objects, and changes to those objects over time.* |
| Schilit et al. [17] | *the constantly evolving execution environment.* |
| Brown et al. [18] | *the location, the identities of individuals surrounding the user, the time of day, season, temperature, etc.* |
| Pascoe [19] | *the subset of physical and conceptual conditions relevant to a particular entity.* |
| Brézillon and Pomerol [20] | *the entirety of knowledge that constrains problem-solving at a particular step without explicit intervention.* |
| Schmidt [21] | *the knowledge regarding the user's and device's state, encompassing surroundings, situation, and tasks, underscoring that context extends beyond mere location.* |
| Winograd [22] | *operating term referring to something is context based on its usage in interpretation, rather than its inherent properties.* |
| Mostéfaoui et al. [23] | *denotes the elements around the focal point, providing further information on where, who, what, so enhancing comprehension.* |
| Chaari et al. [24] | *the set of parameters external to the application that can influence its behaviour by establishing new perspectives on its data and services.* |
| Strassner et al. [25] | *a collection of measured (perceived facts) and inferred knowledge (derived from learning and computational reasoning applied to past and present contexts) that defines the state and environment in which an entity exists or has existed.* |

**Table 2.2.** Compilation of context definitions.

An examination of the above definitions of the term *'context'* reveals a significant evolution in its meaning, particularly in light of advances in pervasive systems development. Initially, the term was associated with the location and identity of users and objects. Over time, its scope has expanded to include additional environmental information, covering both computational and physical environments. More generally, we conclude that all these definitions agree that context encompasses any information useful for describing a situation related to communication between users, services, and the surrounding environment. The key difference between these definitions lies in the specificity of what constitutes a particular environment. Some focus on the user environment, while others emphasize the application environment.

In summary, context refers to any information that helps characterize the situation in which different system entities interact. This includes various aspects, such as the location of the interaction, the identity of individuals or groups involved, and the state or condition of both humans and the computational or physical objects in the environment.

## 3.2 Context-awareness

This section presents the key definitions of to the term *context-awareness* from various perspectives, as summarized in Table 2.3.

| Reference | Context-awareness essence |
|---|---|
| Dey [15] | *a system qualifies as context-aware if it utilizes context to deliver pertinent information and/or services to the user, with relevance based on the user's task.* |
| Schilit and Theimer [16] | *the capability of mobile user's applications to identify and react to changes in their surrounding environment.* |
| Schilit et al. [17] | *context-awareness allows systems to adjust based on the user's location, the presence of nearby individuals, available devices, and temporal changes to these factors. A system possessing these capabilities can analyze the computing environment and respond to changes in it.* |
| Brown et al. [18] | *characterized context-aware applications as systems that adapt their behavior based on the user's context.* |
| Chaari et al. [24] | *context-aware systems are those capable of sensing the user's situational context within their environment and subsequently adapting the system's behavior — including services, data, and interfaces — without requiring explicit user intervention.* |
| Ryan et al. [26] | *a computer's capability of sensing and reacting to information regarding its environment, including factors such as location, time, temperature, and user identity.* |
| Salber et al. [27] | *the capacity to offer optimal flexibility of a computational service through real-time context sensing.* |
| Lieberman and Selker [28] | *in contrast to traditional applications that obtain explicit input data from users to generate explicit output, a context-aware application additionally considers implicit data, typically gathered autonomously.* |
| Burrell and Gay [29] | *the use of environmental features, such as user's location, time, identity, and activity, to make the computing device able to deliver information pertinent to the current context.* |
| Rohn [30] | *a system qualifies as context-aware if it is able to extract, interpret, and utilizes context information, thereby adapting its functionality to the current context of use.* |
| Barkhuus [31] | *context-awareness is the application's ability to detect and react to environment variables autonomously.* |
| Abbas et al. [32] | *context-aware systems can provide users with information that is relevant and tailored to their needs, as well as the context that affects their behavior during interactions with information systems.* |
| Christopoulou [33] | *a mobile application is considered as context-aware when it utilizes context to deliver pertinent information or facilitate services for users, with relevance based upon the user's current task and profile.* |
| Sindico [34] | *Ability of a system to modify its behaviors based on the characteristics of the environment.* |

**Table 2.3.** Some definitions related to context-awareness.

The definitions presented in Table 2.3 highlight two fundamental aspects of context-awareness: the ability to perceive context and the ability to act accordingly. A context-aware system not only senses environmental conditions but also adapts to changes in those conditions. More broadly, context-awareness has become increasingly important in the field of distributed systems since the 1990s, offering a potential solution to many challenges posed by the use of mobile terminals in dynamic environments (see Table 2.4).

| Mobile computing | Pervasive computing | Context-aware pervasive computing |
|---|---|---|
| Mobile | Ubiquitous | Integrated |
| Wireless | Interactive | Context-aware |
| Networked | Interoperable | Personalized |
| Location-aware | Distributed | Adaptive |
| Security | Scalable | Anticipatory |

**Table 2.4.** Ambient intelligence positioning [12].

### 3.3 The Importance and Benefits of Context-aware Pervasive Systems

By leveraging sensors, data analysis techniques, and machine learning algorithms, context-aware pervasive systems deliver seamless and personalized services. Their importance and benefits include the following:

- **Enhanced user experience:** Context-aware pervasive systems significantly improve user experiences by tailoring services and to individual preferences and situations. For instance, in a smart home, the system can automatically adjust lighting, temperature, and music based on the user's presence and preferences, creating a comfortable and personalized environment [35].
- **Improved efficiency:** Context-aware pervasive systems optimize resource allocation, automate routine tasks, and provide timely assistance. For example, smart home systems can regulate temperature and lighting based on occupancy, reducing energy consumption and enhancing comfort [35].
- **Proactive assistance and decision support:** Context-aware pervasive systems anticipate user needs and provide proactive support. For example, a healthcare monitoring system can detect abnormal vital signs and alert caregivers, enabling timely interventions. Similarly, navigation systems can provide real-time traffic updates and suggest alternative routes to avoid congestion [36].
- **Seamless and ubiquitous connectivity:** These systems enable smooth interaction between devices, services, and environments. They also facilitate seamless switching between different devices and platforms, allowing users to continue their tasks easily. For instance, a multimedia streaming system can adjust content and quality based on

device capabilities and network conditions, ensuring an uninterrupted viewing experience [16].

- **Adaptive service delivery:** By considering user preferences, location, and other contextual factors, context-aware systems provide personalized and relevant services. This enhances the effectiveness of advertising, recommendations, and information sharing, improving user engagement and satisfaction [35].
- **Intelligent decision-making:** These systems analyze and interpret complex contextual data to support intelligent decision-making. They provide valuable insights and recommendations across domains such as healthcare, logistics, and emergency management.
- **Context-aware social interactions:** These systems facilitate meaningful context-aware social interactions by providing relevant information about individuals, locations, or events. They enhance social networking, collaboration, and community engagement, such as during conferences or on online platforms [37].

## 4. Context Management

The shift toward context-aware pervasive computing paradigm introduces two major challenges [14]:

1. **Diverse user devices and applications:** The increasing variety of devices – such as smartphones, tablets, smartwatches, and emerging technologies – requires a flexible and adaptable interaction model. This model should ensure a seamless and consistent user experience across devices, maintaining user-friendliness regardless of the device in use.
2. **Understanding and adapting to environmental contexts:** Applications must recognize and adapt to the diverse environments in which they are used. Contextual factors such as location, network connectivity, user preferences, and available resources can significantly impact user experience. By understanding these factors, applications can dynamically adjust their behavior and interface to provide personalized, relevant experiences.

Addressing these challenges requires a balance between accommodating device diversity and optimizing application performance across various environments. This can be achieved through efficient context management techniques. By integrating these techniques, applications can seamlessly adapt to a wide range of devices and environmental conditions, enhancing user experience. This holistic approach ensures smooth interactions across devices while optimizing performance in response to the specific contextual nuances of each environment.

## 4.1 Context Classification

Context information is typically classified into four categories, as illustrated in Figure 2.2 [14].

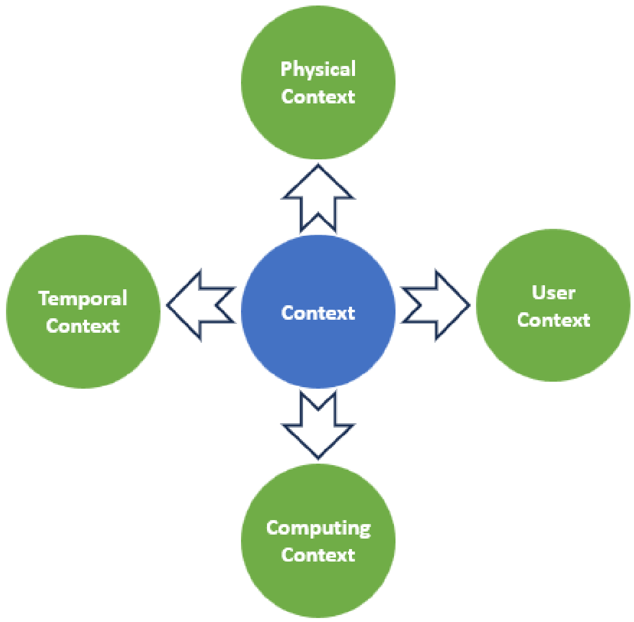



**Figure 2.2.** Context classification [14].

**1. Physical context:** Encompasses information about physical environment in which interactions occur. It includes factors such as temperature, lighting, noise levels, and other environmental conditions. such as temperature, lighting, noise level, etc.

**2. User context:** Pertains to users and their associated information, such as profiles, locations, preferences, and activities. It includes details about the characteristics and attributes of individuals involved in the interaction.

**3. Computing context:** Involves the computing resources available within the environment, such as network connectivity, device capabilities, printers, and communication bandwidth. It captures the technical aspects of the computing environment that influence interactions.

**4. Temporal context:** Relates to time and includes aspects such as season, year, month, week, day, and time of day. It also encompasses events and schedules, helping to account for the dynamic and time-sensitive nature of interactions.

It should be noted that context may also be categorized from various viewpoints. Below, we outline alternative context classifications found in the literature:

- Bill Schilit [17] categorized context into three types:
    - **Computing context:** The networked and computing environment.
    - **User context:** The user's location and related conditions.

- **Physical context:** The physical environmental factors.

– Chen and Kotz [38] defined two categories:

- **Active context:** Elements that directly affect application behavior.
- **Passive context:** Elements that are essential but not immediately critical to the program; it provides information for later retrieval.

– Petrelli et al. [39] identified two types:

- **Material context:** Includes location, devices, and existing platforms.
- **Social context:** Includes relationships among individuals.

– Dey [15] categorized context into two different categories:

- **Primary context:** includes information about location, identity, time, and behavior.
- **Secondary context:** Includes information inferred from primary context.

– Hofer et al. [40] defined two types:

- **Physical context:** Accessible via physical sensors (e.g., temperature, humidity).
- **Logical context:** Involves interaction information, such as the user's emotional state and intentions.

– Henricksen [41] classified context into four categories:

- **Sensed context:** Information directly perceived by sensors, such as temperature or humidity.
- **Static context:** Information that remains constant over time, such as sensor identifiers or personal identity.
- **Profile context:** Information that changes infrequently, such as sensor location or user status.
- **Derived context:** Information computed from primary context, such as the distance between two sensors.

– Razzaque et al. [42] proposed a taxonomy with six categories:

- **User context:** Includes user profiles, identity, and interactions with others.
- **Physical context:** Integrates physical environmental information, such as location, temperature, and noise level.

- **Network context:** Provides network details such as connection, bandwidth, and protocols.
- **Activity context:** Involves events in the environment, such as user movements or environmental changes.
- **Device context:** Identifies usable devices and their details (e.g., ID, location, battery level).
- **Service context:** Encompasses the features and functionalities provided by the system.

– In [43], authors presented a comprehensive classification based on various parameters influencing a user's situation:

- **Surroundings:** Includes time, location, etc.
- **Application:** Encompasses software, hardware, etc.
- **Individual:** Covers identity, preferences, etc.

– Geihs and Wagner [44] assert that contextual information includes:

- The state of execution environment.
- The state of computing device.
- User preferences.

None of these classification schemes can be deemed universally superior, as each serves specific purposes and is suited to different situations. Context can be addressed from the preferred viewpoint by selecting the most appropriate classification scheme for the given application. In the following chapters, we adopt the definition provided in [14] as it is broad enough to encompass any kind of information, without being limited to specific purposes.

## 4.2 Context Modeling Approaches

Context modeling refers to the process of structuring and organizing context information in a meaningful way that can be understood and utilized by context-aware systems. Once detected, context information should be stored for future processing, necessitating a structure or model that stores this information in a machine-readable form. This structured representation is known as *context modeling*. As mentioned in [45], various approaches to context modeling exist, differing in how they represent and exchange context information.

- **Key-value model:** The key-value model offers a simple and straightforward representation of context elements by capturing them using key-value pairs. For instance, the user's location can be represented by the following key-value pairs: "*latitude: 37.8849*" and "*longitude: -122.1564*". This model is particularly useful for expressing context elements with discrete values and requires minimal structural complexity. Furthermore, it is easy to implement and is considered the simplest data structure for modeling context information. However, it is unsuitable for representing complex structures [45].

- **Object-oriented model:** The object-oriented model, inspired by object-oriented programming principles, represents context types as classes and their values as objects. This model considers both relevant attributes and behaviors, utilizing concepts such as encapsulation, inheritance, and polymorphism. For example, a weather context object might include attributes like temperature, humidity, and wind speed, alongside methods for accessing and manipulating these attributes. The primary advantages of this model are encapsulation and reusability. Despite these benefits, it is less suited for knowledge sharing in open and dynamic environments [45].

- **Markup schema-based model:** The markup schema-based model follows a hierarchical approach to representing context information. Context and its attributes are structured using a hierarchy of tags, facilitating efficient organization, categorization, retrieval, and manipulation. This model is well-suited for distributed systems and provides a better representation of context information. However, it struggles with describing complex relationships between context elements [45].

- **Graphic model:** The graphical model uses visual representations, often employing the Unified Modeling Language (UML), to depict context information. This approach models the relationships and dependencies between different context elements in a clear and intuitive manner. For instance, a graphical model could illustrate connections between sensors, devices, and users in a smart environment, simplifying system design and comprehension [46].

- **Logical model:** The logical model represents context information formally through expressions, facts, and rule-based inference systems. This approach enables rigorous reasoning and inference capabilities. By relying on logical expressions and rules, new facts can be deduced from existing context information. For example, a logical model might infer whether a user is awake or asleep, based on sensor inputs and historical data. The primary advantage of this approach is its high degree of formality. However, it is less effective in describing relationships between continuous data [46].

- **Ontology-based model:** The ontology-based model uses ontologies to describe context information and the relationships between them. An ontology is a formal representation of knowledge that defines concepts, relationships, and properties within a specific domain. This approach provides a shared vocabulary and conceptual framework for describing and reasoning about context values. Ontology-based models enable efficient reasoning and knowledge representation, offering the most comprehensive degree of semantic expressiveness. However, designing ontologies can be complex and time-consuming [46].

As discussed in [46], each context modeling has its own strengths and weaknesses, as summarized in Table 2.5.

| Comparison criterion | Key-value | Logic | Graphical | Markup-scheme | Ontology-based |
|---|---|---|---|---|---|
| Formalism | None | High | Low | Medium | High |
| Simplicity | High | Medium | Medium | Medium | Low |
| Expressiveness | Low | Medium | High | Medium | High |
| Reasoning | Low | Medium | Low | High | Medium |
| Design tools | None | Low | High | High | High |

**Table 2.5.** Comparison of context-modeling approaches [46].

## 5. Elements of Context-aware Pervasive Systems

The design of context-aware pervasive systems typically follows a three-layered architecture consisting of three key elements: *sensing*, *thinking*, and *acting*, as illustrated in Figure 2.3. These layers are functionally divided into five primary components [47]. While other architectural designs exist, the three-layered architecture is the most widely adopted.

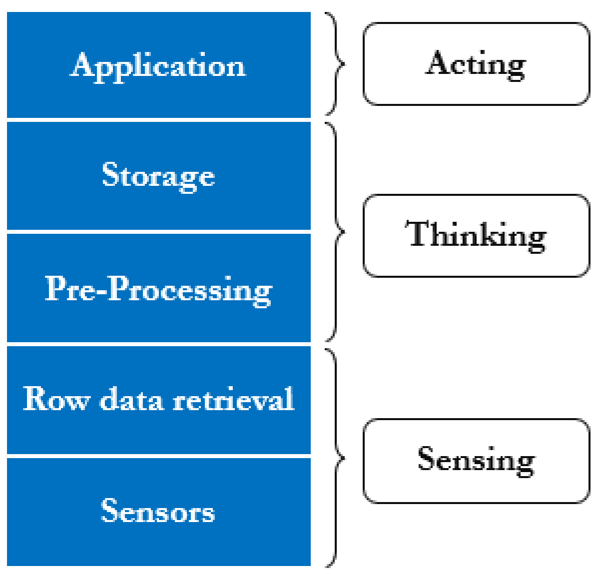



**Figure 2.3.** Basic elements of context-aware pervasive applications [47].

### 5.1 Sensing

*Sensing* is the initial layer of a context-aware pervasive system. It involves gathering information or data, specifically about the physical world, for use by the system. These data are collected from various sources, including sensors and sensing devices. Sensing is crucial for context identification and consists of two main components: *sensors* and *raw data retrieval components*.

### 5.1.1 Sensors Component

Context-aware pervasive systems utilize a variety of sensing technologies to capture contextual information from the surrounding environment and user interactions. It is important to note that "sensors" include not only hardware devices but also any source that may offer relevant contextual information. Sensors are categorized into three groups based on their data-gathering methods: *physical*, *logical*, and *virtual* [48].

- **Physical sensors:** These are hardware devices that capture physical data, such as location, acceleration, and temperature. They are the most common type of sensors used in context-aware systems.
- **Virtual sensors:** These derive contextual information through software applications or services. For example, user activity can be recognized by monitoring mouse movements and keyboard inputs.
- **Logical sensors:** These combine information from physical and virtual sensors with additional data from other resources to deliver integrated contextual information.

### 5.1.2 Raw Data Retrieval Component

This component focuses on extracting unprocessed contextual data from various sources, such as physical sensors and APIs for virtual and logical sensors. It organizes the data extraction to ensure that it is ready for use in further processing [48].

## 5.2 Thinking

The *thinking* layer represents the core intelligence of context-aware pervasive systems. It involves analyzing collected context information through algorithms and decision-making mechanisms to infer the user's current situation and needs. This layer aims to provide suitable responses to users and the environment. It comprises two components: *storage* and *preprocessing components* [48].

### 5.2.1 Storage Component

This component organizes collected data into a specified format (i.e., a context model) to enable efficient data retrieval and processing. Raw sensor data, often technical and unsuitable for direct application use, undergoes transformation to increase its abstraction level. Clients can access contextual information in two modes: *synchronous* and *asynchronous* [49].

- **Synchronous mode:** The client requests context changes via remote method calls by sending a message and waiting for a response.
- **Asynchronous mode:** Clients subscribe to particular events of interest and are notified immediately when those events occur.

### 5.2.2 Preprocessing Component

The preprocessing component interprets contextual data collected by sensors. This is achieved through operations such as aggregation or composition, which combine individual sensed data into high-level contextual information. The aim is to derive meaningful insights of the collected data. For example, temperature data collected over time by sensors in a building can reveal temperature trends or ranges in a specific area.

Additionally, this component analyzes stored data to gain further knowledge or predict future values, which aids in decision-making. This capability is critical for determining appropriate responses or actions for users and the environment.

## 5.3 Acting

The *acting* layer represents the application interface where actions are executed in response to different events and context changes. Based on the information from the *thinking* layer, the system adapts its behavior to provide personalized services. Adaptations can range from simple adjustments, such as changing device settings, to complex actions, such as recommending relevant content or dynamically reconfiguring application functionality. For instance, if a mobile device's light sensor detects poor illumination, text may be displayed in a more suitable color contrast.

# 6. Context-aware Pervasive Applications

Context-aware pervasive applications are utilized across various domains. Below, we explore examples of such systems in the areas of location-based services, smart homes and buildings, and healthcare and wellness. It is essential to address critical aspects such as privacy, security, and ethical concerns during the development and deployment of these applications to maximize their benefits while safeguarding user rights and welfare.

## 6.1 Location-based Services

Location-based services are among the most prevalent and widely adopted applications of context-aware pervasive systems. They employ user location data to provide personalized, location-specific content, recommendations, and assistance. Important examples include:

- **Navigation:** Context-aware systems employ GPS, Wi-Fi, or cellular network signals to accurately determine user location. This enables them to provide turn-by-turn directions, real-time traffic updates, and personalized route suggestions.
- **Social networking:** Location-based social networking applications enhance social interactions by allowing users to connect with nearby friends, discover local events, and share location-tagged content [47].

- **Advertising:** In this domain, context-aware systems deliver targeted advertisements based on users' current location, preferences, and past behaviors. For example, mobile applications can display relevant promotions from nearby stores to users in close proximity [47].
- **Emergency services:** Context-aware systems improve emergency response by incorporating location information. Authorities can quickly locate individuals in distress and dispatch assistance. For instance, emergency call centers can use GPS to pinpoint the caller's location, enabling faster responses and potentially saving lives [41].

## 6.2 Smart Homes and Buildings

Context-aware pervasive systems play a vital role in creating intelligent and adaptive environments in smart homes and buildings. These systems use contextual data to enhance comfort, convenience, energy efficiency, and security.

- **Smart homes:** Context-aware systems can automatically monitor factors such as occupancy, lighting conditions, temperature, and user preferences to create personalized living environments. For example, the system can adjust lights and thermostats based on user presence or learn preferences over time to optimize comfort. Additionally, these systems improve energy efficiency by managing devices and appliances intelligently. For instance, they can turn off lights in unoccupied rooms or adjust heating and cooling based on occupancy patterns [50].
- **Smart buildings:** In smart buildings, context-aware systems enhance space utilization and energy efficiency by adjusting lighting, heating, and ventilation based on occupancy patterns. They also bolster security by detecting abnormal activities or unauthorized access, such as unusual movement or the absence of authorized individuals [50].

## 6.3 Healthcare and Wellness Systems

Context-aware pervasive systems hold significant potential in healthcare and wellness, enabling personalized and proactive care. These systems leverage contextual data to monitor health, provide timely interventions, and support wellness management [51].

- **Health monitoring:** Context-aware systems collect and analyze health-related data such as vital signs, activity levels, and environmental conditions. This information is used to detect and predict events like falls [24], abnormal heart rates, or changes in activity patterns, enabling swift interventions [51].
- **Remote patient monitoring:** These systems facilitate remote health monitoring by collecting data from wearable devices, sensors, and mobile applications. Context-aware

algorithms identify deviations from normal patterns and trigger appropriate responses, reducing hospital readmissions and improving patient outcomes [51].

- **Wellness management:** Wellness applications use contextual data to provide personalized recommendations for healthy living. For instance, they can suggest exercise routines based on activity levels and preferences or provide medication reminders tailored to individual needs [51].
- **Healthcare decision support:** By analyzing patient contexts and historical data, context-aware systems can assist healthcare providers in diagnosis, treatment selection, and care planning. This enhances the effectiveness and efficiency of healthcare delivery [51].

## 7. Conclusion

By combining the ubiquity of pervasive systems with the adaptive intelligence of ambient intelligence, context-aware pervasive systems represent a major breakthrough in the evolution of computing technologies. These systems leverage context awareness to deliver personalized, efficient, and dynamic interactions, improving user experiences across diverse environments and applications. Through the principles, modeling approaches, and design elements discussed in this chapter, context-awareness is more than a technical capability but rather a framework for building systems that respond intuitively to user needs and environmental conditions. Whether applied in healthcare, education, smart cities, or other domains, these systems unlock new possibilities for seamless integration between users, devices, and their surroundings.

As mobile computing and related technologies continue to advance, the future of context-aware pervasive systems will likely feature even greater sophistication in sensing, interpretation, and adaptability. This progress will empower devices to not only understand and respond to their current contexts but also anticipate and proactively address users' evolving needs to realize environments that are truly intelligent, intuitive, and user-centered. This is the subject of the next chapter in which we address how context-aware pervasive systems enhance the delivery and usability of multimedia documents.

# Chapter III

# Multimedia Document Adaptation in Context-Aware Pervasive Systems

## Summary

# CHAPTER III

## Multimedia Document Adaptation in Context-Aware Pervasive Systems

### 1. Introduction

Context-aware pervasive systems mark a significant advancement in computing technologies, creating environments enriched with communication, computation, and decision-making capabilities. These systems enable users to access information anytime, anywhere, and in anyway, seamlessly integrating the ubiquity of pervasive systems with the adaptive intelligence of ambient intelligence. By leveraging context-awareness, these systems collect and interpret information about the user's environment, preferences, and device capabilities, delivering personalized, efficient, and dynamic interactions that enhance user experiences.

In this chapter, we focus on efficient access to multimedia documents within context-aware pervasive systems. Multimedia documents contain diverse objects such as text, images, audio, and video, and they play a crucial role in various domains, including e-learning, healthcare, news, and tourism. The proliferation of devices such as laptops, smartphones, tablets, and smart TVs has made these documents widely accessible across platforms, with pervasive computing enhancing their compatibility and presentation. However, evolving user contexts – such as changes in noise levels, location, or device constraints – can pose challenges to the effective execution or visualization of multimedia content. For instance, auditory content may need to be converted to text in public settings, or video content from a tablet could be displayed on a larger smart TV at home to enhance usability.

This chapter highlights multimedia content adaptation as a practical and effective solution to address these challenges, ensuring compatibility and optimizing user experiences within context-aware pervasive systems. We begin by introducing foundational concepts related to multimedia document adaptation. A comprehensive literature review follows, detailing existing approaches in the field. Finally, we present a case study that illustrates the adaptation of multimedia content in context-aware pervasive learning environments.

## 2. Multimedia Document Adaptation: Basic Concepts

This section introduces the fundamentals concepts of multimedia document adaptation as well as languages and tools for creating multimedia presentations.

### 2.1 Media Contents

The term '*media content*' refers to diverse forms of data used to transmit, process, or store information through computing technologies for various purposes, such as sharing or consuming by users or systems. Media content is typically categorized by its ***type*** and ***format***, which determine the methods required for its processing and presentation. Media content type refers to the way media data is identified and categorized based on its form and purpose. Media content format specifies the technical design of media files, including their structure, encoding, and file extensions. The goal is to ensure seamless access and usability across different systems, applications, devices, and digital environments. Below are the primary types of media content with examples of common formats [52]:

- **Text:** Comprises written information, such as articles, books, and newspapers, presented as characters, symbols, and sentences. It is used to display and communicate textual information through various visual representations. Examples of formats for this media type includes Plain text files, PDFs, or HTML pages.
- **Audio:** Encompasses sound-based contents such as speech or music. Examples of formats for this media type includes MP3, OGG, or WMA files.
- **Images:** Visual representations in static form of many entities and objects, such as pictures, graphics, photos, and illustrations to enrich the viewer's experience. They can be created according to various formats such as JPEG, PNG, GIF, and SVG files [53].
- **Videos:** Visual representations in moving form, such as films and live streams, which may include synchronized audio. They can be created according to various formats including MP4, AVI, MOV, WMV, and FLV files [53].
- **Animations:** Sequences of images that create the illusion of motion through sequential frames or computer-generated graphics. Animations are often used to represent animated sequences or to illustrate processes or concepts visually. They can be generated according to different formats such as animated GIFs, Flash, or 3D animations files.
- **Interactive media:** Refers to media content that allows users interaction or engagement, often through input devices like touchscreens, keyboards, or voice commands. Interactive web pages and virtual reality environments are good examples of such media type.

## 2.2 Multimedia Documents

It is difficult to find a precise and complete definition of the term '*multimedia*'. In general, this word designates the integration of multiple forms of media contents to enable and improve interactions with the user [54], [55]. More specifically, a multimedia document integrates multiple types of media objects – such as text, image, audio and video, animations, and interactive elements – to create a rich, engaging, and dynamic presentation. Each media object has a set of properties that specify its technical features and behavior, categorized as:

- **Generic properties:** Applicable to various media types (e.g., size).
- **Specific properties:** Unique to each media type (e.g., resolution for images, frame rate for videos).

Beyond merely aggregating media objects, multimedia documents organize them through synchronization and linking relationships across four dimensions: *temporal*, *spatial*, *logical*, and *hypermedia*. These relationships are defined during document creation to specify the temporal synchronization, spatial presentation, logical organization and interconnection of media objects [56]. Figure 3.1 represents the general description of a multimedia document.

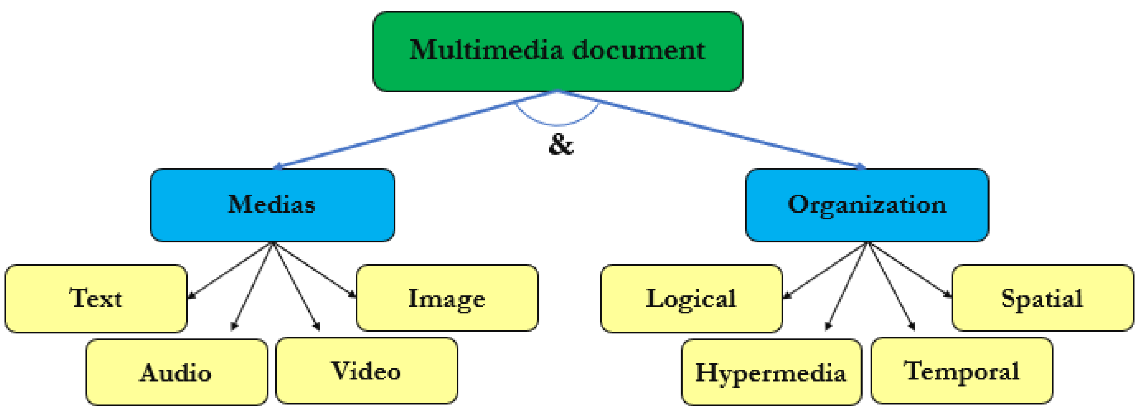



**Figure 3.1.** The general description of a multimedia document [57].

### 2.2.1. Temporal Dimension

The temporal dimension concerns the synchronization of media objects in time. Regardless of their granularity, media objects are temporally linked to define an overall presentation order. This order is often modeled using a temporal scenario, ensuring the different media objects are presented in a coordinated manner by aligning their timing and duration. Temporal synchronization is crucial to maintain the intended narration, rhythm, and timing of multimedia content.

Figure 3.2 presents a timeline as an example of the temporal dimension in a multimedia document. The timeline includes four media objects – Title, Speech, Author, and Demo – with the following temporal organization:

- The *Title* object plays from 0 to 4 seconds.
- The *Speech* object plays from 4 to 18 seconds.
- The *Author* object plays from 4 to 10 seconds.
- The *Demo* object plays from 10 to 20 seconds.

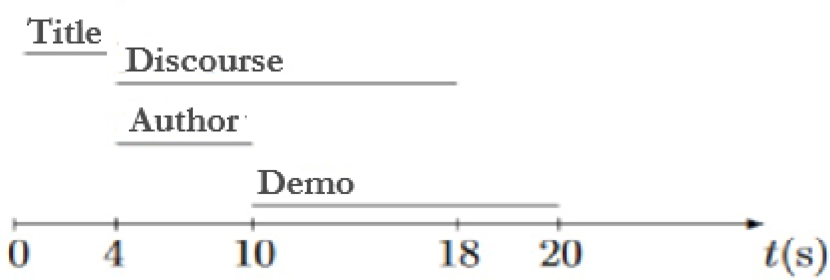



**Figure 3.2.** Temporal dimension of a typical multimedia document [56].

### 2.2.2. Spatial Dimension

The spatial dimension concerns the layout and positioning of media objects within the display space. It determines how the objects are visually arranged, considering factors such as size, aspect ratio, and visual hierarchy. The spatial layout plays a significant role in organizing and presenting multimedia content in an aesthetic and intuitive manner.

On the left side, Figure 3.3 presents the spatial structure of a typical multimedia presentation comprising three media objects: two images and a text. On the right side of this example, the spatial dimension of the main environment is described to show how these three media objects are arranged within the display space [56].


**Figure 3.3.** Spatial dimension of a typical multimedia document [55].

### 2.2.3. Logical Dimension

The logical dimension involves grouping media objects under a shared entity, category, or purpose. For example, multiple images can be grouped to form a photo album, or a video can be segmented into different scenes. This grouping facilitates the organization and structuring of related content, creating logical connections and associations between different media objects. This leads to a hierarchical structure for the entire document.

Figure 3.4 presents a typical logical structure where the root represents the entire multimedia document. This document can be divided into several parts (e.g., Introduction, Chapter) and sub-parts (e.g., Sections). Each part or sub-part can, in turn, contain a finite number of media objects [55].

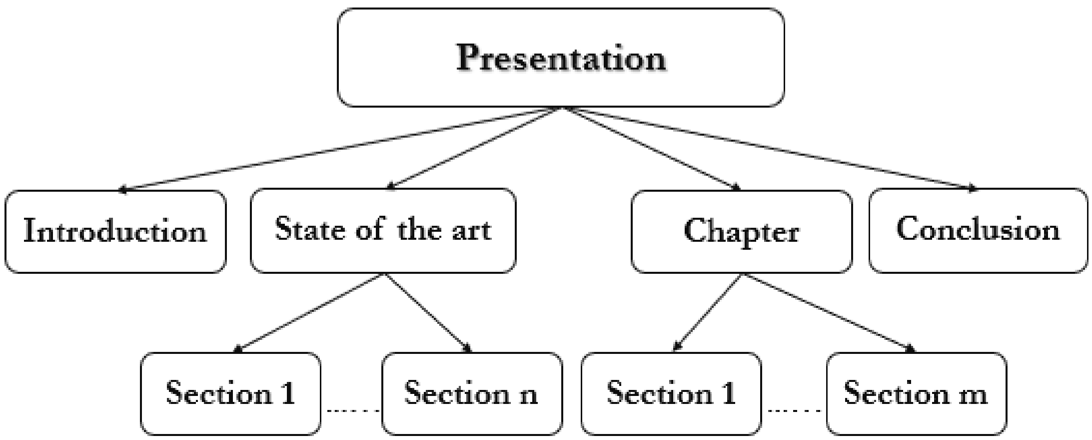



**Figure 3.4.** Logical dimension of a typical multimedia document [55].

### 2.2.4. Hypermedia Relationship

The hypermedia dimension introduces interactive elements into a multimedia document by providing links, buttons, or menus that enable users to navigate through its content. These hypermedia links encode details about the temporal and spatial presentation of media objects, facilitating non-linear exploration and enhancing engagement through options for user-directed browsing and interaction [56].

For example, Figure 3.5 depicts three hypermedia links $l_1$, $l_2$ and $l_3$. The anchors of these hypermedia links are respectively the media objects *Poster*, *Summary* and *Date*. Their targets correspond to *Another presentation* and *the trailer*.

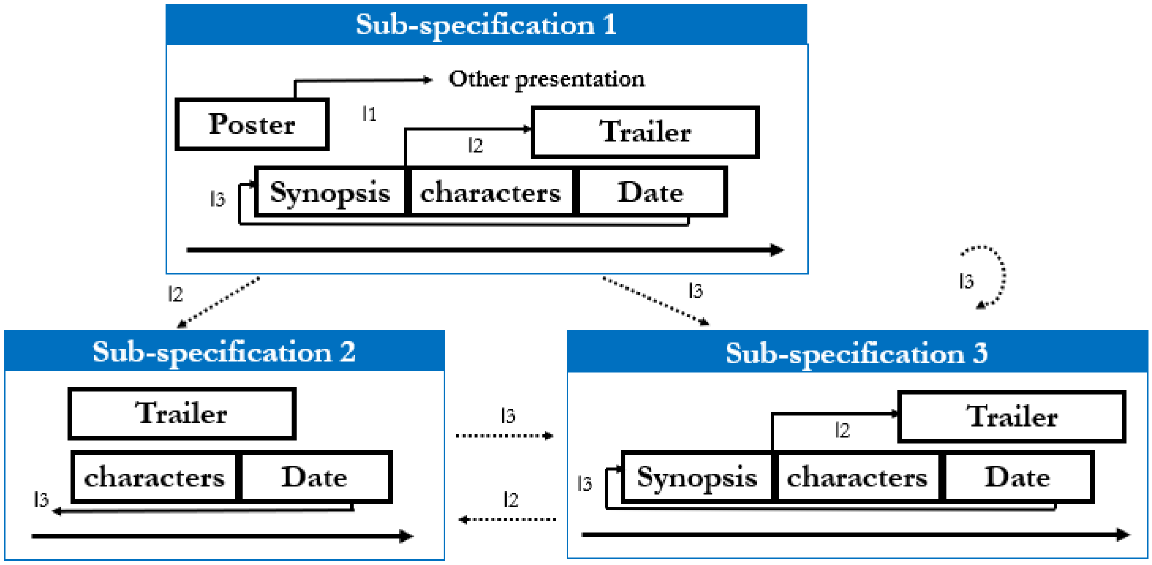



**Figure 3.5.** Hypermedia dimension of a typical multimedia document [56].

## 2.3 Languages And Tools for Creating Multimedia Presentations

Several languages are available for creating multimedia presentations, particularly those based on XML (Extensible Markup Language), each with its own strengths and features [58]. The choice of a multimedia presentation language depends on specific application requirements, such as the type of media objects, the level of interactivity, and the need for dynamic content generation [58]. Below are some of the most well-known XML-based and declarative languages for multimedia presentations:

1. **SMIL (Synchronized Multimedia Integration Language):** SMIL is an XML markup language recommended by W3C for describing multimedia presentations. It supports features such as timing, layout, animations, and visual transitions [59]. It is media-agnostic as it can handle and synchronize diverse media types without prior knowledge of their specifics [60]. SMIL is widely used for creating multimedia presentations and has been extended for dynamic content adaptation based on user profiles or display configurations [61].
2. **NCL (Nested Context Language):** NCL is a declarative XML-based language designed for creating interactive and synchronized multimedia presentations. It emphasizes the strict separation of content and structure, allowing non-invasive control over presentation, linking, and layout. Its nested architecture enables the hierarchical organization of multimedia content, making it well-suited for many systems such as **interactive television (iTV)**, multimedia content delivery, and e-learning platforms [62].
3. **SVG (Scalable Vector Graphics): SVG** is an XML-based format for creating scalable 2D vector graphics. Unlike raster formats (e.g., JPEG or PNG), SVG uses

mathematical definitions for shapes and colors, ensuring scalability without quality loss. It is widely used in multimedia presentations to create interactive graphics, animations, and diagrams, particularly for web-based content. Recent advancements aim to enhance SVG's efficiency and utility [59], [63], [64], [65].

4. **X3D (Extensible 3D Graphics):** X3D is an XML-based standard for 3D graphics and interactive content. It supports real-time interactivity, making it suitable for virtual reality, 3D web content, and interactive simulations. X3D has been applied in fields, including medical simulation [66], interactive training and simulation [67] [68], and the development of conventional GUI libraries [69].
5. **DITA (Darwin Information Typing Architecture):** DITA is an open-standard XML-based architecture for designing, authoring, managing, and delivering technical content. It is mainly used for structured documentation, such as user manuals, help systems, and other forms. DITA separates content from presentation, allowing content reuse across documents and formats. This makes it ideal for technical documentation and e-learning presentations. DITA has been applied in various domains, such as document automation [70], scientific grammar [71], and multilingual parallel texts [70], [72].
6. **XHTML & SMIL:** XHTML is an XML-based version of HTML designed for creating web pages with stricter syntax rules, ensuring well-formed and compatible multimedia presentations across different browsers and devices. Combined with SMIL, it creates complex and interactive multimedia presentations, blending the structure and layout of web elements with synchronized media playback [73] [74], [75].
7. **IT languages and dedicated tools:** With the advancement of programming languages, it is now possible to handle multimedia tasks through dedicated libraries or frameworks, such as *PyGame* and *Tkinter* for Python, and *JavaFX*, *Java3D*, and *AWT*/*Swing* for Java. Combined with CSS and JavaScript, HTML5 is highly versatile for multimedia presentations, with native support for audio, video, and animations, ensuring cross-platform compatibility and extensive browser support. It is also worth mentioning JavaScript libraries like *Three.js* and *PixiJS*, which can handle multimedia presentations, particularly for interactive 3D graphics and animations. Lastly, multimedia presentations can also be created using dedicated tools like *MS PowerPoint* and *Beamer* (a LaTeX-based presentation library).

These languages enhance multimedia presentations by managing synchronization, layout, timing, and transitions for various media types. Table 3.1 outlines their descriptions, strengths, and best use cases.

| Language | Description | Strengths | Best use cases |
|---|---|---|---|
| **SMIL** | A W3C-recommended XML-based language designed for multimedia presentations, allowing synchronization of different media types (audio, video, images, and text). | • Time-based synchronization.<br>• Accessibility features like subtitles and captions.<br>• Lightweight and cross-platform. | • Web-based multimedia presentations.<br>• Interactive tutorials and slideshows. |
| **NCL** | An XML-based language specialized for interactive multimedia applications, particularly in **Digital TV systems** (ISDB-T). | • Advanced interactivity and event-driven multimedia.<br>• Support nested multimedia elements and hierarchical structures.<br>• Integration of distributed content. | • Interactive TV applications.<br>• Complex multimedia projects requiring high interactivity. |
| **SVG** | An XML-based language for describing 2D vector graphics, animations, and text. | • High-quality vector graphics.<br>• Animation and interactivity using **SMIL animations** or **JavaScript**.<br>• Integrates well with web technologies. | • Graphics-intensive multimedia presentations.<br>• Interactive diagrams and charts. |
| **X3D** | An XML-based language for defining interactive 3D graphics and multimedia scenes. | • Rich support for 3D modeling.<br>• Can integrate 2D media with 3D environments.<br>• Interactive capabilities | • Multimedia presentations with 3D components.<br>Virtual reality or augmented reality experiences. |
| **DITA** | An XML-based framework for authoring and publishing structured content, including multimedia elements. | • Content reuse and modularity.<br>• Integration with multimedia for technical documentation. | • Multimedia-embedded technical documents.<br>• Content-driven presentations. |
| **XHTML+ SMIL** | Combines XHTML for text and layout with SMIL for multimedia synchronization. | • Unified framework for web and multimedia content.<br>• Enhanced compatibility with browsers. | • Web-based multimedia content blending text and synchronized media. |
| **Other tools** | Programming languages and dedicated tools | • Dedicated libraries<br>• Generating tools | • Various multimedia processing |

**Table 3.1.** Comparison of multimedia presentations languages and tools.

## 3. Multimedia Documents Adaptation in Context-Aware Pervasive Systems

Multimedia document adaptation (MDA) refers to dynamically modifying multimedia content based on context changes in pervasive systems. The goal is to deliver suitable content that conforms to the current contextual situations and constraints, such as device features, surrounding environment, and user preferences (see Figure 3.6). Playing multimedia documents in pervasive systems, where the context continuously changes, may lead to conflicts when document requirements contradict contextual constraints. For instance, let consider playing video contents in public places (e.g., buses, or libraries). In this case, auditory contents should not be played there. Consequently, it is possible to adapt the content by muting audio and subtitling videos. This ensures that the content remains accessible while respecting the constraints of the surrounding environment [14].

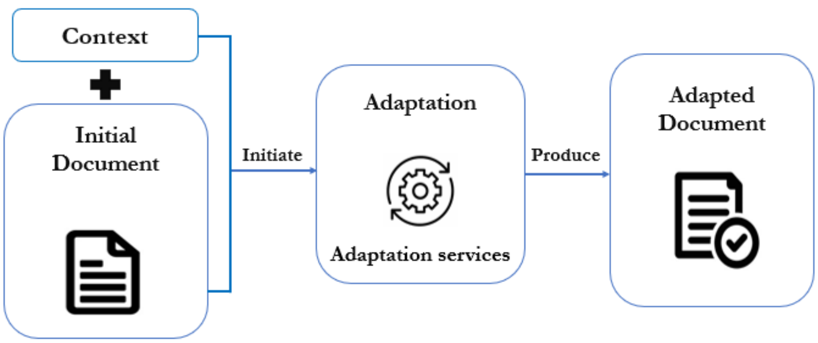



**Figure 3.6.** Multimedia documents adaptation in context-aware pervasive systems.

### 3.1 Levels of Multimedia Documents Adaptation

Adaptation operations can be categorized into two levels: *local adaptation* and *global adaptation*.

- **Local adaptation:** Local adaptation applies to individual media objects separately by performing physical transformations. It includes three main operations: *transmoding*, *transcoding*, and *transformation* (see Figure 3.7) [76], [77].

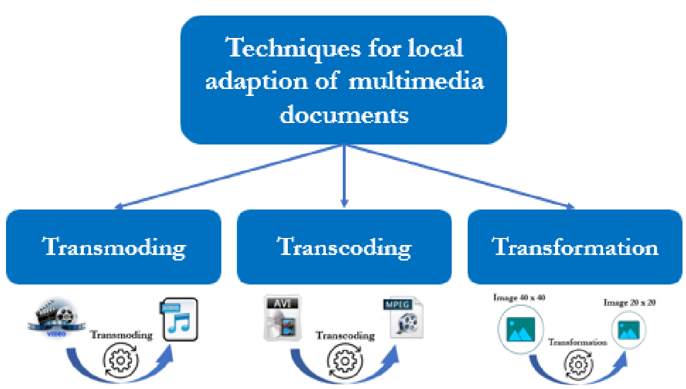



**Figure 3.7.** Techniques for local adaption of multimedia documents.

– **Transmoding:** Transmoding alters the type of a media object by changing its modality. An example of such an operation is the conversion of a video into an audio format as this may be useful in scenarios where visual content cannot be displayed (see Figure 3.8) [78].

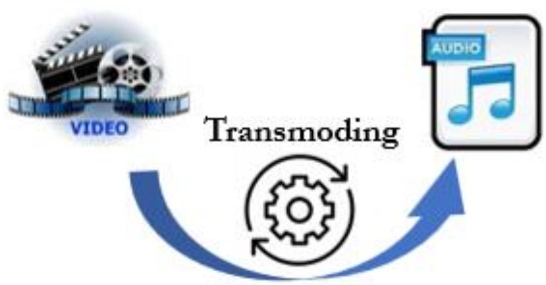



**Figure 3.8.** Transmoding operation of a video into an audio.

– **Transcoding:** Transcoding changes the format of a media object while preserving its modality. It also attempts to ensure the compatibility across various devices and platforms. An example is converting a video from AVI to MPEG format (see Figure 3.9) [78].

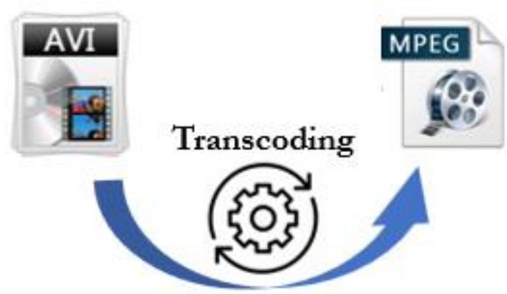



**Figure 3.9.** Transcoding operation of a video from AVI to MPEG format.

– **Transformation:** Transformation modifies the content of a media object while maintaining its type and format. An example is resizing an image while retaining its original format, as shown in Figure 3.10 [78].

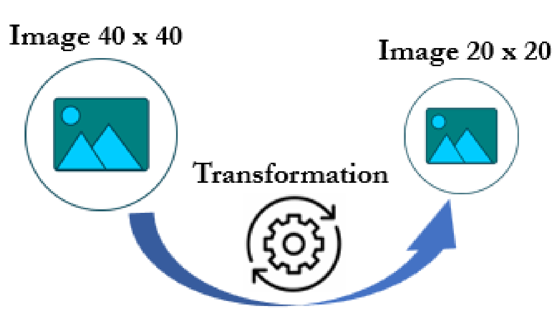



**Figure 3.10.** Transformation of an image from 40×40 pixels to 20×20 pixels.

- **Global adaptation:** Global adaptation affects the organization of media objects within the documents. It modifies the temporal, spatial, or hypermedia dimensions of the media objects. The aim is to maintain the semantic integrity of the adapted document while accommodating the new context (see Figure 3.11) [7, 8]. Examples of such an operation include:
  - Temporal adaptation that adjusts the timing and synchronization of media objects.
  - Spatial adaptation that rearranges the layout of media objects to fit different display sizes or orientations.
  - Hypermedia adaptation that modifies navigation links or relationships between objects.

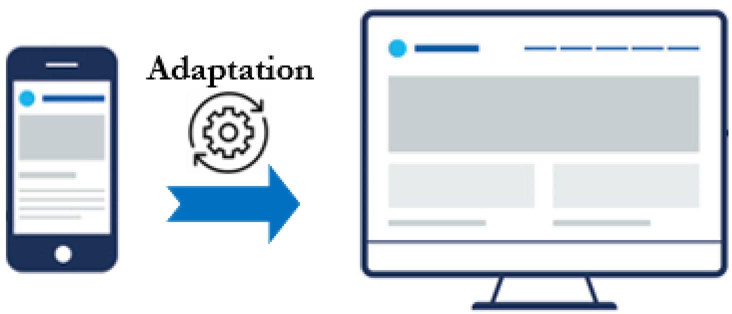



**Figure 3.11.** Example of a global adaptation in a multimedia document.

## 3.2 Adaptation Services

Performing adaptation actions often requires the composition of multiple adaptation services. Each adaptation service is defined by the following attributes [77]:

- **Service ID:** A unique identifier for the service.
- **Input parameters:** Characteristics of the media object content to be processed by the service.
- **Output parameters:** Characteristics of the adapted media object generated by the service.
- **Action property:** Defines specific properties of the document being processed by the service (e.g., file size, format).
- **Action type:** Specifies the type of action performed namely transformation, transmoding, and transcoding.
- **Quality parameters:** Represent non-functional properties of the service, such as price, reliability, and availability.

Adaptation services can be implemented using various technologies, including web services, cloud platforms, and APIs [79]. The organization of these services is often structured as a class hierarchy, enabling efficient management and reuse [78] (see Figure 3.12).

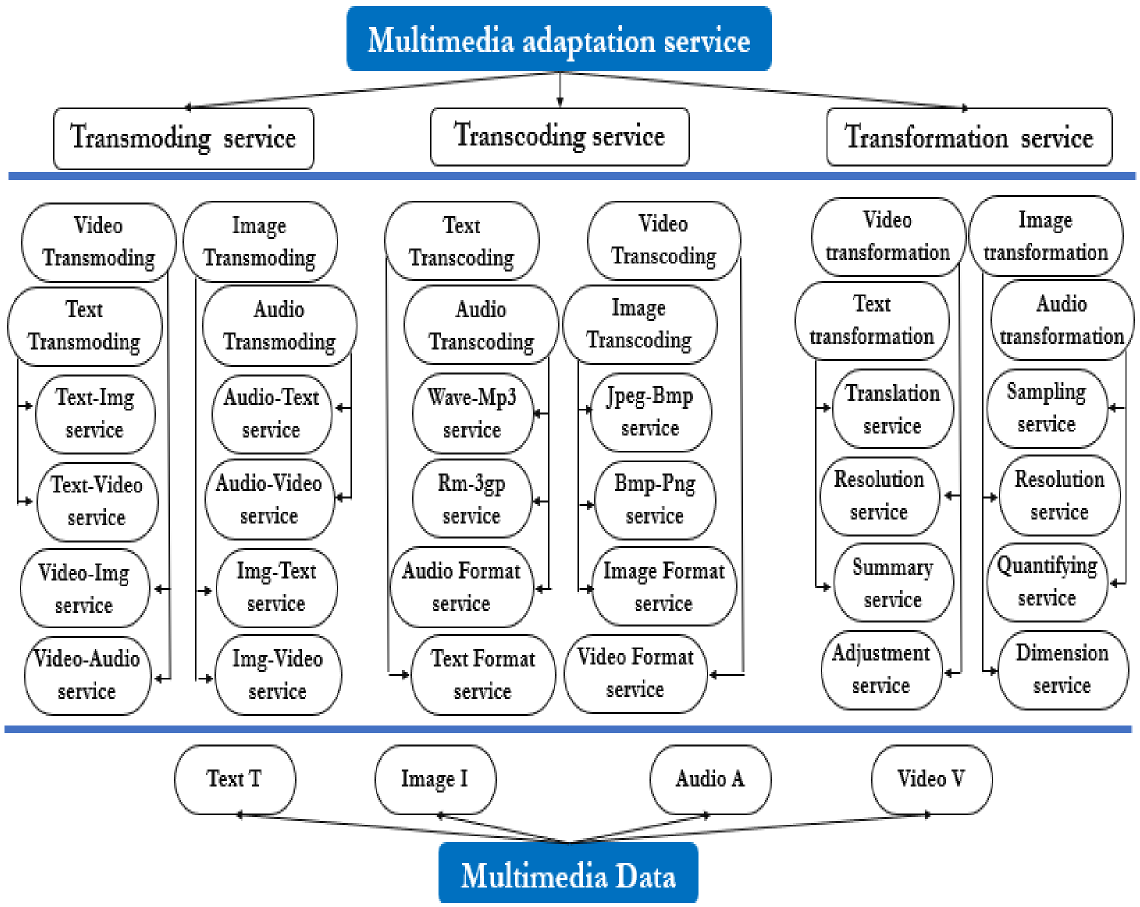



Figure 3. 12. A comprehensive hierarchical description of adaptation services [79].

## 3.3 Quality of Service in Multimedia Documents Adaptation

In MDA processes, adaptation services are first described as abstract tasks within a workflow model. Adaptation plans (or paths) are then generated, where each abstract service is bound to a concrete service selected from a repository of candidates (see Figure 3.13).

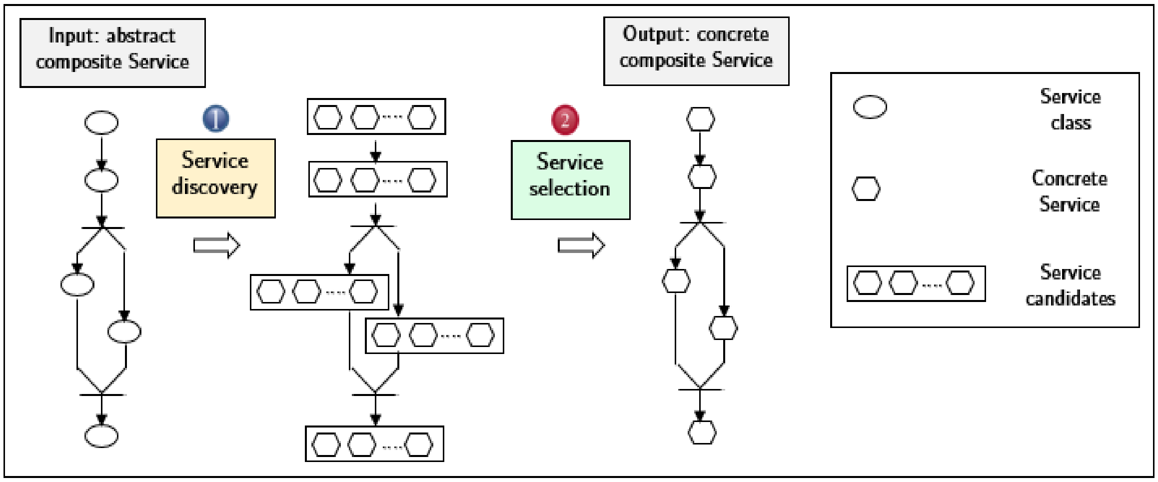



**Figure 3.13.** Adaptation services discovery and selection.

As discussed in [80], the quality of service in any given MDA process is defined by evaluating the adaptation services based on two types of quality parameters: *cost* and *benefit.*

- **Cost parameters:** Characteristics of the adaptation services, such as price, response time, reliability, etc.
- **Benefit parameters:** Characteristics of media objects, such as image resolution, video speed, etc.

These two types of parameters are used to select the best adaptation path based on specific contextual parameters, such as user preferences (e.g. resolution, compression rate, response time, budget, etc.) and available hardware resources (e.g. battery level, available memory space, bandwidth, etc.) [80]. For example, a user may prioritize high-resolution video playback, while the system considers available bandwidth and battery constraints to select a compatible service.

## 4. Context-Aware Multimedia Documents Adaptation Processes: State of the Art

Several adaptation approaches have been proposed to provide documents adapted to contextual situations and constraints in pervasive systems [81], [82], [83], [84], [85], [86]. These approaches differ in how the adaptation process is conducted and the document models they target.

### 4.1 The General Scheme of Context-Aware Multimedia Document Adaptation Processes

As discussed in Section 5 of Chapter 2, context-aware pervasive systems are built around three primary layers: *sensing*, *reasoning*, and *acting*. These layers are detailed below in the case of MDA processes, as shown in Figure 3.14.

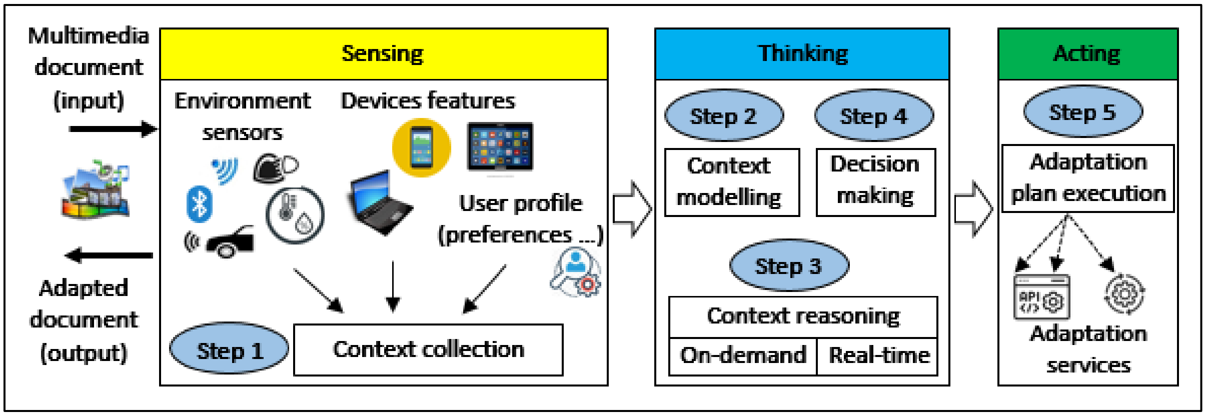



**Figure 3.14.** The general architecture of a standard MDA process

– ***Sensing layer:*** this layer includes a single step: *context collection*. This step updates the context elements by acquiring data from the physical world using physical, logical, or virtual sensors. The context includes different types of information such ***physical context*** (e.g., lighting, noise level, etc.), ***user-related context*** (e.g., profiles, locations, etc.), ***computational context*** (e.g., network connectivity, communication bandwidth, etc.) and ***temporal context*** (e.g., day, time of day, etc.)[87].

– ***Thinking layer:*** this layer comprises three steps: *context modeling*, *context reasoning*, and *decision-making*.

- ***Context modeling:*** Raw context data, collected from various sources (low-level context data), is organized using abstract context models (e.g., key-value models, ontology-based models). This allows processing context information at a higher level, enabling reasoning and decision-making [88]. Generally, these data are obtained either by events triggered by context changes or by time driven methods according to which the context values are calculated during periodic checks.

- ***Context reasoning:*** High-level context data is analyzed to identify constraints or conflicts that may hinder playing the documents properly. Reasoning can occur in two forms:

    - **Real-time reasoning:** Continuously executed, responding to context changes as they occur either as an event driven or a time driven context.

    - **On-demand reasoning:** Triggered by specific user or application requests, such as resource discovery or environmental recommendations.

  In the end of this step, a set of adaptation actions are inferred to provide adapted documents, often expressed as "*if* ... *then*" rules (e.g., *If* (the user is in a public place), *then* mute the audio and enable subtitles).

- ***Decision-making:*** The objective of this step is twofold: the determination of the adaptation actions to execute and the generation of the adaptation plan. Three modes are possible:

  a) **Automatic mode:** Adaptation actions are determined without user input.

  b) **Semi-automatic mode:** The system suggests adaptation actions while the user selects the most suitable ones.

  c) **Manual mode:** The user defines all adaptation actions based on specific needs.

  The adaptation plan is generated by binding each action to possible execution paths, considering available services and QoS parameters (e.g., response time, availability) to select the best path.

– ***Acting layer:*** This layer executes the adaptation plan using selected adaptation services. These services are applied to individual media objects to produce adapted media content (local adaptation). The adapted content is then organized into a document suitable for the current context (global adaptation).

- **Explanatory example**

Consider a mobile user consulting online courses. For the sake of simplicity, we assume that the context elements include the location, current activity and preferred language of the user. Table 2.2 summarizes typical context values, conflicts, and adaptation actions:

| Context element | Value | Conflict | Adaptation action |
|---|---|---|---|
| Location | Bus | Auditory contents should not be played | Exclude audio |
| Language | Spanish | The content cannot be understood | Language translation |
| User activity | Driving | Hands and eyes cannot be used properly | Voice-commands |

**Table 3.2.** Some typical context elements, conflicts and adaptation actions.

## 4.2 Classification of Multimedia Documents Adaptation Processes

Depending on where decision-making and adaptation actions take place, MDA processes can be divided into four broad categories [45]. The selection of a specific approach depends on factors such as device capabilities, communication bandwidth, and available resources.

- ***Client-side adaptation:*** In this approach, client devices running the documents handle the entire adaptation process themselves (e.g., [57]). This implies that the devices should have the resources and capabilities to adapt the documents according to their contextual situations and constraints.
    - **Advantages:** Provides responsiveness, allowing real-time adaptation decisions without relying on remote systems.
    - **Disadvantages:** Limited by the device's hardware and energy resources (e.g., battery, CPU, and RAM), making it unsuitable for resource-constrained devices.
- ***Server-side adaptation:*** In this approach, client devices send adaptation requests to a dedicated server, which performs the entire adaptation process (e.g., [9], [10]).
    - **Advantages:** Offloads computational tasks from clients, improving their performance and efficiency.
    - **Disadvantages:** May overload the server, leading to delays, especially when handling large volumes of adaptation requests. Thus, real-time and instantaneous responsiveness can be affected.
- ***Proxy-based adaptation:*** Here, a proxy acts as an intermediary between the client and server (e.g., [89]). The proxy analyzes client requests, negotiates with the server to determine appropriate adaptation actions, and performs some adaptations itself before delivering the results to the client.
    - **Advantages:** Reduces the workload on both clients and the server by leveraging the proxy's computational resources.
    - **Disadvantages:** Increases communication overhead due to additional coordination between the client, proxy, and server, which may impact latency.
- ***Peer-to-peer adaptation:*** In this approach, devices playing the multimedia documents form a peer-to-peer network, enabling direct communication and collaboration (e.g., [90]). Thus, devices can share and execute adaptation services dynamically.
    - **Advantages:** Offers scalability and flexibility by distributing the workload among devices. Each device can utilize its own or others' adaptation services, expanding the pool of available resources.
    - **Disadvantages:** Managing the peer-to-peer network and ensuring service availability can be challenging, as they depend on the reliability and number of connected nodes.

Table 3.3 provides a summary of the approaches described above.

| Approach | Category | Overview |
|---|---|---|
| [89] | Proxy-based | A proxy-based adaptation approach, using a multi-agent system to optimize multimedia content delivery to handheld devices. |
| [91] | Server-side | An innovative, knowledge-driven paradigm for developing next-generation multimedia adaptation servers, capable of generating intricate adaptation chains from semantically annotated transformation operations. |
| [92] | Proxy-based | This approach utilizes adaptation agents that implement sequential decision-making policies in uncertain environments. Based on the degree of observability of the context, whether fully or partially observable. |
| [93] | Peer-to-peer | A service-oriented architecture for the adaptation of multimedia documents that introduces a negotiation protocol to assist the adaptation manager in selecting the optimal adaptor. |
| [94] | Server-side | A probable architecture entails server-side processing for the rendering of VR content and the adaptation of multimedia materials to cater to the distinct requirements of students or learners. |
| [95] | Server-side | A framework named MMSA (Metamodel Multimedia Software Architecture) which employs a hierarchical media structure to model the adaptation process, considering the organization of media and flows. |
| [96] | Server-side | A server-side approach which present a service-oriented framework for media streaming and digital item adaptation. |
| [90] | Peer-to-peer | A service-based approach for MDA in which adaptation services are available locally and remotely on multiple platforms. |
| [97] | Proxy-based | A proxy-based approach for automatic generation of multimedia presentations based on SMIL (Synchronized Multimedia Integration Language) standard. |
| [98] | Client-side | A client-side approach presents a localization architecture for an m-tourism services delivery platform. It allows clients to reach various multimedia descriptions according to their profiles. The aim of this approach is to deliver services for nomads (e-tourists) according to their localization and according to the results given by the search engine. This engine is based on a quantitative similarity measure. |

| | | |
|---|---|---|
| [99] | Proxy-based | A service-based context modeling approach for MDA that links the context information using semantic generic profiles. |
| [100] | Server-side | This approach focuses on the integration of media with virtual 3D environments in real-time. This suggests that media content is likely processed and rendered on the server before being delivered to clients. |
| [101] | Server-side | This study presents a method for the dynamic adaptation of multimedia documents, structured as an over-constrained constraint satisfaction problem (OCSP), to guarantee a particular level of service for the presentation of the adapted document. |
| [102] | Server-side | This approach focuses on adapting class-level and decision-level multimodal fusion under the assumption that inputs to the fusion models (called cues) are provided by the same algorithms in all contexts. |
| [103] | Server-side | An approach based on the use of recommendations and social networks for real-time adaptation of multimedia content using semantic representation approaches to define contextual constraints and community interests. |
| [9] | Server-side | An MDA approach based on formal concepts related to technical adaptations to select relevant policies by detecting the conflicts between documents properties and users preferences. |
| [104] | Server-side | An MDA approach that involves social impact inferred from communities on Twitter and Facebook, to make an effective composition of adaptation services using semantic relationships. |
| [10] | Server-side | A handicap-based MDA that exploits an ontology to provide context-aware assistance by binding each context constraint to a physical handicap and thus to an adaptation action. |
| [105] | Server-side | A context-aware MDA approach based on cooperative multi-agents systems that uses semantic web services to predict users context and provide adapted contents accordingly. |
| [106] | Server-side | An XML-based personalization of course content in ubiquitous learning, considering learning styles and context-awareness |
| [57] | Client-side | An XML-based multimedia specification that describes the structure of web contents according to users context |

| | | |
|---|---|---|
| | | and profile. The content is adapted by modifying their spatial structures. |
| [107] | Server-side | MDA approach that focuses on content classification, which based on classification metrics to define its temporal and spatial complexity, those metrics helps to decide the type of adaptation that should be performed on the content to cope with existent restrictions imposed by the distribution chain and the consumption context. |
| [108] | Sever-side | A framework that allows the build of learning systems for various learning contexts. It offers superior application management and a personalized learning strategy suited to the learner's context. |
| [109] | Server-side | A multimedia adaptation approach that integrates multi-agent systems with semantic web services to determine the user's current context. |
| [110] | Server-side | A server-side approach that represent an E-Learning framework based on a mathematical model for learner categorization using machine learning techniques (a hybrid of case based reasoning and neural networks). |
| [11] | Server-side | A graph-based MDA that allows the inference of multiple adaptation actions using semantic reasoning upon users context. |
| [111] | Server-side | This approach proposes a dynamic clustering and filtering mechanism for situation rules that considers the user's domain, role, current location, and time during situation processing and reasoning. It also involves mechanisms for simultaneous event detection and parallel situation identification. |
| [112] | Server-side | This approach focuses on summarization, which is a process of extracting the most representative words from a bucket of multimedia documents. |
| [113] | Server-side | A server-side approach that proposes a five-module architecture to generate both textual and visual summaries from multimedia sources. The modules include video temporal segmentation, visual summarization, textual segmentation, textual summarization, and multimodal summarization by cross-domain alignment. |
| [114] | Server-side | This approach is a segment-level cross-domain semantics alignment model for the task of multimodal multimedia summarization with multimodal output. It consists of |

| | | |
|---|---|---|
| | | five modules: video temporal segmentation, visual summarization, textual segmentation, textual summarization, and cross-domain alignment. |
| [115] | Server-side | Server-side approach that combines the integration of multimedia documents in a 3D urban scene with user guidance modes to allow the user to effectively consume and understand the multimedia content. |
| [116] | Server-side | A novel multi-viewpoint ontological approach for adapting multimedia documents; each viewpoint of the multimedia document may correspond to a physical disability, thereby starting an adaptation action through multi-viewpoint ontological reasoning. |

Table 3.3. A summary of some MDA approaches

## 5. Case Study: MDA for Learning in Context-Aware Pervasive Systems

This section introduces the concept of *ubiquitous learning*, its fundamental components, and the design principles for multimedia adaptation systems adapted to learners' contexts in a pervasive environment.

### 5.1 Overview of Ubiquitous Learning

Ubiquitous learning (u-learning) refers to an adaptive learning environment supported by computers, mobile devices, and connected objects, combined with wireless communication to provide learning materials anytime, anywhere, and anyway. As a context-aware system, u-learning can detect learners' situations and provide personalized guidance and support. For example, if a learner watches an educational video on a noisy bus, the system may reduce sound and enable subtitles.

A typical u-learning environment comprises the following elements:

- **Sensors:** Collect contextual data about learners (e.g., location) and their environment (e.g., temperature, noise levels).
- **Servers:** Store and provide active and passive support, including adaptation decisions.
- **Mobile devices:** Enable learners to access learning materials through the internet.
- **Wireless networks:** Facilitate communication between devices, sensors, and servers.

Key criteria for an effective context-aware u-learning environment include ability to:

- Sense learners' situations and enabling real-world learning activities.

- Adapt to learners' behaviors and contexts in both virtual and real-world environments.
- Provide personalized guidance based on learners' contexts, profiles, and histories.
- Ensure seamless learning experiences across locations within a predefined area.
- Adapt content for various mobile device functionalities.

## 5.2 Adapting Multimedia Content for Ubiquitous Learning

Technological advancements have revolutionized distance learning, including u-learning, which revolves around two core elements: *learning objects* and *learning context*. Figure 3.15 illustrates the general architecture of a u-learning system.

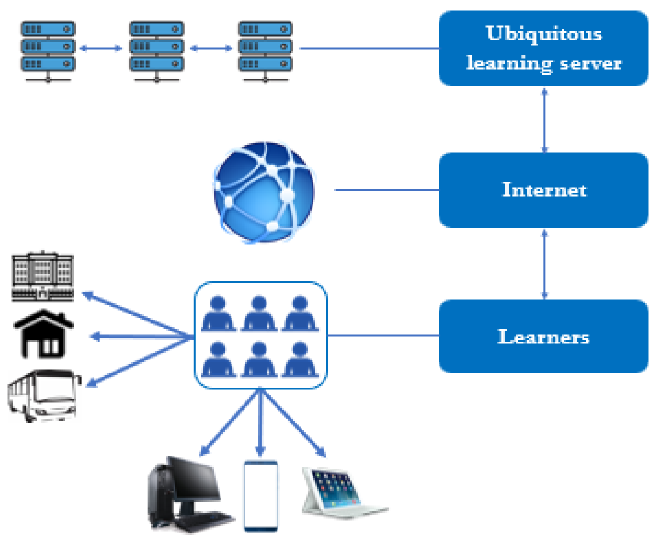



**Figure 3.15.** General architecture of ubiquitous learning systems.

### 5.2.1 Learning Objects

Learning objects are entities used for learning, education or training purposes, often reusable to achieve specific instructional objectives. These include various multimedia content such as video, text, audio, images, slides, animations, and simulations. According to [106], effective learning objects should be:

- **Modular:** Presented in small and self-contained units.
- **Reusable:** Suitable for multiple learning media and contexts.
- **Aggregated:** Grouped into larger content collections.
- **Searchable:** Tagged with metadata for easy retrieval.

Popular models for representing learning objects employ XML and HTML, which ensure flexible data storage and interoperability.

## 5.3 Learning Context

In ubiquitous learning systems, learners are different from those using traditional e-learning systems. This is because contextual information is incorporated throughout the learning process to enhance accessibility to relevant learning objects. The learning context describes the learner's situation during an activity [106]. It includes two types of information:

- **Instructional information:** Includes learning style models used to personalize learning paths.
- **Non-instructional information:** Covers environmental and personal attributes such as location, device type, and surrounding conditions.

By combining instructional and non-instructional information, u-learning systems support learning activities through the adaption of learning materials presentation to both learners' preferences and environmental requirements, ensuring a personalized experience.

Learning style models are employed to personalize learning paths by identifying how learners process information during their activities, based on their preferences and behaviors. These features vary from learner to another and may also evolve over, experience, or learning objectives. One widely recognized model is the *Felder-Silverman model*, which classifies learners into four dimensions of learning preferences, as shown in Table 3.4 [106].

| **Continuum** | **Preference** | **Learner style** |
|---|---|---|
| Sensing-intuitive | How the learner takes in information | - *Sensing:* the learner likes better concrete and practical thinking, and is interested in facts and procedures<br>- *Intuitive:* the learner likes better conceptual thinking, and is interested in theories, definitions, and meanings |
| Visual-verbal | How information is presented to the learner | - *Visual:* the learner likes better visual representations, figures, diagrams, and flowcharts<br>- *Verbal:* the learner likes better written and oral explanations |
| Active-reflective | How the learner processes information | - *Active:* the learner likes better to try things and to experiment with new concepts<br>- *Reflective:* the learner likes better to learn individually and to think before trying. |
| Sequential-global | How the learner organizes himself and progresses to understand the information | - *Sequential:* the learner likes better linear thinking and orderly learns in small incremental steps<br>- *Global:* the learner likes better holistic thinking and learns in big leaps |

**Table 3.4.** Classification of learners according to the Felder-Silverman learning model [106].

### 5.3.1. Adaptation of learning materials

In ubiquitous learning systems, learning objects need to be adapted at any time and in anywhere, depending on the learning context (see Figure 3.16). A context-aware content adaptation system revolves around several parameters that help define a learner's context, in order to provide personalized learning materials accordingly. Two types of context-aware adaptation in u-learning systems are distinguished: *the adaptation based on instructional context information* and *the adaptation based on non-instructional context information* [106].

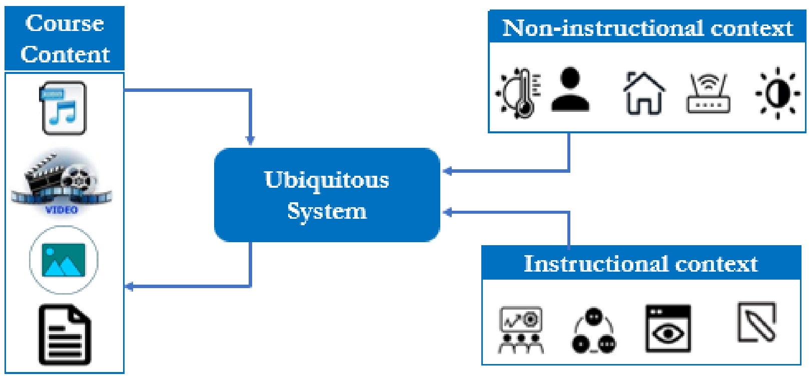



**Figure 3.16.** Learning adaptation based instructional and non-instructional context.

- **Instructional context-based adaptation**

Adaptation based on instructional context information focuses on generating learning material parts with a well-defined sequencing (path) adapted to the learner's style specified in the profile. Thus, this adaptation aims to provide learning objects with contents that meet the instructional goals set without worrying about the environment requirements. For instance, consider two learners A and B. Learner A has a verbal cognitive style while learner B has a visual cognitive style. Figure 3.17 shows how the content is adapted for these two learners.

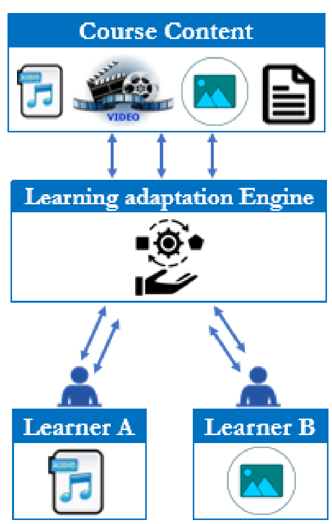



Figure 3.17. Adapted contents for learners A and B.

- **Non-instructional context-based adaptation**

Adaptation based on non-instructional context information seeks to present personalized learning material paths that accommodate the learner's current contextual situation, including environmental factors. As a learner's context can dynamically change over time, certain constraints may arise, hindering the proper delivery of learning materials. These constraints occur when the features of the learning objects conflict with the current learning context. In such cases, the adaptation operates in two phases as described in subsection 6.1:

- **Local adaptation:** Applied to the media objects within the learning materials.
- **Global adaptation:** Generates personalized learning material paths.

For example, consider Learner A, who adopts a verbal cognitive style. If this learner is in a laboratory where auditory content is unsuitable, the system suggests replacing audio content with text, muting video content, and providing subtitles to ensure compatibility with the environment.

## 6. Conclusion

In this chapter, we examined the adaptation of multimedia documents in context-aware pervasive systems, emphasizing its role in enhancing user experiences and ensuring seamless content delivery across diverse devices and contexts. By addressing the challenges posed by dynamic context, multimedia document adaptation emerges as an effective solution for maintaining usability and accessibility.

We began by outlining the foundational concepts of multimedia document adaptation, establishing a solid theoretical framework. This was followed by a comprehensive literature review that highlighted existing approaches, helping to identify gaps that still exist in the field. Finally, through a case study on multimedia content adaptation in context-aware pervasive learning environments, we demonstrated the potential of these systems to dynamically adapt content to users' needs, thereby improving engagement and usability.

Despite the notable results already achieved, context-aware multimedia adaptation still offers opportunities for growth, particularly in expanding the adaptability of systems to handle more complex and domain-specific contexts. This will ensure the delivery of highly personalized, efficient, and responsive interactions. To this end, exploring the use of historical user data and integrating advanced techniques such as machine learning could further refine adaptation processes. These enhancements aim to make context-aware multimedia adaptation systems more robust, scalable, and impactful in real-world applications, a topic that will be explored in the next chapter.

# Chapter IV

## Management of historical data in context-aware MDA processes

Summary

# Chapter IV

## Management of historical data in context-aware MDA processes

### 1. Introduction

In multimedia adaptation processes (MDA) processes, systems detect the user's context, reasons on it, and executes actions to deliver an adapted document. While existing adaptation approaches have achieved notable results, they primarily rely on real-time data collected from various sensing sources to adapt documents in response to context changes. Thus, these approaches often overlook the value of historical user data accumulated over time, which could significantly enhance the adaptation process. Historical data provides opportunities to refine system behavior through advanced tasks, such as applying machine learning techniques for classification or prediction and learning adaptation rules.

This chapter addresses this gap by proposing a software component designed to manage historical user data generated during MDA processes for further processing. This stored data includes both context values and their corresponding adaptation actions, enabling more personalized adaptation rules. Depending on the adaptation process category, we introduce three data management variants – client-side management, proxy management, and server-side management – as well as their hybridization to ensure compatibility with a wide range of adaptation approaches.

To achieve this, we first model the context and the required adaptation actions using an object-oriented approach. We then translate this modeling into relational and NoSQL schemas. Additionally, algorithms for data storage, retrieval, and analysis are developed to support the proposed framework. The effectiveness of our approach is validated through a series of test scenarios, building upon the prototype implementation from the previous chapter. Finally, we demonstrate the practical benefits of our proposal by exploring use cases involving the generation of context-aware adaptation rules.

## 2. Involving Historical Users Data in Context-Aware MDA Processes

A review of several MDA approaches [2-21] reveals that most of them focus primarily on context collection, representation, and interpretation. The reasoning mechanisms generally rely on user profiles and current context values, performing adaptation tasks at a single location (e.g., client, proxy, or server). Therefore, these approaches typically lack mechanisms to store and analyze historical context values and corresponding adaptation actions, which limits their ability to incorporate prior knowledge into the adaptation process. Many context-aware pervasive applications log historical data to provide insights into user decisions, enabling future improvements. We cite as examples machine-learning techniques for IoT cultural data [119], context-aware intrusion detection [120], smartphone data analytics [24-27], activity recognition [3], and healthcare support [4].

Similarly, HUD can contribute in improving the overall behaviour of MDA processes by performing several advantageous tasks such as machine-learning techniques, statistical methods for building recommendation systems, adaptation rules personalization, etc. In this case, two types of HUD can be considered either jointly or separately:

- The first type of HUD concerns the multimedia documents consulted by emphasizing on general information about users and resources accessed.
- The second type of HUD is related to specific features of ubiquitous systems (e.g., sensors, human-computer interactions, etc.) insofar as the adaptation process should deal with context values and adaptation actions.

Regarding the first type of HUD, several researches in other fields have already been conducted in this respect; web log mining is a good example [5]. In contrast, there is still a general dearth of MDA approaches treating the second type of HUD. Indeed, even though there are some MDA-based applications that involves certain HUD (e.g. [124], [125]), they however do not provide effective mechanisms to manage such data since they only focus on analyzing them either through datasets or by certain collected data. Generally speaking, log data are extremely diverse and processing them is unfortunately quite complex. This is due to the lack of standards and agreements outlining the granularity, structure, content, format, and level of details provided by log events [6]. This raises certain issues related to their management:

- The first concern is about the determination of which data will be stored; this is a crucial question that depends on the intended purposes of their storage.
- The second concern is about the format in which the data will be stored. This aspect has a direct link with efficient data storage and retrieval for further processing.
- The third concern is about the place where the data will be stored. This is a very important aspect since it may lead to dealing with large volumes of data.

- The last concern is about the way in which the data stored will be leveraged in tasks integrating users' experiences into MDA processes, to improve their effectiveness.

Based on this perspective, we propose to design a software component that handles HUD resulting from the execution of MDA processes for further processing. Compared to other MDA-based approaches, our proposal should be generic insofar as it can be integrated into different MDA processes categories in such a way that *it allows the storage, retrieval and analysis of historical data as log data resulting from the accumulation of the decisions made by users*. This way, the proposed component will provide MDA processes with more options to efficiently process HUD in different locations namely clients, proxy and servers sides. To do so, we rely on a well-devised, flexible and agile architecture that can easily change form (polymorphism) and adapt to context changes (e.g. computation resources) and users preferences / requirements (e.g. personalized processes, data privacy and sharing).

## 3. The General Architecture of the Proposed HUD Manager

Our proposal consists of a software component for managing HUD generated by MDA processes. The goal is to facilitate the storage, retrieval and analysis of HUD to involve them in users' decision-making. This HUD manager complements existing adaptation approaches by addressing some of their limitations as:

- Most MDA approaches rely solely on on-demand and/or real-time reasoning, without incorporating HUD. Our proposal enables MDA processes to leverage HUD for various tasks, such as machine learning, thereby enhancing their quality.
- Adaptation tasks are often executed in a single location (client, proxy, or server), in particular the client-server paradigm which is the most common due to simplicity and computational resource availability. However, other paradigms can be effective for challenges like workload balancing. Even so, the use of the other paradigms can also be effective for difficulties like workload balancing. Indeed, by relying on an effective hybrid approach, MDA processes can be executed in a decentralized scheme by dynamically switching between adaptation categories as needed.

In this regard, the proposed architecture is designed to be flexible, agile, and adaptable, offering the following benefits:

a) Compatibility with all MDA categories.

b) Customization based on computational resources and user preferences (e.g., data privacy and sharing).

c) Versatility to manage HUD across various devices (e.g., laptops, servers).

Figure 4.1 shows the architecture of of the proposed HUD manager and its potential interactions with MDA processes. The architecture consists of four main components.

1) ***Database:*** Serves as the storage medium for HUD.
2) ***Data logger:*** Records HUD systematically in the database whenever new data is acquired by the system.
3) ***Data retriever:*** Retrieves HUD from the database for use by the data analyzer.
4) ***Data analyzer:*** Comprises a set of data analysis algorithms for various tasks such as machine learning, data mining, and statistical analysis.

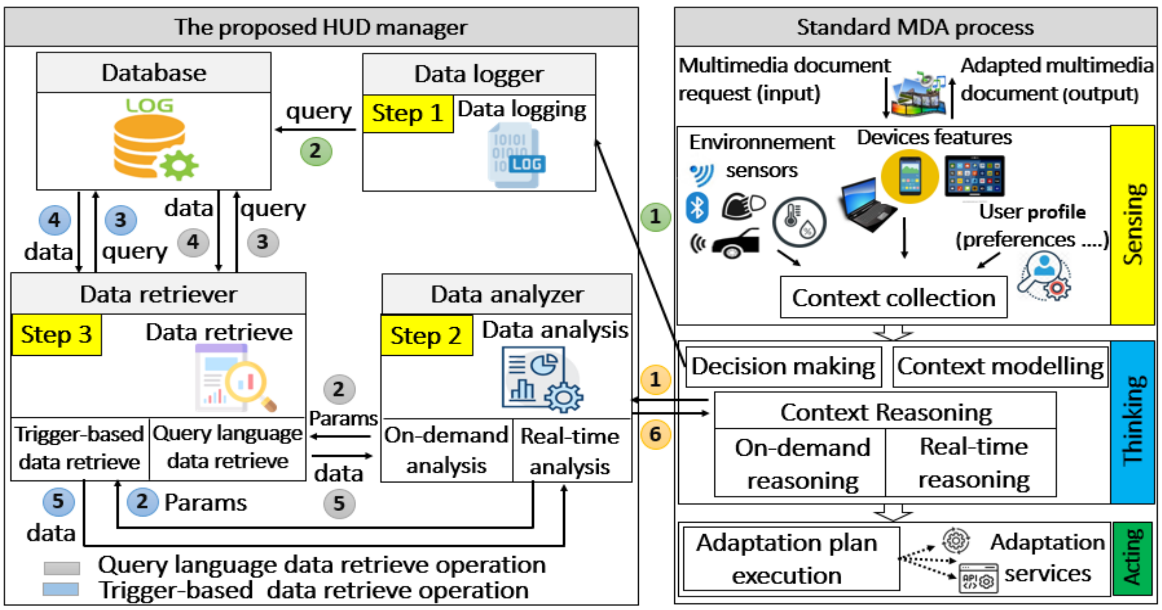



**Figure 4.1.** Detailed architecture of the proposed component.

The proposed HUD manager interacts with MDA processes via the *Thinking layer*, as illustrated in Figure 4.1. This layer includes context-reasoning and decision-making modules, which forms the core intelligence of the adaptation system.

- On the one hand, the decision-making module invokes the data logger to store HUD, mainly context values and users decisions (Step 1 in Figure 4.1).
- On the other hand, the context-reasoning module utilizes the data analyser's functionalities to support future decision-making (Step 2 in Figure 3.1).

Next, we summarize the three steps executed by the proposed HUD manager.

– ***Step 1: HUD Logging***

After generating an adaptation plan, the decision-making module calls the data logger to store information about the context values and adaptation actions. The data logger uses a query language that depends on the implementation (e.g., SQL, XPath), to insert this data into the database as new entries.

– ***Step 2: Data Analysis***

The data analyzer enhances the context-reasoning module's effectiveness through a bag of data analysis algorithms. Hence, it calls the data retriever to acquire data for analysis according to two methods: *real-time analysis* and *on-demand analysis*. Real-time analysis refers to event driven methods triggered whenever new entries are inserted into the DB; they are continuously executed to update the context-reasoning module, as needed, depending on application purposes. These algorithms should be lightweight to maintain the responsiveness of the system. A typical example is statistical methods as they can track pattern frequencies using dynamic data structures. On-demand reasoning is initiated by user or application request, for many purposes such as machine-learning algorithms and recommendation systems. These algorithms can be heavyweight and executed online or offline as they do not influence the responsiveness of the system.

– ***Step 3: Data Retrieve***

The data retriever is designed according to the Data Access Object (DAO) pattern to abstract database operations, allowing seamless interaction with different database types. It consists of a set of callable methods that retrieve HUD according to different input parameters using query languages (e.g. SQL, XPath, etc.), depending on the DB adopted. Two types of data retrieval methods are proposed: *trigger-based retrieval* and *query-based retrieval*. Trigger-based retrieval automatically executes predefined methods when specific DB update conditions are met. Query-based retrieval accesses data using parameterized methods, enabling flexible and efficient queries.

Depending on where HUD is stored and processed, the HUD manager supports three methods in addition to their hybridization, providing various options to meet users' requirements and application specifications

- ***Client-side management:*** HUD are stored, retrieved and analysed directly on the client devices.
- ***Server-side management:*** The HUD manager resides on a server, which handles storage, retrieval, and analysis. Clients interact with the server by invoking the data logger and data analyser.
- ***Proxy-based management:*** The HUD manager operates on a proxy positioned between the client and server, facilitating intermediate processing.
- ***Hybrid management:*** Components of the HUD manager namely *database*, *data logger*, *data retriever,* and *data analyser* are distributed across clients, proxies, and servers. This flexible approach adapts to computational resources, user preferences, and context changes.

## 4. Implementation of the Proposed Approach

To demonstrate the feasibility of our proposal, we developed a real prototype to adapt multimedia documents (pedagogical materials) for ubiquitous learning purposes based on contextual situations while storing the HUD. From a functional standpoint, most MDA processes perform well. However, prototypes developed to validate adaptation approaches often rely on simulations to collect context elements, particularly for data from connected objects [9], [10], [11]. While simulations are useful in many scenarios, developing a real prototype provides a more practical and viable solution, especially for addressing technical challenges. We note that this implementation only focuses on non-instructional context. This is because the adaptation based on instructional context primarily generates multimedia documents tailored to users' learning styles without addressing environmental requirements.

### 4.1 Context Sensing

Context values are obtained from various hardware and software sources. Initially, these values are collected as raw data, representing only measured inputs. These raw values are then used as input for the reasoning phase, where adaptation rules are constructed. The context elements utilized in our prototype are detailed in Table 4.1.

| Context class | | Attribute | Domain of values | Source |
|---|---|---|---|---|
| Non-instructional | Physical | Noise | *value* ≥ 0 | Sensors |
| | | Luminosity | *value* ≥ 0 | |
| | Computational | Battery level | *level* ∈ [0, 100%] | Device |
| | | CPU load | *load* ∈ [0, 100%] | |
| | | Memory usage | *usage* ∈ [0, 100%] | |
| | | Screen size | *(width, height) (inch)* | |
| | | Smart TV | *availability* ∈ {*available*, *unavailable*} | |
| | | Network | *Bandwidth (value ≥ 0)* <br> *Type* ∈ {*WI-FI, mobile data, …*} | |
| | Temporal | Time | (*date*, *time*) | Device |
| | User | Preferred language | *list of languages* | Profile |
| | | Agenda | *list of timed tasks / places* | |
| | | Location | *Lis of places* {*home, lab, public place, …*} | Profile |
| Instructional | | Sensing-intuitive | {*sensing, intuitive*} | Profile |
| | | Visual-verbal | {*visual, verbal*} | |
| | | Active-reflective | {*active, reflective*} | |
| | | Sequential-global | {*sequential, global*} | |
| | | Cognitive-state | {*beginner, intermediate, expert*} | |

**Table 4.1.** Different context classes features.

## 4.2 Reasoning on the Context

Reasoning on context values occurs in two phases: *the qualification of context values* and *the inference of adaptation actions*. The first phase aims to qualify the quantitative values of the context to ensure they can be effectively used during the inference of adaptation actions and HUD analysis. Table 4.2 provides a typical quantification of context values, which are usually defined empirically based on observations and practical experience. Context elements whose value domains are already qualitative remain unchanged (e.g., location, network, preferred language, or sensing-intuitive traits).

| Attribute | Quantitative value $v$ | Qualitative value |
|---|---|---|
| Noise | $v < v_{min}$ | Quiet |
| | $v \in [v_{min}, v_{max}]$ | Less noisy |
| | $v > v_{max}$ | Noisy |
| Luminosity | $v < v_{min}$ | Penumbra |
| | $v \in [v_{min}, v_{max}]$ | Luminous |
| | $v > v_{max}$ | Very luminous |
| Battery level | $v < v_{min}$ | Low |
| | $v \in [v_{min}, v_{max}]$ | Medium |
| | $v > v_{max}$ | Full |
| CPU load | $v < v_{min}$ | Loaded |
| | $v \in [v_{min}, v_{max}]$ | Medium |
| | $v > v_{max}$ | Over loaded |
| Memory usage | $v < v_{min}$ | Loaded |
| | $v \in [v_{min}, v_{max}]$ | Medium |
| | $v > v_{max}$ | Overloaded |
| Screen size | $v < v_{min}$ | Narrow |
| | $v \in [v_{min}, v_{max}]$ | Wide |
| Network Bandwidth | $v < v_{min}$ | Week |
| | $v \geq v_{min}$ | Strong |
| Time | $v \in [T_{min}, T_{max}]$ | Day |
| | $v \notin [T_{min}, T_{max}]$ | Night |

**Table 4.2.** Qualification of quantitative values of typical context elements.

Once the context values are qualified, the next phase is to infer the adaptation actions. Each adaptation action is represented as a rule in the form "*if* (*conflicts*), *then* (*actions*)". Table 4.3 provides details of a subset of typical adaptation actions, illustrating how conflicts are resolved through specific rules. These rules are inspired by those proposed in [10], [11].

| Rule ID | Conflicts | Adaptation actions |
|---|---|---|
| R1 | *if* (public place *or* noisy environment) *then* | Speech to text |
| R2 | *if* (very luminous environment) *then* | Contrast adjustment |
| R3 | *if* (low battery *or* overloaded memory) *then* | Speech to Text<br>Video to images |
| R4 | *if* (inappropriate language) *then* | Language translation |
| R5 | *if* (overloaded CPU) *then* | Format change<br>Video to images |
| R6 | *if* (smart TV available *and* small screen) *then* | Video in smart TV |
| R7 | *if* (busy agenda) *then* | Summarize text |
| R8 | *if* (communication bandwidth is bad) *then* | Format change |

**Table 4.3.** Some typical rules for carrying out adaptation actions.

## 4.3 Execution of the Adaptation Actions

Our prototype supports the adaptation of web content based on user context, without being tied to a specific domain of application. The goal is to ensure the generality and adaptability of the prototype so it can be used across a wide range of applications, whether they involve multimedia documents with simple objects (e.g., text, images, videos) or complex compositions combining various types of content.

In general, several types of document sharing exist depending on application scope. We cite as examples File Transfer Protocol (FTP), peer-to-peer sharing, and cloud-based sharing (e.g., [126], [127]). For our prototype, we adopted the HTTP protocol for sharing multimedia documents hosted on a server, due to its simplicity of implementation.

The adaptation of multimedia documents involves executing a series of adaptation actions. Each action is carried out through one or more adaptation services, depending on the nature of the media objects in the documents. The description of the adaptation actions given in Table 4.3 is as follows:

- ***Speech to text:*** Mutes auditory objects by replacing them with textual content (e.g., muting video sounds and adding subtitles, or converting audio files to text).
- ***Contrast adjustment:*** Enhances the contrast of visual objects based on the brightness ratio.
- ***Video to images:*** Converts videos into sequences of images.
- ***Language translation:*** Translates media objects written in a source language into the target language specified in the user's profile.
- ***Format change:*** Changes the format of a media object to another format (e.g., mp3 to mp4).

- ***Video in smart TV:*** Plays video objects on a smart TV to take advantage of the larger screen size.
- ***Text summarizing:*** Summarizes the content of text objects.

## 4.4 Hardware and Software Configuration

The hardware configuration includes an *Arduino UNO board* connected to three sensors: *sound*, *GPS*, and *photoresistor light*. Our prototype supports WiFi and Bluetooth Low Energy (BLE) connectivity. WiFi, one of the most popular wireless communication protocols, is highly compatible with a wide range of devices. BLE-based devices are widely adopted in IoT implementations due to their low power consumption and consistent connectivity. These protocols are used within the sensing layer shown in Figure 4.1 to collect context elements from various sources. They also facilitate intercommunication between wired and wireless devices. The prototype can be easily expanded to support additional communication protocols, such as Zigbee, to meet potential compatibility demands with installations based on other ubiquitous IoT protocols.

The hardware configuration also includes two *HP Z640 stations* (*Xeon* CPU with 20 cores and 64 GB of RAM) used as a proxy and a server. A *Samsung Galaxy A10s* smartphone (*Helio P22* CPU and 2 GB of RAM) serves as the client device, where documents are executed. Adaptation services are invoked using the following Java APIs:

- *Google cloud speech API* (details at: [https://cloud.google.com/speech-to-text/])
- *Yandex translate* (available at: [https://tech.yandex.com/translate/])
- *Fast Forward Moving Pictures Expert Group (FFMPEG)* (downloadable from: [https://www.ffmpeg.org/])
- *Jave* (downloadable from: [http://www.sauronsoftware.it/projects/jave/]).

The DB component is implemented using both *relational* and *NoSQL* databases, which operate independently of specific data analysis frameworks to allow for a comparative evaluation.

- On the one hand, relational databases efficiently handle structured data, offering robust storage and retrieval functionalities
- On the other hand, NoSQL DBs can handle large volumes of data at high speed, with scalable architectures suitable for varying degrees of data structuredness.

Besides, there have been significant efforts made to bridge oriented-object, ontology-based and key-value paradigms with relational and NoSQL databases (e.g., [7], [8]). The stored data include qualitative values of the context elements and the corresponding adaptation actions resulting from context changes over time. The structural design of the HUD is represented by the class diagram shown in Figure 4.2.

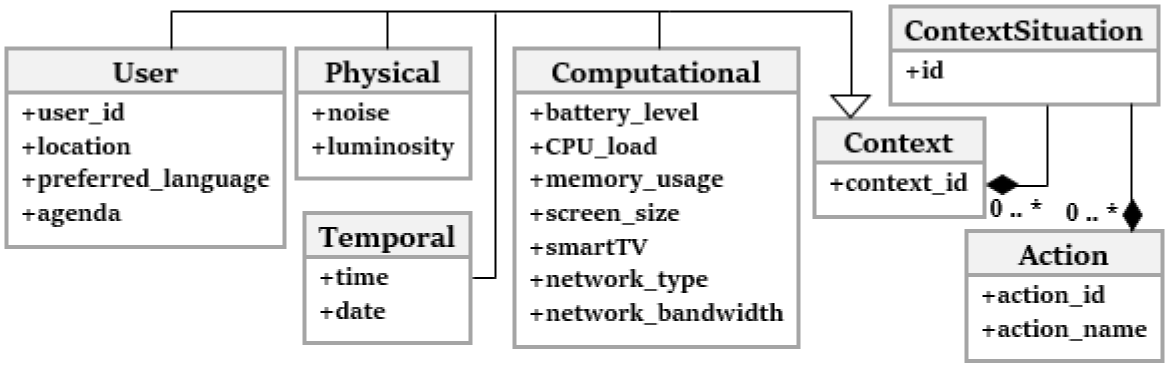



Figure 4.2. The structural design of the HUD.

Depending on where the HUD are stored, multiple database configurations have been implemented (see Table 4.4). For relational databases:

- **SQLite** is used for client-side methods
- **MySQL** is used for both proxy-based and server-side methods.

For NoSQL databases, JSON files are used across all methods.

| Management method | Relational model | NoSQL model |
|---|---|---|
| Client-side | SQLite database | JSON file |
| Proxy-based | MySQL database | JSON file |
| Server-side | MySQL database | JSON file |

**Table 4.4.** Technical choices to implement the database component in the HUD manager.

The implementation of the remaining components – namely the data logger, data retriever, and data analyzer – relies on the features of the object-oriented paradigm. Specifically, Java interfaces are used to define process specifications, allowing for flexible and modular implementations that can adapt to various requirements. Table 4.5 provides an overview of the implementation details, categorized by the different data management methods.

| Management method | Technical choice for implementation |
|---|---|
| Client-side | Java language for Android systems |
| Proxy-based | Java language for standard systems |
| Server-side | Java language for standard systems |

**Table 4.5.** Technical choices for implementing the components of the HUD manager.

## 5. Results and Discussion

To validate our proposal, three tests were conducted:

1) **Case study 1:** This test demonstrates how data are organized in the DB and provides a typical example of the adaptation result of a multimedia document containing learning materials.
2) **Performance evaluation:** This test measures and evaluates the proposal's performance in terms of time and data size required for processing HUD.
3) **Case study 2:** This test illustrates the usefulness of storing HUD by considering a specific example of how HUD contributes to personalizing adaptation rules related to context-aware learning tasks.

### 5.1 Formatting and Storing the HUD

This section describes how HUD are formatted and stored in the DB. Intuitively, two methods can be used to transform the HUD modeled in Figure 4.2 into relational models or JSON objects:

1) **Tabular format:** In this approach, log data are structured in a columnar format by grouping together all attributes of HUD classes in a single table. This results in a set of structured entries (rows), making the data well-suited for applying data analysis techniques.
2) **Structred objects:** Here, log data are represented as structured entities (e.g., linked tables, nested objects) using mapping procedures while maintaining the relationships among the data. This format is more suitable for data readability and high-level specification (e.g., [7], [8]).

In our work, we adopt the tabular format since our primary goal is to perform data analysis. Below, we present an illustrative scenario in which HUD are stored by executing an adaptation process triggered under specific conditions. For this purpose, we rely on the prototype developed by running a client-side adaptation system that deals with web pages contents according to the context situations of users while handling their HUD.

- **Case Study 1: Validation Scenario**

*Mr. Ahmed is a medical-researcher at University and frequently connects to medical forums for discussion. Given the nature of his activities, Ahmed often travels to forums, conferences, and similar events. While traveling, he wants to consult a multimedia document related to radiology. The document is written in French; it consists of text, audio, image and video contents, as illustrated in Figure 4.3.*

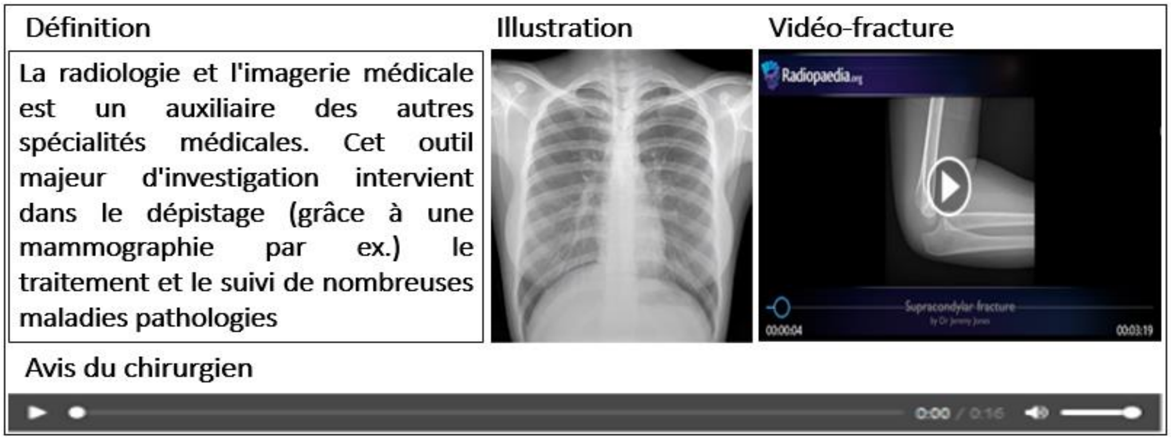



**Figure 4.3.** Structure of the original multimedia document.

Let assume that Mr. Ahmed is in a conference room and is using his tablet to access the document. The battery level of his tablet indicates it has reached 25%. The adaptation system infers the following contextual constraints: *public place*, *low battery* and *inappropriate language*. As a result, Rules $R_{i\,=\,1,\,3,\,4}$ (refer to Table 4.3) are executed to generate an adapted document, as shown in Figure 4.4. Simultaneously, the system invokes the data logger to store the log data in the DB, according to the methods shown in Section 3. A typical HUD input content is encoded as in the following JSON code.

{"ContextSituation": {"situation_id": "1", "context_id": "5", "user_id": "1", **"preferred_languge": "inappropriate language"**, **"location": "public place"**, "agenda": "busy", **"battery_level": "low"**, "CPU_load": "low", "memory_usage": "low", "screen_size": "small", "network_type": "mobile data", "network_bandwidth": "good", "smartTV": "unavailable", "noise": "quiet", "luminosity": "luminous", "time": "9:33:54", "video_in_TV": "false", **"language_translate": "true"**, "txt_summarize": "false", **"speech2txt": "true"**, "contrast_adjust": "false", "format_change": "false", **"video2image": "true"**}

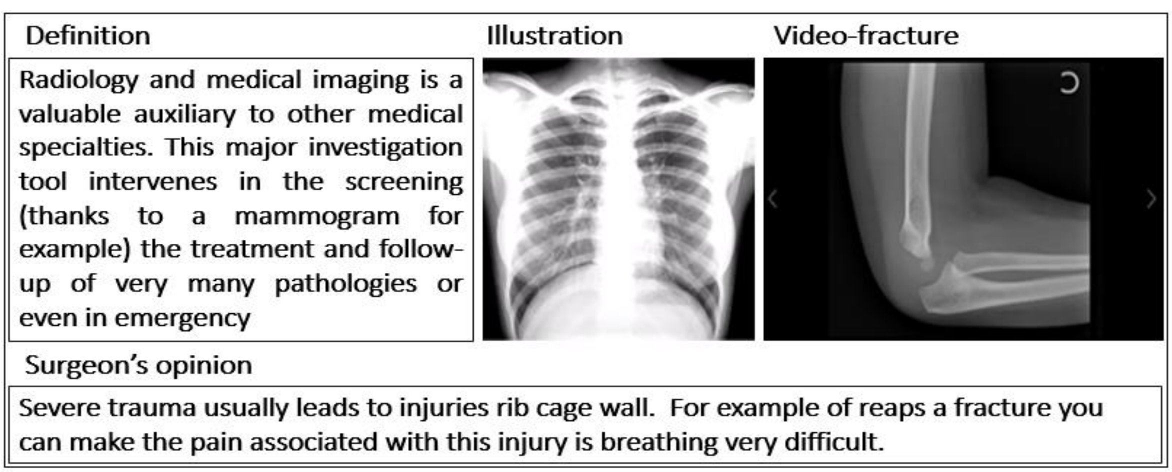



**Figure 4.4.** Structure of the resulting adapted document.

## 5.2 Performance Analysis

### 5.2.1 Running Time and Data Size Measurements

The performance of the DB models listed in Table 4.4 is evaluated by measuring the running time and data size required for processing HUD entries as the number of entries in DBs increases. Specifically, the evaluation focuses on:

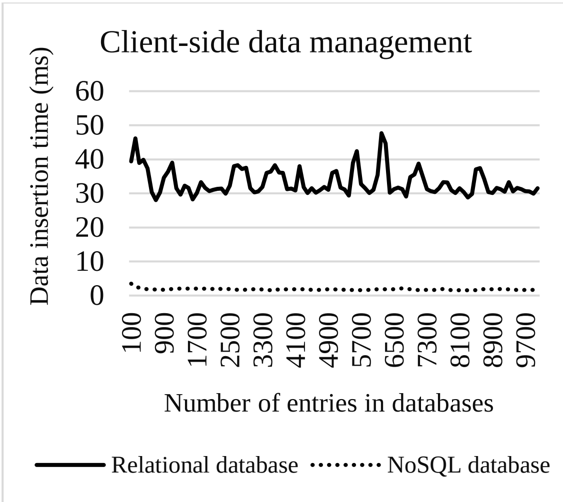
- Average time for inserting one HUD entry into the DB.
- Required time for loading the DB content into memory for data analysis
- Storage space needed for storing the HUD entries into the DB.

The tests are conducted using the hardware configuration outlined in Subsection 4.4. Figure 4.5 presents the results, assuming each user stores a maximum of 10,000 entries. For the evaluations, the users' number is set to **1** for client-side HUD management and to **50** for proxy-based / server-side HUD management. Figure 4.5 includes six subfigures as follows:

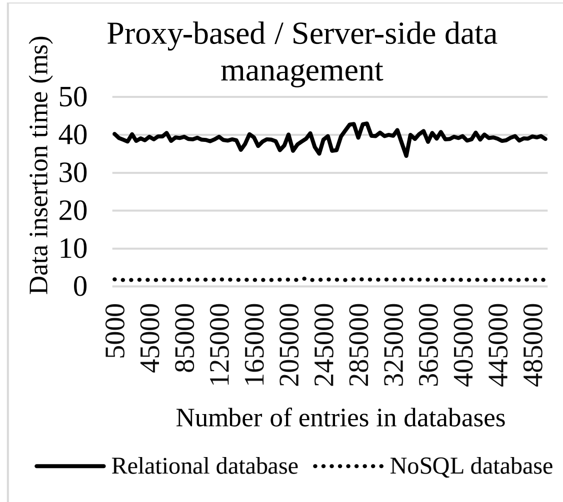
- Subfigures (a) and (b) plot the average time required to insert a new entry into DBs.
- Subfigures (c) and (d) plot the average time required to retrieve entries from DBs.
- Subfigures (e) and (f) plot the size of the data stored in DBs.

From the curves illustrated in Figure 3.5, the following observations can be made considering both cases (i.e., client-side and proxy-based / server-side HUD management):

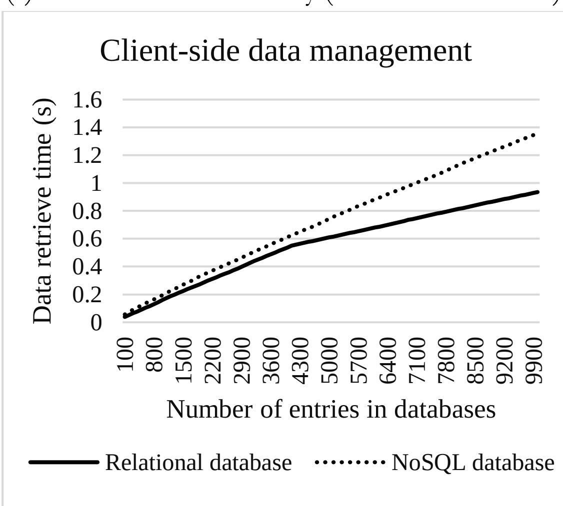
1) **Insertion time:** The time required to store one entry remains nearly constant regardless of DB size (see Figure 4.5 (a) and (b)). In particular, NoSQL databases outperform relational DBs for data storage due to their simpler data storage mechanisms. This is because relational database management systems use advanced techniques to optimize the data storage, which slightly affects the processing time.
2) **Retrieval time:** The time required to retrieve and load data increases linearly with DB sizes (see Figure 4.5 (c) and (d)). This is expected as the volume of processed data becomes larger. Nevertheless, relational DBs outperform NoSQL DBs for data retrieval as the latter often parse data into objects when loading them into memory, which adds overhead.
3) **Storage space:** The storage space required increases linearly with DB size (see Figure 4.5 (e) and (f)). Besides, relational DBs require less storage space given that NoSQL DBs store entries as pairs of keys (i.e., objects with attributes and values). To reduce storage requirements in NoSQL DBs, we need to minimize the descriptors length. Despite this, NoSQL DBs may excel, as they do not require dedicated configurations for exporting or accessing data in addition to their simplicity of implementation.



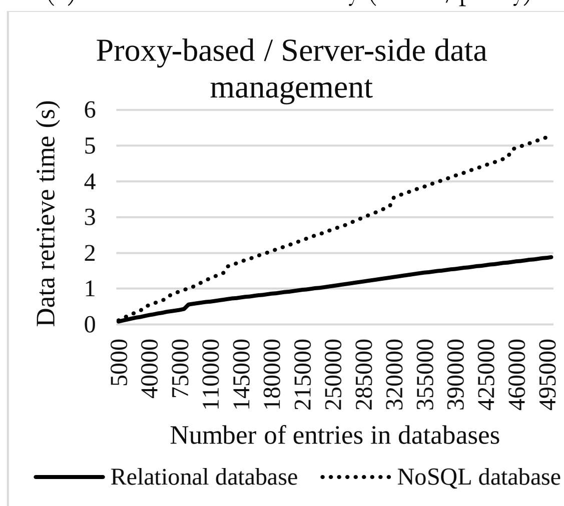

(a) Time to insert one entry (client-side methods).



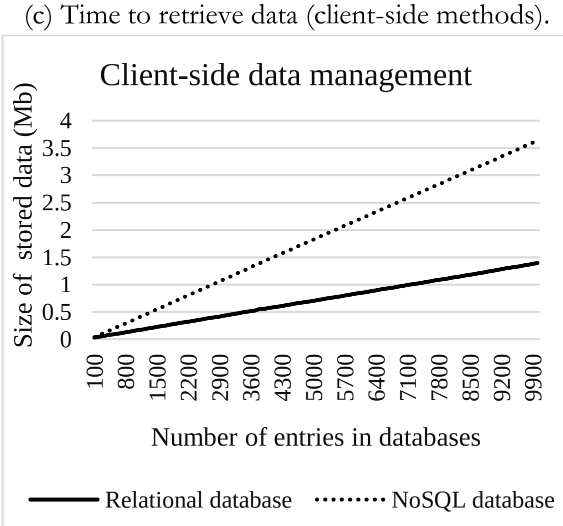

(b) Time to insert one entry (server/proxy).



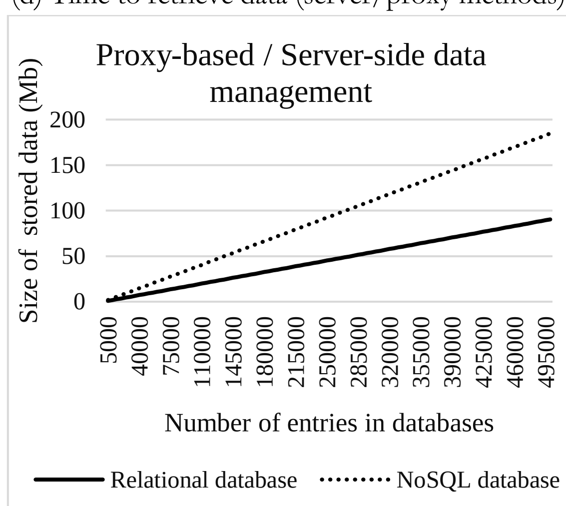

(c) Time to retrieve data (client-side methods).




(d) Time to retrieve data (server/proxy methods).




(e) Data size in client-side methods.




(f) Data size in server/proxy methods.

**Figure 4.5.** Performance analysis results for processing HUD.

Overall, the general performances of the proposed HUD management methods depend mainly on the available computational resources. For instance, during the experiments, we could observe:

1) **Client-side limitations:**
    - The client-side method was unable to store more than 55,000 entries in SQLite DBs.
    - It also could not load more than 150,000 entries into memory when using NoSQL DBs.
2) **Proxy-based limitations:**
    - The proxy-based method faced a limitation in loading, being unable to process more than 800,000 entries in memory.

These limitations highlight the importance for addressing efficiency-related challenges, including:

- **Data storage and retrieval:** Optimizing the capacity and speed of storage systems.
- **Data analysis:** Ensuring algorithms can handle large datasets effectively.
- **Scalability:** Designing solutions that adapt to growing data volumes and resource constraints.

### 5.2.2 Comparison of the Proposed HUD Management Methods

In this section, we compare the HUD management methods detailed in Section 3 based on their ability to meet the following criteria: *data volume*, *data performance*, *data retrieval*, *data analysis*, *scalability*, *data sharing*, and *data security*. These criteria depend on certain factors that affect the general performance of the HUD management system such as data storage location, data size, computation resources and number of users. Table 4.6 provides details about each criterion used for the comparison.

Table 4.7 summarizes the comparison between the HUD management methods according to the criteria presented in Table 4.6. These statistics help determine the most effective approaches for managing and processing HUD. Overall, **server-side methods** excel due to their computational resources, capacity to handle a large number of clients, and ability to store vast data volumes. Conversely, **client-side methods** offer superior data security, as data are stored locally on users' devices, ensuring access only with user permission.

| Criterion | Overview | Influencing factors | Qualitative evaluation values | | |
|---|---|---|---|---|---|
| | | | **High value** | **Mediocre value** | **Low value** |
| Data volume | Data volume the system can process (storage and retrieval). | Computation resources<br>Number of users<br>Data size | Big data | Large | Small |
| Data performance | Performance for storing / retrieving large volumes of data. | | Excellent | Good | Average |
| Data retrieval | Ability to load large volumes of data for processing. | | Excellent | Average | Limited |
| Data analysis | Computational ability to perform analysis tasks upon HUD. | | Easy | Feasible | Difficult |
| Scalability | Number of users the system can handle. | | Scalable | Less scalable | Not scalable |
| Data sharing | Ability to share HUD among users to enhance experiences. | Number of users | Suitable | Less suitable | Unsuitable |
| Data security | Ability to deal with privacy and security of users' data. | HUD storage location | Strong | Weak | Very weak |

**Table 4.6.** Criteria for comparing the proposed HUD management methods.

| Method | Security | Volume | Performance | Retrieval | Analysis | Sharing | Scalability |
|---|---|---|---|---|---|---|---|
| Client | Strong | Small | Average | Limited | Difficult | Unsuitable | Not scalable |
| Proxy | Weak | Large | Good | Average | Feasible | Less suitable | Less scalable |
| Server | Weak | Big data | Excellent | Excellent | Easy | Suitable | Scalable |
| P-2-P | Weak | Small | Average | Limited | Difficult | Less suitable | Less scalable |

**Table 4.7.** Comparison between the proposed HUD management methods.

However, the performance of these methods can be influenced by factors such as computation resources, workload, load balancing, fault tolerance, and service availability. To address these challenges, hybrid management methods – combining client-side, server-side, proxy-based, and peer-to-peer paradigms – can provide an efficient solution. Indeed, the hybrid methods allow for optimal placement of HUD manager components in different locations, depending on the requirements of applications and users.

### 5.2.3 Comparison of Our Proposal Over Other Approaches Exploiting HUD

Finally, we compare our proposal with three research-works [35], [124], [125] that exploits HUD in pervasive systems. The goal is to show the strengths of our proposed approach as well as the value added to improve MDA processes. Table 4.8 outlines the features of the considered approaches in comparison to our own. The comparison focuses on the following criteria:

- ***Approach scope:*** Describes the scope of each research work.
- ***Approach category:*** Indicates whether the HUD management method is server-side, proxy-based, or client-side.
- ***Efficient historical data management:*** Identifies whether the approach provides mechanisms for effective HUD handling.
- ***Technical details:*** Specifies whether the research work includes technical implementation details about HUD handling.
- ***Historical data use:*** Explains how HUD is leveraged for specific tasks.
- ***Historical data involved:*** Refers to the elements included in the HUD.
- ***Approach flexibility:*** Assesses the adaptability and generality of the approach across various application categories.

| **Approach scope** | **Category** | **HUD management** | **Technical details** | **HUD involved** | **HUD use** | **Flexibility** |
|---|---|---|---|---|---|---|
| adaptation of multimedia content (movies) to users' needs in smart homes [124] | Client-side | Yes | No | Context values<br>Multimedia contents | multimedia content recommendation | No |
| personalized recommendation system for learning materials in an ubiquitous learning environment [125] | Server-side | No | No | Learner profile<br>Learner model | learning materials recommendation | No |
| context-aware middleware supporting reasoning mechanisms for user interaction in cultural spaces [35] | Server-side | Yes | Yes | Context values<br>Users actions | user interaction enhancement | No |
| **Our proposal is a context-aware MDA architecture supporting reasoning mechanisms on context and HUD** | Adaptable to all categories | Yes | Yes | Context values<br>Adaptation actions | Efficient multimedia content adaptation | Yes |

**Table 4.8.** Characteristics of our proposal and some pervasive systems involving HUD.

By leveraging hybrid management methods (combining client-side, server-side, proxy-based, and peer-to-peer paradigms), our proposal overcomes limitations of standalone methods (e.g., workload balancing and service availability). The integration of cloud computing techniques further enhances data handling capabilities. Based on the criteria in Table 4.6, the key advantages of our approach over those discussed in [35], [124], [125] are summarized in Table 4.9. It should be noted that these values are assigned according to the details provided in the corresponding references. Moreover, the lack of technical details and difficulty in reconstructing these context-aware systems precluded direct numerical comparisons.

| Approach | Security | Volume | Performance | Retrieval | Analysis | Sharing | Scalability |
|---|---|---|---|---|---|---|---|
| [124] | Strong | Small | Average | Average | Feasible | Unsuitable | Not scalable |
| [125] | Weak | Small | Average | Average | Feasible | Suitable | Not scalable |
| [35] | Weak | Big data | Good | Average | Feasible | Suitable | Scalable |
| **Our proposal** | **Strong** | **Big data** | **Excellent** | **Excellent** | **Easy** | **Suitable** | **Scalable** |

**Table 4.9.** Comparison of our proposal against some approaches involving HUD.

In view of the foregoing, we assess our proposal over related-works. The goal is to show the added values so as to improve MDA processes. The main advantages of our approach are summarized as follows:

1) **Complementary to existing approaches:** Our proposal enhances MDA processes by efficiently handling HUD, as most existing approaches do not address adequately.
2) **High adaptability:** The proposal manages HUD at a high level of abstraction, making it compatible with various MDA approaches.
3) **Flexibility:** It provides diverse options for storing, retrieving, and analyzing HUD. These features help overcome the issues outlined in Table 3.7 through an efficient placement of the HUD manager components based on applications and users needs.

### 5.3 Case Study 2 HUD Analysis for Context-aware Adaptation Rules Learning

This section presents a case study to demonstrate the usefulness of storing and processing HUD. The following scenarios illustrate how HUD contributes to the personalization of adaptation rules through rule-learning mechanisms. This task refers to the prediction of adaptation actions using data-driven approaches (i.e. based on data analysis and interpretation) rather than explicitly predefining them.

- ***Scenario 1:*** *Mr. Amine is a mobile user. While moving, he accesses multimedia documents containing various learning materials using mobile data from his smartphone – a paid and limited service. To minimize mobile data consumption, he always asks the system to adapt the documents by converting media objects to low-quality format.*

– ***Scenario 2:*** *Mrs. Sarah is a university student. Back home, she always revises her online courses using her laptop. After a busy daily schedule, she typically feels tired. Consequently, she asks the system to adapt the target documents by summarizing their contents for easier comprehension.*

These scenarios are implemented by storing HUD in a NoSQL database, denoted by *DB*, hosted on a proxy. For each scenario, we store two types of HUD:

1) **Scenario-specific entries:** A sufficient number of entries are created based on the assumptions described in the scenarios.

2) **Random entries:** Random entries are added to the database to simulate dynamicity (i.e., noise) during data analysis.

To infer new personalized adaptation rules, the process depicted in Figure 4.6 is proposed.

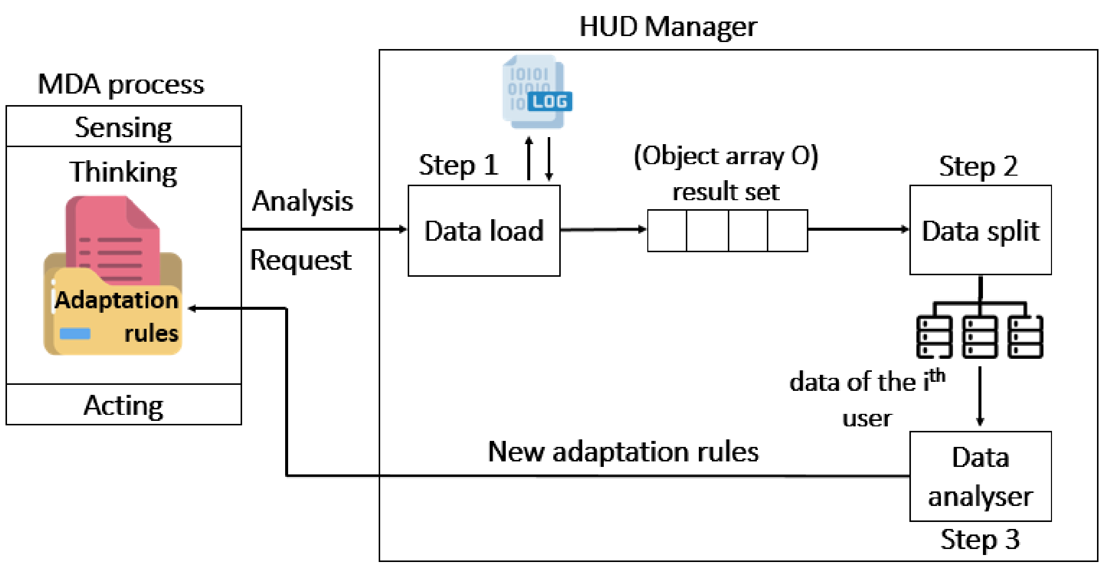



**Figure 4.6.** Process phases for adaptation rules learning from HUD.

The steps are as follows:

1) **Data retrieval:** The data analyzer invokes the data retriever to load database *DB* into memory as an array of objects, denoted by *O*.

2) **Data segmentation:** Array *O* is divided into subparts $O_{i=1..n}$ according to user *id*; here *n* represents the total number of users.

3) **Conflicts and adaptation action analysis:** the elements of each subpart $O_{i=1..n}$, denoted by $I_i=\{o_{ij}\}$, are analyzed. The analysis extracts associated conflicts and adaptation actions in each subpart $O_{i=1..n}$ through the following backtracking algorithm.

**Backtracking algorithm**

**Input variables:**

- Let $X = (x_1, x_2, \ldots, x_m)$ be a vector where each element $x_i \in X$ corresponds to a context element; $m$ is the number of context elements. The possible values for $x_i$ include all qualitative values for the corresponding context element in addition to the *null value*. For example, if $x_i$ represents *battery level*, then $Domain\ (x_i) = \{null, low, medium, full\}$.
- Let $A = \{a_1, a_2, \ldots, a_k\}$ represent the set of adaptation actions, where $k$ is the total number of actions. Each $a_{j=1..k} \in A$ can be 0 or 1, indicating whether adaptation action $a_j$ is considered for execution.

**Rule generation:**

The backtracking algorithm goes through paths built by combining the possible values for context elements $x \in X$ and adaptation actions $a \in A$. When the confidence degree of any given combination $(x, a)$ exceeds a threshold $T$, it will be considered as a personalized rule for the $i^{th}$ user.

**Confidence degree:**

The confidence degree is defined as the ratio of:

- the number of objects in $I_i=\{o_{ij}\}$ that include non-null values from vector $x$ and executed actions from set $a$
- the total number of objects in $I_i=\{o_{ij}\}$ containing all non-null values from vector $x$.

**Rule format:**

The resulting rules are expressed as:

- IF $(\bigwedge_{i=1}^{|x|} c_i \in x \,/\, c_i \neq \text{null})$ THEN $v_{j=1..k} \in a$ where $v_j = 1$.

**Redundancy check:**

Each time a potential rule is identified, a redundancy check is performed to decide whether the rule should be retained or discarded.

Table 4.10 provides examples of typical rules retained during the analysis phase. The above scenarios highlight the importance of processing HUD to develop intelligent context-aware applications through the learning of adaptation rules, which can predict user behaviors.

| Rule ID | Conflicts | Adaptation actions | Scenario | Confidence |
|---|---|---|---|---|
| R1 | *if* (internet data is limited) *then* | Format change | Scenario 1 | 95% |
| R2 | *if* (night time *and* busy agenda) *then* | Summarize text | Scenario 2 | 92% |

**Table 4.10.** Typical rules retained during the analysis phase.

The main advantages of this approach are summarized as follows:

- **Unsupervised machine learning:** Using unsupervised machine learning helps infer context-aware adaptation rules directly from HUD rather than predefining them. This eliminates the need for pre-trained datasets to perform the learning task.
- **Efficient data processing:** It is possible to limit log data processing to specific subparts by employing a sliding window approach (e.g., focusing only on recent data).
- **Reusability:** The system is adaptable to other rule-based context-aware pervasive environments, enhancing its versatility and utility.

Despite these advantages, there are also notable limitations:

- **Inefficient association rule algorithms:** The system does not employ advanced algorithms such as the frequent pattern growth algorithm. Instead, it relies on backtracking, which is unsuitable for processing large volumes of data. More effective algorithms are necessary to extract personalized adaptation rules on a large scale.
- **Blind correlation limitations:** The system is based on unweighted correlations, which can lead to inefficiencies. For example, it may take significant time to replace outdated frequent rules with more recent ones, affecting system responsiveness.
- **Lack of experience sharing:** The system does not support experience sharing between users, a feature that helps designing rule-based recommendation systems.

## 6. Conclusion

In this chapter, we investigated the management of HUD in MDA processes within context-aware pervasive systems. By leveraging both relational and NoSQL DBs, we demonstrated how HUD can be efficiently stored and utilized to enhance adaptation processes. Starting with an object-oriented modeling approach for context and adaptation actions, we translated these models into robust DB schemas, laying the foundation for a flexible and efficient management framework. To ensure adaptability across various strategies, we proposed four management methods: client-side, proxy-based, server-side, and hybrid. These methods were validated through real-world scenarios implemented in a prototype system, showcasing their effectiveness in supporting adaptation rules learning and dynamic system responsiveness.

While the feasibility of the proposed approach has been demonstrated, the developed prototype relies only on a limited subset of general adaptation rules that are not always adapted to specific application domains. To address real-world needs, the adaptation rules base can be expanded by incorporating additional contextual elements specific to domain requirements. Furthermore, to overcome the limitations discussed in subsection 5.3 and enhance the system's adaptability, the next chapters will explore the application of machine learning techniques and data analysis algorithms. These enhancements will enable learning from HUD, paving the way for more robust and domain-specific solutions.

# Chapter V

# Rule-learning for the personalization of adaptation actions

## Summary

# Chapter V

# Rule-learning for the personalization of adaptation actions

## 1. Introduction

In the previous chapters, we explored the importance of managing historical user data in the context of multimedia adaptation. This chapter focuses on the use of rule-based data mining algorithms, by means of the Eclat, FP-Growth, ID3 decision tree and sequential covering algorithms, to effectively manage historical user data in the context of context-aware multimedia adaptation. These algorithms play a crucial role in extracting meaningful relationships and insights from the data, enabling personalized experiences for users.

We present four methods that leverage HUD, collected through multimedia adaptation processes, to personalize context-aware adaptation rules via rule-learning mechanisms. This involves employing rule-based data mining classifiers, a widely used approach in machine learning and data mining. By using Eclat, sequential covering, FP-Growth, and decision tree algorithms, this work adopts a data-driven methodology that relies on analysis and interpretation rather than predefined rules. Our proposal has been validated through a client-side application that implements the proposed algorithms. This application systematically records context values and corresponding actions, providing the necessary HUD for the rule-learning system to generate adaptation rules.

The remainder of this chapter is structured as follows. Section 2 provides the main basics of association rule mining techniques and reviews some related works and domains applications. Section 3 introduces association rule mining algorithms specially: Eclat, sequential covering, FP-Growth, and decision tree algorithms. Section 4 gives the details of the proposed approach for the personalization of adaptation rules of multimedia documents. Section 5 focuses on validation aspects of the proposal by discussing its implementation as well as the obtained results. The chapter ends with some concluding remarks.

## 2. Association Rule Mining

Association rule mining is a data mining technique used to discover relationships between items in large datasets. For example, in a supermarket, it can identify the likelihood that a customer will purchase bread along with milk. This knowledge can help store owners optimize product placement or design targeted marketing campaigns, such as offering discount coupons for specific item combinations. Association rule mining gained prominence through its application in analyzing shopping cart contents and serves as a foundational technique in data mining, underpinning many other approaches [128].

Association rules are conditional statements that describe the likelihood of relationships between items. The process of association rule mining employs machine learning models to identify co-occurrences within a database, generating "if-then" rules that reveal these relationships [129].

### 2.1 Key concepts: support, confidence, lift:

Key concepts such as support, confidence, and lift are crucial metrics for evaluating the strength and relevance of the association rules generated by mining algorithms. These metrics help in identifying the most significant relationships within the data [128], [130], [131], [132], [133], [134].

- ***Support:*** Support measures how often an itemset appears in a dataset. It is calculated as the ratio of transactions containing the itemset to the total number of transactions. A minimum support threshold is set to filter out infrequent itemsets. For example, Lee et al. [135] used a minimum support threshold of 0.01% to generate association rules.
- ***Confidence:*** Confidence, often viewed as conditional probability, quantifies the likelihood that an itemset occurs given the presence of another. It is calculated as the ratio of transactions containing both itemsets to the number of transactions containing the antecedent itemset. A minimum confidence threshold ensures that only reliable associations are considered. Lee et al. [135] applied a 1% confidence threshold to evaluate the strength of generated rules.
- ***Lift:*** Lift quantifies the degree of dependence or correlation between two itemsets. It compares the actual confidence of a rule to the expected confidence (assuming no association).
  - **Lift = 1: No correlation**.
  - **Lift > 1: Positive correlation**; the occurrence of one item increases the likelihood of the other.
  - **Lift < 1: Negative correlation**; the presence of one item decreases the likelihood of the other.

For example, a lift value greater than 1 indicates a meaningful association, while a lift value less than 1 suggests that the items are less likely to co-occur [135].

## 2.2 How do association rules work?

Association rule mining (ARM) involves two essential phases: *the identification of frequent itemsets* and *the generation of association rules* [128].

1. **Identification of Frequent Itemsets:** This phase identifies items that frequently appear together in the dataset. These itemsets are selected based on a minimum support threshold, which filters out infrequent itemsets, ensuring that only relevant patterns are retained (i.e., those that co-occur in transactions at or above a defined support threshold) [129].
2. **Generation of Association Rules:** Association rules are derived from the frequent itemsets based on a specified confidence threshold. In this phase, each itemset that meets the support threshold is evaluated to identify associations that satisfy the confidence threshold [129]. For example, if two items co-occur in more than 50% of transactions where one of them is present, this could serve as a confidence threshold.

An association rule consists of two main components:

- **Antecedent (If):** This is the item or set of items present in the dataset.
- **Consequent (Then):** is the item or set of items that co-occur with the antecedent.

**Example:** consider the rule: If a customer buys bread, then he will likely buy butter. In this case, the antecedent is 'Bread' while the consequent is 'Butter'. Data analysts evaluate these rules by analyzing their **frequency** (support) and **reliability** (confidence).

### Application Domains

ARM is employed in various domains to uncover patterns and relationships in datasets. Key applications include:

- **Market Basket Analysis:** Market basket analysis employs ARM to discover which items are commonly bought together in a shopping transaction database, which helps to optimize product placement and allow effective promotions [136].
- **Customer Segmentation:** ARM aids in grouping customers with similar purchasing behaviors, enabling businesses to adapt marketing strategies and enhance customer engagement [137].
- **Recommender Systems:** in recommender systems, ARM is used to suggest products or services to customers by analyzing their past purchases or browsing behavior, making recommendations more personalized and relevant [137].

- **Medical Diagnostics:** ARM helps to identify relationships between symptoms, diseases, and treatments to improve diagnostic accuracy and decision-making.
- **Sports Analytics:** by analyzing player or team performance patterns ARM is a useful method to produce strategies and training programs.

These applications highlight ARM's versatility in identifying meaningful patterns and generating actionable insights, making it a valuable tool for predictions and recommendations.

## 3. Association Rule Mining Algorithms

The objective of ARM is to discover rules that meet or exceed user-defined thresholds for support (MinSup) and confidence (MinConf). ARM identifies relationships between items that frequently occur together with a specified level of reliability [136]. It is an unsupervised technique, meaning it does not require pre-labeled data, can handle data of varying lengths, and generates interpretable rules. The primary goal is to uncover patterns and associations among items in a transactional database, offering valuable insights into underlying relationships [129]. In the following sections, we present four widely used ARM algorithms: ECLAT, FP-Growth, Decision Trees, and Sequential Covering.

### 3.1 ECLAT algorithm

The ECLAT (Equivalence Class Transformation) algorithm is widely used for association rule mining due to its efficiency and effectiveness. It employs a vertical data format, where transactions are represented by the list of transaction IDs (TIDs) in which each item appears [138]. This structure minimizes the need for multiple database scans and facilitates efficient support calculation for itemsets [137], [139]. The ECLAT algorithm has several advantages over other algorithms such as Apriori. Firstly, it only scans the database once, reducing the time complexity from $O(t^2)$ to $O(t)$ compared to Apriori, where the database is scanned repeatedly [135], [138]. Secondly, ECLAT uses a vertical data structure, enabling faster generation of association rules by traversing the data structure. Finally, ECLAT is more memory-efficient than Apriori, as it reduces the number of frequent pattern candidates and uses an early pruning strategy to optimize memory usage [138], [135], [131], [140].

- **How ECLAT algorithm works?**

The Eclat algorithm consists of several steps, which can be summarized as follows:

1. **Preprocessing**: Convert the transaction database into a vertical format, where each item is linked to its corresponding transaction IDs (TID-lists) [132].
2. **Frequent Itemset Generation**: Identify all frequent k-itemsets by intersecting TID-lists of $k-1$ itemsets to calculate support, where $sup(x) = |tid(x)|$ [133].

3. **Candidate Generation**: Generate $k + 1 - itemsets$ from frequent $k\ itemsets$ [133].
4. **Pruning**: Eliminates itemsets that do not meet the minimum support threshold, retaining only frequent $k + 1$ itemsets [133].
5. **Iteration**: Repeat steps 2-4 until no more frequent itemsets are generated [133].

By following these sequential steps ECLAT algorithm transforms the data, identifies frequent itemsets, and iterates through the candidate generation process until no new frequent itemsets are produced.

***ECLAT Algorithm Pseudo-code***

***Input:*** Database $D$, Minimum support threshold $MinSup$
***Output:*** Frequent itemsets $F$
**Procedure ECLAT:**
**1.** Initialize $F = \{\}$ (set of all frequent itemsets).

**2.** For each item $i$ in $D$:

a. $Compute\ T(i) = \{transaction\ IDs\ containing\ i\}$.
b. $If\ |T(i)| \geq MinSup$:
$Add\ \{i\}\ to\ F$.
$Add\ \{i, T(i)\}$ to initial equivalence class.

**3.** Perform depth-first search (DFS) traversal:

$Call\ DFS(\{\ \}, equivalence_{class})$
$\boldsymbol{Procedure}\ DFS(current_{itemset}, equivalence_{class})$:
$For\ each\ item\ i\ in\ equivalence\_class$:
a. $New\_itemset = current\_itemset \cup \{i\}$.
b. $New\_tids = intersection\ of\ T(current\_itemset)\ and\ T(i)$.
c. $If\ |New\_tids| >= minsup$:
$Add\ New\_itemset\ to\ F$.
$Add\ New\_itemset\ to\ a\ new\ equivalence\ class$.
$Call\ DFS(New\_itemset, new\ equivalence\_class)$.
***End Procedure***

## 3.2 FP-Growth Algorithm

The FP-growth algorithm is a widely recognized technique in association rule mining, with applications across various domains such as data mining, information retrieval, and urban smart community systems [134], [140], [141]. Unlike traditional methods this algorithm is built on the principle of frequent pattern growth, utilizing a specialized tree structure known as the

FP-tree. The FP-tree efficiently compresses the dataset and facilitates the extraction of frequent itemsets, enabling the algorithm to mine large-scale data more effectively [134], [141], [142]. The FP-tree has two main components: a header table and the tree structure itself:

- **Header Table**: lists the frequent items ordered by their support counts.
- **Tree Structure**: represents the relationships between items, with each node corresponding to an item and branches indicating associations between items.

- **How FP-Growth algorithm works:**

The FP-growth algorithm involves two main phases: **FP-tree construction** and **FP-tree mining,** these phases can be summarized as follow [134], [141]:

**Phase 1: FP-Tree Construction:** this phase encompasses two steps detailed as follow:

1. **Cleaning and Sorting**:
   - Remove infrequent items based on a minimum support threshold.
   - Sort remaining items in descending order of frequency.
2. **Building FP-Tree and Header Table**:
   - To build the FP-tree, the algorithm first scans the database to count the frequency of each item and filter out those that aren't frequent enough. Then it performs a second scan to construct the tree, ensuring that only relevant data is included.
   - Map cleaned transactions to the FP-tree. If an item already exists in the branch, it shares the same node or element, and the count is increased. Otherwise, the item will be located on the new constructed branch.
   - Build a header table with linked lists which facilitates the rapid identification of an item's occurrence inside the tree without requiring a complete traversal of it [128].

**Phase 2: FP-Tree Mining:** this phase also has two main steps, as follow [134], [141], [142]:

1. **Conditional FP-Trees**:
   - FP-Growth algorithm traverses the FP-tree, collecting items along the path from each node to the root. The collection of items from these paths forms the "conditional pattern base," which is then used to recursively find more frequent itemsets.
   - Split the FP-tree into smaller conditional FP-trees for each frequent item.
   - Generate conditional pattern bases (prefix paths) for these trees.

2. **Recursive Extraction**:
    - Use depth-first search to derive frequent patterns from conditional trees.

***FP-Growth algorithm pseudo-code***

```
Input:
 - D: Transaction database
 - minsup: Minimum support threshold
Output:
 - F: Set of all frequent itemset
 Procedure FP − Growth(D, minsup):
    1. Scan D and calculate the frequency of each item.
    2. Remove items with support < minsup.
    3. Sort remaining items in each transaction in descending order of frequency.
    4. Construct the FP-tree:
        Create the root of the tree (null).
        For each transaction in D:
        Add items to the tree in order (from step 3).
        Update node counts and maintain pointers to the same items (header table).
    5. Extract frequent itemsets from the FP-tree:
      Call RecursiveFP(Tree, HeaderTable, ∅).
Procedure RecursiveFP(Tree, HeaderTable, Prefix):
    For each item i in HeaderTable (processed in reverse order):
      New_prefix = Prefix ∪ {i}.
      Add New_prefix to F.
      Construct the conditional pattern base for i:
          Traverse paths ending in i to extract prefix paths and their counts.
      Construct the conditional FP-tree for i using the conditional pattern base.
      If the conditional FP-tree is not empty:
          Call
RecursiveFP(Conditional_Tree, Conditional_HeaderTable, New_prefix).
End Procedure
```

The FP-growth offers several advantages across multiple domains applications, such as:

- Eliminates candidate generation, this process enhances efficiency and scalability, particularly for extensive datasets.
- Requires only two database scans and avoids the need for generating candidate itemsets, making it suitable for large datasets.
- Handles high-dimensional data effectively.

- Capability to manage high-dimensional data and is applicable for both frequent itemset mining and association rule mining [5].
- Applicable in diverse fields including information retrieval [143], urban smart systems and geographical queries reformulation [141], cancer patient data analysis [144], online education [145], and TCM constitution analysis [146].

### 3.3 Decision Tree Algorithm

The decision tree algorithm is a powerful tool for decision-making and rule discovery, with applications spanning machine learning, image processing, and pattern recognition [147], [148], [149], [150], [151], [152], [153], [154]. At its core, a decision tree is a hierarchical structure that organizes data into decisions based on specific features, making it suitable for both classification and regression tasks [150]. Decision trees are particularly effective for classification problems and are widely used in data mining. A decision tree consists of two types of nodes:

- **Decision Nodes**: Represent decision points and have multiple branches.
- **Leaf Nodes**: Represent outcomes and do not have further branches.

Figure 5.1 illustrates an example of a decision tree. As shown, a decision tree is a graphical representation of all potential solutions to a problem or decision based on specified conditions. The tree starts with a root node, which splits into branches, forming a tree-like structure. Each decision node poses a question, and depending on the response (e.g., Yes/No), it subsequently divides the tree branches into subtrees.

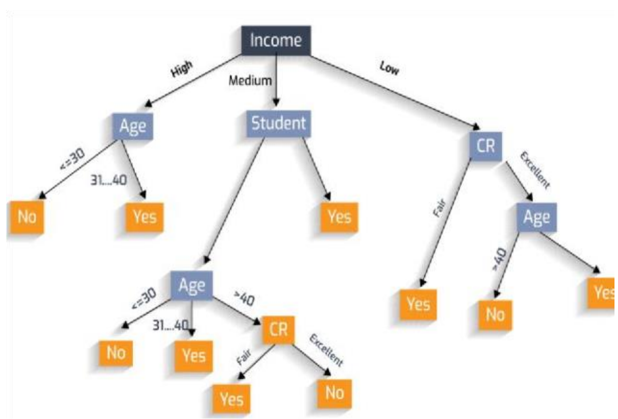



**Figure 5.1.** Example of a typical decision tree.

- **Decision Tree Terminologies**

A decision tree integrates a series of fundamental tests, where each test compares a feature's value to a threshold value [147]. The following statement are the main terminology involved in the decision tree algorithm:

- **Root Node**: The starting point of the tree that represents the entire dataset, which is divided into subsets.
- **Leaf Node**: The terminal node representing the final output. Once a leaf node is reached, no further splitting occurs.
- **Splitting**: The process of dividing the decision root/node into sub-nodes based on specific conditions.
- **Branch/Subtree**: A smaller tree formed from a split node.
- **Pruning**: is the process of removing the unwanted branches from the tree.
- **Parent/Child Node**: The root node of the tree is called the parent node, and its split nodes are the child nodes.

- **Types of Decision Tree Algorithms**

Several types of decision tree algorithms are commonly used, including:

- **ID3** (Iterative Dichotomizer 3): Focuses on information gain for selecting attributes.
- **C4.5**: An improvement over ID3 with support for continuous variables and pruning.
- **CART** (Classification and Regression Tree): Uses the Gini index for splits and is applicable to both regression and classification tasks.
- **CHAID** (Chi-squared Automatic Interaction Detector): Employs the Chi-square test for categorical variables.
- **QUEST** (Quick Unbiased Efficient Statistical Tree): Specializes in linear combinations for binary splits.
- **MARS** (Multivariate Adaptive Regression Splines): Extends decision trees for regression tasks.

| Algorithm | Measure for variable selection | Input variables | Split at each node | Pruning | Dependent variable |
|---|---|---|---|---|---|
| **ID3** | Entropy and info-gain | Categorical | Greedy, highest Information Gain | No | Categorical |
| **CART** | Gini Index | Categorical/ Continuous | Binary (linear combinations) | Pre - pruning | Categorical/ Continuous |
| **C4.5** | Entropy and info-gain | Categorical/ Continuous | Multiple | Pre - pruning | Categorical/ Continuous |
| **CHAID** | Chi-Squared | Categorical | Multiple | Chi-square | Categorical |
| **QUEST** | Chi-square (categorical); Jway ANOVA (continuous). | Categorical/ Continuous | Binary (linear combinations) | Post-pruning | Categorical |

**Table 5.1.** Comparison of common decision tree algorithms based on the specified criteria.

- **Entropy and Information Gain**

Entropy measures the randomness or impurity in a dataset [147]. It is quantified within a range of (0 to 1), and is optimized when close to zero, it is calculated using equation (1):

$$\boldsymbol{Entropy\ (S) = -\sum_{i=1}^{c} P_i\, log_2(P_i)} \qquad (1)$$

In this context, $\boldsymbol{P_i}$ represents the proportion of the dataset belonging to $\boldsymbol{i^{th}}$ attribute value.

Information gain, often referred to as mutual information, evaluates how much entropy is reduced by a split. It represents a state of order; a higher value indicates a more favorable condition. is defined based on the concept of entropy [148], as illustrated in equation (2):

$$\boldsymbol{Gain\ (S, A) = Entropy\ (S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} Entropy\ (S_v)} \qquad (2)$$

Here, $\boldsymbol{S_v}$ is the subset of $\boldsymbol{S}$ corresponding to the value $\boldsymbol{v}$ of attribute $\boldsymbol{A}$, and $\boldsymbol{V(A)}$ is the range of $\boldsymbol{A}$ [148].

- **How do Decision Tree Algorithm work?**

The tree is constructed through successive splits, where data is divided based on specific features. Internal nodes represent decision points, while leaf nodes indicate outcomes or classifications [147], [148]. In practical terms, the path from the root node to any given leaf outlines a set of conditions that must be met for an item to be classified into a particular class. These conditions can be written as a series of (*if-then*) rules, where each rule explains the logic behind the classification decision [147]. For instance, in a loan classification problem, the algorithm might classify applicants using rules such as: *IF income > $50,000 AND credit score > 700, THEN approve loan.* These rules form a "rule base," making decision trees interpretable and actionable [147].

The process is as follows:

1. Begin with the root node containing the complete dataset.
2. Use an attribute selection measure (e.g., information gain) to determine the best attribute for splitting.
3. Split the dataset into subsets based on the chosen attribute.
4. Recursively repeat the process until leaf nodes are reached, where no further splitting is possible.

- **Advantages of decision trees**
  - **Interpretability**: Decision trees are easy to understand and explain. They allow tracing how a decision was made, making them valuable in fields like healthcare, finance, and law [147].
  - **Versatility**: Suitable for both classification and regression tasks, decision trees are widely applied in fields like image processing, text classification, and natural language processing.
  - **Rule Extraction**: Decision trees can be converted into simple "IF-THEN" rules.
  - **Scalability**: Effective for large datasets, particularly when combined with ensemble methods like bagging, boosting, and random forests [152].
  - **Application Diversity**: Decision trees are employed in numerous domains, including: identifying patterns and extracting knowledge from large and complex datasets [23][24], text classification [148], recruitment process [151], natural language processing, and time series forecasting [29].

## 3.4 Sequential Covering Algorithm:

The Sequential Covering Algorithm is a rule induction technique based on the ***divide and conquer*** heuristic. It works by incrementally learning a single rule that covers a portion of the dataset. The rule is developed using a standard rule induction process, which is guided by a quality metric that helps determine the best possible rule [155]. It facilitates rule discovery and decision-making by producing rules that classify instances using interpretable conditions.

- **How do Sequential Covering Algorithm work?**

The algorithm begins with an empty rule set and a complete dataset. It then uses a rule induction method, applying an evaluation metric to expand each rule to a predefined length. Once a rule is created, the algorithm removes the data points it covers from the dataset. This process continues until all relevant patterns have been addressed [156]. The algorithm operates in two main stages: **growing** and **pruning**.

1. **Growing**: In this stage, the algorithm starts with an empty rule and progressively adds conditions, which are simple "if-then" statements based on data features. The goal is to identify the conditions that lead to the most accurate classification of the data.
2. **Pruning**: After the rule has been grown, the pruning stage begins. Here, the algorithm refines the rule by removing unnecessary and redundant conditions. The objective is to reduce complexity while maintaining or improving accuracy.

This process of growing and pruning allows the Sequential Covering Algorithm to refine its rules over time, ensuring they are both accurate and easy to understand, making it a useful tool for decision-making and rule discovery in various applications. The steps involved in SCA algorithm can be summarized as follows:

- **Initialization**: start with an empty rule and a complete dataset.
- **Rule Creation**: use a standard rule induction process, guided by a quality metric (e.g., accuracy or coverage), to add conditions iteratively.
- **Dataset Reduction**: remove all examples covered by the newly created rule from the dataset.
- **Iteration**: repeat the process until the entire dataset is covered or no further rules can be generated.

- **Advantages of Sequential Covering Algorithm**

  1. **Human-Readable Rules**: SCA generates interpretable "if-then" rules that are easy to understand and apply, making it ideal for use cases requiring transparency, such as healthcare or finance.
  2. **Clarity**: Rules focus on attributes most relevant to decision-making, offering actionable insights.
  3. **Efficiency in Decision-Making**: The algorithm creates a structured rule set, which simplifies the process of classifying or predicting new instances.
  4. **Flexibility**: The algorithm is widely used in fields such as machine learning, data mining, and artificial intelligence [155], [156], [157], [158], [158], [159], [160]. Applications include fraud detection, credit risk evaluation, and medical diagnostics. For instance, SCA has been employed to predict case numbers and categorize COVID-19 patients using real-world X-ray datasets [161].

## 4. The proposed rule-learning approaches

The proposed mechanism for personalizing context-aware adaptation rules for multimedia documents integrates rule-based learning to identify relationships between various elements in the log data. This approach leverages our proposed HUD manager as a client-side application, utilizing the data retriever and data analyzer components. As a result, each user can store and analyze historical data on its own devices. The key idea of our proposed approach is to adapt the use of ECLAT, FP-Growth, decision trees (ID3), and sequential covering algorithms to uncover relationships between context elements values and adaptation actions over the HUD database. These relationships are expressed as **"IF** (***Antecedent***) **THEN** (***Consequent***)**"** rules, where:

- **Antecedent**: conjunctive conditions on context values.
- **Consequent**: the corresponding adaptation actions.

These algorithm can be adapted to incorporate an adaptive learning approach, enabling the generation of personalized adaptation rules for multimedia documents. Their simplicity, effectiveness, and interpretability of the resulting rules make them highly suitable for this purpose. Interpretable rules provide valuable insights into the decision-making process, which is crucial for transparency and effective rule induction. These algorithms are particularly suitable for our study for two key reasons:

1. **Efficiency**: The rule-learning algorithm must run continuously as the HUD is generated each time the context changes. This requires a faster algorithm that can operate periodically.
2. **Resource Adaptability**: These algorithms must function on devices with varying computational capabilities, ranging from resource-constrained devices to those with high computational power. This necessitates an algorithm that minimizes computational resource consumption.

## 4.1 Logical Structure of log data

The logical structure of log data is presented as shown in Table 5.2, where each entry contains the values of context elements and the corresponding user-performed adaptations actions.

| **Context elements** | | | | **Actions** | | | |
|---|---|---|---|---|---|---|---|
| **$CE_1$** | **$CE_2$** | ......... | **$CE_n$** | **Action 1** | **Action 2** | ......... | **Action m** |
| $V_{11}$ | $V_{21}$ | ......... | $V_{n1}$ | {yes/no} | {yes/no} | {yes/no} | {yes/no} |
| . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . |
| $V_{1k}$ | $V_{2k}$ | ......... | $V_{nk}$ | {yes/no} | {yes/no} | {yes/no} | {yes/no} |

**Table 5.2.** The logical structure of the raw log data.

Table 5.3 provides a typical example of a subset of context element values and the corresponding adaptation actions performed.

| **Context elements** | | | | **Actions** | |
|---|---|---|---|---|---|
| **Battery** | **Network** | **Location** | **Language** | **Speech2Text** | ***Language translation*** |
| Full | WI-FI | Home | Inappropriate | 0 | 1 |
| Medium | Mobile data | Bus | Appropriate | 0 | 0 |
| Low | Mobile data | Public place | Inappropriate | 1 | 1 |

**Table 5.3.** Typical example of context elements values and adaptation actions.

## 4.2 Rule-learning process based on the ECLAT Algorithm

The rule learning process proposed in this Section is based on the ECLAT algorithm, as outlined in the following steps:

***ECLAT algorithm for the personalization of adaptation rules***

**Input:**

- ***logfile***: Contains context values and adaptation actions.
- ***minsup***: Minimum support threshold.

**Output:**

- ***Rules***: A set of adaptation rules in the form *"IF (Antecedent) THEN (Consequent)"*.

***Procedure Extract_Rules:***

1. Parse ***logfile*** into a dataset $DB$:

   Each row represents a transaction containing context elements values and adaptation actions.

   Separate context values as potential antecedents and adaptation actions as potential consequents.

2. Initialize $F\_items = \{\}$ (set of all frequent itemsets).
3. Identify single-item frequent sets:

   For each item $i$ (context value or action) in $DB$:

   a. $Compute\ T(i) = \{transaction\ IDs\ containing\ i\}$.

   b. $If\ |T(i)| \geq \boldsymbol{minsup}$:

   $Add\ \{i\}\ to\ F\_items$.

   $Add\ \{i, T(i)\}$ to initial equivalence class.

4. Perform depth-first search (DFS) traversal:

   $Call\ DFS(\{\}, equivalence\_class)$

5. Generate adaptation rules from frequent itemsets:

   $For\ each\ frequent\ itemset\ FI\ in\ F\_items$:

   Split $FI$ into antecedent (A) and consequent (C):

   - $A = \{context\ values\ in\ FI\ \}$
   - $C = \{adaptation\ actions\ in\ FI\ \}$

   If both $A$ and $C$ are non-empty:

   Add rule "*IF* $(A)$ *THEN* $(C)$" to ***Rules***.

***Procedure*** $DFS(current\_itemset, equivalence\_class)$:

For each item $i$ in equivalence_class:

$New\_itemset = current\_itemset \cup \{i\}$.

$New\_tids = intersection\ of\ T(current\_itemset)\ and\ T(i)$.

$If\ |New\_tids| \geq minsup$:

$Add\ New\_itemset\ to\ F\_items$.

$Add\ New\_itemset\ to\ a\ new\ equivalence\ class$.

Call $DFS(New\_itemset, new\ equivalence\_class)$.

***End Procedure***

## 4.3 Rule-learning process based on the FP Growth Algorithm

The rule learning process proposed in this Section is based on the FP-Growth algorithm, as outlined in the following steps:

***FP-Growth algorithm for the personalization of adaptation rules***
**Input:**
- ***logfile***: Contains context values and adaptation actions.
- ***minsup***: Minimum support threshold.

**Output:**
- ***Rules***: A set of adaptation rules

***Procedure*** $FP - Growth(\boldsymbol{logfile}, \boldsymbol{minsup})$

1. Parse ***logfile*** into a dataset $\boldsymbol{DB}$:
   Each row represents a transaction containing context values (antecedents) and adaptation actions (consequents).
   Separate context values as potential antecedents and adaptation actions as potential consequents.
2. Scan D and calculate the frequency of each item (context values and actions).
3. Remove items with $support < minsup$.
4. Sort remaining items in each transaction in descending order of frequency.
5. Construct the FP-tree:
   Create the root of the tree (null).
   For each transaction in $\boldsymbol{DB}$:
      Add items to the tree in order (from step 4).
      Update node counts and maintain pointers to the same items (header table).
6. Extract adaptation rules from the FP-tree:

Call $RecursiveFP(Tree, HeaderTable, \emptyset)$.

***Procedure*** $RecursiveFP(Tree, HeaderTable, Prefix)$:
   For each item $i$ in $HeaderTable$ (processed in reverse order):
      $New\_prefix = Prefix \cup \{i\}$.
      Construct the conditional pattern base for $i$:
         Traverse paths ending in $i$ to extract prefix paths and their counts.
      Construct the conditional FP-tree for $i$ using the conditional pattern base.
      If the conditional FP-tree is not empty:
         Call $RecursiveFP(Conditional\_Tree, Conditional\_HeaderTable, New\_prefix)$.
      Generate rules:
         Split $New\_prefix$ into antecedent (A) and consequent (C):
         $A = \{context\ values\ in\ New\_prefix\}$.
         $C = \{adaptation\ actions\ in\ New\_prefix\}$.
      If both $A$ and $C$ are non-empty:
         Add rule "*IF* $(A)$ *THEN* $(C)$" to ***Rules***.

***End Procedure***

## 4.4 Rule-learning process based on sequential covering algorithm

The rule learning process proposed in this Section is based on the sequential covering algorithm, as outlined in the following steps:

**Sequential Covering Algorithm**
**Input**:
$\boldsymbol{Log\_data}$: Logfile containing context values and corresponding adaptation actions
$A$: Set of all attributes.
$minSupport$ : Minimum acceptable accuracy for a rule.
**Output**: $R$: A set of rules that classify the instances in $\boldsymbol{Log\_data}$.
***Procedure:***
**Function** $SequentialCovering(\boldsymbol{Log\_data}, A, minSupport)$
Initialize an empty rule set $R = \{\}$.
While $\boldsymbol{Log\_data}$ is not empty:
Learn a new rule $r$ using the function $LearnOneRule(\boldsymbol{Log\_data}, A)$.
If $Accuracy(r) \leq \ minSupport$, stop.
$R = R \cup \{r\}$.
Remove instances from $\boldsymbol{Log\_data}$ that are covered by $r$.
Return $R$.
**Function** $LearnOneRule(\boldsymbol{Log\_data}, A)$:
Initialize an empty rule $r$.
While $r$ does not classify a single adaptation action of instances in $\boldsymbol{Log\_data}$:
Evaluate all possible attributes (context elements) and their values.
Select the attribute that provides the highest **Information Gain** or **Support**
Add the best attribute-value condition to the rule $r$.
For each instance in $\boldsymbol{Log\_data}$, check whether it satisfies the condition rule $r$.
Return $r$.
***End Procedure***

The sequential learning algorithm shown in the pseudo-code above is used to learn rules by considering one adaptation action at a time. For each adaptation action *Ai*, we search for rules that exclusively cover tuples from this action and not any other adaptation actions. The learning process includes the following operations:

- **Initialization**: This step has the same role as in the basic version of the sequential covering algorithm, starting with an empty rule set.
- **Rule Generation**: The algorithm selects an uncovered adaptation action from the HUD and generates a rule to cover it.
- **Rule Refinement**: The algorithm iteratively refines a generated rule by adding conditions to it.

- **Rule Evaluation**: The algorithm evaluates the quality of a given rule using a predefined evaluation metric. If the rule meets the quality threshold, it is added to the adaptation rule set. Otherwise, the process continues to refine the rule.
- **Stopping Criterion**: The process continues to iteratively generate, refine, and evaluate each rule until a certain stopping criterion is met. Once the stopping criterion is satisfied, the rule set is considered complete.

The sequential covering algorithm allows the rule set to evolve and improve over time based on the content of the HUD. This iterative process of learning and updating the rules can lead to better classification, especially in dynamic environments where contextual data change frequently over time.

## 4.5 Rule-learning process based on decision tree algorithm (ID3)

The ID3 decision tree algorithm is well-suited for generating adaptation rules in multimedia documents contexts as it uses information gain to iteratively select the most relevant attributes, creating a tree that represents clear and interpretable decision rules. These rules can classify context elements and predict corresponding adaptation actions, providing transparency and insights into the decision-making process. This makes ID3 an effective tool for scenarios requiring both accuracy and explainability in rule generation. The following algorithm represent the pseudo code for our proposed ID3 decision tree algorithm:

***Information Gain Pseudocode***

$InformationGain(Log\ File, TargetAdaptationAction, Context\_attribute)$

**Gain** = null

for $value$ in $ContextAttributeValues(Log\ File, Context\_attribute)$:

$$\boldsymbol{Gain}(\text{Target\_adaptation\_action}, \text{Context\_attribute}) = Entropy(Target\_adaptation\_action) - \sum_{Av\in(AttributeValues)} \frac{|AttEx_{Av}|}{|TotalAttN|} Entropy(Context\_attribute_{Av})$$

return **Gain**

Where: $TotalAttN$ = Total Attribute number

$AttEx$ = Number of times the attribute value equal to "yes" or "true"

***Entropy Pseudocode***

$Entropy\ (Target\_adaptation\_action)$

$Entropy\ =\ 0$

$$Entropy(\text{Target\_adaptation\_action}) = Entropy(\text{yes}, \text{no}) = -\frac{PositiveValue}{Total}\log_2\frac{PositiveValue}{Total} - \frac{NegativeValue}{Total}\log_2\frac{NegativeValue}{Total}$$

**Return** $Entropy$

***ID3 Pseudo code:***

**Input:**

- $LogFile$: A dataset containing context values and corresponding adaptation actions.
- $TargetAdaptationAction$: The adaptation action to be predicted.
- $Context\_attributes$: A list of context attributes that can be used for splitting.

**Output:**

- $\boldsymbol{Decision\_Tree}$: A tree.

***Procedure ID3***($Log\ File, TargetAdaptationAction, Context\_attributes$)

Create a root node for the tree.

If all rows in $LogFile$ have the same adaptation action $A_i$:

Return (Root node = $A_i$);

If $Context\_attributes$ is empty:

Root = the most frequent adaptation action in $LogFile$

Return the root.

Else: Calculate the Information Gain for each attribute in $Context\_attributes$.

Select the attribute $C_i$ with the highest Information Gain.

Set $C_i$ as the decision attribute for the root node.

For each value $v$ of attribute $C_i$:

Create a branch below the root node for value $v$.

Let $SubsetDv$ be the subset of $LogFile$ where $C_i = v$.

If $SubsetDv$ is empty:

Attach a leaf node to this branch with the most common class in $LogFile$.

Else:

Call ID3($SubsetDv$, $TargetAdaptationAction$, $Context\_attributes - C_i$).

Attach the resulting subtree to the branch.

Return the root node as the $\boldsymbol{Decision\_Tree}$

## 5. Experiments, results and discussion

To validate our proposal, we implemented a prototype that adapts multimedia documents (web pages containing pedagogical content) according to users' contextual changes, while storing their historical data in a log file. The following scenario illustrate typical cases of how HUDs are used to personalize adaptation rules through the rule learning mechanisms.

*"Sarah is a graduate student pursuing a degree in environmental science. In the morning, while commuting to the university, she listens to podcasts related to her field of study and asks the system to transcribe the audio content into text so she can review it on her phone. During her lunch break at the campus café, Sarah quickly reviews lecture notes on her tablet and requests the system to highlight important points for easier reading. In the afternoon, while waiting for her next class in a quiet corner of the library, she watches a recorded lecture on her laptop and asks the system to slow down*

***the pace of the video to match her preferred learning speed. In the evening, back at her apartment, Sarah reviews the day's lessons while cooking dinner and requests the system to summarize complex concepts and display relevant diagrams on her smart TV."***

Table 5.4 presents a typical example of Sarah's log file, which includes the most frequent context element values and the corresponding adaptation actions that occur when Sarah interacts with others in her daily life. The log file captures various context parameters such as location, and activity, along with the adaptation actions, such as changing the display settings, adjusting the audio output, or altering content presentation based on Sarah's current context. For illustration purposes, we have selected only a small portion of the log file, focusing on a few key interactions, to highlight how the system adapts in real-time based on Sarah's HUD. This small subset demonstrates how the system tailors its responses to Sarah's needs throughout different moments of her day.

| Noise | Battery level | Screen size | Smart_TV | Network | Location | Speech to_text | Video-in-smart-TV |
|---|---|---|---|---|---|---|---|
| Quiet | Low | Narrow | Available | WiFI | Home | 0 | 1 |
| Quiet | Full | Wide | Not Available | WiFI | Laboratoire | 1 | 0 |
| Quiet | Medium | Narrow | Available | WiFI | Meeting room | 1 | 1 |
| L Noisy | Medium | Narrow | Not Available | Mobile data | Public Place | 0 | 0 |
| Quiet | Medium | Wide | Not Available | WiFI | Laboratoire | 1 | 0 |
| Noisy | Medium | Narrow | Not Available | Mobile data | Public Place | 1 | 0 |
| Noisy | Low | Narrow | Not Available | Mobile data | Public Place | 1 | 0 |
| Quiet | Full | Wide | Available | WiFI | Meeting room | 1 | 1 |
| Quiet | Medium | Narrow | Available | Mobile data | Meeting room | 0 | 1 |
| Noisy | Medium | Narrow | Not Available | Mobile data | Public Place | 1 | 0 |
| Noisy | Medium | Narrow | Not Available | Mobile data | Public Place | 1 | 0 |
| Quiet | Medium | Wide | Available | WiFI | Home | 1 | 0 |
| Quiet | Medium | Wide | Not Available | Mobile data | Meeting room | 1 | 0 |
| Noisy | Low | Narrow | Not Available | Mobile data | Public Place | 1 | 0 |
| L Noisy | Full | Wide | Available | WiFI | Home | 0 | 1 |

**Table 5.4.** Typical example of small portion of the log file.

## 5.1 Results of the Attribute-Based Split Algorithms

Table 5.5 shows the results of the attribute-based split algorithms (ID3 and Sequential Covering)

| Algorithm | Rule Level | Generated Rules | Description |
|---|---|---|---|
| ID3 (Decision Tree) | **One-Level Rules** | – if (Battery level = Full) then (Speech_to_text = 1)<br>– if (Screen size = Narrow) then (Video-in-smart-TV = 1)<br>– if (Location = Home) then (Video-in-smart-TV = 1) | Single-attribute combinations that directly map to adaptation actions. |
| | **Two-Level Rules** | – if (Battery level = Low and Network = Mobile data) then (Speech_to_text = 1)<br>– if (Battery level = Full and Network = WiFi) then (Speech_to_text = 1)<br>– if (Location = Home and Screen size = Wide) then (Video-in-smart-TV = 1) | Combinations of two attributes for more refined adaptation decisions. |
| | **Three-Level Rules** | – if (Battery level = Full and Screen size = Narrow and Network = WiFi) then (Video-in-smart-TV = 1)<br>– if (Noise = Quiet and Battery level = Medium and Location = Meeting room and Smart_TV = Available) then (Video-in-smart-TV = 1) | Complex rules involving three or more attributes to maximize personalization. |
| Sequential Covering | **One-Level Rules** | – if (Battery level = Full) then (Speech_to_text = 1)<br>– if (Location = Home) then (Video-in-smart-TV = 1) | Single-step rules that provide initial coverage for adaptation actions. |
| | **Two-Level Rules** | – if (Battery level = Full and Location = Meeting room) then (Speech_to_text = 1)<br>– if (Network = WiFi and Battery level = Full) then (Speech_to_text = 1)<br>– if (Battery level = Medium and Network = WiFi) then (Speech_to_text = 1) | Two-attribute rules refined iteratively to improve coverage. |
| | **Three-Level Rules** | – if (Battery level = Full and Screen size = Wide and Location = Home) then (Video-in-smart-TV = 1)<br>– if (Noise = Quiet and Network = WiFi and Screen size = Wide) then (Speech_to_text = 1)<br>– if (Battery level = Full and Smart_TV = Available and Screen size = Narrow) then (Video-in-smart-TV = 1) | Multi-level rules that capture deeper relationships between context elements. |

**Table 5.5.** Results of the attribute-based split algorithms (ID3 and Sequential Covering)

## 5.2 Results of the Frequent Itemset-Based Split Algorithms

Table 5. shows the results of the frequent itemset-based algorithms (Eclat and FP-Growth).

| Algorithm | Frequent Itemsets/Patterns | Rule Level | Generated Rules |
|---|---|---|---|
| **Eclat** | {Noise = Quiet, Battery = Medium, Video-in-smart-TV} (Support = 3) {Noise = Noisy, Battery = Medium, Speech_to_text} (Support = 2) Battery = Medium, Smart_TV = Available, Video-in-smart-TV} (Support = 2) | **One-Level Rules** | – if (Location = Home) then (Video-in-smart-TV) |
| | | **Two-Level Rules** | – if (Battery level = Full and Screen size = Wide) then (Video-in-smart-TV)<br>– if (Battery level = Medium and Network = WiFi) then (Speech_to_text) |
| | | **Three-Level Rules** | – if (Battery level = Full and Screen size = Wide and Location = Home) then (Video-in-smart-TV) |
| **FP-Growth** | {Noise = Quiet, Battery = Medium, Location = Meeting room, Video-in-smart-TV} (Support = 2) {Noise = Noisy, Battery = Medium, Location = Public Place, Speech_to_text} (Support = 2) {Battery = Full, Smart_TV = Available, Video-in-smart-TV} (Support = 1) | **One-Level Rules** | – if (Battery level = Full) then (Speech_to_text)<br>– if (Location = Home) then (Video-in-smart-TV) |
| | | **Two-Level Rules** | – if (Battery level = Medium and Network = WiFi) then (Speech_to_text)<br>– if (Location = Home and Network = WiFi) then (Video-in-smart-TV)<br>– if (Battery = Full and Smart_TV = Available) then (Video-in-smart-TV) |
| | | **Three-Level Rules** | – if (Battery level = Full and Screen size = Wide and Location = Home) then (Video-in-smart-TV)<br>– if (Noise = Quiet and Battery level = Medium and Location = Meeting room) then (Video-in-smart-TV)<br>– if (Noise = Noisy and Battery level = Medium and Location = Public Place) then (Speech_to_text) |

**Table 5.6.** Results of the frequent itemset-based algorithms (Eclat and FP-Growth)

**Summary of adaptation rules results**

These adaptation rules are one-level and multi-level rules generated using the different algorithms: ***ID3*** and ***Sequential Covering*** algorithms focus on ***attribute-based splits***, while ***Eclat*** and ***FP-Growth*** algorithms focus on ***frequent itemsets***. By analyzing the frequency and relevance of these rules in the log data, this can make the system's behavior more effectively, personalizing it to user needs in dynamic environments.

– **ID3**: Focuses on creating decision trees, often it generates rules based on single attributes or combinations of attributes with clear decision boundaries.

- **Sequential Covering**: Generates rules sequentially, focusing on covering the instances of a specific adaptation action and refining them iteratively.
- **Eclat**: Focuses on finding frequent patterns in the data, generating rules based on these patterns. Its primary strength is in finding co-occurrence of context attributes.
- **FP-Growth**: Like Eclat, it finds frequent patterns and generates rules based on these patterns. However, it is optimized for large datasets.

## 5.3 Discussion

Table 5.7 present the comparison between the attribute-based split algorithms (ID3 and Sequential Covering) while Table 5.8 present the comparison between the frequent itemset-based algorithms (Eclat and FP-Growth). Those tables provide some advantages and limitations of each algorithm and their suitability for extracting adaptation rules in the context of personalized multimedia documents adaptation:

| **Algorithm** | **ID3 (Decision Tree)** | **Sequential Covering** |
|---|---|---|
| **Advantages** | – Focuses on maximizing information gain<br>– Faster to generate initial rules<br>– Easy to interpret and understand<br>– Maximizes information gain for clear decision paths<br>– Works well with categorical data | – Flexible and adaptable<br>– Produces high-quality rule sets<br>– Iterative rule refinement<br>– Adapts better to dynamic contexts |
| **Limitations** | – Prone to overfitting with noisy or large datasets | – Slower convergence due to iterative learning<br>– Computationally expensive for large datasets |
| **Best Use Case** | Best for **medium-sized categorical datasets** where clear decision-making paths are needed | Suitable for **dynamic, evolving systems** that require continuous learning and rule refinement |

**Table 5.7.** Comparison between the attribute-based split algorithms.

| **Algorithm** | **Eclat** | **FP-Growth** |
|---|---|---|
| **Advantages** | – Efficient in finding frequent itemsets<br>– Focuses on high-frequency rules<br>– Generates rules based on common context combinations<br>– Simpler data structure | – Scalable and efficient for large datasets due to FP-tree structure<br>– Generates frequent itemsets efficiently<br>– Handles both sparse and dense datasets<br>– Generates more comprehensive rules |
| **Limitations** | – Limited rule complexity<br>– Not optimized for decision-making clarity<br>– May miss rare patterns | – Focuses mainly on frequent patterns, missing rare or complex rules<br>– May generate too many redundant rules |
| **Best Use Case** | Ideal for large datasets where identifying high-frequency patterns is crucial | Suitable for large-scale systems with abundant context data, focusing on efficiency and pattern discovery |

**Table 5.8.** Comparison between the frequent itemset-based algorithms.

Table 5.9. shows the overall comparison between association rules algorithms involves in our case study. We could say that each algorithm has strengths and weaknesses, but the choice of the best algorithm depends on the specific requirements of the multimedia adaptation system.

| **Algorithm** | **Strengths** | **Weaknesses** | **Best Use Case** |
|---|---|---|---|
| **ID3** | Simple, interpretable rules | Prone to overfitting | Small to medium-sized datasets with clear attribute splits |
| **Sequential Covering** | Produces high-quality rules iteratively | Slower convergence | Adaptive systems with evolving user interactions |
| **Eclat** | Efficient for frequent itemset generation. | Limited rule complexity | Systems focused on identifying frequent context combinations |
| **FP-Growth** | Scalable and efficient | May generate too many rules | Large-scale systems with frequent and complex context patterns |

**Table 5.9.** Summary of the overall comparison.

## 6. Conclusion

Data mining algorithms, such as FP-Growth, Eclat algorithm, decision tree, and sequential covering algorithm, play a vital role in managing historical user data for context-aware multimedia adaptation. These algorithms enable the extraction of patterns, preferences, and behavior insights from the log data, facilitating personalized experiences for users.

Through this chapter we proposed four association rules mining algorithms typically FP-Growth, Eclat algorithm, decision tree, and sequential covering algorithm to extract useful relationship from the log file. The experimental results shows that the ID3 is best for simple systems where interpretability is the priority and the data is mostly categorical. Sequential Covering is ideal when you need to adapt and refine rules iteratively, especially for dynamic systems that evolve over time. FP-Growth is highly suitable for large-scale systems where you need to find frequent patterns quickly and efficiently. Eclat is great when identifying frequent patterns is your primary goal, especially in large datasets. Thus, the choice of the best algorithm depends on the specific system requirements: ID3 for simplicity, Sequential Covering for adaptability, FP-Growth for scalability, and Eclat for frequent pattern mining.

However, these algorithms are designed for client-side applications, which focus on processing individual user log files. This approach limits the potential for users to benefit from shared experiences, and collaborative learning derived from collective data. To overcome this limitation, the next chapter introduces a server-side application powered by latent semantic analysis (LSA). This approach enables the aggregation and analysis of data across multiple users, fostering shared knowledge and uncovering broader patterns that can enhance personalization. By leveraging LSA, the system can provide more intelligent, adaptive, and community-driven multimedia adaptation experiences.

# Chapter VI

# Latent Semantic Analysis for the Classification of Similar Contextual Situations and User Behaviors

## Summary

# CHAPTER VI

## Latent semantic analysis for the classification of similar contextual situations and user behaviors

### 1. Introduction

Latent Semantic Analysis (LSA), a powerful technique in the field of Natural Language Processing (NLP) and information retrieval, has gained significant attention for its ability to uncover meaningful relationships between terms and documents. By reducing the complexity of data while preserving core information, LSA provides insights into latent structures within large datasets. Grounded in strong mathematical principles, LSA enables the discovery of connections by analyzing the contextual factors surrounding the data. The idea of LSA is to transform text items, such as words and documents, into a latent semantic space. In this space, the proximity of vectors indicates that the original items share comparable semantic features.

LSA has been widely used in various text processing tasks, particularly in e-learning environments for instructional purposes. In the context of learning within context-aware pervasive systems, LSA offers an effective method for extracting valuable insights from learners' log data, shedding light on their interaction patterns. While several analysis techniques have been applied to instructional contexts, few have focused on non-instructional contexts. Instructional context is undeniably important, but learners' behaviors and styles do not always align with the environmental requirements. This chapter, therefore, focuses on leveraging LSA to analyze learner behaviors in context-aware pervasive systems, particularly with regard to non-instructional contexts. The objective is to present an efficient approach for uncovering hidden semantic structures and extracting meaningful patterns from HUD, eventually enhancing the learning experience.

The chapter begins by introducing the fundamentals of LSA and its typical applications. Next, it details the steps of the proposed LSA-based analysis approach. Finally, it demonstrates the effectiveness of our approach through numerical results and discussions.

## 2. Latent Semantic Analysis (LSA)

Developed in the late 1980s, latent semantic analysis (LSA) is a well-established approach in natural language processing and information retrieval. It gained prominence during the 1990s and early 2000s as the availability of digital text corpora expanded, enabling its broader adoption in addressing large-scale text analysis tasks. In LSA-based processes, latent structures in extensive text corpora are uncovered by identifying patterns and correlations between terms and documents, thereby improving the retrieval, organization, and understanding of textual information [162]. Its main principle involves reducing the high-dimensional term-document matrix into a lower-dimensional latent space. This is achieved by utilizing the implicit higher-order structure in the associations between terms and documents, enabling the discovery of approximate semantic relationships within the data [163]. LSA involves a three-step process:

1) **Dimensional model creation:** A singular value decomposition (SVD) is performed on the data matrix to create a dimensional model. SVD outputs a diagonal matrix containing singular values and two orthogonal rotation matrices. The number of dimensions in the data corresponds to the number of non-zero singular values.
2) **Dimensionality reduction:** The number of dimensions is reduced by removing the smallest singular values and the corresponding rows and columns of the rotation matrices
3) **Data reconstruction:** The reduced matrices are combined to approximate the original matrix. This reconstruction allows documents to gain frequencies for terms absent in the original data, enabling broader semantic associations.

The LSA process is conceptually similar to principal components analysis and factor analysis, widely used in social and behavioral sciences. These techniques share a focus on reducing data complexity while retaining key patterns. Figure 6.1 provides a schematic overview of LSA, illustrating how a query is compared with a corpus [163].

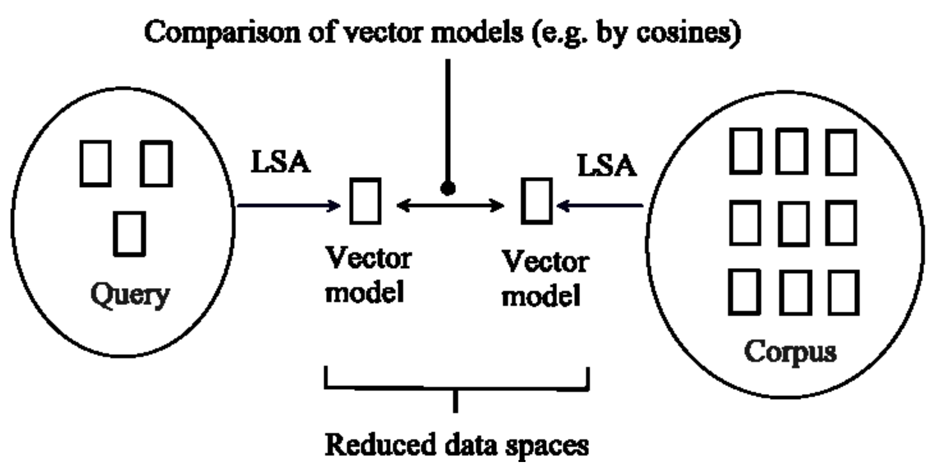



**Figure 6.1.** Overview of the LSA process.

## 2.1 Singular Value Decomposition (SVD)

SVD is a key mathematical approach in LSA; it decomposes a term-document matrix into linearly independent components, which can be interpreted as vectors derived from the raw data. This transformation significantly reduces the dimensionality of the original matrix, enabling a more compact and efficient representation [164]. Beyond dimensionality reduction, SVD provides another critical benefit: it enhances semantic relationships between documents. Even if identical terminology is absent, SVD captures underlying patterns, making documents on similar topics more comparable in the reduced-dimensional space. This property is especially useful for discovering latent associations in text corpora [164].

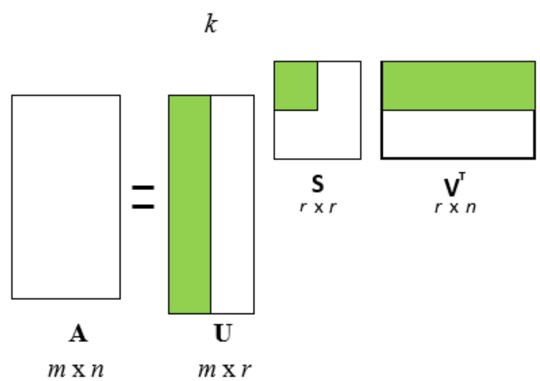



**Figure 6.2.** Singular values decomposition technique.

As shown in Figure 6.2 and equation (1), the original matrix $A$ is divided into three matrices ($U, S \text{ and } V^T$).

$$A = U_{m\times r}\, S_{r\times r}\, V^T_{r\times n} \qquad (1)$$

Where:

- $A$ is the initial matrix of dimension $m \times n$, represented by $m$ words and $n$ documents.
- Matrix $U$ is an orthogonal matrix consisting of vectors of terms, where the columns are eigenvectors of $A * A^T$.
- $V$ is an orthogonal matrix of document vectors, with columns that are eigenvectors of the vector space $A^T * A$.
- An eigenvalue-containing diagonal matrix denoted as $S$.

## 2.2 Document-Term Matrix

LSA begins with a collection of text documents and the words they contain. The word frequencies in the documents are stored in a matrix known as the term-document matrix. A document is represented by a column vector of term frequencies (document vector), while a term is represented by a row vector of frequencies across documents (term vector). The order of words in the document is irrelevant, LSA does not consider syntactic information.

The dimensions of the term-document matrix, referred to as $A$, are reduced in two phases. The first phase involves applying SVD to the data matrix. In SVD, matrix $A$ is decomposed into three components: $A = U \times S \times V$, where $U$ and $V$ are orthonormal matrices while $S$ is a diagonal matrix containing singular values. The number of singular values greater than zero defines the number of dimensions in the matrix. Consider the values in $S$ as delineating orthogonal axes in a high-dimensional space, with the values representing the lengths of these axes.

To reduce dimensionality, only the longest axes are retained by eliminating the corresponding rows and columns from matrices $S$, $U$, and $V$. The original matrix is then reconstructed using the reduced matrices. In the reconstructed matrix, a document vector may have a frequency for a word $w$ that was absent in the original document. A search for "all documents about word $w$" may return documents that do not contain the keyword $w$, but instead phrases commonly associated with word $w$. The reconstructed term-document matrix can be used to compute several metrics, including the correlation between document vectors. A higher correlation indicates greater similarity between the documents.

## 3. Applications of LSA

LSA has been widely used in text mining and natural language processing to extract relationships between terms and documents. Its versatility spans disciplines such as information retrieval, artificial intelligence, psychology, cognitive science, education, and business systems. The following sections highlight key applications of LSA in e-learning and text summarization.

### 3.1 LSA in E-learning

In e-learning, LSA enhances educational tools by improving information retrieval, contextual understanding, and personalized learning. Studies have demonstrated its utility in analyzing learner behavior, tracking conceptual development, and automating essay grading.

- **Analyzing learner behavior:** A study by [165] applied LSA to log files from online learning platforms. By uncovering latent relationships between user actions, the researchers identified behavioral patterns helping to better understand student engagement, predict learning outcomes, and provide insights that can enhance personalized learning experiences. This research inspired by the use of LSA in analyzing log files for workflow discovery aiding in the optimization and management of business processes [164].
- **Tracking conceptual development:** CONSPECT, a tool introduced and evaluated in [165], integrates LSA with network analysis to monitor learners' conceptual growth.

LSA examines the meaning within the textual data generated by learners during their learning process, while network analysis offers a visual framework for exploring the semantic structures uncovered by LSA. By visualizing semantic structures, CONSPECT helps tutors and learners understand the progression of knowledge with a minimal workload.

- **Automated essay grading:** A method proposed in [166] uses LSA to assess the semantic content of student essays in open-mind text answer. By preprocessing text and leveraging LSA's ability to analyze latent meaning, this approach offers a scalable and fair alternative to manual grading.

## 3.2 LSA in Text Summarization

LSA has been extensively studied for automatic text summarization, where it identifies key sentences that represent the core meaning of documents.

- **Summarizing news articles:** Research [167] applied LSA to automatic summarization of news articles, extracting representative sentences by analyzing latent semantic structures (i.e., identifying key concepts and relationships between words and sentences). This approach prioritizes underlying meanings rather than surface-level features [167].
- **Political articles summarization:** Authors in [168] presented a method for automatic text summarization that leverages machine learning and LSA. The idea consisted in deriving the semantic matrix of a document or a corpus and uses semantic sentence representation to construct a semantic text relationship map. The proposal was validated on a corpus composed of 100 political articles.
- **Legal document summarization:** A tool designed for Philippine court cases [169] leverages LSA to produce concise summaries of lengthy legal texts, saving time for legal professionals researchers, and the general public and improving access to critical information. The underlying meaning of texts was uncovered by analyzing relationships between terms and documents, allowing the system to identify and prioritize key sentences relevant to the case.
- **Arabic text summarization:** A system proposed in [170] addresses the challenges of Arabic text processing, such as complex morphology and rich diacritics. By applying LSA, the authors created summaries that matched human-generated ones in relevance and informativeness.

Table 6.1 reviews some of application domain of LSA.

| Domain | Description | Examples of Applications |
|---|---|---|
| Education [5] [165] [166] [171] | Analyzes and evaluates learning materials, student progress, and conceptual understanding. | - Automatic essay grading<br>- Personalized learning systems<br>- Paper evaluation |
| Information retrieval [172] [173] | Enhances search engines and document retrieval by identifying semantic relationships among terms. | - Search engine optimization<br>- Document ranking |
| NLP , [169], [170], [175] | Improves the understanding of text semantics for various language-related tasks. | - Text summarization<br>- Machine translation<br>- Sentiment analysis<br>- Text categorization |
| Business Intelligence [164], [176] | Extracts insights from textual data to support decision-making and strategy development. | - Feedback analysis<br>- Market research<br>- Topic modeling in reviews |

**Table 6.1.** Summary of LSA application domain.

## 4. Proposed Approach

This section presents our final contribution, where LSA is employed to analyze HUD generated from multimedia document adaptation processes. The goal is to extract user behaviors related to remote learning in context-aware pervasive systems.

### 4.1 Motivation and Objective of the Contribution

In context-aware pervasive environments, learning activities extend beyond traditional boundaries. Therefore, the collection and analysis of learners' historical data – including preferences and interactions within their environment – is crucial for enhancing learning experiences. LSA provides an effective approach for extracting knowledge from these data and obtaining relevant information regarding learners' interaction patterns for several reasons:

- **Latent semantic relationships with instructional context:** LSA can uncover hidden semantic relationships between learner actions and instructional contexts. This allows for the identification of behaviors related to frequently studied topics and knowledge acquisition pathways.
- **Latent semantic relationships with non-instructional context:** LSA can also uncover semantic relationships between learner actions and non-instructional contexts. This helps identify how learners interact with learning materials considering environmental conditions.
- **Personalization of learning environments:** By analyzing the semantic structures in learners' interactions, LSA facilitates the adaptation of feedback and recommendations to individual needs, ensuring timely and relevant support for learners.

- **Identification of collaborative learning patterns:** LSA enables the discovery of collaborative learning patterns by analyzing semantic similarities in log data from various learners. This helps determine effective group dynamics and collaboration styles, guiding the design of group activities to leverage these strengths.

In summary, LSA enables the efficient extraction and interpretation of learner log data in ubiquitous learning environments. This capability enhances personalized learning, supports data-driven decision-making, and fosters a more interactive and effective educational experience within a flexible and interconnected learning environment.

Many studies have been conducted with respect to instructional context. However, few of them have considered non-instructional contexts. While instructional context is important, learners' styles do not always fit the environmental requirements, including user and application needs. For instance, consider a learner who adopts a visual cognitive style. If this learner is driving, where the eyes and hands cannot be properly used, the system could suggest converting textual learning objects into audio content and using voice commands to ensure compatibility with the environment. Starting from this standpoint, our contribution focuses on using LSA to analyze learner behaviors in context-aware pervasive systems, ***specifically in relation to non-instructional context***. The aim is to provide a comprehensive classification of both *similar contextual situations* and *learner clusters*, which helps enhance content delivery, context understanding, and personalized user experiences. The rationale behind this is that LSA is a well-suited technique for applications that uncover relationships between terms and documents. Similarly, our approach treats each learner's historical data as a textual document, with context elements and corresponding actions serving as words. Thus, our proposal attempts to offer a robust method for uncovering hidden semantic structures and extracting meaningful patterns from HUD. It involves server-side management of the HUD, particularly for the analysis task. This fits with ubiquitous learning architectures, which mainly rely on servers to store and deliver learning materials (refer to Section 7 of Chapter 2).

## 4.2 Details of the Proposed LSA-Based Analysis Approach

The proposed LSA-based approach consists of an analysis process comprising three steps: *Pre-LSA analysis*, *LSA analysis* and *Post-LSA analysis*; its general scheme is given in Figure 6.3.

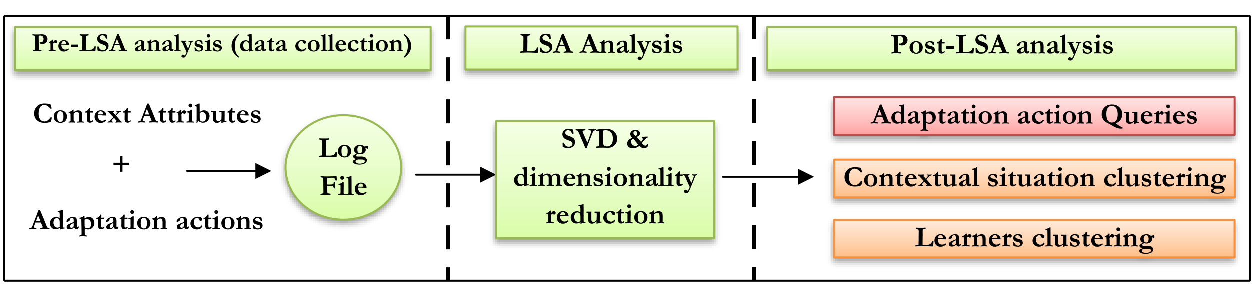



**Figure 6.3.** General scheme of the proposed LSA-based approach.

### 4.2.1 Pre-LSA Analysis (Data Collection) Step

Pre-LSA analysis leverages the HUD manager discussed in Sections 4 and 5 of Chapter 4 to construct a corpus (data collection) using the data stored in the log database. It is performed in three phases: *data retrieval*, *data preprocessing*, and *data segmentation*.

- **Data Retrieval**

The data analyzer invokes the data retriever to load HUD from the database into memory as a log file denoted by *LF*. The HUD consist mainly of the values of the context elements and the corresponding adaptation actions performed by each user according to changes in context over time. Log file *LF* stores the HUD as an aggregation of raw data about all users in a tabular format, so its logical structure takes the form shown in Table 6.2. Each entry (row) *i* in the log data represents an interaction with the learning environment by the corresponding user id (learner): *contextual situations*.

| Context elements | | | | | Actions | | | |
|---|---|---|---|---|---|---|---|---|
| **User ID** | **$C_1$** | **$C_2$** | **....** | **$C_g$** | **$A_1$** | **$A_2$** | **.....** | **$A_h$** |
| $U_i$ | $V_{11}$ | $V_{21}$ | .... | $V_{g1}$ | {yes/no} | {yes/no} | {yes/no} | {yes/no} |
| . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . |
| $U_n$ | $V_{1k}$ | $V_{2k}$ | .... | $V_{gk}$ | {yes/no} | {yes/no} | {yes/no} | {yes/no} |

**Table 6.2.** The logical structure of the raw log data.

Table 6.3 shows a typical example for a subset of context elements values and adaptation actions performed accordingly.

| Context elements | | | | | Actions | |
|---|---|---|---|---|---|---|
| User ID | Battery | Network | Screen size | Location | Speech2Text | Text-summary |
| $U_2$ | Full | Mobile data | Narrow | Public place | Yes | No |
| $U_1$ | Low | Mobile data | Narrow | Bus | Yes | Yes |
| $U_4$ | Full | WI-FI | Wide | Home | No | Yes |

**Table 6.3.** Typical example of context elements values and adaptation actions.

Formally, contextual situations are defined according to the following formalisms:

- Let $CXT = (c_1, c_2, \ldots, c_g)$ be a vector where each element $c_i \in CXT$ corresponds to a context element from log file *LF*; *g* is the number of context elements. The possible values for $c_i$ include all qualitative values for the corresponding context element, as well as to the *null value* (refer to Section 4 of Chapter 3). For example, if $c_i$ represents *battery level*, then *Domain* ($c_i$) = {*null*, *low*, *medium*, *full*}.

- Similarly, let $ACT = (a_1, a_2, \ldots, a_h)$ represents a vector of adaptation actions, where $h$ is the total number of actions. Each $a_{j=1..h} \in ACT$ can be 0 or 1, indicating whether adaptation action $a_j$ is considered for execution.
- Let $CS = \{cs_1, cs_2, \ldots, cs_k\}$ be the set of all context situations in log file *LF*. Each contextual situation $cs_{v=1..k} \in CXT \times ACT$. For instance, context situation $cs_1$ from Table 5.3 is represented as follows: $cs_1$ = ($U_1$, Full, Mobile data, Narrow, Public place, Yes, No), indicating that learner whose *id* = $U_1$, uses a smartphone through mobile data with a full battery and a narrow screen, and is in a public place, while *Speech2Text* adaptation action was executed.

Where $g$ is the number of context elements considered, $h$ is the number of adaptation actions, and $k$ is the number of contextual situations.

- **Data Preprocessing**

Each contextual situation $cs_i \in CS$ contains textual data related to context elements values including user id and the corresponding adaptation actions. To make log data more machine readable and thus easier to process in subsequent steps, contextual situations $cs_{v=1..k} \in CS$ are converted into numerical format, particularly for context elements values. This is achieved by replacing the textual representation of each qualitative value of a context element (which is human-readable) with different numerical values. For instance, if $c_i \in CXT$ represents *battery level*, then *Domain* $(c_i)$ = {*null*, *low*, *medium*, *full*}, and the corresponding numerical representation of this domain would be *Numerical-domain* $(c_i) = \{0, 1, 2, 3\}$. Hence, context situation $cs_1$ from Table 6.3 becomes: $cs_1$ = (1, 3, 1, 0, 0, 1, 0).

- **Data Segmentation**

Log file *LF* is divided into sub-logs $LF_{i=1..n}$ according to user *id*; here $n$ represents the total number of users (learners). Each sub-log file $LF_i$ is considered as a separate document in the corpus. Figure 6.4 illustrates an example of a log file as well as its segmentation into sub-logs.

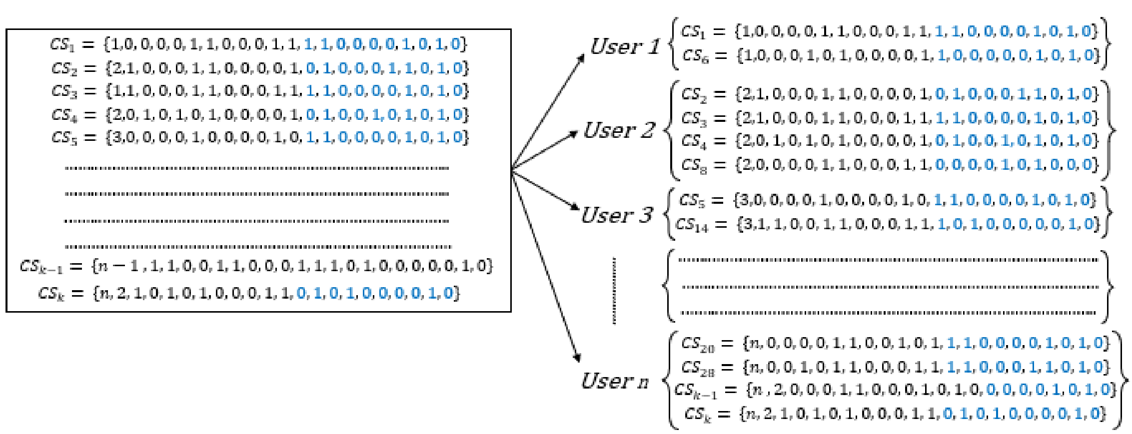



**Figure 6.4.** Partitioning LF into sub-logs.

### 4.2.2 LSA Analysis Step

In LSA analysis step, we apply LSA to the corpus constructed in the previous step. It is performed in four phases: *definition of the dictionary of terms*, *construction of the occurrence matrix*, *decomposition of the occurrence matrix*, and *extraction of the compressed matrix*.

- **Definition of the Dictionary of Terms**

LSA uncovers latent structures in extensive text corpora by identifying patterns and correlations between terms and documents. As the corpus is constructed in the previous step, we need to define the dictionary of terms required for the analysis.

Let $T = \{t_1, t_2, \dots t_m\}$ be the set of $m$ terms. Each term $t \in T$ is represented as pair $(c_i, a_j)$, where $c_i$ is an element of the context elements vector $CXT$, and $a_j$ is an element of the actions vector $ACT$. Semantically, each term $t = (c_i, a_j)$ is interpreted as the occurrence of the value of context element $c_i$ with adaptation action $a_j$ in the same contextual situation in sub-logs $LF_{i=1..n}$, such that $a_j = 1$ (action $a_j$ is executed). It is important to note that the number of terms $m$ depends on the number of actions $h$, the number of context elements $g$, and the cardinality of the domains of definition of each context element in vector $CXT$.

- **Construction of the Occurrence Matrix**

In this phase, the occurrence matrix, denoted by $A$, is constructed by calculating the number of occurrences of each term $t \in T$ in each sub-log $LF_{i=1..n}$ (i.e., number of times that term $t$ occurs in each sub-log $LF_{i=1..n}$), as shown in Table 5.4. Matrix $A$ is of size $(m \times n)$.

| | $LF_1$ **(user 1)** | $LF_2$ **(user 2)** | ……………… | $LF_n$ **(user** $n$**)** |
|---|---|---|---|---|
| $t_1$ | Occurrence $(t_1, LF_1)$ | Occurrence $(t_1, LF_2)$ | ……………… | Occurrence $(t_1, LF_n)$ |
| $t_2$ | Occurrence $(t_2, LF_1)$ | Occurrence $(t_2, LF_2)$ | ……………… | Occurrence $(t_2, LF_n)$ |
| .<br>.<br>. | …………..<br>…………..<br>………….. | …………..<br>…………..<br>………….. | ………………<br>………………<br>……………… | …………..<br>…………..<br>………….. |
| $t_m$ | Occurrence $(t_m, LF_1)$ | Occurrence $(t_m, LF_2)$ | ……………… | Occurrence $(t_m, LF_n)$ |

**Table 6.4.** The structure of the occurrence matrix A.

- **Decomposition of the Occurrence Matrix**

The next phase is to decompose the occurrence matrix $A$ using SVD, as discussed in Section 2, through factorization into three matrices $U$, $S$, and $V$, such that $A = U \times S \times V^T$. Matrix $A$ $(m \times n)$ represents the original occurrence matrix, matrix $U$ $(m \times r)$ is composed of vectors representing latent features of rows (contextual situations), matrix $U$ $(r \times r)$ consists of **singular values** indicating the importance of each latent feature, and matrix $V^T$ $(r \times n)$ is the transposed matrix of sub-log vectors indicating the latent features for columns (users).

- **Extraction of the Compressed Matrix**

We then proceed to generate the compressed matrix *A'* by reducing the rank $k$ of matrix *A* once the factorization step has been completed. A large amount of information is preserved in the reduced matrix *A'*, which also functions as the most accurate approximation of the original occurrence matrix *A*. Matrix *A'* involves eigenvalues according to equation 3:

$$A' = U\,(k \times m) \times S\,(k \times k) \times V\,(k \times n) \qquad (3)$$

### 4.2.3 Post-LSA Analysis Step

The last step leverages compressed matrix *A'* to uncover hidden correlations leading to clustering contextual situations and learners. To do, we calculate the correlation between users and the correlation between the contextual situations. This is performed by determining the angle between pairs of vectors either vertically, denoted by $V1$ ($x_1$, $x_2$), or horizontally, denoted by $V2$ ($y_1$, $y_2$) in order to assess their similarity. The first correlation shows the similarities between users, which is very helpful for finding similar behavior of users and simplify the process of recommending the best adaptation actions for users who has the same behaviors. The second correlation simplifies the understanding of the contextual situations and classify the similar ones, for better recommendation of the useful adaptation actions. Two techniques are commonly used for calculating the similarities: *Cosine* and *Pearson law*s.

- **Cosine Law**

The cosine law is calculated by the formula given in equation 4. The measurement of the cosine takes a value between -1 and 1. The interpretation of the correlation value is as follows:

- The two vectors are completely separated when this value is 0.
- The two vectors have a high degree of similarity when the cosine has a value of 1.
- The two vectors are not aligned when this value is -1.

$$cos(v1, v2) = \frac{v1, v2}{\|v1\|\|v2\|} = \frac{x_1y_1 + x_2y_2}{\sqrt{(x_1^2 + x_2^2)}\sqrt{(y_1^2 + y_2^2)}} \qquad (4)$$

One of the most significant scenarios that might take place is the shift of two vectors that have the same value. When it comes to this case, the similarities that exist between them ought to remain unchanged. This is not something that the cosine law can supply. If vectors $V1$ and $V2$ are shifted by a value τ, the degree of similarity between them will take on a different form.

- **Pearson Law**

Pearson law, whose formula is given by equation 5, is able to preserve the similarity between vectors effectively. Unlike cosine similarity, which is highly affected by the magnitude and position of vectors, Pearson correlation measures similarity based on the relative direction of the vectors rather than their absolute magnitudes. In other words, despite the fact that the scale and location of the vectors that are being compared may change, the correlation measure does not change and is not influenced by the cosine component. A more accurate reflection of the similarities between vectors is provided. This feature make Pearson correlation more suitable for measuring similarity between users or conditions in systems that involve continuous changes or heterogeneous data. Another advantage is when vectors are not equal in size or range, Pearson correlation provides a more reliable measure of their relationship. It ensures that differences in magnitude do not affect the final similarity interpretation. This is why we opt for the use of Pearson law in our work.

$$corr(a,b) = \frac{\sum i(a_i - a)(b_i - b)}{\sqrt{\sum(a_i - a)^2}\sqrt{\sum(b_i - b)^2}} \quad (5)$$

## 5. Experiment, Results and Discussion

This section shows the effectiveness of our proposal through scenarios, test and discussions.

### *5.1* Case Study for Experiment Settings

Due to the lack of datasets, on the one hand, and, the difficulty of providing the necessary hardware resources to scale for a large number of users, on the other hand, our approach is tested and validated through simulation. To this end, we rely on the following scenario to generate reliable synthetic log data that mimic as much as possible real-world situations.

Consider a learning environment, where five (05) learners (Ahmed, Noah, Emma, Sophia, and Sarah) interact with their devices and with each other. The system automatically adapts multimedia content (learning materials) using features such as speech-to-text, contrast adjustment, and video streaming on smart TVs. Each learner has a learning style that shapes its interactions with learning materials.

*Ahmed is an engineering student who follows a cognitive-state learning style. Ahmed always reviews his lectures on his tablet during his morning commute. The system detects glare on his tablet screen and asks him to adjust the contrast for better visibility. On his way home, Ahmed always takes a short break to watch his favorite TV show before reviewing his lectures. The system sometimes detects that his phone's battery is low and the room's lighting is dim, so it asks him to adjust the screen contrast to protect his eyes and reduce battery consumption. When he is revising the lectures, Ahmed always prefers to collaborate with other students; thus, he highlights the sections he does not understand and shares them with Emma, knowing that the material is related to her psychology project.*

*Emma, is a psychology student who follows a visual-verbal learning style. She always goes to the university's cafeteria studying during her lunch break. She opens the shared content from Ahmed on her smartphone but struggles to hear the audio due to ambient noise. Thus,* *the system asks her to provide a speech-to-text transcription of the lecture, enabling Emma to follow along. The system also creates visual diagrams summarizing the theoretical concepts. Emma integrates these visuals with her verbal notes and shares them back with Ahmed, combining her preference for visual-verbal learning with his practical focus.*

*Noah, a design student who follows an active-reflective learning style. He periodically watches architecture tutorials on his smartphone. Due to the narrow screen size of his smartphone, the system asks him to stream the tutorial on a smart TV in his study room. The system also activates annotation tools, allowing him to highlight design elements while pausing and reflecting on important sections. He shares the annotated video with Sophia to get her perspective on how the designs might align with broader economic trends, blending his active learning style with reflective pauses.*

*Sophia, an economics student with sequential-global learning style, prefers detailed analysis through visual presentations. She accesses the annotated tutorials videos shared by Noah and other learners in the library; thus, she should avoid playing the audio to maintain the library's silence. The system asks her to generate a detailed transcript of the video and highlights economic implications discussed in the content. Sophia adds her observations on market trends and financial data, sharing the enriched content back with Noah to contribute to his understanding.*

*Sarah, a literature student whose learning state is currently sensing-intuitive. Unlike the others, Sarah enjoys delving into texts and analyzing literary ideas and themes through deep reading and focusing on textual details over documents. Regularly every morning, while sitting in the library, Sarah begins reading novels on her tablet. Since she tends to read for long periods, the system sometimes detects the extended reading time; thus, it asks her to adjust the screen's brightness and contrast to be more comfortable for her eyes, reducing any harsh brightness or contrast that could negatively affect her comfort. When she backs home after a long day, Sarah always continues analyzing novels on her tablet. Based on the ambient lighting in her room, the system asks her to adjust the screen lighting to match the surrounding environment. Additionally, if the room is dark, the system asks her to increase the brightness slightly to avoid eyestrain.*

To simulate these scenarios, we developed simple web pages according to the needs of Ahmed, Noah, Emma, Sophia, and Sarah and asked five volunteer students to play their roles for several times to collect, store and analyze the log data using our LSA-based approach.

## 5.2 Results

In the following, we present the obtained results in each step of the proposed approach.

### 5.2.1 Results of the Pre-LSA Analysis Step

Pre-LSA analysis includes three phases: *data retrieval*, *data preprocessing*, and *data segmentation*. First, log file *LF* is constructed using an invocation to the data retriever that loads the HUD from the database. For the sake of simplicity and feasibility, we only consider the following contextual situations settings:

- Context elements vector *CXT* = (*location*, *battery level*, *screen size*, *network type*).
- Adaptation actions vector *ACT* = (Speech-to-text, Contrast-adjustment, Video-in-Smart-TV).
- Set of context situations in log file *LF* $CS = \{cs_1, cs_2, \ldots, cs_k\}$.

Where number of context elements considered $g = 4$, number of adaptation actions $h = 3$, and number of contextual situations $k = 100$. Then, the log data are preprocessed by replacing the textual representation qualitative values of context elements with numerical values. Table 6.5 summarizes the resulting values.

| Notation | Context situation element | Domain of definition | Numerical domain of definition | Context situation element type |
|---|---|---|---|---|
| $c_1$ | Location | Public place | 0 | Context element |
| | | Library | 1 | |
| | | Home | 2 | |
| $c_2$ | Battery level | Low | 0 | |
| | | Medium | 1 | |
| | | Full | 2 | |
| $c_3$ | Screen size | Narrow | 0 | |
| | | Wide | 1 | |
| $c_4$ | Network Type | Mobile data | 0 | |
| | | WI-FI | 1 | |
| $a_1$ | Speech-to-text | Yes | 1 | Action |
| $a_2$ | Contrast-adjustment | | | |
| $a_3$ | Video-in-Smart-TV | No | 0 | |

**Table 6.5.** The resulting numerical representation of contextual situations.

- **Data Segmentation**

Finally, log file *LF* is divided into sub-logs $LF_{i=1..5}$ according to user *id*, to construct the corpus. Figure 6.4 illustrates subparts of the sub-logs resulting from its segmentation.

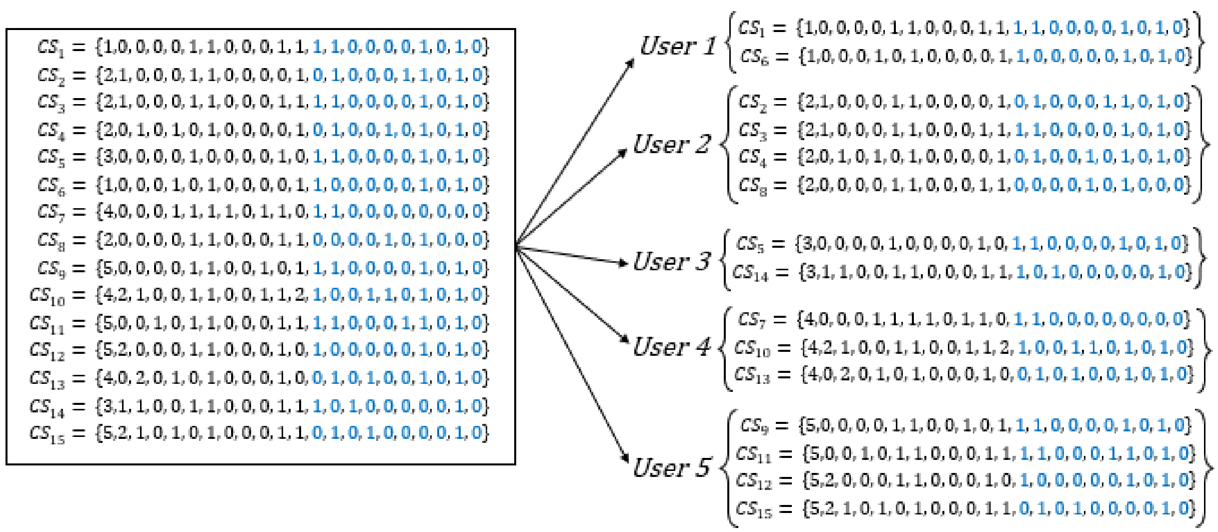



**Figure 6.5.** Partitioning LF into five sub-logs.

### 5.2.2 Results of the LSA Analysis Step

LSA analysis step comprises four phases: *definition of the dictionary of terms*, *construction of the occurrence matrix*, *decomposition of the occurrence matrix*, and *extraction of the compressed matrix*.

- **Definition of the Dictionary of Terms**

As the corpus constructed in the previous step, we need to define the dictionary of terms required for the analysis. The set of terms is defined as $T = \{t = (c_i, a_j) \ / \ c_i \in CXT$ and $a_j \in ACT\}$. Thus, Set $T$ contains $m = 30$ terms (see Table 5.6).

- **Construction of the Occurrence Matrix**

The resulting occurrence matrix $A$ $(m \times n)$ is constructed by calculating the number of occurrences of each term $t \in T$ in each sub-log $LF_{i=1..n}$. Table 6.6 represents part from content of Matrix $A$ (due to space limitation).

| Contextual Situation | $LF_1$ (user 1) | $LF_2$ (user 2) | $LF_3$ (user 3) | $LF_4$ (user 4) | $LF_5$ (user 5) |
|---|---|---|---|---|---|
| $t_{01} = (c_1(0), a_1)$ | 2 | 5 | 2 | 1 | 4 |
| $t_{02} = (c_1(1), a_1)$ | 2 | 2 | 2 | 1 | 2 |
| $t_{03} = (c_1(2), a_1)$ | 2 | 4 | 3 | 1 | 1 |
| ……. | . | . | . | . | . |
| ……. | . | . | . | . | . |
| ……. | . | . | . | . | . |
| $t_{29} = (c_4(0), a_3)$ | 2 | 4 | 0 | 4 | 0 |
| $t_{30} = (c_4(1), a_3)$ | 5 | 3 | 6 | 5 | 3 |

**Table 6.6.** Part from the content of occurrence matrix A.

- **Decomposition of the Occurrence Matrix**

The next phase is to decompose the occurrence matrix *A* using SVD, as discussed in Section 2, through factorization into three matrices *U*, *S*, and *V*, such that $A = U \times S \times V^T$. Matrix *A* ($m \times n$) represents the original occurrence matrix, matrix *U* ($m \times r$) is composed of vectors representing latent features of rows (contextual situations), matrix *U* ($r \times r$) consists of **singular values** indicating the importance of each latent feature, and matrix $V^T$ ($r \times n$) is the transposed matrix of sub-log vectors indicating the latent features for columns (users). Tables 5.7, 5.8, and 5.9 represent parts of the values of Matrices *U*, *S*, and $V^T$.

| *m* \ *r* | 1 | 2 | 3 | ............. | *m*/2 | | *m*-1 | *m* |
|---|---|---|---|---|---|---|---|---|
| 1 | -0.1825 | -0.2628 | 0.0184 | ............. | -0.2087 | ............. | -0.0722 | -0.0297 |
| 2 | -0.1132 | -0.0455 | -0.0609 | ............. | 0.2989 | ............. | 0.2391 | -0.3909 |
| 3 | -0.1476 | -0.1224 | 0.0844 | ............. | 0.2041 | ............. | 0.3257 | -0.4247 |
| . | . | . | . | ............. | . | ............. | . | . |
| . | . | . | . | ............. | . | ............. | . | . |
| . | . | . | . | ............. | . | ............. | . | . |
| *m* | -0.2692 | 0.2623 | -0.008 | ............. | 0.1478 | ............. | 0.1865 | 0.657 |

**Table 6.7.** Part from the content of matrix U.

| | | | | |
|---|---|---|---|---|
| 35.5184 | 0 | 0 | 0 | 0 |
| 0 | 9.2633 | 0 | 0 | 0 |
| 0 | 0 | 6.486 | 0 | 0 |
| 0 | 0 | 0 | 5.3673 | 0 |
| 0 | 0 | 0 | 0 | 4.8741 |

**Table 6.8.** Content of matrix S.

| | | | | |
|---|---|---|---|---|
| -0.4518 | -0.5792 | -0.4253 | -0.3858 | -0.3615 |
| 0.2124 | -0.4756 | -0.0542 | 0.7999 | -0.2933 |
| -0.8335 | 0.2489 | 0.3287 | 0.349 | -0.116 |
| 0.045 | 0.6078 | -0.6549 | 0.1509 | -0.4206 |
| -0.2323 | -0.0835 | -0.5285 | 0.2585 | 0.77 |

**Table 6.9.** Content of matrix $V^T$.

- **Extraction of the Compressed Matrix**

Compressed matrix *A'* is generated by reducing the rank *k* after factorization of matrix *A*. The value for rank *k* is often fixed empirically ($k= 2$ in our case). Hence, matrix $A' = U_{m\times k} \times S_{k\times k} \times V_{k\times n}$. Table 6.10 represents part from the content of occurrence matrix *A'*.

| | | | | |
|---|---|---|---|---|
| 2.411551 | 4.912235 | 2.888785 | 0.553525 | 3.05729 |
| 1.727022 | 2.529235 | 1.732841 | 1.214037 | 1.577097 |
| . | . | . | . | . |
| . | . | . | . | . |
| . | . | . | . | . |
| 2.517694 | 1.939734 | 1.932376 | 3.215038 | 1.216754 |
| 4.835992 | 4.382456 | 3.934835 | 5.632415 | 2.743852 |

**Table 6.10.** Part from the content of occurrence matrix A'.

### 5.2.3 Results of Post-LSA Analysis Step

By using Pearson law, the correlation matrix of users $COR_u$ ($n \times n$) is calculated; its content is given in Table 6.11. This is a symmetrical matrix which represents the similarity degree between any two users *i* and *j* by value $COR_u$ (*i*, *j*).

| | User1 | User2 | User3 | User4 | User5 |
|---|---|---|---|---|---|
| User1 | 1 | 0.641475 | **0.909569** | 0.785065 | 0.645468 |
| User2 | | 1 | **0.902255** | 0.028419 | **0.999986** |
| User3 | | | 1 | 0.456671 | **0.904492** |
| User4 | | | | 1 | 0.033633 |
| User5 | | | | | 1 |

**Table 6.11.** The correlation matrix of users $COR_u$ ($n \times n$).

The clustering of users is carried out by grouping them into subsets such that the correlation between any two users in the same subset is greater than a threshold $Th_{cor}$. Figure 6.6 illustrates the graphical representation of the resulting clusters for users, performed according to two value 0.9 for parameter $Th_{cor}$.

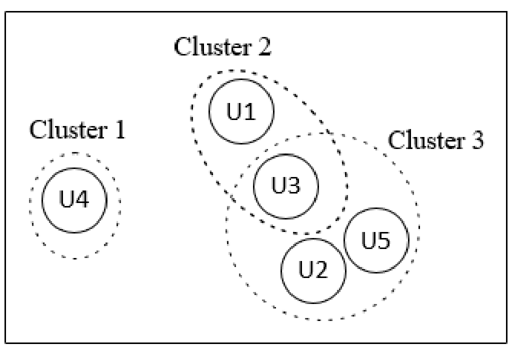



**Figure 6.6.** The graphical representation of users clustering.

Similarly, the correlation matrix of contextual situations $COR_{cs}$ ($m \times m$) is calculated using Pearson law. Due to space limitation the content of matrix $COR_{cs}$ ($m \times m$) can be found in Appendix A. The clustering of contextual situations is also carried out by grouping them into subsets by considering value 0.9 for threshold $Th_{cor}$. Figure 6.7 illustrates the graphical representation of the resulting clusters for contextual situations.

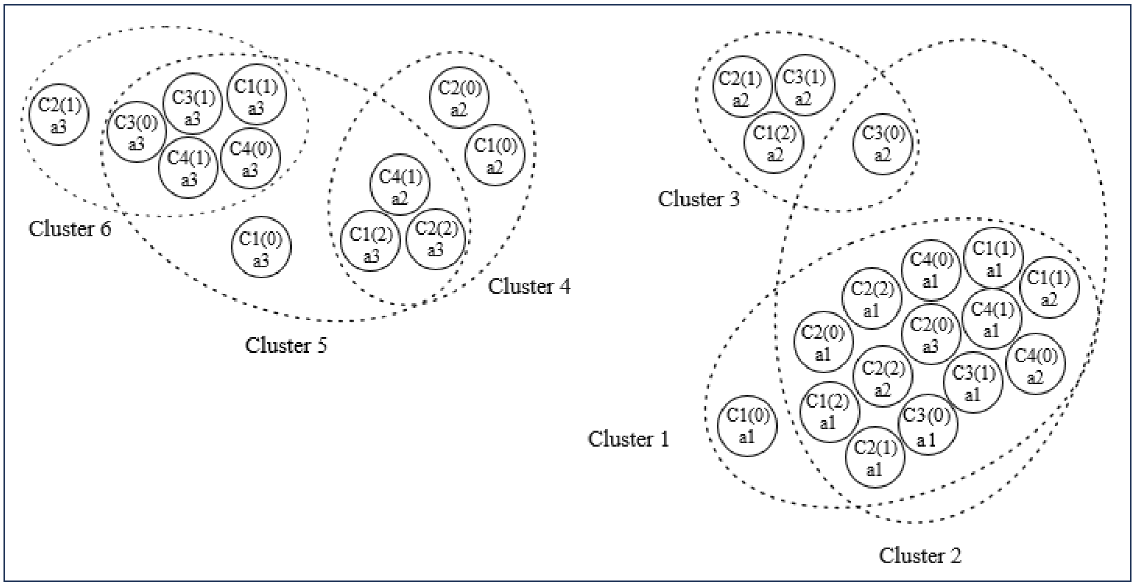



**Figure 6.7.** The graphical representation of contextual situation clustering.

- **Discussion**

As illustrated in Figure 6.6, learners $U_{i=1..5}$ are grouped into three distinct clusters based on their behaviors regarding the presentation of learning materials, considering the environment's requirements. These clusters are as follows: **Cluster 1 (User 4)**, **Cluster 2 (User 1 and User 3)**, and **Cluster 3 (User 2, User 3, and User 5)**. Below are some observations and insights regarding the learners' behaviors based on their clustering:

– **Cluster 1 (U4):** Learner 4 (U4) is isolated from the other users, suggesting that U4 has a unique learning style, preferences, or behaviors that do not align with those of the other learners. This learner can be associated with Sarah, a sensing-intuitive learner, who excels in independent learning environments. Sarah balances attention to detail (sensing style) with a focus on broader concepts (intuitive).

– **Cluster 2 (U1, U3):** Learners 1 (U1) and 3 (U3) are grouped together, indicating that they share similar learning behaviors or preferences. They may prefer similar learning activities, such as collaborative or reflective learning tasks. This cluster can be described as:

  - **Emma (U1 - Visual-Verbal):** Emma shares a cluster with Ahmed, as she focuses on visual-verbal methods, which complement Ahmed's need for clear and structured content. Emma benefits from diagrams paired with textual explanations, enhancing her learning experience.

  - **Ahmed (U3 - Beginner):** As a beginner, Ahmed requires foundational material, simplified explanations, and guidance. He thrives in collaborative work, which helps him achieve his learning objectives.

- **Cluster 3 (U2, U3, U5):** Learners 2 (U2), 3 (U3), and 5 (U5) form a separate group, with slight overlap with Cluster 2. This overlap indicates shared traits or transitional behaviors between the learners in these two clusters. This cluster can be described as:
  - **Ahmed (U3), Noah (U2), and Sophia (U5):** The overlap between these users suggests shared learning traits or transitional behaviors, particularly for Ahmed. This grouping indicates potential for collaboration between beginner learners (Ahmed) and those with visual-verbal (U1) and sequential-global (U2 and U5) learning styles.
  - **Noah (U2 - Active-Reflective):** Noah thrives on hands-on learning and reflective pauses, which allow him to process information. His position with Sophia suggests that shared learning activities, where both action and structure play important roles, are beneficial.
  - **Sophia (U5 - Sequential-Global):** Sophia's sequential-global style involves a preference for mastering step-by-step processes and ensuring visibility in her learning path. Her collaboration with Noah could foster synergistic learning, with Noah's active approach complementing Sophia's structured, sequential style.

- **General Observations:**
  - The clustering reveals a diversity of learning behaviors among the users.
  - Learners U1, U3, U2, and U5 tend to exhibit social or collaborative learning tendencies, as evidenced by their proximity within the clusters. In contrast, U4 appears to prefer independent or solitary learning, as indicated by their isolation in Cluster 1.
  - The overlap between Cluster 2 and Cluster 3 suggests the potential for shared resources or adaptive strategies that can cater to the needs of multiple clusters.

More broadly, this clustering approach offers valuable insights and key advantages, including: recommending adaptive learning actions, grouping learners for collaborative tasks, facilitating effective comparisons between learners based on their behaviors, and adapting resources based on the specific needs of each cluster.

As illustrated in Figure 6.7, terms $t_{j=1..30}$ are grouped into six distinct clusters based on the execution of a given action according to a given value of a context element, considering the environment's requirements. Below are some key observations and insights:

- **Cluster 1:** Analysis of this cluster reveals that it is suitable for scenarios where accessibility is a priority or where audio output is less feasible. This cluster involves the use of the Speech-to-Text adaptation action, which is particularly useful in environments like libraries where silence is required.
- **Cluster 2:** This cluster is similar to Cluster 1 but with additional conditions, ensuring adaptability across diverse environmental and device contexts. It focuses on enhancing visibility through the Contrast Adjustment adaptation action, accommodating various screen sizes.
- **Cluster 3:** In this cluster, the system predominantly relies on the Contrast Adjustment adaptation action to optimize visual accessibility across different environmental and device conditions.
- **Cluster 4:** This cluster encompasses multiple contextual situations that involve both the Contrast Adjustment and Video on Smart TV adaptation actions, highlighting how the system interacts with learners based on their visual learning preferences.
- **Cluster 5:** This cluster recommends the use of Video on Smart TV adaptation, catering to user preferences for high-quality, large-screen video experiences. It also suggests adjusting screen contrast when large screens are not available.
- **Cluster 6:** This cluster is characterized by situations where the system exclusively recommends using large screen sizes via the Video on Smart TV adaptation action.

## 6. Conclusion

In this chapter, we explored the application of LSA to analyze learner behaviors within context-aware pervasive systems, with a particular focus on non-instructional contexts. The goal was to provide an efficient method for uncovering latent semantic structures and extracting meaningful patterns from HUD. Specifically, our approach aimed to offer a comprehensive classification of both similar contextual situations and learner clusters, thereby enhancing content delivery, context understanding, and personalized user experiences.

We treated each learner's historical data as textual data, with context elements and corresponding adaptation actions serving as terms. LSA was then applied to this corpus to uncover hidden relationships and semantic structures. Correlations were calculated to identify clusters of similar contextual situations and learners, providing valuable insights into learning behaviors.

Overall, our proposed approach offers an effective method for revealing latent patterns in HUD, which can significantly improve the personalization and adaptability of learning activities in context-aware pervasive systems.

# Chapter VII

# Conclusions and Future Work

## Summary

# CHAPITRE VII

## Conclusions and Future Work

In this final chapter, we conclude the core debate introduced in this thesis regarding the adaptation of multimedia documents within context-aware pervasive systems, with a particular focus on the adaptation of learning materials. We provide a summary of the research conducted, assessing how the established objectives have effectively been achieved. Additionally, we evaluate our work with respect to multimedia adaptation and, more broadly, within the domain of ambient intelligence. Finally, we outline a series of unfinished tasks, leaving the door open for future exploration.

## 1. Research Summary

### 1.1 Objective and Methodology

The main objective of this doctoral thesis was to enhance access to multimedia documents in context-aware pervasive systems by utilizing historical user data. The research focused on two key aspects: efficiently managing historical user data (content, format, and storage location) and leveraging these data to improve adaptation processes. To achieve this objective, we followed a methodology that includes:

**(1) Storing Historical Data**: Using both SQL (relational) and NoSQL (non-relational) databases to store historical user data generated through multimedia document adaptation, were stored in. These databases were chosen for their ability to handle large volumes of data efficiently. The approach integrated context modeling techniques with both types of databases.

**(2) Processing Historical Data**: Developing a set of functions for logging, retrieving, and analyzing historical data, according to three management methods were proposed: client-side, server-side, and proxy-based, with hybrid solutions for specific needs. These functions support various adaptation processes, addressing challenges related to system performance (e.g., data volume, processing) and user requirements (e.g., privacy, security).

**(3) Leveraging Historical Data**: Exploring how historical user data can enhance intelligent context-aware applications through machine learning, data mining, and natural language processing techniques. The aim is to focus on client-side data management for personalizing adaptation rules, and on server-side management for discovering hidden patterns and clustering similar contextual situations and user behaviors.

**(4) Validation with Prototype and Real-World Application**: Demonstrating the feasibility aspect of the proposals through a prototype that both adapts multimedia documents based on context and stores corresponding historical data. While simulations are commonly used, the real prototype addresses practical challenges. Additionally, the prototype focuses on adapting multimedia contents related to distance learning materials based on non-instructional context, as the study does not address adaptations for learning styles, but rather for environmental needs.

## 1.2 Contributions

Our initial goal led us to propose three primary contributions.

**(1) Design of a Software Component for Managing Historical User Data:** The first contribution was the development of a software component to manage historical user data generated from multimedia document adaptation processes. The component allowed for the storage, retrieval, and analysis of context values and adaptation actions. It features flexible data management through three variants: client-side, proxy-based, and server-side management. It also integrated relational and NoSQL schemas and was validated through experiments, aiming to enhance adaptation approaches by leveraging historical data for various beneficial tasks such as machine learning and recommender systems.

**(2) A Rule-Learning Approach for Personalizing Adaptation Actions:** The second contribution introduced a rule-learning method that personalized context-aware adaptation rules using historical user data. By analyzing past decisions, this approach predicted future user actions using data-driven techniques, such as Eclat, sequential covering, FP-Growth, and decision trees. The system operated as an integrated component of the historical data manager, providing unsupervised machine learning without the need for pre-trained datasets.

**(3) Latent Semantic Analysis for Classifying Context Situations and User Behaviors:** The final contribution employed LSA, commonly used in text mining, to classify similar contextual situations and user behaviors. By treating historical data as textual documents, this technique helped uncover hidden semantic structures and extract meaningful patterns, improving understanding of context, content delivery, and user personalization.

## 2. Added Values and Achievements

The successful achievement of the objectives outlined in this thesis has significantly enriched MDA processes and, more broadly, contributed valuable elements to the field of context-aware pervasive systems. Below are the key added values and achievements of this work:

(1) **Generalized Proposal for MDA Processes:** Compared to other MDA-based approaches, our proposal stood out due to its general applicability across various MDA process categories. It facilitated the storage, retrieval, and analysis of historical data (log data) derived from accumulated user decisions, all while handling data at a high level of abstraction, irrespective of specific MDA approaches. This allowed the proposed component to offer MDA processes more options for efficiently processing historical user data across different locations, including client-side, proxy-side, server-side, and hybrid solutions.

(2) **Flexible and Agile Historical Data Manager:** The proposed historical data manager was built on a well-devised, flexible, and agile architecture that can adapt to changing contexts, such as computational resources or user preferences (e.g., personalized processes, data privacy, and sharing). Most existing context-aware pervasive systems involving HUD lack detailed design and technical specifications for managing log data. In contrast, our work provided a comprehensive description of the design requirements and technical implementation, ensuring a robust and adaptable data management approach.

(3) **Context-Aware Focus on Users and Environment:** While most existing multimedia adaptation approaches emphasize general user information and resource access, our approach shifted focus to user and environmental requirements, which are more specific to context-aware pervasive systems. These included sensors, human-computer interactions, and other context values that influence the adaptation process. Despite some MDA-based applications involving HUD, many do not effectively manage such data, often analyzing it through static datasets or specific collected data without a comprehensive approach to contextual adaptation.

(4) **Incorporating Association Rule Mining (ARM):** A significant contribution of our work was the application of ARM techniques to analyze historical log data. ARM offers multiple advantages, including the discovery of hidden patterns, improved decision-making, personalized recommendations, simplicity, and interpretability. These features made ARM particularly beneficial for enhancing the efficiency of multimedia adaptation processes by uncovering meaningful patterns in user behavior and context.

(5) **Leveraging Latent Semantic Analysis (LSA) for User Behavior Analysis:** LSA was another key technique employed in this work to analyze learner behaviors. LSA

provides several advantages, especially in areas like natural language processing, information retrieval, and knowledge discovery. These include dimensionality reduction, capturing latent semantic relationships, contextual understanding, improved clustering, and noise reduction. While many MDA-based approaches focus on instructional context for learner behavior analysis, our work distinguished itself by examining non-instructional behaviors. This is crucial in context-aware pervasive systems, where learning often occurs in mobile environments that present unique constraints and challenges.

## 3. Implications and Practical Application

Our research spans a variety of application domains; however, we focus on the following three key implications and applications.

1. **Distance Learning Platforms (e.g., MOOCs)**: In this domain, multimedia document adaptation plays a crucial role in tailoring educational content to individual learners' contexts. By considering factors such as learners' device type, location, and preferences, our framework ensures that the multimedia material (videos, readings, quizzes) is presented in a way that enhances engagement, accessibility, and learning outcomes. This personalized adaptation not only improves accessibility for learners with diverse needs but also supports better retention and comprehension by offering content suited to the learner's current context.
2. **Tourism and Cultural Spaces**: In the realm of tourism, particularly in museums or historical sites, multimedia documents can be adapted to provide richer, more immersive experiences through context-aware systems. Using virtual and augmented reality, our framework dynamically adjusts multimedia content based on the visitor's location, preferences, and interaction patterns. For instance, a visitor standing in front of a particular artifact may receive an augmented reality (AR) experience that displays additional multimedia content like videos, interactive maps, or 3D models, enhancing their understanding of the exhibit. In virtual tourism, the content can adapt to the virtual environment and the user's preferences, allowing for a more engaging and personalized exploration of cultural landmarks. This kind of adaptation allows for deeper engagement and interaction, catering to the diverse interests and needs of each visitor.
3. **Private Smart Spaces**: In private smart spaces, such as smart homes and smart libraries, multimedia documents can be automatically adjusted based on a variety of factors, including user behavior, device capabilities, and environmental conditions. For example, in a smart home, multimedia content like video or audio can be adapted depending on the user's physical presence in a room, the time of day, or even the current activity (e.g., reading, cooking, or exercising). In smart libraries, where users may be accessing digital documents, content could be tailored to their reading habits,

device usage, or even the ambient lighting conditions, ensuring optimal readability and comfort. This adaptation of content in smart spaces not only provides greater convenience but also fosters a more efficient and user-centric experience by anticipating needs and adjusting multimedia delivery accordingly.

## 4. Limitations and Constraints of the Work

Despite the significance of the finding achieved in this research work, several limitations and constraints still need to be addressed:

- **Data Availability:** Although this study deals with real-world and daily life issues and scenarios, the availability of reliable datasets is a serious concern. Indeed, in the absence of real smart environment, the only way to validate the results is to rely on simulations. Generally, prototyping may help in such cases but does not completely solve the problem on its own.
- **Security and Privacy Concerns:** While the primary focus of this research was the management and use of historical users data, particularly for adapting learning materials contents, ensuring the protection of sensitive learners information is critical. Thus, accurate implementation requires compliance with privacy laws and robust security measures to prevent unauthorized access.
- **Generalizability of Results:** While the analysis performed well on small systems, their effectiveness across other larger systems is uncertain. This due to the fact that it is very challenging to provide the necessary hardware resources to scale for a large number of users.
- **Computational Resources:** As the historical users data grow over time, their analysis becomes more difficult with respect to computation resources. Devices with limited resources may find it challenging to analyze large datasets efficiently, which could negatively influence scalability and slow down systems performance.
- **Real-World Application:** Deploying these models in operational systems introduces additional challenges as this will require significant investments in infrastructures.

## 5. Future Directions of the Research

While this work has addressed several key aspects and synthesized relevant analyses, there are still ongoing challenges and opportunities for improvement. We outline the following future directions for enhancing this research:

(1) **Enhancing the Performance of Analysis Mechanisms:** As the volume of historical user data continues to grow, the performance of the algorithms proposed in this thesis may require optimization from various perspectives, including:

- **Effectiveness:** As the data and time complexity increase, especially on client-side devices with limited computational resources, performance issues may arise. To address this, dynamic data structures (e.g., matrices, trees) could be employed to be update automatically the elements involved in analysis tasks whenever new data entries are inserted into the log database.

- **Efficiency:** As the system is based on unweighted (blind) correlations, it may take a long time to overlook the most frequent old values for both context elements and adaptation actions so to replace them with more recent values. To improve efficiency, the system could adopt a sliding window approach to focus on more recent data and discard irrelevant older records. Additionally, weighting log entries based on their timestamps would prioritize more recent user behavior and decisions.

(2) **Scalability Issues:** As the practical implementation of the multimedia adaptation system is still under test, scaling the adaptation process to accommodate larger systems or increasing numbers of users remains a significant challenge. Potential solutions include:

- **Data Augmentation:** Employing data augmentation techniques to generate synthetic log data that closely resembles real-world patterns and user behavior, ensuring that datasets remain reliable and realistic.

- **Simulation Platforms:** Using simulation environments designed for Internet of Things (IoT) and Cloud computing could provide valuable insights and aid in testing the system's scalability across varying contexts.

(3) **Involving machine learning and deep learning mechanisms:** Deep learning can be effectively used in multimedia document adaptation within context-aware pervasive systems by harnessing its capacity to process vast amounts of diverse data and uncover complex patterns without the need for explicit programming. This capability enables deep learning to address adaptation from multiple perspectives, including contextual and semantic understanding. By integrating deep learning into the adaptation process, these systems can deliver highly personalized, context-sensitive experiences that evolve in response to changing user needs and environmental conditions. As a result, the adaptation process becomes more dynamic and efficient, significantly enhancing content accessibility, usability, and overall user satisfaction while leveraging both historical and real-time user data.

Finally, we acknowledge that this work is inherently partial and not exhaustive. However, we hope that it will serve as a valuable contribution to the community focused on context-aware pervasive systems and multimedia document adaptation. Even if it only serves as a foundation, we trust that this research will inspire further exploration and development in these areas.

## Bibliography


[1] F. Piccialli, S. Cuomo, V. S. D. Cola, and G. Casolla, "A machine learning approach for IoT cultural data," *J Ambient Intell Human Comput*, vol. 15, no. 2, Art. no. 2, Feb. 2024, doi: 10.1007/s12652-019-01452-6.

[2] I. H. Sarker, "Context-aware rule learning from smartphone data: survey, challenges and future directions," *J Big Data*, vol. 6, no. 1, p. 95, Dec. 2019, doi: 10.1186/s40537-019-0258-4.

[3] Y. Asim, M. A. Azam, M. Ehatisham-ul-Haq, U. Naeem, and A. Khalid, "Context-Aware Human Activity Recognition (CAHAR) in-the-Wild Using Smartphone Accelerometer," *IEEE Sensors J.*, vol. 20, no. 8, pp. 4361–4371, Apr. 2020, doi: 10.1109/JSEN.2020.2964278.

[4] A. Haque, A. Milstein, and L. Fei-Fei, "Illuminating the dark spaces of healthcare with ambient intelligence," *Nature*, vol. 585, no. 7824, pp. 193–202, Sep. 2020, doi: 10.1038/s41586-020-2669-y.

[5] I. N. Milat, H. Seridi, and A. Moudjari, "Discovering Learners Behaviour Patterns From Log Files Using LSA:," *International Journal of Distance Education Technologies*, vol. 18, no. 2, pp. 90–113, Apr. 2020, doi: 10.4018/IJDET.2020040106.

[6] F. Skopik, M. Wurzenberger, and M. Landauer, *Smart Log Data Analytics: Techniques for Advanced Security Analysis*. Cham: Springer International Publishing, 2021. doi: 10.1007/978-3-030-74450-2.

[7] F. Abdelhedi, A. A. Brahim, F. Atigui, and G. Zurfluh, "Logical Unified Modeling for NoSQL Databases:," in *Proceedings of the 19th International Conference on Enterprise Information Systems*, Porto, Portugal: SCITEPRESS - Science and Technology Publications, 2017, pp. 249–256. doi: 10.5220/0006311702490256.

[8] W. Deng, "Object-Oriented Database and O/R Mapping Technology," in *Big Data Analytics for Cyber-Physical System in Smart City*, vol. 1303, M. Atiquzzaman, N. Yen, and Z. Xu, Eds., in Advances in Intelligent Systems and Computing, vol. 1303. , Singapore: Springer Singapore, 2021, pp. 800–806. doi: 10.1007/978-981-33-4572-0_115.

[9] F. Bettou and M. Boufaida, "An Adaptation Architecture Dedicated to Personalized Management of Multimedia Documents:," *International Journal of Multimedia Data Engineering and Management*, vol. 8, no. 1, pp. 21–41, Jan. 2017, doi: 10.4018/IJMDEM.2017010102.

[10] A. Saighi, R. Philippe, N. Ghoualmi, S. Laborie, and Z. Laboudi, "HaMA: A Handicap-based Architecture for Multimedia Document Adaptation," *International Journal of Multimedia Data Engineering and Management*, vol. 8, no. 3, pp. 55–96, Jul. 2017, doi: 10.4018/IJMDEM.2017070104.

[11] A. Saighi, Z. Laboudi, R. Philippe, S. Laborie, N. Ghoualmi-Zine, "On using multiple disabilities profiles to adapt multimedia documents," *International Journal of Information Technology and Web Engineering*, doi: doi: 10.4018/IJITWE.2020070103.

[12] J. C. Augusto, "Ambient Intelligence: Basic Concepts and Applications," in *Software and Data Technologies*, vol. 10, J. Filipe, B. Shishkov, and M. Helfert, Eds., in Communications in Computer and Information Science, vol. 10. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 16–26. doi: 10.1007/978-3-540-70621-2_2.

[13] A. Bimpas, J. Violos, A. Leivadeas, and I. Varlamis, "Leveraging pervasive computing for ambient intelligence: A survey on recent advancements, applications and open challenges," *Computer Networks*, vol. 239, p. 110156, Feb. 2024, doi: 10.1016/j.comnet.2023.110156.

[14] P. N. Mahalle and P. S. Dhotre, *Context-Aware Pervasive Systems and Applications*, vol. 169. in Intelligent Systems Reference Library, vol. 169. Singapore: Springer Singapore, 2020. doi: 10.1007/978-981-32-9952-8.

[15] A. K. Dey, "Understanding and Using Context".

[16] B. N. Schilit and M. M. Theimer, "Disseminating active map information to mobile hosts," *IEEE Network*, vol. 8, no. 5, pp. 22–32, 1994, doi: 10.1109/65.313011.

[17] B. Schilit, N. Adams, and R. Want, "Context-Aware Computing Applications".
[18] P. J. Brown, J. D. Bovey, and Xian Chen, "Context-aware applications: from the laboratory to the marketplace," *IEEE Pers. Commun.*, vol. 4, no. 5, pp. 58–64, Oct. 1997, doi: 10.1109/98.626984.
[19] J. Pascoe, "Adding generic contextual capabilities to wearable computers," in *Digest of Papers. Second International Symposium on Wearable Computers (Cat. No.98EX215)*, Pittsburgh, PA, USA: IEEE Comput. Soc, 1998, pp. 92–99. doi: 10.1109/ISWC.1998.729534.
[20] P. Brézillon and J.-C. Pomerol, "Contextual knowledge and proceduralized context," in *AAAI Conference on Artificial Intelligence*, 1999. [Online]. Available: https://api.semanticscholar.org/CorpusID:18620700
[21] A. Schmidt, "Implicit human computer interaction through context," *Personal Technologies*, vol. 4, no. 2–3, pp. 191–199, Jun. 2000, doi: 10.1007/BF01324126.
[22] T. Winograd, "Architectures for Context," *Human–Computer Interaction*, vol. 16, no. 2–4, pp. 401–419, Dec. 2001, doi: 10.1207/S15327051HCI16234_18.
[23] G. K. Mostifaoui, J. Pasquier-Rocha, and P. Brkzillon, "Context-Aware Computing: A Guide for the Pervasive Computing Community".
[24] T. Chaari, F. Laforest, and A. Flory, "Adaptation des applications au contexte en utilisant les services web," in *Proceedings of the 2nd French-speaking conference on Mobility and uibquity computing - UbiMob '05*, Grenoble, France: ACM Press, 2005, p. 111. doi: 10.1145/1102613.1102638.
[25] J. Strassner and D. O'Sullivan, "Knowledge management for context-aware, policy-based ubiquitous computing systems," in *Proceedings of the 6th international workshop on Managing ubiquitous communications and services*, Barcelona Spain: ACM, Jun. 2009, pp. 67–76. doi: 10.1145/1555321.1555335.
[26] N. Ryan, J. Pascoe, and D. Morse, "Enhanced Reality Fieldwork: the Context Aware Archaeological Assistant".
[27] D. Salber, A. K. Dey, and G. D. Abowd, "Ubiquitous Computing: Defining an HCI Research Agenda for an Emerging Interaction Paradigm".
[28] H. Lieberman and T. Selker, "Out of context: Computer systems that adapt to, and learn from, context," *IBM Syst. J.*, vol. 39, no. 3.4, pp. 617–632, 2000, doi: 10.1147/sj.393.0617.
[29] J. Burrell and G. K. Gay, "Collectively Defining Context in a Mobile, Networked Computing Environment".
[30] E. Rohn, "Predicting Context Aware Computing Performance".
[31] L. Barkhuus, "Context Information in Mobile Telephony," in *Human-Computer Interaction with Mobile Devices and Services*, vol. 2795, L. Chittaro, Ed., in Lecture Notes in Computer Science, vol. 2795. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 451–455. doi: 10.1007/978-3-540-45233-1_45.
[32] K. Abbas, C. Verdier, and A. Flory, "Exploiting Profile Modeling for Web-Based Information Systems," in *Web Information Systems Engineering – WISE 2007 Workshops*, vol. 4832, M. Weske, M.-S. Hacid, and C. Godart, Eds., in Lecture Notes in Computer Science, vol. 4832. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 313–324. doi: 10.1007/978-3-540-77010-7_30.
[33] E. Christopoulou, "Context as a Necessity in Mobile Applications".
[34] A. Sindico and V. Grassi, "Model driven development of context aware software systems," in *International Workshop on Context-Oriented Programming - COP '09*, Genova, Italy: ACM Press, 2009, pp. 1–5. doi: 10.1145/1562112.1562119.
[35] K. Michalakis and G. Caridakis, "Enhancing user interaction with context-awareness in cultural spaces," *Pers Ubiquit Comput*, vol. 27, no. 2, pp. 379–399, Apr. 2023, doi: 10.1007/s00779-022-01698-6.
[36] C. Mosquera-Lopez *et al.*, "Automated Detection of Real-World Falls: Modeled From People With Multiple Sclerosis," *IEEE J. Biomed. Health Inform.*, vol. 25, no. 6, pp. 1975–1984, Jun. 2021, doi: 10.1109/JBHI.2020.3041035.

[37] A. Alti, S. Laborie, and P. Roose, "A Community-Based Semantic Social Context-Aware Driven Adaptation for Multimedia Documents," *Int. J. Virtual Communities Soc. Netw.*, vol. 7, no. 2, pp. 31–49, Apr. 2015.
[38] G. Chen and D. Kotz, "A Survey of Context-Aware Mobile Computing Research".
[39] D. Petrelli, E. Not, M. Zancanaro, C. Strapparava, and O. Stock, "Modelling and Adapting to Context," *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 20–24, Feb. 2001, doi: 10.1007/s007790170023.
[40] T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann, and W. Retschitzegger, "Context-awareness on mobile devices - the hydrogen approach," in *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, Big Island, HI, USA: IEEE, 2003, p. 10 pp. doi: 10.1109/HICSS.2003.1174831.
[41] K. Henricksen, "A framework for context-aware pervasive computing applications," PhD Thesis, The University of Queensland, 2003. doi: 10.14264/106832.
[42] M. A. Razzaque, S. Dobson, and P. Nixon, "Categorization and Modelling of Quality in Context Information".
[43] "DEVELOPMENT OF CONTEXT-AWARE APPLICATIONS IN UBIQUITOUS INFORMATION SYSTEMS:," in *Proceedings of the 13th International Conference on Enterprise Information Systems*, Beijing, China: SciTePress - Science and and Technology Publications, 2011, pp. 223–228. doi: 10.5220/0003488802230228.
[44] K. Geihs and M. Wagner, "Context-Awareness for Self-adaptive Applications in Ubiquitous Computing Environments," in *Context-Aware Systems and Applications*, vol. 109, P. C. Vinh, N. M. Hung, N. T. Tung, and J. Suzuki, Eds., in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 109. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 108–120. doi: 10.1007/978-3-642-36642-0_11.
[45] T. Strang and C. Linnhoff-Popien, "A Context Modeling Survey".
[46] G. Nalepa and S. Bobek, "Rule-based solution for context-aware reasoning on mobile devices," *ComSIS*, vol. 11, no. 1, pp. 171–193, 2014, doi: 10.2298/CSIS130209002N.
[47] P. Temdee and R. Prasad, *Context-Aware Communication and Computing: Applications for Smart Environment*. in Springer Series in Wireless Technology. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-319-59035-6.
[48] A. J. Perez and S. Zeadally, "Recent Advances in Wearable Sensing Technologies," *Sensors*, vol. 21, no. 20, p. 6828, Oct. 2021, doi: 10.3390/s21206828.
[49] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *IJAHUC*, vol. 2, no. 4, p. 263, 2007, doi: 10.1504/IJAHUC.2007.014070.
[50] R. Sánchez-Corcuera *et al.*, "Smart cities survey: Technologies, application domains and challenges for the cities of the future," *International Journal of Distributed Sensor Networks*, vol. 15, no. 6, p. 155014771985398, Jun. 2019, doi: 10.1177/1550147719853984.
[51] S. C. Christopoulou, "Impacts on Context Aware Systems in Evidence-Based Health Informatics: A Review," *Healthcare*, vol. 10, no. 4, p. 685, Apr. 2022, doi: 10.3390/healthcare10040685.
[52] M. Dobie, R. Tansley, D. Joyce, M. Weal, P. Lewis, and W. Hall, "A Flexible Architecture for Content and Concept Based Multimedia Information Exploration," presented at the Challenge of Image Retrieval, Feb. 1999. doi: 10.14236/ewic/CIR1999.5.
[53] M. J. Swain, "Image and Video Searching on the World Wide Web," presented at the Challenge of Image Retrieval, Feb. 1999. doi: 10.14236/ewic/CIR1999.11.
[54] BETTOU Farida, "Gestion de la Qualité de Service dans les Applications Multimédia basée sur les préférences de l'utilisateur et orientée vers la Programmation Orientée Aspect," Université Constantine 2 - Abdelhamid Mehri, Constantine, 2027. [Online]. Available: https://www.univ-constantine2.dz/files/Theses/Informatique/BETTOU-Farida.pdf
[55] S. Laborie, J. Euzenat, and N. Layaïda, "Adaptation spatiale efficace de documents SMIL Effective spatial adaptation of SMIL documents".

[56] LABORIE Sébastien, “Adaptation de documents multimédia : Approche sémantique de la dimension spatiotemporelle des documents SMIL,” Mémoire de Master de recherche, Université Joseph Fourier-INPG, Grenoble, 2004. [Online]. Available: https://exmo.inria.fr/files/reports/m2r-laborie.pdf
[57] Y. Belhadad, A. Refoufi, and P. Roose, “Spatial reasoning about multimedia document for a profile based adaptation: Combining distances, directions and topologies,” *Multimed Tools Appl*, vol. 77, no. 23, pp. 30437–30474, Dec. 2018, doi: 10.1007/s11042-018-6080-8.
[58] N. Zulkiply, “Examining the redundancy effect in a multimedia presentation on retention of French vocabulary,” *ILS*, vol. 3, no. 2, Dec. 2014, doi: 10.33736/ils.1659.2014.
[59] Liang Wang, “Across Browsers SVG Implementation.” Apr. 2009. [Online]. Available: https://arxiv.org/pdf/1101.0243
[60] A. P. D. Godoy, C. C. Viel, E. L. Melo, D. R. C. Dias, L. C. Trevelin, and C. A. C. Teixeira, “Multimedia Presentation Integrating Media with Virtual 3D Realistic Environment Produced in Real Time with High Performance Processing,” *JIS*, vol. 5, no. 1, Jul. 2014, doi: 10.5753/jis.2014.641.
[61] A. Martin, H. Iribas, I. Alberdi, and N. Aginako, “Automatic Multimedia Creation Enriched with Dynamic Conceptual Data,” *IJIMAI*, vol. 1, no. 7, p. 44, 2012, doi: 10.9781/ijimai.2012.175.
[62] D. F. L. Souza, L. S. Machado, and T. A. Tavares, “3D Technologies to Extend Brazilian DTV Middleware,” *JIS*, vol. 2, no. 1, May 2011, doi: 10.5753/jis.2011.564.
[63] A. Almutairi, “A Comparative Study on Steganography Digital Images: A Case Study of Scalable Vector Graphics (SVG) and Portable Network Graphics (PNG) Images Formats,” *ijacsa*, vol. 9, no. 1, 2018, doi: 10.14569/IJACSA.2018.090123.
[64] T. Hu, R. Yi, B. Qian, J. Zhang, P. L. Rosin, and Y.-K. Lai, “SuperSVG: Superpixel-Based Scalable Vector Graphics Synthesis,” in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2024, pp. 24892–24901. doi: 10.1109/CVPR52733.2024.02351.
[65] T. Zhang, H. Liu, P. Zhang, Y. Cheng, and H. Wang, “Beyond Pixels: Exploring Human-Readable SVG Generation for Simple Images with Vision Language Models,” Nov. 27, 2023, *arXiv*: arXiv:2311.15543. doi: 10.48550/arXiv.2311.15543.
[66] F. G. Hamza-Lup, S. Farrar, and E. Leon, “Interactive X-ray and proton therapy training and simulation,” *Int J CARS*, vol. 10, no. 10, pp. 1675–1683, Oct. 2015, doi: 10.1007/s11548-015-1229-7.
[67] F. P. A. Vogt, C. I. Owen, L. Verdes-Montenegro, and S. Borthakur, “Advanced Data Visualization in Astrophysics: the X3D Pathway,” *ApJ*, vol. 818, no. 2, p. 115, Feb. 2016, doi: 10.3847/0004-637X/818/2/115.
[68] P. Rojtberg and B. Audenrith, “x3ogre: Connecting X3D to a state of the art rendering engine,” in *Proceedings of the 22nd International Conference on 3D Web Technology*, Jun. 2017, pp. 1–5. doi: 10.1145/3055624.3075949.
[69] I. Sopin and F. G. Hamza-Lup, “Extending the Web3D: design of conventional GUI libraries in X3D,” in *Proceedings of the 15th International Conference on Web 3D Technology*, Los Angeles California: ACM, Jul. 2010, pp. 137–146. doi: 10.1145/1836049.1836070.
[70] M. A. Achachlouei, O. Patil, T. Joshi, and V. N. Nair, “Document Automation Architectures: Updated Survey in Light of Large Language Models”.
[71] Utrecht University, NL and J. Odijk, Eds., “Enriching a Scientific Grammar with Links to Linguistic Resources: The Taalportaal,” in *CLARIN in the Low Countries*, Ubiquity Press, 2017, pp. 299–310. doi: 10.5334/bbi.24.
[72] M. A. Achachlouei, O. Patil, T. Joshi, and V. N. Nair, “Document Automation Architectures and Technologies: A Survey”.
[73] J. Mikác, C. Roisin, and B. L. Duc, “An export architecture for a multimedia authoring environment,” *Proceedings of the eighth ACM symposium on Document engineering*, 2008, [Online]. Available: https://api.semanticscholar.org/CorpusID:2998776

[74] M. Atay, "DEVELOPMENT AND MAINTENANCE OF XML-BASED VERSUS HTML-BASED WEBSITES: A CASE STUDY".

[75] M. D. Steinberg, "Software solutions for form-based collection of data and the semantic enrichment of form data," Jan. 30, 2019, *arXiv*: arXiv:1901.11053. doi: 10.48550/arXiv.1901.11053.

[76] M. Derdour, P. Roose, M. Dalmau, and N. Ghoualmi-Zine, "An adaptation platform for multimedia applications CSC (component, service, connector)," *Journal of Systems and Information Technology*, vol. 14, no. 1, pp. 4–22, Mar. 2012, doi: 10.1108/13287261211221119.

[77] Q. P. Hai, S. Laborie, and P. Roose, "On-the-fly Multimedia Document Adaptation Architecture," *Procedia Computer Science*, vol. 10, pp. 1188–1193, 2012, doi: 10.1016/j.procs.2012.06.171.

[78] M. Derdour, P. Roose, M. Dalmau, and N. Ghoualmi-Zine, "An adaptation platform for multimedia applications CSC (component, service, connector)," *Journal of Systems and Information Technology*, vol. 14, no. 1, Art. no. 1, Mar. 2012, doi: 10.1108/13287261211221119.

[79] Zhijun Lei and N. D. Georganas, "Context-based media adaptation in pervasive computing," in *Canadian Conference on Electrical and Computer Engineering 2001. Conference Proceedings (Cat. No.01TH8555)*, Toronto, Ont., Canada: IEEE, 2001, pp. 913–918. doi: 10.1109/CCECE.2001.933563.

[80] Z. Laboudi, A. Moudjari, A. Saighi, A. Draa, and S. Hadjadj, "An adaptive context-aware optimization framework for multimedia adaptation service selection," *Neural Comput & Applic*, vol. 34, no. 17, pp. 14239–14251, Sep. 2022, doi: 10.1007/s00521-021-06644-w.

[81] S. Kumar and M. Singh, "Big data analytics for healthcare industry: Impact, applications, and tools," *Big Data Mining and Analytics*, vol. 2, no. 1, pp. 48–57, 2019, doi: 10.26599/BDMA.2018.9020031.

[82] L. M. Ang, K. P. Seng, G. K. Ijemaru, and A. M. Zungeru, "Deployment of IoV for Smart Cities: Applications, Architecture, and Challenges," *IEEE Access*, vol. 7, pp. 6473–6492, 2019, doi: 10.1109/ACCESS.2018.2887076.

[83] B. P. L. Lau *et al.*, "A survey of data fusion in smart city applications," *Information Fusion*, vol. 52, no. January, pp. 357–374, 2019, doi: 10.1016/j.inffus.2019.05.004.

[84] Y. Wu *et al.*, "Large scale incremental learning," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 374–382, 2019, doi: 10.1109/CVPR.2019.00046.

[85] A. Mosavi, S. Shamshirband, E. Salwana, K. wing Chau, and J. H. M. Tah, "Prediction of multi-inputs bubble column reactor using a novel hybrid model of computational fluid dynamics and machine learning," *Engineering Applications of Computational Fluid Mechanics*, vol. 13, no. 1, pp. 482–492, 2019, doi: 10.1080/19942060.2019.1613448.

[86] V. Palanisamy and R. Thirunavukarasu, "Implications of big data analytics in developing healthcare frameworks – A review," *Journal of King Saud University - Computer and Information Sciences*, vol. 31, no. 4, pp. 415–425, 2019, doi: 10.1016/j.jksuci.2017.12.007.

[87] P. N. Mahalle and P. S. Dhotre, *Context-Aware Pervasive Systems and Applications*, vol. 169. in Intelligent Systems Reference Library, vol. 169. Singapore: Springer Singapore, 2020. doi: 10.1007/978-981-32-9952-8.

[88] I. E. Guabassi, Z. Bousalem, M. Al Achhab, I. Jellouli, and B. E. El Mohajir, "Identifying learning style through eye tracking technology in adaptive learning systems," *IJECE*, vol. 9, no. 5, p. 4408, Oct. 2019, doi: 10.11591/ijece.v9i5.pp4408-4416.

[89] G. Leetch and E. Mangina, "A Multi-Agent System to Stream Multimedia to Handheld Devices," in *Sixth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'05)*, Las Vegas, NV, USA: IEEE, 2005, pp. 2–10. doi: 10.1109/ICCIMA.2005.6.

[90] Q. P. Hai, S. Laborie, and P. Roose, "On-the-fly Multimedia Document Adaptation Architecture," *Procedia Computer Science*, vol. 10, pp. 1188–1193, 2012, doi: 10.1016/j.procs.2012.06.171.

[91] D. Jannach and K. Leopold, "Knowledge-based multimedia adaptation for ubiquitous multimedia consumption," *Journal of Network and Computer Applications*, vol. 30, no. 3, pp. 958–982, Aug. 2007, doi: 10.1016/j.jnca.2005.12.007.
[92] C. Plesca, V. Charvillat, and R. Grigoras, "Adapting Content Delivery to Limited Resources and Inferred User Interest," *International Journal of Digital Multimedia Broadcasting*, vol. 2008, pp. 1–13, 2008, doi: 10.1155/2008/171385.
[93] Z. Kazi-Aoul, I. M. Demeure, and J.-C. Moissinac, "PAAM: a web services oriented architecture for the adaptation of composed multimedia documents," 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:64468735
[94] P. N. M. Sampaio, R. I. C. De Freitas, and G. N. P. Cardoso, "Applying Multimedia and Virtual Reality for Learning Environments," *Int. J. Emerg. Technol. Learn.*, vol. 4, no. 6, p. 32, Oct. 2009, doi: 10.3991/ijet.v4s2.912.
[95] M. Derdour, P. Roose, M. Dalmau, N. Ghoualmi Zine, and A. Alti, "MMSA: Metamodel Multimedia Software Architecture," *Advances in Multimedia*, vol. 2010, pp. 1–17, 2010, doi: 10.1155/2010/386035.
[96] S. Tönnies, B. Köhncke, P. Hennig, I. Brunkhorst, and W.-T. Balke, "A Service Oriented Architecture for Personalized Universal Media Access," *Future Internet*, vol. 3, no. 2, pp. 87–116, Apr. 2011, doi: 10.3390/fi3020087.
[97] A. Martin, H. Iribas, I. Alberdi, and N. Aginako, "Automatic Multimedia Creation Enriched with Dynamic Conceptual Data," *IJIMAI*, vol. 1, no. 7, p. 44, 2012, doi: 10.9781/ijimai.2012.175.
[98] F. M. Bouyakoub and A. Belkhir, "A Service Discovery Approach Based on a Quantitative Similarity Measure for M-Tourism Platforms," *CN*, vol. 04, no. 03, pp. 227–239, 2012, doi: 10.4236/cn.2012.43027.
[99] C. Dromzée, S. Laborie, P. Roose, Ed., "A semantic generic profile for multimedia document adaptation," in *Intelligent Multimedia Technologies for Networking Applications: Techniques and Tools*, IGI Global, 2013. doi: 10.4018/978-1-4666-2833-5.CH009.
[100] A. P. D. Godoy, C. C. Viel, E. L. Melo, D. R. C. Dias, L. C. Trevelin, and C. A. C. Teixeira, "Multimedia Presentation Integrating Media with Virtual 3D Realistic Environment Produced in Real Time with High Performance Processing," *JIS*, vol. 5, no. 1, Jul. 2014, doi: 10.5753/jis.2014.641.
[101] A.-E. Maredj and N. Tonkin, "CSP-based adaptation of multimedia document composition," in *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, Anaheim, CA, USA: IEEE, Feb. 2015, pp. 232–235. doi: 10.1109/ICOSC.2015.7050811.
[102] E. Vildjiounaite, G. Gimel'farb, V. Kyllönen, and J. Peltola, "Lightweight Adaptation of Classifiers to Users and Contexts: Trends of the Emerging Domain," *The Scientific World Journal*, vol. 2015, no. 1, p. 434826, Jan. 2015, doi: 10.1155/2015/434826.
[103] A. Alti, S. Laborie, and P. Roose, "A Community-Based Semantic Social Context-Aware Driven Adaptation for Multimedia Documents," *Int. J. Virtual Communities Soc. Netw.*, vol. 7, no. 2, pp. 31–49, Apr. 2015.
[104] A. Adel, R. Philippe, and L. Sébastien, "Multimedia Documents Adaptation Based on Semantic Multi-Partite Social Context-Aware Networks:," *International Journal of Virtual Communities and Social Networking*, vol. 9, no. 3, pp. 44–59, Jul. 2017, doi: 10.4018/IJVCSN.2017070104.
[105] H. Khallouki and M. Bahaj, "Multimedia documents adaptive platform using multi-agent system and mobile ubiquitous environment," in *2017 Intelligent Systems and Computer Vision (ISCV)*, Fez, Morocco: IEEE, Apr. 2017, pp. 1–5. doi: 10.1109/ISACV.2017.8054915.
[106] I. El Guabassi, Z. Bousalem, M. Al Achhab, I. Jellouli, and B. E. El Mohajir, "Personalized adaptive content system for context-aware ubiquitous learning," *Procedia Computer Science*, vol. 127, pp. 444–453, 2018, doi: 10.1016/j.procs.2018.01.142.
[107] R. Fernandes and M. T. Andrade, "Multimedia Content Classification Metrics for Content Adaptation," *UPjeng*, vol. 2, no. 2, pp. 14–25, Mar. 2018, doi: 10.24840/2183-6493_002.002_0003.

[108] R. A. W. Tortorella, D. Kinshuk, and N.-S. Chen, "Framework for designing context-aware learning systems," *Educ Inf Technol*, vol. 23, no. 1, pp. 143–164, Jan. 2018, doi: 10.1007/s10639-017-9591-4.

[109] H. Khallouki and M. Bahaj, "Multimodal Generic Framework for Multimedia Documents Adaptation," *IJIMAI*, vol. 5, no. 4, p. 122, 2019, doi: 10.9781/ijimai.2018.02.009.

[110] S. Sarwar, Z. U. Qayyum, R. García-Castro, M. Safyan, and R. F. Munir, "Ontology based E-learning framework: A personalized, adaptive and context aware model," *Multimed Tools Appl*, vol. 78, no. 24, pp. 34745–34771, Dec. 2019, doi: 10.1007/s11042-019-08125-8.

[111] A. Lakehal, A. Alti, S. Laborie, and P. Roose, "A Semantic Agile Approach for Reconfigurable Distributed Applications in Pervasive Environments:," *International Journal of Ambient Computing and Intelligence*, vol. 11, no. 2, pp. 48–67, Apr. 2020, doi: 10.4018/IJACI.2020040103.

[112] A. R. Khan, U. Rashid, K. Saleem, and A. Ahmed, "An architecture for non-linear discovery of aggregated multimedia document web search results," *PeerJ Computer Science*, vol. 7, p. e449, Apr. 2021, doi: 10.7717/peerj-cs.449.

[113] J. Qiu *et al.*, "MHMS: Multimodal Hierarchical Multimedia Summarization," 2022, *arXiv*. doi: 10.48550/ARXIV.2204.03734.

[114] J. Qiu *et al.*, "Semantics-Consistent Cross-domain Summarization via Optimal Transport Alignment," 2022, *arXiv*. doi: 10.48550/ARXIV.2210.04722.

[115] C. Gautier, J. Delanoy, and G. Gesquière, "INTEGRATING MULTIMEDIA DOCUMENTS IN 3D CITY MODELS FOR A BETTER UNDERSTANDING OF TERRITORIES," *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. X-4/W2-2022, pp. 69–76, Oct. 2022, doi: 10.5194/isprs-annals-X-4-W2-2022-69-2022.

[116] F. Bettou and B. Boulkroun, "A Multi-Viewpoint Approach For Semantic Multimedia Documents Adaptation," Aug. 2023. doi: 10.11159/cist23.107.

[117] D. Jannach and K. Leopold, "Knowledge-based multimedia adaptation for ubiquitous multimedia consumption," *Journal of Network and Computer Applications*, vol. 30, no. 3, pp. 958–982, Aug. 2007, doi: 10.1016/j.jnca.2005.12.007.

[118] F. Bettou and B. Boulkroun, "A Multi-Viewpoint Approach For Semantic Multimedia Documents Adaptation," presented at the The 9th World Congress on Electrical Engineering and Computer Systems and Science, Aug. 2023. doi: 10.11159/cist23.107.

[119] F. Piccialli, S. Cuomo, V. S. D. Cola, and G. Casolla, "A machine learning approach for IoT cultural data," *J Ambient Intell Human Comput*, vol. 15, no. 2, pp. 1715–1726, Feb. 2024, doi: 10.1007/s12652-019-01452-6.

[120] S.-T. Park, G. Li, and J.-C. Hong, "A study on smart factory-based ambient intelligence context-aware intrusion detection system using machine learning," *J Ambient Intell Human Comput*, vol. 11, no. 4, pp. 1405–1412, Apr. 2020, doi: 10.1007/s12652-018-0998-6.

[121] Iqbal Sarker, Alan Colman, Jun Han, Paul Watters, *Automated Rule-based Services with Intelligent Decision-Making*, 1st ed. 2021. in Context-Aware Machine Learning and Mobile Data Analytics: Springer; 1st ed. 2021 edition (December 2, 2021).

[122] I. H. Sarker, "Data Science and Analytics: An Overview from Data-Driven Smart Computing, Decision-Making and Applications Perspective," *SN COMPUT. SCI.*, vol. 2, no. 5, p. 377, Sep. 2021, doi: 10.1007/s42979-021-00765-8.

[123] I. H. Sarker, A. I. Khan, Y. B. Abushark, and F. Alsolami, "Mobile Expert System: Exploring Context-Aware Machine Learning Rules for Personalized Decision-Making in Mobile Applications," *Symmetry*, vol. 13, no. 10, p. 1975, Oct. 2021, doi: 10.3390/sym13101975.

[124] I. Đuric, D. Barac, Z. Bogdanovic, A. Labus, and B. Radenkovic, "Model of an intelligent smart home system based on ambient intelligence and user profiling," *J Ambient Intell Human Comput*, vol. 14, no. 5, pp. 5137–5149, May 2023, doi: 10.1007/s12652-021-03081-4.

[125] S. Pal, P. Kanti Dutta Pramanik, A. Nayyar, and P. Choudhury, "A Personalised Recommendation Framework for Ubiquitous Learning System," in *2021 6th International Conference on Intelligent Information Technology*, Ho Chi Minh Viet Nam: ACM, Feb. 2021, pp. 63–72. doi: 10.1145/3460179.3460190.

[126] Y.-M. Chen and W.-C. Wu, "An anonymous DRM scheme for sharing multimedia files in P2P networks," *Multimed Tools Appl*, vol. 69, no. 3, pp. 1041–1065, Apr. 2014, doi: 10.1007/s11042-012-1166-1.
[127] N. Chidambaram, P. Raj, K. Thenmozhi, S. Rajagopalan, and R. Amirtharajan, "A cloud compatible DNA coded security solution for multimedia file sharing & storage," *Multimed Tools Appl*, vol. 78, no. 23, pp. 33837–33863, Dec. 2019, doi: 10.1007/s11042-019-08166-z.
[128] A. Bala and M. Z. Shuaibu, "Performance Analysis of Apriori and FP-Growth Algorithms (Association Rule Mining)," vol. 7, 2016.
[129] M. Sinthuja, N. Puviarasan, and P. Aruna, "Evaluating the Performance of Association Rule Mining Algorithms," 2017.
[130] V. Sarlis, G. Papageorgiou, and C. Tjortjis, "Leveraging Sports Analytics and Association Rule Mining to Uncover Recovery and Economic Impacts in NBA Basketball," *Data*, vol. 9, no. 7, p. 83, Jun. 2024, doi: 10.3390/data9070083.
[131] A. Naghshzan, S. Khalilazar, P. Poilane, O. Baysal, L. Guerrouj, and F. Khomh, "Leveraging Data Mining Algorithms to Recommend Source Code Changes," 2023, *arXiv*. doi: 10.48550/ARXIV.2305.00323.
[132] J. Reynaldo and D. Boy Tonara, "Data Mining Application using Association Rule Mining ECLAT Algorithm Based on SPMF," *MATEC Web Conf.*, vol. 164, p. 01019, 2018, doi: 10.1051/matecconf/201816401019.
[133] N. Rostamzadeh, S. S. Abdullah, K. Sedig, A. X. Garg, and E. McArthur, "Visual Analytics for Predicting Disease Outcomes Using Laboratory Test Results," *Informatics*, vol. 9, no. 1, p. 17, Feb. 2022, doi: 10.3390/informatics9010017.
[134] L. Shabtay, R. Yaari, and I. Dattner, "A Guided FP-growth algorithm for multitude-targeted mining of big data," *arXiv: Databases*, 2018, [Online]. Available: https://api.semanticscholar.org/CorpusID:49566914
[135] L. Jia, L. Xiang, and X. Liu, "An Improved Eclat Algorithm Based on Tissue-Like P System with Active Membranes," *Processes*, vol. 7, no. 9, p. 555, Aug. 2019, doi: 10.3390/pr7090555.
[136] M. Martinez, B. Escobar, G.-D. Maria-Elena, and D. P. Pinto-Roa, "Market basket analysis with association rules in the retail sector using Orange. Case Study: Appliances Sales Company," *CLEIej*, vol. 24, no. 2, Aug. 2021, doi: 10.19153/cleiej.24.2.12.
[137] M. R. Al-Bana, M. S. Farhan, and N. A. Othman, "An Efficient Spark-Based Hybrid Frequent Itemset Mining Algorithm for Big Data," *Data*, vol. 7, no. 1, p. 11, Jan. 2022, doi: 10.3390/data7010011.
[138] R. Wahyuningsih, A. Suharsono, and N. Iriawan, "COMPARISON OF MARKET BASKET ANALYSIS METHOD USING APRIORI ALGORITHM, FREQUENT PATTERN GROWTH (FP- GROWTH) AND EQUIVALENCE CLASS TRANSFORMATION (ECLAT) (CASE STUDY: SUPERMARKET 'X' TRANSACTION DATA FOR 2021)," *BFJ*, vol. 8, no. 2, pp. 192–201, Nov. 2023, doi: 10.33086/bfj.v8i2.5226.
[139] M. Kudriavtsev, M. Bezbradica, and A. McCarren, "Exploring the Trie of Rules: a fast data structure for the representation of association rules," 2023, *arXiv*. doi: 10.48550/ARXIV.2310.17355.
[140] M. S. Danessh, C. Balasubramanian, and K. Duraiswamy, "Similarity Data Item Set Approach: An Encoded Temporal Data Base Technique," vol. 2, no. 3, 2010.
[141] O. El Midaoui, B. El Ghali, and A. El Qadi, "Geographical queries reformulation using a parallel association rules generator to build spatial taxonomies," *IJECE*, vol. 11, no. 3, p. 2586, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2586-2594.
[142] J. W. Li, N. Yu, J. W. Jiang, X. Li, Y. Ma, and W. D. Chen, "RESEARCH ON STUDENT BEHAVIOR INFERENCE METHOD BASED ON FP-GROWTH ALGORITHM," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XLII-3/W10, pp. 981–985, Feb. 2020, doi: 10.5194/isprs-archives-XLII-3-W10-981-2020.

[143] L. Afuan, A. Ashari, and Y. Suyanto, "Query Expansion in Information Retrieval using Frequent Pattern (FP) Growth Algorithm for Frequent Itemset Search and Association Rules Mining," *ijacsa*, vol. 10, no. 2, 2019, doi: 10.14569/IJACSA.2019.0100235.
[144] R. Alharith, M. Khalil, A. O. Ibrahim, and S. H. Babiker, "Extraction of Association Rules from Cancer Patient's Records using F-P Growth Algorithm," *ITM Web Conf.*, vol. 63, p. 01017, 2024, doi: 10.1051/itmconf/20246301017.
[145] K. Dahdouh, A. Dakkak, L. Oughdir, and A. Ibriz, "Improving Online Education Using Big Data Technologies," in *The Role of Technology in Education*, F. Altlnay, Ed., IntechOpen, 2020. doi: 10.5772/intechopen.88463.
[146] M. Li, X. Lv, Y. Liu, L. Wang, and J. Song, "TCM Constitution Analysis Method Based on Parallel FP-Growth Algorithm in Hadoop Framework," *Journal of Healthcare Engineering*, vol. 2022, pp. 1–14, Aug. 2022, doi: 10.1155/2022/9006096.
[147] A. Pakzad and M. Analoui, "A rule-based/BPSO approach to produce low-dimensional semantic basis vectors set," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, no. 7, pp. 2586–2604, Nov. 2022, doi: 10.55730/1300-0632.3957.
[148] M. Moshkov, B. Zielosko, and E. T. Tetteh, "Selected Data Mining Tools for Data Analysis in Distributed Environment," *Entropy*, vol. 24, no. 10, p. 1401, Oct. 2022, doi: 10.3390/e24101401.
[149] J. B. Machado, "Automatic Risk Identification in Software Projects: an Approach based on Inductive Learning," *IIM*, vol. 04, no. 05, pp. 291–295, 2012, doi: 10.4236/iim.2012.425041.
[150] S. Liu, Z. Yang, Y. Li, and S. Wang, "Decision Tree-Based Sensitive Information Identification and Encrypted Transmission System," *Entropy*, vol. 22, no. 2, p. 192, Feb. 2020, doi: 10.3390/e22020192.
[151] Department of Computer Science and Engineering, National Engineering College, Tamil Nadu, India, S. N., R. K., and Sri Vidhya College of Engineering and Technology, Tamil Nadu, India, "KNOWLEDGE ENGINEERING TO AID THE RECRUITMENT PROCESS OF AN INDUSTRY BY IDENTIFYING SUPERIOR SELECTION CRITERIA," *IJSC*, vol. 01, no. 03, pp. 138–144, Jan. 2011, doi: 10.21917/ijsc.2011.0022.
[152] Z. R. Rise and M. M. Ershadi, "Application of Ensemble Learning in CXR Classification for Enhancing COVID-19 Diagnosis," *Qeios*, Apr. 2024, doi: 10.32388/1NMNYE.
[153] K.-C. Cheng, M.-J. Huang, C.-K. Fu, K.-H. Wang, H.-M. Wang, and L.-H. Lin, "Establishing a Multiple-Criteria Decision-Making Model for Stock Investment Decisions Using Data Mining Techniques," *Sustainability*, vol. 13, no. 6, p. 3100, Mar. 2021, doi: 10.3390/su13063100.
[154] J. M. Rudd and H. "Gene" Ray, "An Empirical Study of Downstream Analysis Effects of Model Pre-Processing Choices," *OJS*, vol. 10, no. 05, pp. 735–809, 2020, doi: 10.4236/ojs.2020.105046.
[155] J. A. Castellanos-Garzón, Y. Mezquita Martín, J. L. Jaimes Sánchez, S. M. López García, and E. Costa, "A Genetic Programming Strategy to Induce Logical Rules for Clinical Data Analysis," *Processes*, vol. 8, no. 12, p. 1565, Nov. 2020, doi: 10.3390/pr8121565.
[156] C. Wen and Y. Lou, "On Finding Bi-objective Pareto-optimal Fraud Prevention Rule Sets for Fintech Applications," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Barcelona Spain: ACM, Aug. 2024, pp. 5959–5968. doi: 10.1145/3637528.3671521.
[157] S. Kim, M. Jeong, and B. C. Ko, "Interpretation and Simplification of Deep Forest," Jan. 20, 2020. doi: 10.36227/techrxiv.11661246.v1.
[158] M. Kozielski, M. Sikora, and Ł. Wawrowski, "Towards consistency of rule-based explainer and black box model -- fusion of rule induction and XAI-based feature importance," 2024, *arXiv*. doi: 10.48550/ARXIV.2407.14543.
[159] N. Burkart and M. F. Huber, "A Survey on the Explainability of Supervised Machine Learning," *jair*, vol. 70, pp. 245–317, Jan. 2021, doi: 10.1613/jair.1.12228.
[160] J. H. Friedman and B. E. Popescu, "Predictive learning via rule ensembles," *Ann. Appl. Stat.*, vol. 2, no. 3, Sep. 2008, doi: 10.1214/07-AOAS148.
[161] N. Youssef, H. Abdulkader, and A. Abdelwahab, "Detecting the Behaviour of COVID-19 Based On Parallel Approach of Sequential Rule Mining Algorithm," *IJCI. International Journal of Computers and Information*, vol. 0, no. 0, pp. 0–0, Nov. 2021, doi: 10.21608/ijci.2021.79097.1051.

[162] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Processes*, vol. 25, no. 2–3, pp. 259–284, Jan. 1998, doi: 10.1080/01638539809545028.
[163] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Am. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, Sep. 1990, doi: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.
[164] A. Moudjari, S. Chikhi, and H. Kheddouci, "Latent semantic analysis for business protocol discovery using log files," *IJWET*, vol. 9, no. 4, p. 365, 2014, doi: 10.1504/IJWET.2014.067550.
[165] F. Wild, D. Haley, and K. Bülow, "Using Latent-Semantic Analysis and Network Analysis for Monitoring Conceptual Development," *JLCL*, vol. 26, no. 1, pp. 9–21, Jul. 2011, doi: 10.21248/jlcl.26.2011.133.
[166] R. E. Febrita and W. F. Mahmudy, "PRE-PROCESSED LATENT SEMANTIC ANALYSIS FOR AUTOMATIC ESSAY GRADING," *kursor*, p. 175, Oct. 2017, doi: 10.28961/kursor.v8i4.110.
[167] S. Basheer, M. Anbarasi, D. G. Sakshi, and V. Vinoth Kumar, "Efficient text summarization method for blind people using text mining techniques," *Int J Speech Technol*, vol. 23, no. 4, pp. 713–725, Dec. 2020, doi: 10.1007/s10772-020-09712-z.
[168] J.-Y. Yeh, H.-R. Ke, W.-P. Yang, and I.-H. Meng, "Text summarization using a trainable summarizer and latent semantic analysis," *Information Processing & Management*, vol. 41, no. 1, pp. 75–95, Jan. 2005, doi: 10.1016/j.ipm.2004.04.003.
[169] R. A. Sagum, P. A. C. Clacio, R. E. R. Cayetano, and A. D. F. Lobrio, "Philippine Court Case Summarizer using Latent Semantic Analysis," *Procedia Computer Science*, vol. 227, pp. 474–481, 2023, doi: 10.1016/j.procs.2023.10.548.
[170] F. Ba-Alwi, G. Gaphari, and F. Al-Duqaimi, "Arabic Text Summarization Using Latent Semantic Analysis," *BJAST*, vol. 10, no. 2, pp. 1–14, Jan. 2015, doi: 10.9734/BJAST/2015/17678.
[171] Y. Kalmukov, "Comparison of Latent Semantic Analysis and Vector Space Model for Automatic Identification of Competent Reviewers to Evaluate Papers," *IJACSA*, vol. 13, no. 2, 2022, doi: 10.14569/IJACSA.2022.0130209.
[172] A. Patil, "Word Significance Analysis in Documents for Information Retrieval by LSA and TF-IDF using Kubeflow," in *Expert Clouds and Applications*, vol. 209, I. Jeena Jacob, F. M. Gonzalez-Longatt, S. Kolandapalayam Shanmugam, and I. Izonin, Eds., in Lecture Notes in Networks and Systems, vol. 209. , Singapore: Springer Singapore, 2022, pp. 335–348. doi: 10.1007/978-981-16-2126-0_29.
[173] A. Pakzad and M. Analoui, "A rule-based/BPSO approach to produce low-dimensional semantic basis vectors set," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, no. 7, Art. no. 7, Nov. 2022, doi: 10.55730/1300-0632.3957.
[174] R. Rajalakshmi, S. Vidhya, D. Harina, R. Karna, and A. Sowmya, "Text Summarization for News Articles using Latent Semantic Analysis Technique," in *4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, in 2023. , Coimbatore, India, 2023, pp. 1421–1425. [Online]. Available: doi:10.1109/ICESC57686.2023.10193508
[175] R. Dzisevic and D. Sesok, "Text Classification using Different Feature Extraction Approaches," in *2019 Open Conference of Electrical, Electronic and Information Sciences (eStream)*, Vilnius, Lithuania: IEEE, Apr. 2019, pp. 1–4. doi: 10.1109/eStream.2019.8732167.
[176] X. Han *et al.*, "Automatic Business Process Structure Discovery using Ordered Neurons LSTM: A Preliminary Study".

## Appendix A

# Implementing Latent Semantic Analysis Steps

### Results of the LSA Analysis Step

Table 1 represent the resulting occurrence matrix $A$ $(m \times n)/(m=30 \times n=5)$.

| Contextual Situation | $LF_1$ (user 1) | $LF_2$ (user 2) | $LF_3$ (user 3) | $LF_4$ (user 4) | $LF_5$ (user 5) |
|---|---|---|---|---|---|
| $t_{01} = (c_1(0), a_1)$ | 2 | 5 | 2 | 1 | 4 |
| $t_{02} = (c_1(1), a_1)$ | 2 | 2 | 2 | 1 | 2 |
| $t_{03} = (c_1(2), a_1)$ | 2 | 4 | 3 | 1 | 1 |
| $t_{04} = (c_1(0), a_2)$ | 1 | 2 | 3 | 3 | 3 |
| $t_{05} = (c_1(1), a_2)$ | 3 | 4 | 2 | 1 | 1 |
| $t_{06} = (c_1(2), a_2)$ | 3 | 3 | 1 | 2 | 2 |
| $t_{07} = (c_1(0), a_3)$ | 1 | 1 | 1 | 1 | 0 |
| $t_{08} = (c_1(1), a_3)$ | 1 | 2 | 1 | 4 | 1 |
| $t_{09} = (c_1(2), a_3)$ | 5 | 4 | 4 | 4 | 2 |
| $t_{10} = (c_2(0), a_1)$ | 2 | 4 | 3 | 2 | 4 |
| $t_{11} = (c_2(1), a_1)$ | 3 | 4 | 3 | 1 | 2 |
| $t_{12} = (c_2(2), a_1)$ | 1 | 3 | 1 | 0 | 1 |
| $t_{13} = (c_2(0), a_2)$ | 5 | 3 | 3 | 3 | 3 |
| $t_{14} = (c_2(1), a_2)$ | 2 | 3 | 3 | 2 | 1 |
| $t_{15} = (c_2(2), a_2)$ | 0 | 3 | 0 | 1 | 2 |
| $t_{16} = (c_2(0), a_3)$ | 1 | 3 | 3 | 0 | 0 |
| $t_{17} = (c_2(1), a_3)$ | 0 | 1 | 2 | 6 | 0 |
| $t_{18} = (c_2(2), a_3)$ | 6 | 3 | 1 | 3 | 3 |
| $t_{19} = (c_3(0), a_1)$ | 2 | 7 | 5 | 2 | 4 |
| $t_{20} = (c_3(1), a_1)$ | 4 | 4 | 2 | 1 | 3 |
| $t_{21} = (c_3(0), a_2)$ | 5 | 7 | 4 | 4 | 3 |
| $t_{22} = (c_3(1), a_2)$ | 2 | 2 | 2 | 2 | 3 |
| $t_{23} = (c_3(0), a_3)$ | 5 | 4 | 4 | 6 | 3 |
| $t_{24} = (c_3(1), a_3)$ | 2 | 3 | 2 | 3 | 0 |
| $t_{25} = (c_4(0), a_1)$ | 3 | 7 | 5 | 2 | 5 |
| $t_{26} = (c_4(1), a_1)$ | 3 | 4 | 2 | 1 | 2 |
| $t_{27} = (c_4(0), a_2)$ | 2 | 5 | 3 | 2 | 3 |
| $t_{28} = (c_4(1), a_2)$ | 5 | 4 | 3 | 4 | 3 |
| $t_{29} = (c_4(0), a_3)$ | 2 | 4 | 0 | 4 | 0 |
| $t_{30} = (c_4(1), a_3)$ | 5 | 3 | 6 | 5 | 3 |

**Table 1.** Content of occurrence matrix $A$.

Table 2 represent the content of matrix $U$ $(m \times m)/(m=30)$.

| m \ r | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -0.183 | -0.263 | 0.0184 | 0.0537 | 0.2871 | -0.188 | 0.0181 | -0.037 | -0.083 | -0.262 | -0.145 | -0.121 | -0.185 | -0.02 | -0.209 | 0.0345 | 0.1438 | -0.28 | -0.258 | -0.27 | -0.234 | -0.179 | -0.126 | 0.024 | -0.341 | -0.187 | -0.217 | -0.197 | -0.072 | -0.03 |
| 2 | -0.113 | -0.046 | -0.061 | -0.129 | 0.0226 | -0.041 | -0.024 | 0.095 | -0.223 | 0.086 | -0.006 | 0.1649 | -0.316 | 0.0149 | 0.2989 | 0.1147 | 0.0942 | -0.361 | 0.3371 | -0.093 | 0.0784 | -0.081 | -0.217 | 0.0651 | 0.2203 | 0.0243 | 0.2011 | -0.203 | 0.2391 | -0.391 |
| 3 | -0.148 | -0.122 | 0.0844 | 0.0535 | -0.278 | 0.1668 | -0.064 | 0.1146 | -0.15 | -0.074 | -0.157 | -0.0009 | -0.02 | -0.218 | 0.2041 | -0.357 | -0.068 | 0.3263 | -0.3 | 0.068 | -0.02 | -0.014 | -0.039 | -0.06 | -0.234 | 0.0018 | -0.081 | 0.0469 | 0.3257 | -0.425 |
| 4 | -0.144 | 0.0668 | 0.208 | -0.282 | 0.2259 | 0.0143 | -0.017 | -0.332 | -0.047 | -0.092 | 0.1309 | 0.1075 | 0.0153 | -0.041 | -0.086 | 0.1219 | -0.613 | 0.0952 | -0.035 | 0.1813 | -0.005 | -0.108 | -0.28 | -0.116 | 0.0102 | 0.1359 | -0.045 | -0.067 | -0.188 | -0.208 |
| 5 | -0.148 | -0.094 | -0.095 | 0.1839 | -0.217 | -0.081 | -0.129 | -0.069 | -0.239 | 0.2791 | -0.11 | -0.11 | -0.002 | -0.148 | 0.0898 | -0.237 | -0.096 | -0.104 | 0.0819 | -0.022 | -0.309 | 0.2485 | -0.09 | -0.348 | 0.1902 | -0.13 | 0.0307 | -0.091 | -0.478 | 0.0405 |
| 6 | -0.141 | 0.0183 | -0.148 | 0.1424 | 0.1192 | 0.9372 | 0.0082 | -0.034 | -0.006 | -0.021 | 0.0032 | -0.013 | -0.039 | 0.0282 | -0.057 | 0.0669 | 0.0169 | -0.116 | 0.0232 | -0.055 | -0.042 | -0.026 | -0.034 | 0.0021 | -0.002 | -0.03 | -0.014 | -0.053 | -0.078 | 0.0465 |
| 7 | -0.052 | 0.0521 | 0.0143 | 0.0277 | -0.12 | -0.008 | 0.9825 | -0.013 | -0.043 | 0.0204 | -0.018 | -0.006 | -0.015 | -0.026 | 0.0188 | -0.032 | -0.028 | -0.014 | -0.0004 | -0.003 | -0.039 | 0.0159 | -0.031 | -0.042 | 0.0099 | -0.014 | -0.0008 | -0.022 | -0.041 | -0.032 |
| 8 | -0.111 | 0.2281 | 0.1962 | 0.147 | 0.1798 | -0.023 | -0.006 | 0.8698 | -0.007 | -0.019 | 0.0446 | 0.0126 | 0.0176 | -0.004 | -0.068 | 0.0531 | -0.197 | 0.0005 | -0.006 | 0.0444 | -0.035 | -0.019 | -0.085 | -0.058 | 0.0129 | 0.0255 | -0.025 | -0.029 | -0.143 | -0.0008 |
| 9 | -0.241 | 0.1679 | -0.107 | -0.037 | -0.212 | -0.041 | -0.035 | -0.01 | 0.8689 | 0.0078 | -0.06 | -0.002 | -0.105 | -0.058 | 0.0585 | -0.048 | -0.022 | -0.108 | 0.0016 | -0.056 | -0.092 | -0.012 | -0.112 | -0.065 | -0.008 | -0.047 | -0.005 | -0.099 | -0.033 | -0.144 |
| 10 | -0.189 | -0.131 | 0.0846 | -0.154 | 0.2489 | 0.0113 | 0.033 | 0.0157 | 0.0653 | 0.8754 | -0.01 | -0.015 | 0.013 | 0.0143 | -0.075 | 0.016 | 0.0405 | 0.0577 | -0.129 | -0.021 | 0.0296 | -0.077 | 0.0353 | 0.0788 | -0.148 | -0.002 | -0.073 | 0.0258 | 0.1078 | 0.0063 |
| 11 | -0.171 | -0.131 | -0.062 | -0.017 | -0.168 | -0.017 | -0.009 | 0.0422 | -0.031 | -0.018 | 0.9398 | -0.042 | -0.024 | -0.029 | -0.0006 | -0.063 | 0.0855 | -0.02 | -0.065 | -0.056 | -0.057 | 0.0039 | 0.0133 | -0.011 | -0.067 | -0.054 | -0.036 | -0.017 | 0.0099 | -0.007 |
| 12 | -0.084 | -0.169 | 0.0194 | 0.1478 | -0.05 | -0.021 | 0.003 | -0.002 | 0.0304 | -0.006 | -0.017 | 0.9488 | 0.0398 | 0.0032 | -0.064 | -0.02 | 0.0429 | 0.0093 | -0.051 | -0.023 | -0.046 | 0.0241 | 0.0476 | -0.016 | -0.039 | -0.036 | -0.04 | 0.018 | -0.074 | 0.1062 |
| 13 | -0.212 | 0.1071 | -0.268 | -0.135 | 0.0181 | -0.048 | -0.003 | 0.029 | -0.083 | -0.042 | -0.042 | 0.0176 | 0.8679 | -0.006 | 0.0434 | 0.0273 | 0.0687 | -0.153 | 0.0224 | -0.089 | -0.029 | -0.066 | -0.085 | 0.028 | -0.029 | -0.037 | 0.0033 | -0.098 | 0.0723 | -0.13 |
| 14 | -0.142 | 0.0153 | 0.0998 | -0.032 | -0.208 | 0.0054 | -0.026 | -0.002 | -0.057 | 0.0014 | -0.046 | -0.023 | -0.009 | 0.9427 | 0.0223 | -0.086 | -0.03 | 0.031 | -0.064 | -0.004 | -0.06 | 0.0162 | -0.029 | -0.057 | -0.042 | -0.026 | -0.03 | -0.015 | -0.022 | -0.064 |
| 15 | -0.08 | -0.131 | 0.1331 | 0.2111 | 0.3176 | -0.025 | 0.0346 | -0.07 | 0.1142 | -0.065 | 0.0466 | -0.036 | 0.0855 | 0.0533 | 0.8403 | 0.0783 | -0.037 | 0.037 | -0.065 | 0.0103 | 0 | -0.021 | 0.0532 | 0.0256 | -0.059 | 0.0044 | -0.063 | 0.042 | -0.102 | 0.1812 |
| 16 | -0.098 | -0.149 | 0.1386 | -0.018 | -0.424 | 0.0265 | -0.036 | 0.0446 | -0.051 | 0.0324 | -0.079 | -0.058 | 0.0294 | -0.08 | 0.0386 | 0.8354 | 0.0317 | 0.0906 | -0.101 | -0.003 | -0.078 | 0.0656 | 0.0333 | -0.076 | -0.056 | -0.048 | -0.037 | 0.025 | -0.019 | -0.024 |
| 17 | -0.105 | 0.4551 | 0.4626 | 0.0379 | 0.0843 | 0.0233 | -0.034 | -0.182 | -0.062 | -0.004 | 0.0496 | 0.044 | 0.0277 | -0.064 | -0.013 | -0.007 | 0.6325 | 0.0951 | -0.031 | 0.1209 | -0.039 | -0.01 | -0.156 | -0.124 | 0.0245 | 0.0652 | -0.022 | -0.027 | -0.148 | -0.132 |
| 18 | -0.2 | 0.1417 | -0.498 | 0.1173 | 0.1873 | -0.121 | 0.011 | -0.015 | -0.068 | -0.019 | -0.004 | 0.0175 | -0.159 | 0.0563 | -0.016 | 0.1451 | 0.0893 | 0.6938 | 0.1273 | -0.133 | -0.049 | -0.07 | -0.106 | 0.0345 | 0.0466 | -0.052 | 0.028 | -0.147 | -0.05 | -0.022 |
| 19 | -0.262 | -0.297 | 0.301 | -0.058 | -0.019 | 0.0314 | 0.0117 | 0.0166 | 0.0677 | -0.105 | -0.052 | -0.078 | 0.0888 | -0.038 | -0.101 | -0.103 | 0.0285 | 0.17 | 0.7658 | -0.0009 | -0.037 | -0.013 | 0.0875 | 0.0084 | -0.201 | -0.034 | -0.124 | 0.0738 | 0.0297 | 0.0663 |
| 20 | -0.182 | -0.134 | -0.259 | 0.0355 | 0.051 | -0.059 | 0.0154 | 0.0398 | -0.003 | -0.044 | -0.035 | -0.031 | -0.06 | 0.0267 | -0.038 | 0.0351 | 0.1405 | -0.123 | -0.011 | 0.9009 | -0.036 | -0.036 | 0.0086 | 0.044 | -0.053 | -0.057 | -0.025 | -0.047 | 0.0137 | 0.0388 |
| 21 | -0.3 | -0.018 | -0.01 | 0.224 | -0.106 | -0.069 | -0.018 | -0.056 | -0.047 | -0.011 | -0.041 | -0.059 | -0.019 | -0.026 | -0.071 | -0.022 | -0.022 | -0.077 | -0.054 | -0.057 | 0.8735 | 0.0105 | -0.046 | -0.075 | -0.044 | -0.067 | -0.058 | -0.058 | -0.167 | 0.0455 |
| 22 | -0.134 | 0.0092 | -0.025 | -0.18 | 0.2336 | 0.0012 | 0.0243 | 0.0125 | 0.0206 | -0.094 | -0.001 | 0.0171 | -0.037 | 0.016 | -0.023 | 0.0418 | 0.0244 | -0.009 | -0.053 | -0.026 | 0.0371 | 0.9188 | -0.015 | 0.0724 | -0.086 | 0.0079 | -0.031 | -0.013 | 0.1148 | -0.057 |
| 23 | -0.272 | 0.309 | -0.017 | -0.06 | 0.0517 | -0.041 | -0.021 | -0.072 | -0.107 | -0.039 | -0.016 | 0.0285 | -0.104 | -0.037 | 0.0185 | 0.0223 | -0.12 | -0.102 | 0.0005 | -0.027 | -0.061 | -0.061 | 0.8454 | -0.044 | -0.017 | -0.01 | -0.015 | -0.107 | -0.035 | -0.164 |
| 24 | -0.131 | 0.1392 | 0.1209 | 0.1968 | -0.204 | -0.03 | -0.038 | -0.081 | -0.075 | 0.0485 | -0.017 | -0.025 | 0.0013 | -0.051 | -0.009 | -0.049 | -0.127 | -0.019 | -0.005 | 0.0125 | -0.105 | 0.0462 | -0.074 | 0.8795 | 0.0323 | -0.025 | -0.016 | -0.042 | -0.181 | -0.013 |
| 25 | -0.285 | -0.306 | 0.1546 | -0.128 | 0.091 | 0.0167 | 0.0267 | 0.0435 | 0.0726 | -0.134 | -0.052 | -0.065 | 0.0464 | -0.013 | -0.098 | -0.059 | 0.0911 | 0.1106 | -0.214 | -0.037 | -0.013 | -0.051 | 0.0835 | 0.0592 | 0.7869 | -0.038 | -0.116 | 0.0523 | 0.094 | 0.0487 |
| 26 | -0.159 | -0.125 | -0.113 | 0.1055 | -0.059 | -0.044 | 0.0004 | 0.0129 | -0.008 | -0.015 | -0.035 | -0.044 | -0.018 | 0.0017 | -0.042 | -0.008 | 0.0778 | -0.064 | -0.031 | -0.063 | -0.06 | 0.002 | 0.0126 | -0.007 | -0.041 | 0.9467 | -0.033 | -0.025 | -0.051 | 0.0564 |
| 27 | -0.195 | -0.151 | 0.1408 | 0.0381 | 0.0738 | -0.006 | 0.0124 | -0.016 | 0.0461 | -0.071 | -0.019 | -0.048 | 0.0449 | -0.006 | -0.091 | -0.022 | 0.0047 | 0.0598 | -0.124 | -0.013 | -0.033 | -0.019 | 0.0366 | 0.0059 | -0.113 | -0.024 | 0.9222 | 0.0261 | -0.024 | 0.0664 |
| 28 | -0.239 | 0.1421 | -0.176 | 0.0062 | 0.054 | -0.062 | -0.006 | -0.027 | -0.073 | -0.034 | -0.023 | 0.0059 | -0.101 | -0.004 | -0.003 | 0.0418 | -0.003 | -0.145 | 0.0178 | -0.069 | -0.057 | -0.052 | -0.097 | -0.009 | -0.017 | -0.035 | -0.011 | 0.903 | -0.028 | -0.078 |
| 29 | -0.134 | 0.1859 | 0.1117 | 0.5822 | 0.0483 | -0.098 | -0.023 | -0.196 | -0.018 | 0.0624 | 0.0518 | -0.04 | 0.0454 | 0.0114 | -0.138 | 0.0752 | -0.214 | -0.098 | 0.0571 | 0.0171 | -0.138 | 0.0567 | -0.087 | -0.15 | 0.0972 | -0.023 | -0.023 | -0.058 | 0.5976 | 0.1656 |
| 30 | -0.269 | 0.2623 | -0.008 | -0.445 | -0.201 | 0.0263 | -0.036 | 0.0431 | -0.164 | -0.053 | -0.085 | 0.0439 | -0.148 | -0.099 | 0.1478 | -0.102 | -0.033 | -0.022 | -0.062 | -0.032 | -0.027 | -0.072 | -0.141 | -0.015 | -0.082 | -0.012 | -0.008 | -0.091 | 0.1865 | 0.657 |



**Table 2.** Part from the content of matrix *U*.

Table 3 represent the content of Singular values matrix *S*.

| | | | | |
|---|---|---|---|---|
| 35.5184 | 0 | 0 | 0 | 0 |
| 0 | 9.2633 | 0 | 0 | 0 |
| 0 | 0 | 6.486 | 0 | 0 |
| 0 | 0 | 0 | 5.3673 | 0 |
| 0 | 0 | 0 | 0 | 4.8741 |

**Table 3.** Content of matrix *S*.

Table 4 represent the content of the transpose matrix of matrix V ($V^T$ matrix).

| | | | | |
|---|---|---|---|---|
| -0.4518 | -0.5792 | -0.4253 | -0.3858 | -0.3615 |
| 0.2124 | -0.4756 | -0.0542 | 0.7999 | -0.2933 |
| -0.8335 | 0.2489 | 0.3287 | 0.349 | -0.116 |
| 0.045 | 0.6078 | -0.6549 | 0.1509 | -0.4206 |
| -0.2323 | -0.0835 | -0.5285 | 0.2585 | 0.77 |

**Table 4.** Content of matrix $V^T$.

Table 5 represent the content of the matrix *A'*.

| | | | | |
|---|---|---|---|---|
| 2.4116 | 4.9122 | 2.8888 | 0.5535 | 3.0573 |
| 1.727 | 2.5292 | 1.7328 | 1.214 | 1.5771 |
| 2.1277 | 3.5757 | 2.2911 | 1.1156 | 2.2277 |
| 2.4486 | 2.6763 | 2.1478 | 2.4737 | 1.6726 |
| 2.1972 | 3.4653 | 2.2887 | 1.34 | 2.1597 |
| 2.3003 | 2.8221 | 2.1223 | 2.0691 | 1.762 |
| 0.9354 | 0.8382 | 0.7578 | 1.0972 | 0.5248 |
| 2.2284 | 1.2765 | 1.5607 | 3.2098 | 0.8042 |
| 4.1897 | 4.2079 | 3.5487 | 4.5397 | 2.6318 |
| 2.7752 | 4.4653 | 2.9208 | 1.6192 | 2.7827 |
| 2.4781 | 4.0851 | 2.6414 | 1.3649 | 2.5454 |
| 1.013 | 2.4667 | 1.3505 | -0.101 | 1.5341 |
| 3.6063 | 3.8812 | 3.1427 | 3.6931 | 2.4259 |
| 2.312 | 2.858 | 2.1404 | 2.0619 | 1.7843 |
| 1.0276 | 2.225 | 1.2758 | 0.1269 | 1.3844 |
| 1.2738 | 2.6625 | 1.549 | 0.2363 | 1.6569 |
| 2.5868 | 0.1633 | 1.3637 | 4.8165 | 0.1169 |
| 3.4931 | 3.4963 | 2.9546 | 3.7947 | 2.1868 |
| 3.619 | 6.695 | 4.1052 | 1.3904 | 4.1689 |
| 2.6489 | 4.3242 | 2.809 | 1.4942 | 2.6945 |
| 4.7729 | 6.2414 | 4.5346 | 3.9743 | 3.8949 |

| | | | | |
|---|---|---|---|---|
| 2.1732 | 2.7223 | 2.0241 | 1.9085 | 1.6994 |
| 4.9792 | 4.2425 | 3.9597 | 6.0223 | 2.6581 |
| 2.3745 | 2.0796 | 1.9075 | 2.8252 | 1.3025 |
| 3.9692 | 7.2049 | 4.4556 | 1.6389 | 4.4868 |
| 2.2986 | 3.8148 | 2.4587 | 1.2449 | 2.3768 |
| 2.8359 | 4.6796 | 3.0243 | 1.5582 | 2.9158 |
| 4.1101 | 4.2846 | 3.5345 | 4.3238 | 2.6788 |
| 2.5177 | 1.9397 | 1.9324 | 3.215 | 1.2168 |
| 4.836 | 4.3825 | 3.9348 | 5.6324 | 2.7439 |
| 2.4116 | 4.9122 | 2.8888 | 0.5535 | 3.0573 |
| 1.727 | 2.5292 | 1.7328 | 1.214 | 1.5771 |
| 2.1277 | 3.5757 | 2.2911 | 1.1156 | 2.2277 |
| 2.4486 | 2.6763 | 2.1478 | 2.4737 | 1.6726 |
| 2.1972 | 3.4653 | 2.2887 | 1.34 | 2.1597 |
| 2.3003 | 2.8221 | 2.1223 | 2.0691 | 1.762 |
| 0.9354 | 0.8382 | 0.7578 | 1.0972 | 0.5248 |

**Table 5.** Content of occurrence matrix *A'*.

**Results of Post-LSA Analysis Step**

Table 6 represent the correlation matrix of users $COR_u$ ($n \times n$)./ ($n$ =5).

| | User1 | User2 | User3 | User4 | User5 |
|---|---|---|---|---|---|
| User1 | 1 | 0.641475 | **0.909569** | 0.785065 | 0.645468 |
| User2 | | 1 | **0.902255** | 0.028419 | **0.999986** |
| User3 | | | 1 | 0.456671 | **0.904492** |
| User4 | | | | 1 | 0.033633 |
| User5 | | | | | 1 |

**Table 6.** The correlation matrix of users $COR_u$ ($n \times n$).

Table 7 represent the correlation matrix of contextual situations $COR_{cs}$ $(m \times m)/(m = 30)$.

| | C1(0) a1 | C1(1) a1 | C1(2) a1 | C1(0) a2 | C1(1) a2 | C1(2) a2 | C1(0) a3 | C1(1) a3 | C1(2) a3 | C2(0) a1 | C2(1) a1 | C2(2) a1 | C2(0) a2 | C2(1) a2 | C2(2) a2 | C2(0) a3 | C2(1) a3 | C2(2) a3 | C3(0) a1 | C3(1) a1 | C3(0) a2 | C3(1) a2 | C3(0) a3 | C3(1) a3 | C4(0) a1 | C4(1) a1 | C4(0) a2 | C4(1) a2 | C4(0) a3 | C4(1) a3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **C1(0) a1** | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **C1(1) a1** | **0.938** | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **C1(2) a1** | **0.989** | **0.979** | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **C1(0) a2** | 0.069 | 0.411 | 0.214 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **C1(1) a2** | **0.975** | **0.992** | **0.997** | 0.291 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | |
| **C1(2) a2** | 0.596 | 0.837 | 0.707 | 0.843 | 0.761 | 1 | | | | | | | | | | | | | | | | | | | | | | | | |
| **C1(0) a3** | -0.534 | -0.209 | -0.405 | 0.806 | -0.331 | 0.361 | 1 | | | | | | | | | | | | | | | | | | | | | | | |
| **C1(1) a3** | -0.809 | -0.555 | -0.714 | 0.531 | -0.656 | -0.009 | **0.929** | 1 | | | | | | | | | | | | | | | | | | | | | | |
| **C1(2) a3** | -0.269 | 0.081 | -0.126 | **0.942** | -0.047 | 0.613 | **0.958** | 0.784 | 1 | | | | | | | | | | | | | | | | | | | | | |
| **C2(0) a1** | **0.980** | **0.988** | **0.999** | 0.265 | **1.000** | 0.742 | -0.357 | -0.677 | -0.074 | 1 | | | | | | | | | | | | | | | | | | | | |
| **C2(1) a1** | **0.986** | **0.983** | **1.000** | 0.236 | **0.998** | 0.722 | -0.385 | -0.699 | -0.104 | **1.000** | 1 | | | | | | | | | | | | | | | | | | | |
| **C2(2) a1** | **0.997** | **0.910** | **0.976** | -0.004 | **0.956** | 0.535 | -0.595 | -0.850 | -0.339 | **0.963** | **0.971** | 1 | | | | | | | | | | | | | | | | | | |
| **C2(0) a2** | -0.001 | 0.345 | 0.145 | **0.997** | 0.223 | 0.802 | 0.846 | 0.589 | **0.963** | 0.196 | 0.166 | -0.074 | 1 | | | | | | | | | | | | | | | | | |
| **C2(1) a2** | 0.623 | 0.855 | 0.731 | 0.824 | 0.782 | **0.999** | 0.328 | -0.044 | 0.586 | 0.765 | 0.745 | 0.564 | 0.781 | 1 | | | | | | | | | | | | | | | | |
| **C2(2) a2** | **1.000** | **0.928** | **0.985** | 0.040 | **0.968** | 0.572 | -0.559 | -0.826 | -0.297 | **0.974** | **0.981** | **0.999** | -0.031 | 0.600 | 1 | | | | | | | | | | | | | | | |
| **C2(0) a3** | **1.000** | **0.933** | **0.987** | 0.056 | **0.972** | 0.585 | -0.545 | -0.816 | -0.282 | **0.978** | 0.984 | 0.998 | -0.014 | 0.613 | **1.00** | 1 | | | | | | | | | | | | | | |
| **C2(1) a3** | -0.899 | -0.692 | -0.825 | 0.375 | -0.778 | -0.184 | 0.851 | **0.985** | 0.664 | -0.795 | -0.813 | -0.929 | 0.439 | -0.217 | -0.911 | -0.905 | 1 | | | | | | | | | | | | | |
| **C2(2) a3** | -0.280 | 0.070 | -0.137 | **0.938** | -0.058 | 0.604 | **0.961** | 0.791 | **1.000** | -0.085 | -0.115 | -0.349 | **0.960** | 0.576 | -0.308 | -0.293 | 0.672 | 1 | | | | | | | | | | | | |
| **C3(0) a1** | **0.998** | **0.957** | **0.996** | 0.130 | **0.986** | 0.643 | -0.482 | -0.772 | -0.210 | **0.990** | **0.994** | **0.991** | 0.059 | 0.669 | **0.996** | **0.997** | -0.871 | -0.221 | 1 | | | | | | | | | | | |
| **C3(1) a1** | **0.984** | **0.985** | **0.999** | 0.247 | **0.999** | 0.730 | -0.374 | -0.690 | -0.092 | **1.000** | **1.000** | **0.968** | 0.178 | 0.753 | **0.978** | **0.981** | -0.806 | -0.103 | **0.993** | 1 | | | | | | | | | | |
| **C3(0) a2** | 0.785 | **0.951** | 0.867 | 0.673 | **0.904** | **0.965** | 0.104 | -0.270 | 0.386 | 0.892 | 0.878 | 0.738 | 0.619 | **0.974** | 0.766 | 0.777 | -0.434 | 0.375 | 0.821 | 0.883 | 1 | | | | | | | | | |
| **C3(1) a2** | 0.668 | 0.884 | 0.769 | 0.789 | 0.818 | **0.996** | 0.272 | -0.102 | 0.537 | 0.802 | 0.783 | 0.612 | 0.743 | **0.998** | 0.646 | 0.658 | -0.274 | 0.527 | 0.712 | 0.790 | **0.985** | 1 | | | | | | | | |
| **C3(0) a3** | -0.604 | -0.290 | -0.481 | 0.753 | -0.410 | 0.281 | **0.996** | **0.957** | **0.930** | -0.435 | -0.462 | -0.660 | 0.798 | 0.247 | -0.627 | -0.614 | **0.892** | **0.934** | -0.554 | -0.451 | 0.020 | 0.190 | 1 | | | | | | | |
| **C3(1) a3** | -0.568 | -0.248 | -0.442 | 0.781 | -0.370 | 0.322 | **0.999** | **0.944** | **0.945** | -0.395 | -0.423 | -0.627 | 0.823 | 0.289 | -0.592 | -0.579 | 0.871 | **0.949** | -0.518 | -0.412 | 0.064 | 0.233 | **0.999** | 1 | | | | | | |
| **C4(0) a1** | **0.997** | **0.961** | **0.997** | 0.144 | **0.989** | 0.654 | -0.470 | -0.762 | -0.196 | **0.992** | **0.996** | **0.989** | 0.074 | 0.680 | **0.995** | **0.996** | -0.864 | -0.207 | **1.000** | **0.994** | 0.829 | 0.722 | -0.542 | -0.505 | 1 | | | | | |
| **C4(1) a1** | **0.987** | **0.981** | **1.000** | 0.228 | **0.998** | 0.716 | -0.392 | -0.704 | -0.112 | **0.999** | **1.000** | **0.973** | 0.159 | 0.740 | **0.982** | **0.985** | -0.817 | -0.123 | **0.995** | **1.000** | 0.874 | 0.778 | -0.468 | -0.430 | **0.996** | 1 | | | | |
| **C4(0) a2** | **0.986** | **0.983** | **1.000** | **0.235** | **0.998** | 0.721 | -0.386 | -0.700 | -0.105 | **1.000** | **1.000** | **0.971** | 0.165 | 0.745 | **0.981** | **0.984** | -0.814 | -0.116 | **0.994** | **1.000** | 0.877 | 0.782 | -0.463 | -0.424 | **0.996** | **1.00** | 1 | | | |
| **C4(1) a2** | -0.136 | 0.216 | 0.010 | **0.979** | 0.090 | 0.715 | **0.910** | 0.692 | **0.991** | 0.062 | 0.032 | -0.208 | **0.991** | 0.690 | -0.165 | -0.149 | 0.556 | **0.989** | -0.076 | 0.044 | 0.507 | 0.647 | 0.872 | 0.892 | -0.061 | 0.024 | 0.031 | 1 | | |
| **C4(0) a3** | -0.695 | -0.403 | -0.582 | 0.669 | -0.516 | 0.164 | **0.979** | **0.985** | 0.880 | -0.539 | -0.564 | -0.745 | 0.720 | 0.130 | -0.716 | -0.704 | **0.940** | 0.885 | -0.650 | -0.555 | -0.100 | 0.071 | **0.993** | **0.987** | -0.639 | -0.571 | -0.565 | 0.807 | 1 | |
| **C4(1) a3** | -0.516 | -0.187 | -0.385 | 0.819 | -0.311 | 0.381 | **1.000** | **0.921** | **0.964** | -0.337 | -0.365 | -0.577 | 0.857 | 0.349 | -0.541 | -0.527 | 0.839 | **0.967** | -0.463 | -0.354 | 0.126 | 0.293 | **0.994** | **0.998** | -0.450 | -0.372 | -0.366 | **0.919** | **0.975** | 1 |



**Table 7.** The correlation matrix of users $COR_u$ $(n \times n)$.