Bulletin of Electrical Engineering and Informatics
Vol. 13, No. 2, April 2024, pp. 1111~1127
ISSN: 2302-9285, DOI: 10.11591/eei.v13i2.5297

On the use of historical data in context-aware multimedia documents adaptation processes

Aziz Smaala[1,2], Zakaria Laboudi[3], Asma Saighi[2,4], Abdelkader Moudjari[5]
[1]Research Laboratory on Computer Science's Complex Systems (RELA(CS)2), University of Oum El Bouaghi, Algeria
[2]Department of Mathematics and Computer Sciences, University of Oum El Bouaghi, Algeria
[3]Department of Networks and Telecommunications, University of Oum El Bouaghi, Algeria
[4]Laboratory of Artificial Intelligence and Autonomous Things (LIAOA), University of Oum El Bouaghi, Algeria
[5]Department of Mathematics and Computer Sciences, Ecole Normale Supérieure, University of Constantine 3, El Khroub, Algeria

Article Info

Article history:

Received Nov 19, 2022
Revised Jul 26, 2023
Accepted Sep 20, 2023

Keywords:

Context-aware adaptation
Historical data
Multimedia contents
Pervasive systems
Rule learning

ABSTRACT

Playing multimedia documents in ubiquitous systems may require content adaptation based on gathered context information and accumulated historical data. Several approaches have already been proposed, in which adaptation actions are performed to provide adapted documents. Nevertheless, these approaches focus mainly on efficient use of context information without involving historical users data to improve the adaptation process. Thus, this paper allows for consideration of historical users data during the execution of the adaptation process. To do so, the context elements and the adaptation actions are first modeled using the oriented-object approach and then converted into relational and NoSQL databases schemes. Finally, algorithms for storing, retrieving and analysing data are designed. The proposal is validated by implementing scenarios through a real prototype. At a first step, the performances are measured to estimate the cost of data processing. The experiments show that NoSQL databases excel in data storage and ease of implementation, while relational databases perform well in data retrieve. At a second step, the proposal usefulness is highlighted by showing how historical data contribute to adaptation rules personalization using data-driven rule learning mechanisms rather than defining them explicitly. The analysis algorithm could retain personalized adaptation rules with confidence degree greater than 90%. Overall, the results are satisfactory.

Corresponding Author:

Zakaria Laboudi
Department of Networks and Telecommunications, University of Oum El Bouaghi
Route de Constantine BP 04000, Oum El Bouaghi, Algeria
Email: laboudizak@gmail.com

1. INTRODUCTION

Ambient intelligence aims to create environments with communication, computation and decision-making capacities. These features help to enhance the way in which users interact with various environments, based on real-time information gathered and historical data accumulated. Ambient intelligence is built around ubiquitous systems (known also as pervasive systems) by introducing a key-element: explicit requirement of intelligence. This way, mobile applications can sense their environment, understand the context of collected data and adapt their behaviour accordingly [1]. The context refers to information used to depict the situation of entities considered relevant to the interaction between a user and its environment.

In this work, we consider the case of playing multimedia documents in ubiquitous systems. These documents include media objects of various natures such as text, image, audio, and video; web pages are a good example. This type of documents plays an important role in several fields such as healthcare, distance

learning and tourism. In pervasive environments, users context may affect the proper presentation of multimedia documents; we cite for instance: noise level, location, screen size, and battery level. One approach is to adapt their content so that they fit as much as possible to the current context. For example, if the user consults multimedia documents through a tablet at home, the system may suggest displaying videos contents through the smart TV since the screen is larger.

In the literature, several approaches have been proposed in order to adapt the content of multimedia documents in context-aware pervasive environment [2]-[21]. These approaches are classified into four categories: server-side adaptation, client-side adaptation, proxy-based adaptation and peer-to-peer adaptation. Within the adaptation process, the user context is first sensed. Then, the constraints making the documents features non-compliant with the context are identified. Finally, adaptation actions are inferred and performed so as to provide adapted documents. Although these approaches show interesting results, they, however, deal only with near real/real-time data collected to adapt the documents in accordance with context changes. In other words, most of them do not handle historical users data (HUD) accumulated within the adaptation process. Indeed, HUD help to improve the overall system behaviour; we mention for instance: applying machine-learning techniques, making adaptation rules learning, and performing data mining processes. Moreover, most of multimedia document adaptation (MDA) processes focus on executing the adaptation module in a single location (clients, proxy or server side), in particular at servers. In several cases, such a choice hinders the efficient execution of MDA processes within context-aware pervasive systems that involve properties such as proactivity, mobility, cross-platform, self-tuning, and adaptation.

Starting from this context, we propose a software component to manage the HUD resulting from the execution of MDA processes for further processing. The proposal allows the storage, retrieval and analysis of historical data as it stores context values and corresponding adaptation actions. To this end, we rely on a flexible and agile architecture that can easily change form (polymorphism) to adapt to different categories of MDA processes, to users preferences (e.g. data sharing and privacy, and personalized processes) and to context requirements (e.g. computation resources). The aim is to provide MDA processes with more options to efficiently process HUD in different locations (clients, proxy and servers sides), depending on different users and context factors. For instance, let consider two users endowed with devices of different capacities: a laptop user and a smartphone user. If the laptop user does not want to share his historical data, it is then better to store and analyze these data through his device. In contrast, if the smartphone user wishes to share his historical data with other users, it is then better to store and analyze these data via a server.

Depending on the category of the adaptation process, three variants of HUD management are proposed in addition to their hybridization: client-side management, proxy-based management, and server-side management. To that end, we first model the context as well as the required adaptation actions using the oriented-object approach. Then, we transform the resulting modelling into relational and NoSQL schemes using conversion procedures. Finally, algorithms for storing, retrieving, and analyzing data are designed. Our proposal is validated through a set of experiments considering scenarios implemented in a real prototype for which the HUD are stored in different ways. In particular, we emphasize the usefulness of our proposal by treating use cases on context-aware adaptation rules learning.

The rest of this paper is organized as follows. Section 2 provides the basics of MDA processes and discusses some related works. Section 3 presents the proposed approach for managing HUD resulting from execution of MDA processes. Section 4 summarizes the experimental part and discusses the obtained numerical results. Finally, the paper ends with some concluding remarks and future directions in section 5.

2. THE COMPREHENSIVE THEORETICAL BASIS

2.1. Multimedia documents adaptation in ubiquitous systems

As the context in pervasive systems is continuously changing over time, some constraints and difficulties may be encountered due to multimedia documents features that conflict with the current context. In this respect, several adaptation approaches have already been proposed, in order to provide documents adapted to contextual situations [2]-[21]. These proposals differ from each other mainly in the way the adaptation process is carried out as well as in the documents types treated. As discussed in [1], context-aware pervasive systems are built around three-layered elements: sensing, thinking and acting. Next, we give details about these elements in the case of MDA processes, as represented in Figure 1.

a. Sensing layer: this layer includes a single step namely context collection. It aims at updating the elements of the context by acquiring data from the physical world using different sources such as sensors. The context includes different types of information that can be classified into physical context (e.g. lighting, and noise level), user-related context (e.g. profiles, and locations), computational context (e.g. network connectivity, and communication bandwidth) and temporal context (e.g. season, year, month, day, and time of day) [1].

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.

b. Thinking layer: this layer comprises three steps namely context-modeling, context-reasoning, and decision-making which act as follows:
   - Context-modeling: the context information collected is raw because it only includes values measured by various sources (low-level context data); hence, it should be analysed further to take desired actions. To this end, the gathered information is organized according to an abstract context model such as key-value models, markup scheme models, graphical models, object-oriented models, logic-based models, and ontology-based models [22] (high-level context data). The goal is to allow processing context information at a higher level that helps to reason upon it.
   - Context-reasoning: the reasoning step involves the high-level context data to identify the constraints and conflicts that hinder playing the documents properly; it usually takes place in two forms: real-time reasoning and on-demand reasoning. Real-time reasoning refers to event driven and time driven context; hence, it is continuously executed in order to infer the required adaptation actions. On-demand reasoning is triggered by a request from a user or application for many purposes such as resources discovery (e.g. surrounding users/objects) and recommendations (e.g. suggestions regarding environmental conditions such as lighting). In the end of this step, a set of adaptation actions are inferred in order to provide adapted documents that fit as much as possible the contextual situation, often through "if ... then" rules.
   - Decision-making: the objective of this step is twofold: the determination of the adaptation actions to be executed and the generation of the adaptation plan. The decision-making step deals with the adaptation actions produced in the context-reasoning step in three possible modes: i) the automatic mode according to which the system keeps these actions without involving the interaction with the user, ii) the semi-automatic mode according to which the system proposes to the user the inferred actions, in order to choose the most suitable among them, and iii) the manual mode according to which the adaptation actions are entirely generated by a request from the user based on specific needs.

     Once the set of adaptation actions has been determined, the adaptation plan is generated by binding each action to a set of possible execution paths, depending on the available candidate adaptation services. Then, the best paths are selected based on a service selection procedure according to QoS parameters (e.g. response time).

c. Acting layer: the last step consists in performing the adaption plan by executing the selected adaptation services, in order to meet the requirements of the environment and the user. These services are applied to media objects composing the documents to produce a set of adapted objects. These contents are then arranged in an adapted document so that it is displayed in a suitable form to the current contextual situation.

Let us consider a mobile user consulting online course. For the sake of simplicity, we assume that the context elements include the location, current activity and preferred language of the user. Table 1 gives some typical context values, conflicts, and adaptation actions related to the MDA process used.

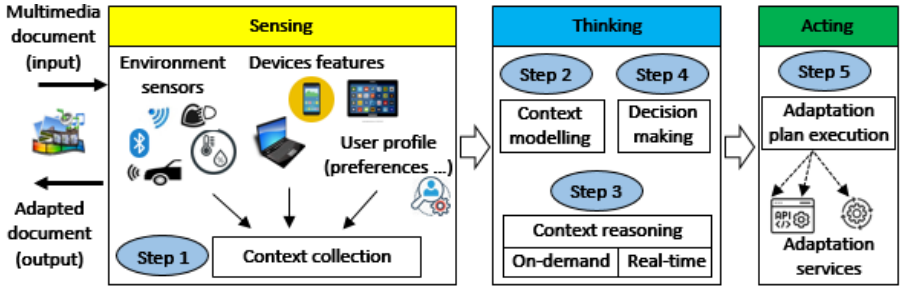

Figure 1. The general architecture of a standard MDA process

Table 1. Some typical context elements, 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

2.2. Involving historical users data in context-aware multimedia documents adaptation processes

In the litterature, much research has been conducted on context-aware maultimeida adaptation [2]-[21]. Depending on where decision-making and adaptation actions take place, MDA processes can be

divided into four broad categories [23]. The adoption of a particular category depends on many factors such as devices computing features, the communication bandwidth, and the available resources.

a. Client-side adaptation: the devices running the documents make the adaptation process by themselves (e.g. [17]). This option allows more responsiveness; however, the limited resources of some devices (e.g. battery, CPU) may be a hindrance to the execution of adaptation actions.
b. Server-side adaptation: the devices playing the documents send adaptation requests to a server that deals with them (e.g. [20]). Although this option frees the client side from a tedious process that could slow down performance, this may affect the server response time due to overload.
c. Proxy-based adaptation: a proxy is placed between the client and the server in order to act as a mediator while running the adaptation process (e.g. [7]). The proxy frees both client and server from intensive tasks by leveraging its computing power; however, this option may involve too much communication for negotiating which adaptation actions to execute.
d. Peer-to-peer adaptation: the devices playing the documents can communicate with multiple platforms, including other devices, to perform the adaptation process (e.g. [5]). This option takes advantages of the peer-to-peer paradigm to ensure a balanced load while executing the adaptation actions. Despite this, the process efficacy depends on the number of connected nodes. Furthermore, the network administration is difficult due to its decentralized nature.

Table 2 provides a comparative summary of some of the MDA approaches cited above. The aim is to compare them according to their main limitations concerning the way the reasoning phase is performed. Such a comparison helps propose improvements and contributions with respect to related research-works.

Table 2. A comparative summary of some MDA approaches

Approach	Category	Overview	Main limitation
Hai *et al.* (2012) [5]	Peer-to-peer	A service-based approach for MDA in which adaptation services are available locally and remotely on multiple platforms.	No support of semantic aspects in profiles and adaptation services.
Dromzée *et al.* (2013) [7]	Proxy-based	A service-based context modeling approach for MDA that links the context information using semantic generic profiles.	No support of reasoning mechanisms.
Bettou and Boufaida (2017) [11]	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.	No support of semantic aspects to deal with context constraints.
Adel *et al.* (2017) [13]	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.	No support of the compatibility between the inferred adaptation actions.
Saighi *et al.* (2017) [12]	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.	Only one adaptation action is inferred per adaptation request (no support of multiple actions).
Khallouki and Bahaj (2017) [14]	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.	No prototype was implemented for the proposal to be tested and evaluated.
El Guabassi *et al.* (2018) [16]	Server-side	An XML-based personalization of course content in ubiquitous learning, considering learning styles and context-awareness	No support of semantic aspects to deal with context elements.
Belhadad *et al.* (2018) [17]	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.	Only the spatial dimension of objects is handled but not their temporal synchronization.
Saighi *et al.* (2020) [20]	Server-side	A graph-based MDA that allows the inference of multiple adaptation actions using semantic reasoning upon users context.	No parallel computing support for reasoning (performances issues).

By analyzing several MDA approaches [2]-[21] (see Table 2), we could check that they focus mainly on context collection, representation, and interpretation. In particular, the reasoning mechanisms only rely on profiles and current context values in such a way that the adaptation task is performed in a single location (i.e. client, proxy or server sides). Thus, the storage and analysis of past context values and corresponding adaptation actions are not supported in order to be involved within adaptation processes. Logging of history information is supported by many context-aware pervasive applications as it helps to provide prior knowledge about users decisions so that they can be taken into account in future processing. We cite as examples machine-learning approaches for: IoT cultural data [24], context-aware intrusion detection [25], smartphone data analytics [26]–[29], activity recognition [30], and healthcare support [31].

Likewise, 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, and adaptation rules personalization. In this case, two types of HUD can be considered either jointly or separately. The first type concerns the multimedia documents consulted by emphasizing on general information about users and resources accessed. The second type is related to

specific features of ubiquitous systems (e.g. sensors, and human-computer interactions.) 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 [32]. 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. [33], [34]), 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 [35]. This raises certain issues related to their management, which are further addressed later in this paper:

- 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 (most likely big data).
- The last concern is about the way in which the data stored will be leveraged in useful tasks by integrating users experiences into MDA processes, in order to improve their effectiveness.

Starting from this context, 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. METHOD

Our proposal consists of a software component for managing HUD resulting from the execution of MDA processes. The aim is to allow their storage, retrieval and analysis so as to involve them in users decision-making according to different ways. This is a complementary element to existing adaptation approaches that attempts to improve their overall performances since in most of them:

a. Decision-making is performed by inferring the adaptation rules using only on-demand and/or real-time reasoning; thus, they do not imply HUD when performing the adaptation task. Our proposal allows MDA processes to leverage from HUD to perform several advantages tasks (e.g. machine learning tasks), in order to improve the adaptation process quality.
b. The whole adaptation task is carried out in a single location (clients, proxy or server sides), in particular the client-server paradigm which is broadly adopted due to implementation simplicity and computation resources power. Even so, the efficient use of other paradigms can also be a suitable choice to deal with several difficulties (e.g. workload balancing). One way to take into account this issue is to rely on an effective hybridizations between different adaptation categories such that the MDA process is executed in a decentralized way by switching from one adaptation category to another in the event that the current solution shows its limitation. In this regard, our proposal is built around a well-devised, flexible and agile architecture that offers the possibility to treat HUD in different locations and in different ways, which:
    - makes the HUD manager potentially supported by all MDA categories.
    - provides more options for MDA approaches, depending on the current computation resources and users preferences (e.g. data privacy and data sharing), and
    - allows HUD to be managed by different devices (e.g. smartphones, laptops, and servers).

3.1. The general architecture of the proposed HUD manager

Now, we give details of the proposed HUD manager. Figure 2 shows its architecture and the potential interactions with MDA processes. The proposal comprises four components as follows:

a. The database: it represents the storage medium for storing the HUD.
b. The data logger: it makes systematic recordings of HUD in the database each time new data is acquired by the system.
c. The data retriever: it allows retrieving HUD from the database to be used by the data analyser.
d. The data analyser: it mainly consists of a bag of data analysis algorithms dedicated to different purposes such as machine learning techniques, and data mining methods.

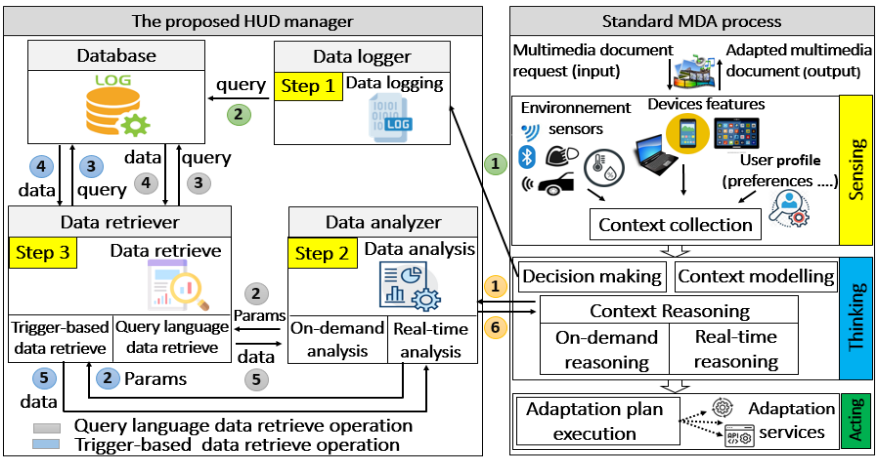

Figure 2. Detailed architecture of the proposed component and its potential interactions with MDA processes

The interaction between the MDA processes explained in subsection 2.1 and the HUD manager takes place through an invocation of the data logger and the data analyser, performed by the thinking layer, as shown in Figure 2. This layer includes the context-reasoning and the decision-making modules that constitute the core of the adaptation system and therefore the source of its intelligence. On the one hand, the decision-making module invokes the data logger to store HUD about the context values and the decisions made by users (step 1 in Figure 2). On the other hand, the context-reasoning module takes advantages of the functionalities provided by the data analyser that contribute in making future decisions (step 2 in Figure 2). Next, we summarize the three steps executed by the proposed HUD manager.

- Step 1: HUD logging

Once the decision-making module has generated the adaptation plan, it calls the data logger component to store information about the corresponding context values and adaptation actions made by the application. In turn, the data logger accesses the DB through a query language–that depends on the implementation–in order to insert these data as new entries (see Figure 2).

- Step 2: data analysis

The data analyzer component provides a bag of data analysis algorithms that can be used by the context-reasoning module, in order to improve its effectiveness. To do so, 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; it is continuously executed in order to update the context-reasoning module as needed, depending on the purposes of the application. Algorithms in this category should be lightweight in order to maintain the responsiveness of the system with respect to application usage. A typical example of this category is the statistical analysis methods based on pattern frequencies for which the algorithm needs only to maintain the frequencies using a dynamic data structure. On-demand reasoning is triggered by a request from a user or application for many purposes such as machine-learning algorithms and recommendation systems. Algorithms in this category can be heavyweight. Moreover, they can be executed online or offline as they do not influence the responsiveness of the system with respect to application usage.

- Step 3: data retriever

The data retriever is designed according to the data access object (DAO) pattern that provides an abstraction between several database types as well as other persistence mechanisms and the other applications parts. This way, calls to the persistence layer goes through the DAO so as data operations are performed without revealing database details. Thus, the data retriever consists of a set of callable methods that retrieve data from the DB according to different input parameters. The implementation of such methods is based on query languages (e.g. SQL, and XPath) depending on the type of DB adopted. Two types of data retrieve methods are proposed: data retrieve based on triggers and data retrieve based on query languages. The first type of data retrieve uses triggered callable methods that are executed when certain conditions related to DB updates are met. The second type of data retrieve is based on parameterized callable methods that access and retrieve data from the DB according to the input parameters.

The proposed HUD manager has several functions. This provides various options that help in efficiently designing MDA processes according to users requirements and context specifications. Depending on where the HUD is stored and processed, three methods in addition to their hybridization are proposed:

a. Client-side management: the client devices store, retrieve and analyse HUD by themselves.

b. Server-side management: the HUD manager is installed in a server that stores, retrieves and analyses the data. The devices playing the documents (i.e. clients) send requests by invoking the data logger, and the data analyser as needed.
c. Proxy-based management: the HUD manager is implemented in a proxy that is placed between the client and the server.
d. Hybrid management: the components of the HUD manager namely the database, the data logger, the data retriever and the data analyser, are separately placed in the client device, proxy or server, depending on the current requirements of the MDA process.

3.2. Implementation of the proposed approach

To show the feasibility of our proposal, we develop a real prototype through which multimedia documents are adapted according to contextual situations while storing the HUD. In fact, from a functional point of view, most MDA processes work very well. However, when it comes to validating a given adaptation approach, the prototypes developed are often based on simulation to collect context elements; mainly for the elements related to connected objects (see for example [11], [12], [20]). Indeed, even if the simulation paradigm is very useful in many situations, the realization of a real prototype remains the most ideal, realistic and viable solution, in particular to deal with technical, and implementation issues.

3.2.1. Context senesing

Context values come from different hardware and software sources. Initially, they are collected as raw data since they only include measured values. These values serve as input data for the reasoning phase, the purpose of which is to build adaptation rules. Table 3 details the context elements used in our prototype.

Table 3. Different context classes features

Context class	Attribute	Domain of values	Source
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 ∈ {0, 1}*	
	Network	*Bandwidth (value ≥ 0)*	
		Type ∈ {WI-FI, mobile data, …}	
User	Preferred language	*List of languages*	Profile
	Agenda	*List of timed tasks / places*	
	Location	*Lis of places {home, lab, public place, meeting room, …}*	Profile/sensors/device
Temporal	Time	*(Date, time)*	Device

3.2.2. Reasoning upon the context

The reasoning upon context values takes place in two phases: qualification of context values and inference of adaptation actions. The first phase aims to qualify the quantitative values of the context so that they are effectively used during adaptation actions inference and HUD analysis. Table 4 shows typical examples of context values quantification; these values are usually defined empirically.

Once the context values are qualified, the second phase is to infer the adaptation actions. Each adaptation action consists of a rule taking the form of "if (conflicts) then (actions)". Table 5 gives details of a subset of typical adaptation actions. To build these rules, we are inspired by those used in [12], [20].

3.2.3. Execution of the adaptation actions

Our prototype supports web contents adaptation according to user contexts without depending on a specific domain of application. The aim is to ensure the genericity of the prototype so that it can be adapted to a wide range of applications using multimedia documents with simple objects (e.g. texts, images, and videos) or those composed of a combination of different types of contents. Generally, there exist several types of document sharing, depending on applications scopes; we cite as examples file transfer protocol (FTP), peer-to-peer sharing and cloud service sharing (e.g. [36], [37]). Regarding our prototype, the HTTP protocol was adopted to share a set of multimedia documents hosted on a server, due to simplicity of implementation.

The adaptation of multimedia documents takes place by involving a set of adaptation actions. Each adaptation action is performed by executing one or several adaptation services, depending on the nature of the media objects in the documents. The description of the adaptation actions given in Table 5 is as follows:

a. Speech to text: it mutes the sounds of auditory objects by replacing them with textual content (sounds on videos are muted and subtitled while audio contents are converted to text).

b. Contrast adjustment: it highlights visual objects contrast according to the brightness ratio.
c. Video to images: it converts videos to images sequences.
d. Language translation: this action translates media objects written in a source language to the target language specified in the users profile.
e. Format change: it changes the format of a media object to another format (e.g. mp3 to mp4).
f. Video in smart TV: this action plays video objects in a smart TV since the screen size is larger.
g. Text summarizing: this action summarizes the content of text objects.

Table 4. Qualification of quantitative values of typical context elements

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

Table 5. Some typical rules for carrying out adaptation actions

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 Video to images
R4	if (inappropriate language) then	Language translation
R5	if (overloaded CPU) then	Format change 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

3.2.4. Hardware and software configuration

The hardware configuration includes an arduino UNO board on which three sensors are connected: sound, GPS, and photoresistor light sensors. Our prototype uses WiFi and bluetooth low energy (BLE) connectivity. On the one hand, WiFi is one of the most popular wireless communication protocols. In addition, it is very compatible with a wide range of devices. On the other hand, BLE-based devices are very adopted in IoT implementations since they provide low power consumption and consistent connectivity. These protocols are used within the sensing layer shown in Figure 1, to collect the context elements from various sources. They are also used in the intercommunication between wired and wireless devices. Our prototype can easily be expanded so that it supports other communication protocols such as zigbee, in order to satisfy the potential demands for compatibility with installations based on other ubiquitous IoT protocols.

The hardware configuration also includes two HP Z640 stations (xeon CPU composed of 20 cores and 64 GB of RAM) used as a proxy and a server, and a smartphone (samsung galaxy A10s with helio P22 CPU and 2 GB of RAM) used as a client device on which the documents are run. Adaptation services calls are performed using the following java APIs: google cloud speech API (details can be found on the website: https://cloud.google.com/speech-to-text/), yandex translate (available at: https://tech.yandex.com/translate/), fast forward moving pictures expert group (FFMPEG) (downloadable at: https://www.ffmpeg.org/) and Jave (downloadable at: http://www.sauronsoftware.it/projects/jave/).

The database component is implemented using relational and *NoSQL* databases as they operate independently of any specific data analysis framework; the aim is to compare the two models. On the one hand, relational databases allow handling important volumes of structured data while offering efficient functionalities for data storage and retrieval. On the other hand, NoSQL databases allow handling large volumes of data at high speed with scalable architectures according to different degrees of data structuredness. Besides, there have been significant efforts made to build bridges between oriented-object, ontology and key-value paradigms on one side, and relational and NoSQL databases on the other side (e.g. [38], [39]). The data to be stored are restricted to qualitative values of the context elements and the corresponding adaptation actions resulting from changes in context over time. The static aspect of the HUD structure is specified by the class diagram shown in Figure 3.

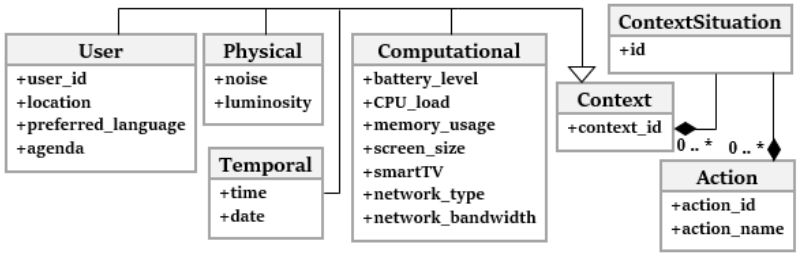

Figure 3. The static aspect of the abstract structure of the HUD

Depending on where the HUD are stored, several database variants are implemented (see Table 6). Concerning the relational databases, we have used SQLite databases for client-side methods and MySQL databases for both proxy-based and server-side methods. Regarding NoSQL databases, we have used JSON files for all methods, in particular JSON files in hadoop distributed file system for sever-side approaches.

Regarding the implementation of the remaining components namely the data logger, retriever and analyser, we rely on the oriented-object paradigm features. More specifically, we use Java interfaces since they help to design processes specifications that can be implemented in different ways. Table 7 shows the implementation details according to the different data management methods.

Table 6. Technical choices to implement the database component in the HUD manager

Management method	Relational model	NoSQL model
Client-side	SQLite database	JSON file
Proxy-based	MySQL database	JSON file
Server-side	MySQL database	JSON file in hadoop distributed file system

Table 7. Technical choices to implement the remaining components in the HUD manager

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 systems based on map/reduce paradigm

4. RESULTS AND DISCUSSION

To validate our proposal, three tests are conducted. The first test consists in a case study through which we show how the data are organized in the database as well as a typical example of the adaptation result of a multimedia document. The second test aims to measure and evaluate the performance of our proposal in terms of processing time and size of data required to process the HUD. Finally, the last test consists in a case study through which we show the usefulness of storing the HUD by considering a typical example of how HUD contribute to the personalization of adaptation rules.

4.1. Formating and storing the HUD

This section shows how HUD are formatted and stored in the database. Intuitively, there are two methods to transform the HUD modeled in Figure 3 into relational model and JSON objects. In the first case, the log data take a columnar form by grouping together all attributes of HUD classes; thus, the data are structured as a set of entries. This format is useful for applying data analysis techniques. In the second case, the log data take the form of structured entities (linked tables, and nested objects) using mapping procedures while maintaining the relationships among them (e.g. [38], [39]). This format is more suitable for data readability at a high specification level. Regarding our work, we adopt the former case since our main goal is to perform data analysis. Below, an illustrative scenario for which the HUD are stored by executing an adaptation process triggered under certain assumptions. 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. We note that the approach is general so that it can easily be adapted to several dedicated MDA processes such as medical applications, ubiquitous learning, and tourism filed. Thus, the scenarios considered only represent demonstrations by way of non-limiting examples.

Mr. John is a medical-researcher at university; he generally connects to medical forums for discussion. Given the nature of his job, John is often on the move for consultations, and conferences. When 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.

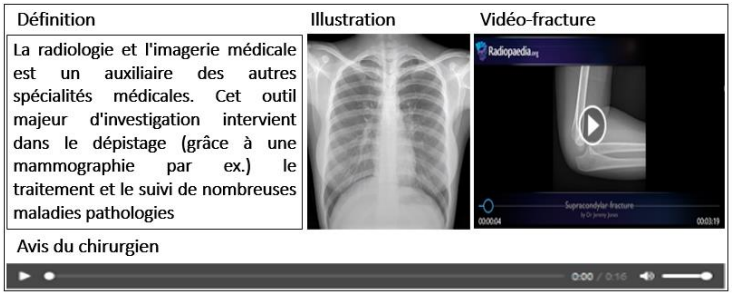

Figure 4. Structure of the original document

Let assume that John is in a conference room and uses his tablet to access the document. The battery level of his tablet indicates that it has reached 25%. The contextual constraints inferred by the adaptation system are as follows: public place, low battery and inappropriate language. Hence, rules $R_{i=1,\ 3,\ 4}$ in Table 5 are executed to provide an adapted document, as shown in Figure 5. On another side, the system invokes the data logger to store the log data in the database, 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"**}

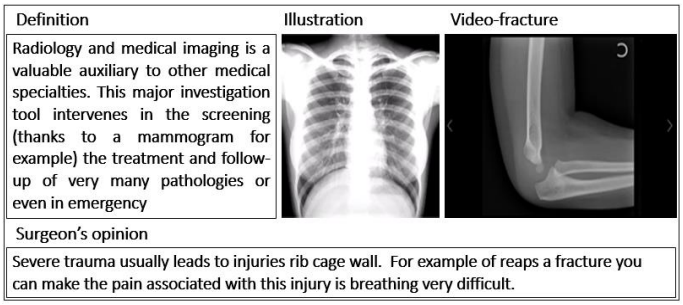

Figure 5. Structure of the resulting adapted document

4.2. Performance analysis

4.2.1. Running time and data size measurements

Next, we evaluate the different database models shown in Table 6. The evaluation of performance is carried out by measuring the running time and the data size required for processing the HUD according to the current number of entries in the database. In particular, we study the average time for inserting one HUD entry in the database, the required time for loading the database content in memory to make data analysis and the storage space required for storing the HUD. The tests are performed using the hardware configuration mentioned in subsection 3.2.4. Figure 6 shows the obtained results by assuming that each user stores a maximum number of entries=10000. The users number is set to 1 for client-side HUD management and to 50 for proxy-based/server-side HUD management. Figure 6 includes:

- Figures 6(a) and (b) to plot the variations of the average time required for inserting a new entry in databases.
- Figures 6(c) and (d) to plot the variations of the average time required for retrieving data entries from databases.
- Finally, Figures 6(e) and (f) to plot the variations of the size of the data stored in databases.

By analyzing the curves illustrated in Figure 6, one observes that for both cases (i.e. client-side, and proxy-based/server-side HUD management):

- The time required for storing one entry is quasi-constant regardless of databases sizes (see Figures 6(a) and (b)). In particular, NoSQL databases are well performing for data storage compared to relational databases. This is because relational database management systems use advanced techniques to optimize the data storage, which affects the processing time.

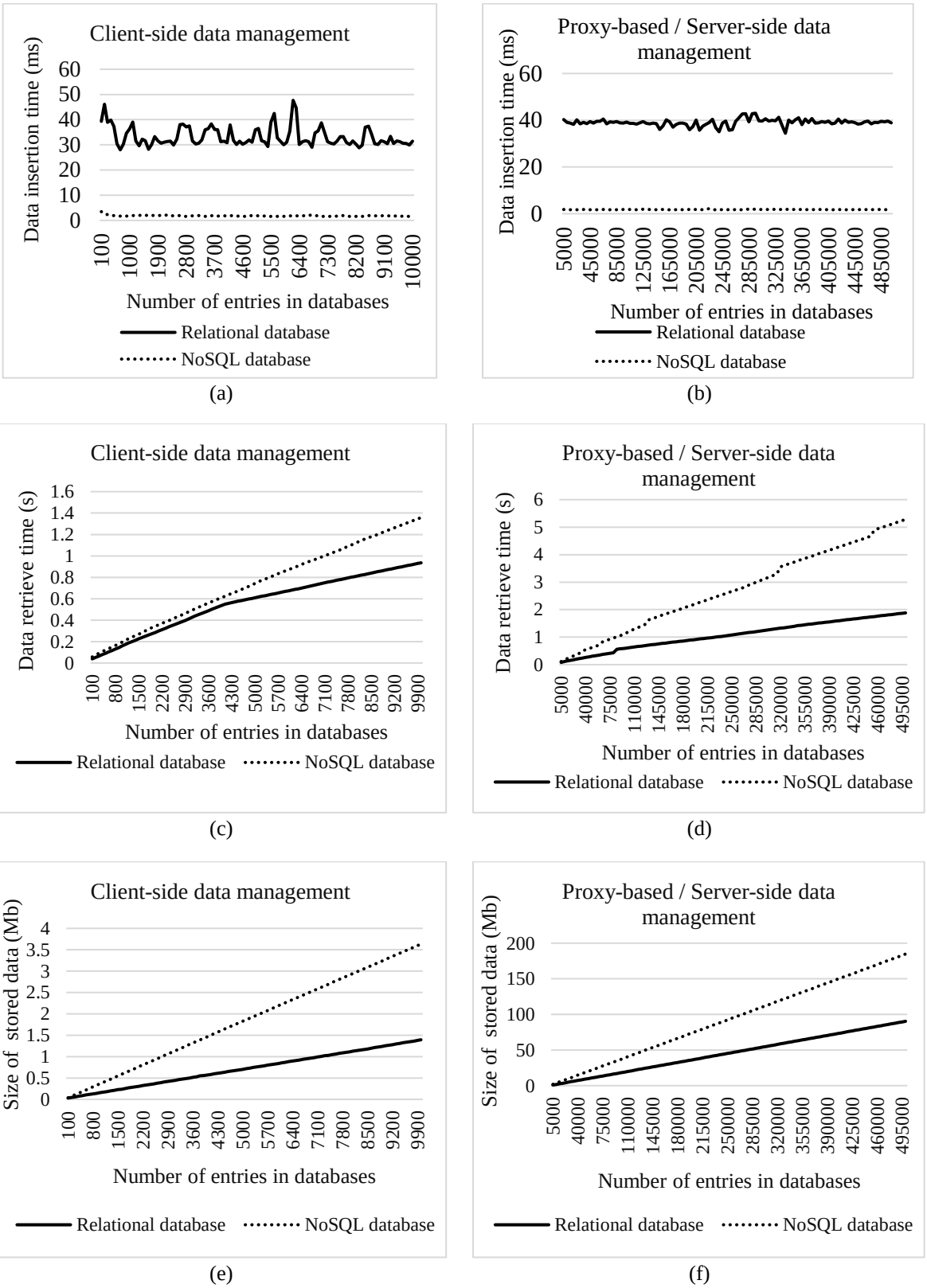

Figure 6. Statistics about running time and data size required for processing HUD: (a) time for inserting one entry in client-side methods, (b) time for inserting one entry in server/proxy methods, (c) time for retrieving data in client-side methods, (d) time for retrieving data in server/proxy methods, (e) data size in client-side methods, and (f) data size in server/proxy methods

- The time required for retrieving and loading data increases linearly in accordance with the growth of databases sizes (see Figures 6(c) and (d)). This is expected as the volume of processed data becomes larger. Nevertheless, relational databases are well performing for data retrieve compared to NoSQL ones since the latter parse data before storing them as objects in memory.
- The required space for storing data increases linearly in accordance with the growth of databases sizes (see Figures 6(e) and (f)). Besides, relational databases require less storage space given that NoSQL databases store entries as combinations of keys (i.e. objects and attributes), and values. To overcome this issue, it is possible to make the descriptors length as shorter as possible. Even so, NoSQL databases may excel since they do not require dedicated configurations either for exporting data or for accessing them in addition to their simplicity of implementation.

Globally, the general performances of the proposed HUD management methods depend mainly on the available computational resources. For instance, during the experiments, we could check that the client-side method could not store more than 55.000 in SQLite databases. It also could not load more than 150.000 entry in memory from NoSQL databases. Likewise, the proxy-based method could not load more than 800.000 entry in memory. This leads us to care about issues related to efficiency of data storage, retrieval, analysis, scalability and others.

4.2.2. Comparison of the proposed HUD management methods

Now, we compare the proposed HUD management methods detailed in subsection 3.1 with each other based on systems 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 influence the general performances of the HUD management system such as data storage location, data size, computation resources, and number of users. Table 8 provides details about each criterion used for comparison purpose.

Table 8. Criteria for comparing the proposed HUD management methods

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	Big data	Large	Small
Data performance	Performance for storing/retrieving large volumes of data.	Number of users	Excellent	Good	Average
Data retrieval	Ability to load large volumes of data for processing.	Data size	Excellent	Average	Limited
Data analysis	Computational ability to perform analysis tasks upon HUD.		Easy	Feasible	Difficult
Scalability	Number of users that 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 9 summarizes the comparison between the proposed HUD management methods according to the criteria presented in Table 8. The statistics given in Table 9 help to determine the way in which the HUD can be effectively managed, and processed. Globally, server-side methods may excel because of the computation resources they have, the number of clients they can handle and the data volume they can store. In contrast, client-side methods show high data security since they store them on clients devices; thus, any access to data goes through client permission.

Even so, these features may be affected by several factors such as the available computation resources, the current workload, load balancing issues, fault tolerance, and service availability. In such cases, hybrid management methods (i.e. those combining client-side, server-side, proxy-based, and peer-to-peer paradigms) help overcome these issues through an efficient placement of the HUD manager components in different locations, depending on the needs and requirements of both applications and users.

Table 9. Comparison between the proposed HUD management methods

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

4.2.3. Comparison of our proposal over other approaches exploiting HUD

Finally, we compare our proposal with three research-works [33], [34], [40] that exploits HUD in pervasive environments (see Table 10). The purpose is to show the strengths of our proposed approach as well as the value added so as to improve MDA processes. To do so, we first consider the following points:

- Approach scope: it describes the scope of a given research-work.
- Approach category: it refers to the HUD management methods discussed in section 3, namely server-side, proxy-based, and client-side.
- Efficient historical data management: it refers to whether a given approach uses/provides efficient mechanisms to handle the HUD.
- Technical details: it refers to whether a given research-work provides technical details about HUD handling.
- Historical data use: it refers to the purpose for which the HUD is leveraged in useful tasks.
- Historical data involved: it refers to the elements composing the HUD.
- Approach flexibility: it refers to whether a given approach is agile and generic so that it can be adapted to a wide range of approach categories.

Table 10. Characteristics of our proposal and some context-aware approaches involving HUD

Approach scope	Category	HUD management	Technical details	HUD involved	HUD use	Flexibility
Adaptation of multimedia content (movies) to users needs in smart homes [33]	Client-side	Yes	No	Context values Multimedia contents	Multimedia content recommendation	No
Personalized recommendation system for learning materials in an ubiquitous learning environment [34]	Server-side	No	No	Learner profile Learner model	Learning materials recommendation	No
Context-aware middleware supporting reasoning mechanisms for user interaction in cultural spaces [40]	Server-side	Yes	Yes	Context values Users actions	User interaction enhancement	No
Our proposal is a generic context-aware MDA architecture supporting reasoning mechanisms upon context values and HUD	Adaptable to all categories (client-side, server-side, proxy-based, peer-to-peer)	Yes	Yes	Context values Adaptation actions	Efficient multimedia content adaptation	Yes

By relying on hybrid management methods (i.e. combining client-side, server-side, proxy-based, and peer-to-peer paradigms) our proposal may provide better data handling because several limitations of client-side, proxy-based, and server-side methods can be reduced (e.g. workload balancing, service availability) in particular if cloud computing techniques are involved. Thus, by considering the criteria given in Table 8, the potential strengths of our approach compared to the works presented in [33], [34], and [40] are summarized in Table 11. It should be noted that these values are assigned based on the details provided in the corresponding references. Moreover, the difficulty of reconstructing these context-awre systems and the lack of several technical details have prevented us from making numerical comparisons with our proposal.

Table 11. Comparison of our proposal against some context-aware approaches involving HUD

Ref	Security	Volume	Performance	Retrieval	Analysis	Sharing	Scalability
[33] (2023)	Strong	Small	Average	Average	Feasible	Unsuitable	Not scalable
[34] (2021)	Weak	Small	Average	Average	Feasible	Suitable	Not scalable
[40] (2023)	Weak	Big data	Good	Average	Feasible	Suitable	Scalable
Our proposal	Strong	Big data	Excellent	Excellent	Easy	Suitable	Scalable

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:

- The proposal can be considered as a complementary element to existing MDA approaches since most of them do not efficiently handle the HUD.
- The proposal shows adaptability insofar as it handles the HUD at a high level of abstraction, regardless of any specific MDA approach (e.g. medical applications, and ubiquitous learning).

- The high flexibility as the proposal offers various options to store, retrieve and analyze the HUD. These features allow overcoming some of the issues discussed in Table 9 through an efficient placement of the HUD manager components in accordance with applications and users needs.

4.3. Case study on HUD analysis: context-aware adaptation rules learning

In what follows, we consider a case study to show the usefulness of storing and processing the HUD. Thus, the scenarios below give a typical example of how HUD contribute to the personalization of adaptation rules through rule learning mechanisms. This task refers to prediction of adaptation actions using data-driven approaches (i.e. based on data analysis and interpretation) rather than explicitly defining them.

- Scenario 1: David is a mobile user. While moving, he accesses multimedia documents on various subjects through the mobile data accessible from his smartphone; this is a paid and limited service. To save mobile data consumption, he always asks the system to adapt the documents by using low quality media objects format.
- Scenario 2: Sarah is a university student. Back home, she has to revise her online courses using her laptop. At that time, Sarah always feels tired due to her busy daily agenda. Therefore, she asks the system to adapt the target documents by summarizing their contents.

These scenarios are implemented in a way that HUD are stored in a NoSQL database, denoted by DB, hosted in the proxy. For each scenario, we store two types of HUD. The first type concerns the storage of a sufficient number of entries according to the assumptions mentioned in the scenarios. The second type consists in storing random entries to endow database DB with dynamicity (i.e. noise) when analysing data. In order to infer new personalized adaptation rules, the process given in Figure 7 is proposed. First, the data analyser invokes the data retriever to load database DB as an array of objects, denoted by O. Then, array O is split into subparts $O_{i=1\ \ldots\ n}$ according to user *id*; *n* is the total number of users. Finally, the elements of each subpart $O_{i=1\ \ldots\ n}$, denoted by $I_i=\{o_{ij}\}$, are analysed to extract associated conflicts and adaptation actions.

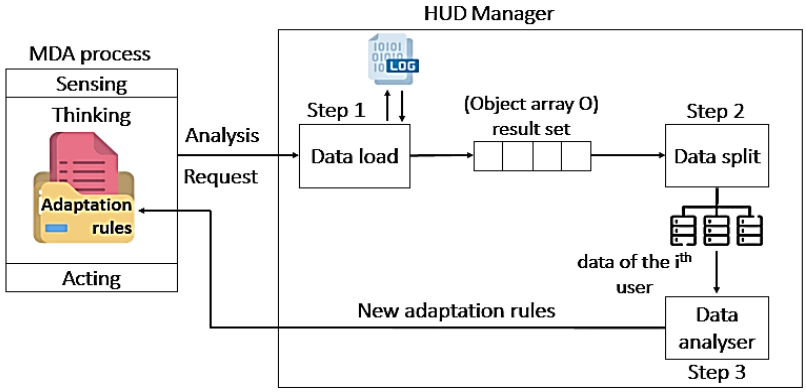

Figure 7. Process phases for adaptation rules learning from HUD

The idea of the analysis algorithm is to implement an efficient backtracking of associated conflicts and actions in each subpart $O_{i=1..\ n}$, as follows:

- Let X=(x_1, x_2, …, x_m) be a vector such that each element $x_i \in X$ is bound to one context element; *m* is the number of context elements. The values taken by element x_i are defined as all possible qualitative values of the corresponding context element in addition to the null value. For example, if x_i refers to battery level then domain (x_i)={*null*, *low*, *medium*, *full*}.
- Let A={a_1, a_2, …, a_k} be a set such that each element refers to one of the adaptation actions; *k* is their number. Each element $a_{j=1..k} \in$ A can be either 0 or 1 to specify whether adaptation action a_j is considered for execution or not.
- The backtracking algorithm goes through paths built by combining the possible values for vector X and adaptation actions from set 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. The confidence degree is defined as the ratio of (the number of objects from $I_i=\{o_{ij}\}$ including non-null values from vector *x* and executed actions from vector *a*) to (the number of objects from $I_i=\{o_{ij}\}$ including all non-null values from vector *x*). The resulting rules take the form "IF ($\bigwedge_{i=1}^{|x|} c_i \in x$ / $c_i \neq$ null) then $v_{j=1..k} \in a$ / v_j = 1". In addition, each time a potential rule is retained, a redundancy check is applied in order to decide whether or not to keep this rule.

Table 12 shows typical rules retained during the analysis phase. The above scenarios show the importance of processing the HUD to build intelligent context-aware applications through adaptation rules learning to predict users behaviours. Thus, the main advantages of such a task are summarized as follows:

- The use of unsupervised machine learning mechanisms; therefore, the context-aware adaptation rules are learnt from HUD instead of defining them in a fixed way. Moreover, there will be no need to pre-trained data (datasets) to perform the learning task.
- Possibility to limit the log data so that only some subparts are processed according to a sliding window (e.g. considering data corresponding to recent periods).
- Reusability since it can be adapted to other rule-based context-aware pervasive environments.

In spite of these advantages, there are still some limitations that should be overcome in future:

- The system does not involve efficient association rules algorithms (e.g. frequent pattern growth algorithm). Indeed, backtracking algorithms cannot be applied to process large volumes of data. That is why we need to develop more efficient and effective algorithms for data analysis in order to extract personalized adaptation rules.
- As the system is based on unweighted (blind) correlations, it may take a long time to overlook the most frequent old rules and replace them with the most recent ones.
- The system lacks of experience sharing between users; this feature helps build rule-based recommendation systems.

Table 12. Typical rules retained during the analysis phase

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

5. CONCLUSION

The aim of this work has been to involve relational and NoSQL databases to efficiently manage HUD resulting from the execution of MDA processes in context-aware pervasive systems. To achieve this goal, the context as well as the adaptation actions were first modeled through the oriented-object approach and then converted into appropriate schemes. To make the proposal adaptable to several adaptation approaches, four management methods have been proposed: client-side management, proxy-based management, and server-side management and hybrid management. The proposal has been validated through scenarios implemented in a real prototype with a special focus on use cases about context-aware adaptation rules learning. In fact, as the aim here was to show only the feasibility of the approach, the developed prototype was built around a restricted subset of general adaptation rules independent of any specific field of application. For real-world applications, the adaptation rules base may be broadened by considering other context elements, depending on the needs of the field of application. In addition, more sophisticated technical choices can be involved, such as *sensing as service paradigm*. Currently, we are working on applying machine-learning techniques along with other efficient data analysis algorithms, in order to perform context-aware adaptation rules learning from HUD.

ACKNOWLEDGEMENTS

The authors would like to thank the direction génerale de la recherche scientifique et du developpement technologique (DGRSDT) in Algeria, for supporting this work as a funding for a doctoral research project at the University of Oum El Bouaghi, Algeria. Also, the authors would like to thank Miss Selma Hadjadj from University of Om El Bouaghi, for her interventions to deal with some technical issues.

REFERENCES

[1] P. N. Mahalle and P. S. Dhotre, "Context-Aware Pervasive Systems," *Intelligent Systems Reference Library*, vol. 169, pp. 49–66, 2020, doi: 10.1007/978-981-32-9952-8_3.

[2] 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, 2007, doi: 10.1016/j.jnca.2005.12.007.

[3] Z. I. Kazi-Aoul, I. Demeure, and Jean-Claude Moissinac, "Paam: a Web Services Oriented Architecture for the Adaptation of Composed Multimedia Documents," in *Parallel and Distributed Compuing and Networks (PDCN)*, 2008.

[4] A. E. Maredj and N. Tonkin, "Semantic adaptation of multimedia documents," *International Arab Journal of Information Technology*, vol. 10, no. 6, pp. 379–398, 2013, doi: 10.20533/ijmip.2042.4647.2012.0012.

[5] 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.

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

[7] C. Dromzée, S. Laborie, and P. Roose, "A semantic generic profile for multimedia document adaptation," *Intelligent Multimedia Technologies for Networking Applications: Techniques and Tools*, pp. 225–246, 2013, doi: 10.4018/978-1-4666-2833-5.ch009.

[8] A. Adel, S. Laborie, and P. Roose, "Automatic adaptation of multimedia documents," *Procedia Computer Science*, vol. 19, pp. 992–997, 2013, doi: 10.1016/j.procs.2013.06.138.

[9] 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*, 2015, pp. 232–235, doi: 10.1109/ICOSC.2015.7050811.

[10] A. Alti, S. Laborie, and P. Roose, "A Community-Based Semantic Social Context-Aware Driven Adaptation for Multimedia Documents," *International Journal of Virtual Communities and Social Networking*, vol. 7, no. 2, pp. 31–49, 2015, doi: 10.4018/ijvcsn.2015040102.

[11] 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, 2017, doi: 10.4018/ijmdem.2017010102.

[12] Z. L. A. Saighi, R. Philippe, N. Ghoualmi, and S. Laborie, "Hama: A handicap-based architecture for multimedia document adaptation," *International Journal of Multimedia Data Engineering and Management*, vol. 8, no. 3, pp. 55–96, 2017.

[13] 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, 2018, doi: 10.4018/ijvcsn.2017070104.

[14] H. Khallouki and M. Bahaj, "Multimedia documents adaptive platform using multi-agent system and mobile ubiquitous environment," *2017 Intelligent Systems and Computer Vision, ISCV 2017*, pp. 1–5, 2017, doi: 10.1109/ISACV.2017.8054915.

[15] O. Benmesbah, L. Mahnane, and M. Hafidi, "Personalized adaptive content system for context-aware mobile," *Proceedings of the 14th International Conference on Mobile Learning 2018, ML 2018*, vol. 127, pp. 139–142, 2018.

[16] I. El Guabassi, M. Al Achhab, I. Jellouli, and B. E. El Mohajir, "Personalized ubiquitous learning via an adaptive engine," *International Journal of Emerging Technologies in Learning*, vol. 13, no. 12, pp. 177–190, 2018, doi: 10.3991/ijet.v13i12.7918.

[17] Y. Belhadad, A. Refoufi, and P. Roose, "Spatial reasoning about multimedia document for a profile based adaptation: Combining distances, directions and topologies," *Multimedia Tools and Applications*, vol. 77, no. 23, pp. 30437–30474, 2018, doi: 10.1007/s11042-018-6080-8.

[18] Z. Bousalem, I. El Guabassi, and I. Cherti, "Toward adaptive and reusable learning content using XML dynamic labeling schemes and relational databases," *Advances in Intelligent Systems and Computing*, vol. 915, pp. 787–799, 2019, doi: 10.1007/978-3-030-11928-7_71.

[19] I. El Guabassi, Z. Bousalem, M. Al Achhab, I. Jellouli, and B. E. E. L. Mohajir, "Identifying learning style through eye tracking technology in adaptive learning systems," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 5, pp. 4408–4416, 2019, doi: 10.11591/ijece.v9i5.pp4408-4416.

[20] A. Saighi, Z. Laboudi, P. Roose, S. Laborie, and N. Ghoualmi-Zine, "On using multiple disabilities profiles to adapt multimedia documents: A novel graph-based method," *International Journal of Information Technology and Web Engineering*, vol. 15, no. 3, pp. 34–60, 2020, doi: 10.4018/IJITWE.2020070103.

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

[22] T. Strang and C. Linnhoff-Popien, "A Context Modeling Survey," in *Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing*, 2004, pp. 1–8.

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

[24] F. Piccialli, S. Cuomo, V. S. di Cola, and G. Casolla, "A machine learning approach for IoT cultural data," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, 2019, doi: 10.1007/s12652-019-01452-6.

[25] 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," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 4, pp. 1405–1412, 2020, doi: 10.1007/s12652-018-0998-6.

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

[27] P. W. Iqbal, A. C. Sarker, and J. Han, "Automated rule-based services with intelligent decision-making," *Springer. Context-aware machine learning and mobile data analytics*, 2022.

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

[29] 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, 2021, doi: 10.3390/sym13101975.

[30] 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 Journal*, vol. 20, no. 8, pp. 4361–4371, 2020, doi: 10.1109/JSEN.2020.2964278.

[31] 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, 2020, doi: 10.1038/s41586-020-2669-y.

[32] 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, 2020, doi: 10.4018/IJDET.2020040106.

[33] 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," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 5, pp. 5137–5149, 2023, doi: 10.1007/s12652-021-03081-4.

[34] S. Pal, P. K. D. Pramanik, A. Nayyar, and P. Choudhury, "A Personalised Recommendation Framework for Ubiquitous Learning System," in *ACM International Conference Proceeding Series*, 2021, pp. 63–72, doi: 10.1145/3460179.3460190.

[35] F. Skopik, M. Wurzenberger, and M. Landauer, "Smart Log Data Analytics," *Smart Log Data Analytics*, 2021, doi: 10.1007/978-3-030-74450-2.

[36] Y. M. Chen and W. C. Wu, "An anonymous DRM scheme for sharing multimedia files in P2P networks," *Multimedia Tools and Applications*, vol. 69, no. 3, pp. 1041–1065, 2014, doi: 10.1007/s11042-012-1166-1.

[37] N. Chidambaram, P. Raj, K. Thenmozhi, S. Rajagopalan, and R. Amirtharajan, "A cloud compatible DNA coded security solution for multimedia file sharing & storage," *Multimedia Tools and Applications*, vol. 78, no. 23, pp. 33837–33863, 2019, doi: 10.1007/s11042-019-08166-z.

[38] F. Abdelhedi, A. A. Brahim, F. Atigui, and G. Zurfluh, "Logical unified modeling for NoSQL databases," in *ICEIS 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems*, 2017, pp. 249–256, doi: 10.5220/0006311702490256.
[39] W. Deng, "Object-Oriented Database and O/R Mapping Technology," *Advances in Intelligent Systems and Computing*, vol. 1303, pp. 800–806, 2021, doi: 10.1007/978-981-33-4572-0_115.
[40] K. Michalakis and G. Caridakis, "Enhancing user interaction with context-awareness in cultural spaces," *Personal and Ubiquitous Computing*, vol. 27, no. 2, pp. 379–399, 2023, doi: 10.1007/s00779-022-01698-6.

BIOGRAPHIES OF AUTHORS

Aziz Smaala received his Bachelor degree in computer sciences from University of Oum El Bouaghi, Algeria in 2014. In 2018, he received a Master degree in distributed architectures from the Department of Mathematics and Computer Science at the University of Oum El Bouaghi, Algeria. In the same year (2018), he joined the University of Oum El Bouaghi as an Engineer in Computer Science. Currently, he is Phd student in the field of Computer Science at University of Oum El Bouaghi under the supervision of Dr. Zakaria Laboudi. In parallel, he joined the Research Laboratory on Computer Sciences Complex Systems ReLa (CS)2, in 2020. His research interests include artificial intelligence, machine learning, data analysis, context-aware pervasive systems, and multimedia content adaptation. He can be contacted at email: azizsmaala@univ-oeb.dz.

Zakaria Laboudi is an associate professor at University of Oum El Bouaghi, Algeria. He received his M.Sc degree in computer sciences from University of Constantine1, Algeria in 2009. He holds a Ph.D degree in Computer Sciences from University of Constantine2, Algeria since 2017. He also holds Habilitation degree (known also as Accreditation to Supervise Research) from University of Oum El Bouaghi, Algeria since 2019. Previously, he joined the MISC Laboratory at University of Constantine2, Algeria, and the ReLa (CS)2 Laboratory at University of Oum El Bouaghi, Algeria, for the period from 2010 to 2022, in which he was invited to collaborate in several research projects and tasks. His current research interests include artificial intelligence, machine learning, optimization, and data analysis, ontology-based computing, context-aware pervasive systems, ubiquitous learning, and multimedia adaptation. He can be contacted at email: laboudi.zakaria@univ-oeb.dz and laboudizak@gmail.com.

Asma Saighi is an associate professor at University of Oum El Bouaghi, Algeria. She received her M.Sc degree in Computer Sciences from University of Oum El Bouaghi, Algeria in 2009. She holds a Ph.D degree in Computer Sciences from University of Badji Mokhtar, Annaba, Algeria since 2018. She also holds Habilitation degree from University of Oum El Bouaghi, Algeria since 2020. She joined the LRS Laboratory at the University of Annaba, Algeria and the ReLa (CS)2 laboratory at University of Oum El Bouaghi, Algeria, for the period from 2010 to 2022. Currently, she is a member of the LIAOA Laboratory at University of Oum El Bouaghi, Algeria. Her current research interests include artificial intelligence, machine learning, service selection, cloud computing, cybersecurity, intrusion detection, data analysis and visualization, semantic and ontology computing, context-aware pervasive systems, multimedia content adaptation, and business intelligence. She can be contacted at email: pgasma_saighi@yahoo.fr.

Abdelkader Moudjari is an associate professor at University of Constantine 3 in Algeria. He holds an M.Sc degree in Computer Sciences from University of Constantine 1, Algeria (2001), and a Ph.D degree in Computer Sciences from University of Constantine 2, Algeria (2016). He also received a Habilitation degree from University of Constantine 2 (2020). Previously, he was a member of the MISC Laboratory at University of Constantine 2 from 2010 to 2021, where he collaborated on various research projects. His current research interests include artificial intelligence, BPM, crowdsourcing paradigm, machine learning, optimization, data mining, e-learning, and file logs analysis. As an active lecturer, he is involved in teaching, supervising, and mentoring computer science students. He can be contacted at email: moudjariabdelkader@gmail.com.