Personalization of Interactive Digital Media in Ubiquitous Educational Environments

Rafael D. Araújo, Taffarel Brant-Ribeiro, Renan G. Cattelan, Sandra A. de Amo
Universidade Federal de Uberlândia
Uberlândia, MG, Brazil
{rdaraujo, tbrantr} @gmail.com, renan@facom.ufu.br, deamo@ufu.br

Hiran N. M. Ferreira
IF Sul de Minas
Passos, MG, Brazil
hiran.ferreira@ifsdeminas.edu.br

Abstract—Multimedia capture and context awareness are two relevant subjects in ubiquitous computing research. While the former results in large amounts of digital media (e.g., video, audio, slides, text comments) being automatically produced, the latter offers the proper means to integrate and access such content. By merging the two concepts, users may benefit from tools that allow them to retrieve, view and interact with the captured media artifacts. In this paper, we present a contextual access architecture for personalization and recommendation of interactive multimedia content captured in a instrumented educational environment. Our approach takes into account context information, user preferences and presentation constraints in order to personalize the access experience, tailoring it to users needs.

Index Terms—Multimedia capture, content recommendation, personalization, interactive digital media, ubiquitous computing

I. INTRODUCTION

The term ubiquitous computing has been widely spread since its first use by Mark Weiser [20]. Since then, it turned into an emerging research area that aims at seamlessly integrating computers into the environment. The ultimate goal of ubiquitous computing is to unobtrusively assist people in their daily tasks, thus making the human-computer interaction process as invisible as possible.

Ubiquitous computing is usually divided into three main research themes [3]: natural interfaces, context awareness, and capture and access (C&A) applications. We explore the last two themes, context awareness and C&A applications, particularly in personalizing and recommending multimedia content captured from environments instrumented with ubiquitous computing devices.

An already known challenge associated with C&A concerns the management of the huge (and continuously increasing) amounts of information generated. Take for example the educational domain, with the capture of a university lecture in a classroom equipped with an electronic whiteboard, microphones and video cameras. In such scenario, several multimedia artifacts can be recorded in a single capture session – slides from the electronic whiteboard, audio streams from the microphones, and video from the cameras. On the other side of this chain, users are interested in accessing relevant pieces of information, aiming for particular media artifacts and with specific needs and tastes in mind. The information stored in an automated C&A system is, thus, useless if we do not connect users with their target information. Functionalities like content recommendation, ranking, personalization and preferences support are highly desirable among users.

A practical example of these demands can be represented by a student willing to find some exam-related material. Based on his context during access (e.g., access location, available time for study and device features), the system could offer him a personalized access experience, providing an interface that points directly to the content of his next scheduled exam. Moreover, personalization in this case could go beyond, adapting the already filtered contents to more specific constraints, like excluding the audio content because the user’s earphones are not available in the device or showing only the captured slides without the video because the connection speed is low.

In this paper, we present the Contextual Access Architecture (CAA), which explores context of access, user preferences and device presentation constraints in order to provide an interactive access experience. CAA allows content recommendation, ranking and personalization of interactive multimedia presentations captured in a instrumented classroom.

The remaining of the paper is structured as follows: in Section II, we detail CAA, presenting its key components and how they work together; in Section III, we describe Classroom eXperience, a lecture capture system which served both as a case study and as a testbed for our proposed architecture; in Section IV, we discuss related work; and, finally, in Section V, we make our final remarks and consider further work.

II. CONTEXTUAL ACCESS ARCHITECTURE

C&A applications are usually structured according to the four phases proposed by Abowd et al. [1]:

- Phase 1 – Pre-production: content is prepared for the capture system;
- Phase 2 – Live recording: relevant media streams are recorded from capture devices;
- Phase 3 – Post-production: media streams are synchronized and integrated;
- Phase 4 – Access: users view the resulting multimedia documents.

According to Kientz [7], the process of access in C&A applications is often left as secondary stage, and the effort invested in their development is not enough. If the access phase is developed with focus on the users, they will be more encouraged to visualize the captured content. Thus, good
retrieval, recommendation and personalization capabilities are key for the success of any C&A system.

Considering once again our educational domain scenario, at many times, the environment in which the student accesses the system presents implicit constraints. For example, if the bandwidth is low, it is not recommended to view the lecture’s recorded video. If device screen is too small, then textual information would be preferred over slides. Constraints like these can therefore directly influence the presentation experience of the captured content to the user.

Our approach takes into account the user’s access context, user preferences and, also, the presentation constraints which might be explicitly provided by the user or automatically inferred by the system.

A. User’s Access Context

There are many definitions for context available in the literature. Among the first who used the term ‘context-aware’, in 1994, for Schilit and Theimer [14], context is defined by some variables such as location, people and nearby objects and their changes. Other authors define context in a similar way, removing or adding some variables such as time of day, temperature, season, among others.

Abowd et al. [2] define context as any information that describes an entity. For them, an entity can be a person, a place or an object. The same authors also define ‘context-aware’ as the use of context to provide relevant information and/or services to the users.

In mobile and ubiquitous computing, the user’s context changes rapidly. Many applications use context variables to customize content presentation to users according to their current situation. This way, context is essential to provide a better user experience when retrieving the previously captured content.

According to Truong et al. [17], there is a minimum set of questions that must be considered when designing context-awareness support for C&A applications:

- **Who** are the users?
- **What** is captured and accessed?
- **When** the capture and access occur?
- **Where** the capture and access occur?
- **How** to perform the capture and access?

Based on Truong et al.’s original mapping, our work considers 7 dimensions\(^1\) for modeling user’s context in C&A applications for educational environments:

- The **type of device**, for example, "Desktop", "Tablet" or "Smartphone";
- The **device’s bandwidth**, measured in Kbps;
- The **screen resolution** of the device, for example, "up to 240x360", "up to 800x600" or "larger than 800x600";
- The access **date and time**;
- The user’s **available time** for viewing the captured content, for example, "up to 15 minutes", "up to 30 minutes", "up to 60 minutes" or "more than 60 minutes";
- The **place** where the user is performing the access, for example, "University", "Home" or "Work"; and,
- The **reason** why the user is accessing the system, for example, "Ordinary studies", "Test", "Quick review" or "Missed lesson";

Figure 1 shows each of these context dimensions, with some examples.

---

\(^1\) This limit can be easily extended according to the application.

---

B. User Preferences

User preferences are a set of personal preferences that, in our case, are used to recommend the captured content that better fit the user needs.

A student who wishes the lectures related to a test would be a good example of such demand. Based on his access context, the system could provide him a personalized access experience, presenting the content that points directly to the contents of the next scheduled test. Further, if the access date is close to that test date and the student is short in time, only the slides of the most important lectures could be presented instead of the complete contents (slides, audio and video) of all lectures.

Similarly, a visually impaired student could register a personal preference in the system stating that he always prefers audio playback over slides browsing interface.

There is a myriad of algorithms and methods that deal with user preferences [11], [5], [9]. Our goal is not to propose a specific algorithm or method, but rather support different ones as alternative mechanisms to collect user information and to identify his personal preferences.

C. Presentation Constraints

Imagine a user’s device connected with low bandwidth or, also, a student who wants to make a quick review before an upcoming class. In both cases, the video stream is not recommended. Thus, captured content should be presented in a personalized way – let’s say, just the lecture slides without the video stream.

Another situation would be, for example, a student using a mobile device with a low screen resolution. Perhaps this student would not see the detailed lecture slides very well; in this situation, the system could recommend only the lecture’s audio.

These are examples of some presentation constraints that may be imposed with regards to the user’s access context or...
even to user preferences. These constraints are not strict, in the sense they only suggest possibilities of personalization for the presentation of the multimedia content retrieved. This is accomplished throughout ranking of possible layouts and the user is free to choose the one that better fits his needs in order to visualize the lecture.

D. Architecture Detailed

All modules of our proposed architecture are loosely coupled, which means the architecture has been designed to allow different implementations of recommendation and personalization algorithms and/or methods, as shown in Figure 2.

![Diagram of the Contextual Access Architecture](image)

The first module, **Context Collection & Inference Module**, is responsible for gathering user’s context and performing some basic inference over it. Depending on how sophisticated are the sensing capabilities of the application, context may be automatically collected, explicitly informed by the user, implicitly inferred by the system, or a mix of them. The case is that, once context is set, it comprises the primary unit of information passed through CAS.

Based on context information, the user’s preferences and presentation constraints are selected next. Two correspondent repositories store preferences and constraints separately. Again, no matter how preferences are modeled, the architecture supports the use of personal profiles. Presentation constraints may be manually defined by users, but the system may also provide some default presentation constraints valid to all users.

Once the system is fully informed about the user’s context, preferences and all relevant presentations constraints, its next module, **Constraints Handling Module**, processes those constraints based on the user context and generates a presentation ranking. Here again, many algorithms for presentation ranking may be applied to the constraints.

Also based on user’s context, preferences and constraints, the **Query Handling Module** is responsible for rewriting the database query to select the better content results for the user. This module searches for lectures in the **Lectures Repository**, resulting in filtered multimedia content.

Finally, the **Presentation Module** is in charge of properly presenting the result set to the user. This module retrieves presentation layouts from a **Stylesheets Repository** and creates a rank, thus sorting the visualization interfaces more likely to fit the user needs. The user, however, is free to choose alternative options to visualize the lecture.

III. **CLASSROOM eXPERIENCE: A CASE STUDY SCENARIO**

We tested our approach with a real world application named **Classroom eXperience (CX)**, a multimedia capture system designed to automatically record lectures. CX performs capture, storage, access and extension of multimedia content in a classroom instrumented with electronic whiteboard, microphones, cameras and projectors. Media streams from each device are captured by specialized software components and, then, integrated and synchronized, generating hypermedia documents in different presentation formats. Built upon the iClass platform [13], CX uses CAA for content personalization, allowing the customization of captured content presentation according to the preferences and access context of each student.

CX assist instructors with ubiquitous computing resources without changing the conventional dynamics of the class, while students are assisted with later access to all information presented during the classes. Next, we detail CX features following the C&A phases previously discussed [1]. We particularly focus on the last phase, Access, where our CAA approach was more widely employed.

A. **Pre-production**

In CX, the pre-production phase is responsible for setting up the content that will be captured. At this stage, through an Web interface (Figure 3), the instructor registers all information relevant to the lecture. Such information is organized into three categories: (mandatory) basic information, additional data (optional meta information), and the lecture content itself.
The first category, basic information, is stored in a relational database containing links to other information. The second category, additional data, is classified and organized using extensions based IEEE Learning Object Metadata [6]. In the third category, content is packed in a single file that comprises slides in pdf, ppt or pptx formats. A specialized component is in charge of transforming this file into a custom template recognized by the capture system.

C. Post-production

The post-production phase starts immediately after live capture is finished. In this phase, the client application integrates all captured media streams, synchronize them and generates documents to be presented to the user. Ordinary presentation formats include HTML, for Web visualization, and NCL, used in Brazil’s Interactive Digital TV Platform [15].

D. Access

The access phase consists of presenting the captured content to students. CX uses our CAA approach to create an user-centered, personalized access experience that encourage the students to visualize and interact with the resulting presentation documents. CX uses the same context dimensions presented in Figure 1. Dimensions Place, Reason and Available Time are explicitly informed by the user, while the other dimensions are automatically inferred by the system.

For handling users preferences, CAA uses CPrefSQL [5], a SQL extension built to support contextual preference queries. CPrefSQL syntax allows us to express conditional preferences through a qualitative approach. The main purpose of using personal preferences in our architecture is to enable users to inform their preferences to the system in order to customize the queries accordingly. For instance, a user with short time for studying could inform he prefers to view easy lessons first (the instructor sets the difficulty level when creating a new lesson in the pre-production phase).

Finally, the presentation constraints provide content personalization by influencing the way in how content can be viewed and what media should be presented according to the user’s context. For example, if the student is using a smartphone at the gym, maybe it would be better to show him the lecture’s audio instead of the lecture’s slides or video.

Constraints are stored as XML documents. The system first defines default constraints applied to all users. As the user uses the system, he can define his own constraints that are added to or replace the default ones. The instructor can also register a calendar for each course, with exams’ and assignments’ dates. The students see these dates in the course’s homepage. Such information is used to recommend content. For example, if the student does not access the system for a week, then logs in the day before a scheduled test; the system may suggest him a summary with the lessons scored by the instructor as important for that test. Recommended content view is not enforced and student can also list the course’s contents chronologically or by title. Figure 5 presents a course’s homepage.

By clicking on a lecture, a popup is displayed with lecture’s general information and its presentation ranking (Figure 6), which lists first the presentation formats more likely to satisfy user’s context and preferences. Each option in the rank is associated with a distinct style-sheet which generates alternative views for the lecture’s media.

CAA supports the implementation of presentation ranking in many ways. Here, it has been implemented using a weighted

Figure 3. Creating a new lesson in Classroom eXperience.

B. Live recording

One of the main challenges of designing a C&A application is to make the live recording phase happen transparently to the involved users. The capture process must be as unobtrusive as possible, with minimal changes to the instructor routine.

The live recording phase occurs as the lecture is presented to students in the classroom. In the beginning, the instructor accesses a Web front-end to select the desired class and its contents are automatically loaded into the electronic whiteboard. From here, besides the instructor using an electronic pen instead of chalk, the lecture goes with no system interference. Behind scenes, a client application downloads all pre-prepared content and interfaces with capture devices available in the environment. The client application is actually a P2P module [4] consisting of three software components, one for each of the individual medias being captured: the audio capture component, the video capture component and the whiteboard component.

Figure 4 shows CX’s instrumented classroom equipped with electronic whiteboard, video camera and projectors. The instructor can interact with the electronic whiteboard as if he were using a traditional whiteboard or chalk board. All strokes written on the electronic whiteboard surface are projected on a second, larger screen for easier viewing by students.

For handling users preferences, CAA uses CPrefSQL [5], a SQL extension built to support contextual preference queries. CPrefSQL syntax allows us to express conditional preferences through a qualitative approach. The main purpose of using personal preferences in our architecture is to enable users to inform their preferences to the system in order to customize the queries accordingly. For instance, a user with short time for studying could inform he prefers to view easy lessons first (the instructor sets the difficulty level when creating a new lesson in the pre-production phase).

Finally, the presentation constraints provide content personalization by influencing the way in how content can be viewed and what media should be presented according to the user’s context. For example, if the student is using a smartphone at the gym, maybe it would be better to show him the lecture’s audio instead of the lecture’s slides or video.

Constraints are stored as XML documents. The system first defines default constraints applied to all users. As the user uses the system, he can define his own constraints that are added to or replace the default ones. The instructor can also register a calendar for each course, with exams’ and assignments’ dates. The students see these dates in the course’s homepage. Such information is used to recommend content. For example, if the student does not access the system for a week, then logs in the day before a scheduled test; the system may suggest him a summary with the lessons scored by the instructor as important for that test. Recommended content view is not enforced and student can also list the course’s contents chronologically or by title. Figure 5 presents a course’s homepage.

By clicking on a lecture, a popup is displayed with lecture’s general information and its presentation ranking (Figure 6), which lists first the presentation formats more likely to satisfy user’s context and preferences. Each option in the rank is associated with a distinct style-sheet which generates alternative views for the lecture’s media.

CAA supports the implementation of presentation ranking in many ways. Here, it has been implemented using a weighted
average algorithm to compute the user’s context similarity. Each context dimension has a specific weight that can be adjusted according to the access statistics.

Consider the presentation constraints set $R$. For each presentation constraint $R_i \in R$, computes:

$$Score(R_i, c) = \sum_{d \in D} w_d \times sim_d(R_i(d), c)$$

where:
- $R$: presentation constraints set
- $D$: user’s context dimensions
- $c$: user’s context instance
- $w_d$: weight of dimension $d$
- $sim_d$: similarity function of dimension $d$ between restriction $R_i$ and context $c$

All context dimensions (Place, Reason, Bandwidth, Available Time, Device, Date and Screen Resolution) have a specific similarity function which is independently computed. Some of them are a boolean function, for instance, the Reason dimension is a simple comparison between the constraint value and the context value, i.e., has it been chosen or not (1 or 0)? In contrast, other functions are calculated in a probability range, a real number between 0 and 1. Additionally, each context dimension has its weight considering the current constraint.

Captured content can be accessed via Web or in the SBTVD (the new Brazilian Digital Television System). Figure 7 shows an HTML page that presents the slides and video of a lecture.

IV. RELATED WORK

A major challenge for C&A applications regards the ability of providing efficient mechanisms to select and present the best content to users [18], [7].

Wang and Wu [19] present a robust learning environment which uses context-aware technology and content recommendation algorithms to assist learning. Among various features, their system uses RFID (Radio-Frequency IDentification) to provide context-aware functionality, and SCORM (Sharable Content Object Reference Model) patterns allow reusability and multiplatform support. As ours, their recommendation system analyses each student’s preferences according to their personal learning profile and recommends learning resources that best fit their needs.
considered. For instance, we are currently working on mining access context, preferences and constraints.

A captured lecture’s content (slides and video) presented as HTML.

Sugita and Yokota [16] introduce the User Interface Switching (UIS) approach, which allows adaptation of multimedia content according to users’ mental and physical abilities, computer facilities, and network QoS. Kinshuk et al. [8] describe an approach for personalization, summarization and adaptative presentation of content in remote interactions. Adaptable collaboration has also been investigated by Zhu and Zhou [21].

Lemos et al. [10] introduce MMedia2U, a mobile system which recommends photos based on the user context. In their work, only three context dimensions are considered: spatial, social and temporal. Context is modeled as an OWL ontology and recommendation is based on similarities. Nevertheless, they do not handle presentation constraints. Lopes et al. [12] present similar results with ontology-based architectural and semantic models.

Our approach differs from the previous discussed ones by considering user preferences strictly related to the context of access in order to derive presentation constraints for personalization of interactive multimedia content.

V. CONCLUSIONS AND FUTURE WORK

CAA can enhance existing automated multimedia capture applications where context and user preferences are relevant to produce personalized views of media artifacts. It supports content recommendation and ranking by design and is a useful tool for ubiquitous educational environments.

Composed by loosely coupled components that can be replaced according to the application needs, CAA allows extension and enables different implementations for each module (e.g., the ranking algorithm).

We showed how CAA integrates to CX lecture capture system. Therefore, instructors can present lectures in an interactive way, while students benefit from a large corpus of captured content for study, with presentation formats tailored to their access context, preferences and constraints.

As those types of applications are able to generate huge volumes of multimedia content, several future research can be considered. For instance, we are currently working on mining techniques for automatic inferring user preferences based and on the development of a social platform for students to extend captured content and improve recommendation capabilities.

ACKNOWLEDGMENTS

The authors would like to thank the Brazilian research agencies CNPq and FAPEMIG for supporting this work.

REFERENCES