Adaptive multimedia systems based on intelligent context management

Dimitris N. Kanellopoulos

Educational Software Development Laboratory (ESDLab),
Department of Mathematics,
University of Patras,
GR 26500, Patras, Greece
E-mail: d_kan2006@yahoo.gr

Abstract: Recently, multimedia computing turns into mobile, pervasive and context-aware. Adaptive multimedia computing is becoming the new paradigm for building complex information systems based on the use of different devices, web services and XML specifications. In dynamic environments, multimedia applications are potentially adaptive and are aware of their changing contexts. An intelligent context-aware infrastructure must be based on an appropriate context model such as to represent, manage and access context information. This paper proposes an ontology-based context model that supports context reasoning and context management for adaptive multimedia systems. The proposed model is flexible as it provides a general framework that can be applied to each multimedia application. Moreover, the model is extensible as it provides mechanisms to add new multimedia domain specifications.

Keywords: context ontology; context management; adaptive multimedia systems; web services.


Biographical note: Dimitris Kanellopoulos holds a PhD degree in Multimedia Communications from the Department of Electrical and Computer Engineering of the University of Patras, Greece. He is a member of the Educational Software Development Laboratory (ESDLab) in the Department of Mathematics at the University of Patras. His research interests include multimedia communications, intelligent information systems, knowledge representation, web engineering and web-based education. He has authored many papers in international journals and conferences at these areas. He serves as an Editor in various academic journals.

1 Introduction

Multimedia referred to as the melding of text, sound, photos and video to create amazing documents and presentations. Today, modern enterprises and organisations are equipped with multimedia systems that provide different types of services in order to meet and satisfy different user requirements. In a multimedia environment, adaptivity is considered in three different variations (Gecsei, 1997).
Adaptive multimedia systems based on intelligent context management

- **Adaptivity of multimedia content and services.** In this case, content and services are adapted according to user preferences and system constraints. Representative examples are adaptive hypermedia, web personalisation and intelligent user interfaces.

- **Adaptivity of multimedia systems and processing.** Multimedia systems adapt their execution at run time according to changing system requirements. In this case, the emphasis is given to the adaptive execution of the processes. Attention is also given on evolving environments like multimedia servers, streaming multimedia presentations, ubiquitous computing (UbiComp) and soft-real time multimedia systems.

- **Adaptivity of networks and communication.** Here, the emphasis is given at the communication and multimedia data transfer, as well as at the ability of the network resources and protocols to adapt their transmission according to the communication requirements and the characteristics of a multimedia connection. User mobility, bandwidth variations during communication and network congestion are taken into account for achieving this type of adaptivity.

Many researchers have investigated the problem of adapting the multimedia content to different networks, terminals and users. The purpose of these efforts is to realise the universal media access (UMA) vision, which means access to content by any terminal and any network (Pereira and Burnett, 2003). In the direction of this target, media adaptation might be performed at different entities (i.e., media servers, gateways, network and terminals) of the digital information distribution chain from content/service generation to end-user terminals. For example, intelligent media servers can adapt the content to different client capabilities (i.e., display size, type of CPU) (Jannach and Leopold, 2007). Similarly, intelligent gateways can adapt the content to different network characteristics, when the properties of the content and networks are properly represented.

Media adaptability can be achieved using media coding, which can be addressed by the MPEG-1–2 and 4 standards (ISO/IEC 14496-1, 2001). Van Beek et al. (2003) discuss how content description can be addressed by the MPEG-7 standard. The MPEG Requirements Group (2003) analyses how the MPEG-21 standard addresses the usage environment (terminal, network, user preferences) and specifying a framework for multimedia delivery and consumption that aims at realising the UMA vision. Magalhães and Pereira (2004) illustrate multimedia customisation using the above concepts, whereas Di Cagno et al. (2006) establish a framework for adaptation at multimedia end-terminals, identifying a set of user-level QoS (Quality of Service) descriptors that can be used for this purpose. Gecsei (1997) states that adaptation as a tool can improve user satisfaction and as a technique can be complementary to resource reservation. To maximise the benefits of adaptivity, multimedia applications should have access to information about the state of the operating environment and to the resource requirements of other competing applications. Khan et al. (1997) proposed a mathematical model (the utility model), which incorporates the dynamics of adaptive multimedia systems. The utility model is based on the concepts of quality profile, quality-resource mapping, session and system utility and of system resources constraints. In adaptive multimedia environments, the key issue is the context management.

Adaptive multimedia systems enforce the vision of customised and context-aware service provisioning (Guan and Lai, 2005). In particular, the multimedia service provided
and its characteristics have to meet users’ profile, in terms of preferences expressed on
the service behaviour and content and to user context. In a context-aware environment,
the user interacts with the service via his device in use or through the network access
characteristics. Adaptation of content according to the end-user physical context and his
personal profile (or user model) constitutes a hot research topic for designing adaptive
multimedia applications. Telecommunications, physical and electronic environments, as
well as the host device characteristics and the user model must be considered.

User modelling is concerned with the identification of outstanding aspects of the user
and the subsequent usage of these in the customisation or personalisation of services
(Kobsa, 2001). The user model might contain rules and heuristics collected from the
continual observation of some particular aspect of the user’s behaviour over a large time
period. However, one popular application of user modelling techniques is the ‘filtering’
or ‘personalisation’ of content retrieved from various sources (e.g., web resources)
(Ardissono et al., 2005). Many multimedia systems adopt the profiling technique through
which user requirements are represented and web pages content is adapted based on user
requirements (Chan, 2000). Through the profiling technique, data are collected and
manipulated with the goal of identifying and describing the profile of an entity such as a
user, an object, a product or a process. A ‘profile’ is a structured representation of the
information that describes the users and their preferences along the services and the
multimedia data they access. Users’ preferences are considered:

1 implicit, if they are inferred by the analysis of personal characteristics and behaviour
2 explicit, if user requirements are extracted directly from the interaction with users
themselves.

According to Dey (2001) quotation…

“Context is any information that can be used to characterise the situation of
entities (i.e., whether a person, place or object) that are considered relevant to
the interaction between a user and an application, including the user and the
application themselves.”

Generally speaking, context is the location, identity and state of people, groups and
computation and physical objects. However, in ubiquitous and mobile computing
applications, other kinds of context can be used like physical (e.g., location and time),
environmental (e.g., weather and light) and personal information (e.g., mood and
activity). In intelligent context-aware computing, ontologies can be very constructive for
three main reasons:

• Ontologies let users of context-aware platforms express their preferences on a
  well-defined domain.
• The context-aware platforms themselves can create a context.
• The clear separation of ‘domain knowledge’ and ‘inference knowledge’ (e.g., logic
  rules by which a multimedia system becomes adaptive) could allow us to build
easy-to-maintain domain knowledge bases and to use highly optimised and domain
independent reasoning engines, which flexibly process the contents of the underlying
knowledge base (Brewster and O’Hara, 2004).

Ontologies constitute an integral part of the semantic web, which is an extension of the
current web in which information is given well-defined meaning (Berners-Lee et al.,
Adaptive multimedia systems based on intelligent context management

2001). Ontologies let applications share, exchange and compose concepts from different domains (Chandrasekaran et al., 1999). Semantic web languages such as the web ontology language (OWL) are used to describe ontologies (Smith et al., 2004).

This paper proposes an ontology-based framework to define and manage context in multimedia environments. The framework emphasises a set of functionalities that are essential to manage context in multimedia systems. The context is specified using a flexible and extensible ontology-based model. The model is flexible, since it provides a general framework that can be applied to each multimedia application in which it is required to manage user context. The model is also extensible, as it provides mechanisms to add new domain specifications on which new aspects of the context can be instantiated. The approach proposed for modelling context relies on the use of ontologies to describe context. Ontologies and OWL can be used to connect different multimedia domain specifications in order to define the general context ontology. In addition, ontologies can help software engineers to develop multimedia applications capable to reason about the context definition for supporting dynamic and adaptive service selection or user profiling.

The structure of the remainder of this paper is as follows. Section 2 discusses related work, while Section 3 describes the context model. Section 4 proposes a framework for managing context information for multimedia applications. Section 5 concludes the paper and draws avenues for further work.

2 Related work

As Jameson (2001) refers, context is specified in three main research fields:
1. ubiquitous and mobile computing
2. context-aware applications
3. pervasive computing.

In ubiquitous and mobile applications, users can access data and information from a great variety of devices and applications with different capabilities and computational resources. In this case, the context is used to deliver the appropriate content using the presentation that is best suited for the user characteristics. In order to define the context information, we have to identify environmental conditions and device characteristics. In most cases, this information is implicit and gathered from sensors and plug-ins that are deployed in the environment surrounding the user. In the ubiquitous web applications (UWA) project (Consortium, 2002), many different models and methodologies were proposed for developing UWA starting from user requirements. The customisation of web applications is achieved from user profiling based on requirements analysis rather than from the management of an actual user context. Capra et al. (2003) propose a reflective architecture for mobile devices and an approach to deal with conflicts between context features and user requirements. O’Grady et al. (2007) presents an architecture (called Gulliver’s Genie) for the development and deployment of intelligent multimedia applications for the ubiquitous and mobile computing community. This architecture supports content adaptivity through context awareness. The Gulliver’s Genie adopts a multi-agent system (MAS) approach in the intelligent delivery of personalised content.
Context is viewed as an aggregation of user profile, user device, network state and physical environment.

Context-aware applications use context to provide relevant information and services to the user (Jameson, 2001). In this case, the problem is not only related to adapt content to the user computational capabilities, but other elements of the context (e.g., user behaviour or preferences) have to be considered in order to choose the most relevant information and service for the user. A typical example of context-aware applications is the virtual traveller’s guides (Cabri et al., 2003). Actually, in these environments the most important factor in defining the context is related to the user preferences in terms of interests or disabilities. Other typical aspects of Ubicomp, such as user location or device in use, become second order concerns that appear only at the moment of providing the content using a customised presentation layer.

The founding idea of pervasive computing is the human-centred computation in which computational resources and services are freely available everywhere and cooperate seamlessly to support users’ tasks (Gu et al., 2004). In order to fulfil this vision, context is not only used to identify the environment and user characteristics for executing tasks, but it is also exploited to infer new knowledge about users to help them in a transparent and proactive manner. Pervasive computing underlines the need to provide formal context descriptions and rules by which new context features can be extracted.

2.1 Context management frameworks (CMF)

The context toolkit framework contains guidelines to help developers in building context-aware applications quickly and from scratch without limiting the complexity of the context space (Dey et al., 2001). Although this proposal represents a great effort in trying to formalise context structure and context-aware computing aspects, it still views the context as a collection of data coming from sensors surrounding users, without considering users’ preferences or other context features. Henricksen and Indulska (2004) discuss a framework for pervasive context-aware applications using the context modelling language (CML). Their framework is oriented toward the use of logics to reason on the context instead of the description of mechanisms to extend and manage context information. The CML represents an ad-hoc language to specify types of information, their classification, dependencies and quality metadata. Kawamiichi et al. (2004) describe a context model for service selection; the model is applied to a middleware for gathering context information for a guidance board PDA application. Additionally, Batini et al. (2007) propose a visual language that allows for adapting the presentation of multimedia data to different physical devices. It is worth mentioning that Bolchini et al. (2004) propose a methodology to support very small databases usage in context-aware applications. Preuveneers and Berbers (2005) present a component-based approach for managing context information. Their novel contribution is that their management system itself can be adapted to a device’s capabilities or service requirements by enabling or disabling certain components or specific properties of certain components. The smart spaces framework GAIA (Ranganathan and Campell, 2003) proposes an infrastructure that supports the gathering of context information from different sensors and the delivery of appropriate context information to Ubicomp applications. In their framework, the working group developed a flexible and expressive
Adaptive multimedia systems based on intelligent context management

2.2 Context ontologies

Wang et al. (2004) created an upper ontology, the CONON context ontology, which captures general features of basic contextual entities, a collection of domain specific ontologies and their features in each sub-domain. The CONON ontologies are serialised in OWL-DL. The CONON supports two types of reasoning:

1. reasoning to detect and correct inconsistent context information
2. reasoning as a means to derive higher-level context information.

Chen et al. (2004) proposed an ontology for context-aware pervasive computing environments used in the CoBrA system. CoBrA provides a set of OWL ontologies developed for modelling physical locations, devices, temporal concepts, privacy requirements and other objects within UbiComp environments. Recently, Doulkeridis et al. (2006) proposed a system architecture for service discovery, based on a novel context-aware service directory. In particular, special context characteristics were used for service discovery, while services were described with web services description language (WSDL) documents (Christensen et al., 2001). The use of ontologies as a way to specify context has emerged as a proper approach to deal with context and rules derived from context specification (Gu et al., 2004). Ontologies really enable the definition of concepts and relations between context elements and provide flexibility and extensibility because they can be combined in order to include descriptions of different domains such as quality and security metadata, in the context specification (Henricksen and Indulska, 2004). Usually, a user profile describes user requirements with a list of \texttt{<attribute value>} pairs, where the \texttt{value} describes a specific user or user class. Each requirement can be classified as:

- \textit{Implicit}: when the user does not specify the value explicitly. The implicit requirements are frequently inferred by the analysis of the user characteristics and the identification of a users’ class to which the user belongs.

- \textit{Explicit}: when the user specifies the value.

2.3 Open issues

In most mobile multimedia applications, context is often provided as a collection of specific user characteristics, such as device and ‘communication path’ features. A lot of applications related to the tourism domain (i.e., virtual traveller’s guides) treat the context as a set of information on users’ location and preferences (Cabri et al., 2003), group belonging, or users’ behaviour (Jameson, 2001) to give tourists information and services closely related to their interests. The literature lacks the presence of frameworks and architectures to manage context definition and expansion for multimedia environments. It is remarkable that the definition and consequent usage of context for ubiquitous, mobile or context-aware applications and pervasive computing is strongly limited and applied to specific domains. In particular, there are no research efforts in the literature that describe the elements that can be contained in a general multimedia context. The objective of this
paper is to provide a general and extensible users context definition that can be used at different architectural levels of adaptive multimedia systems. In this work, we adopt the notion of context as any information that can be used to characterise persons, places or objects that are considered relevant to the interaction between a user and an application, including users and applications themselves.

3 The proposed context ontology for adaptive multimedia systems

The goal of this paper is to provide the elements and guidelines to define and create a user profile in any multimedia domain. In order to describe the multimedia context, we use an approach based on ontologies. This is because ontologies allow a flexible and extensible definition of the concepts characterising a specific element that in this case is a general context. In order to formalise the proposed ontologies, we have used the OWL language. The ontology representation schema adopted here includes ‘structural’ axioms, that is, axioms drawn from the proper structure of the ontology (i.e. axioms that result from the relations concept has attribute, concept A is a concept B, concept A is a part of concept B, etc.). With the term ‘axiom’ we mean all the rules and/or constraints among the elements considered in the model: concepts, attributes, relationships and attribute values. It should be clarified that defining ontologies without non-structural axioms does not mean that this sort of axioms cannot be defined as a (part of the) specification of a conceptualisation.

Figure 1 depicts the context ontology, which specifies the general concepts defined independently of a domain of application and that can be used in different multimedia application domains. The context here defined is composed of three fundamental elements:

1. the user profile
2. the surroundings
3. the communication path.

- The ‘user profile’ describes the properties associated with the user and it is distinguished in domain-independent user profile and domain-dependent user profile. The former contains all the characteristics that describe the users along their social distinctive properties. It includes personal data, their physical abilities and general interests. Characteristics are usually expressed as
  \(<\text{attribute value}>\) pairs. The latter refers to the specific multimedia domain in which the context is considered and it usually contains the preferences along specific multimedia services. They are usually expressed as both
  \(<\text{attribute-operator-value}>\); the specification of operators, such as
  =, >, or < allows the definition of complex constraints along the service characteristics. An example of constraint in the tourism domain is
  ‘hotels.location=Athens’; this indicates that in the booking service
  the user wants to select only hotels, which are located in Athens.

- The ‘surroundings’ collects the information about the geographical position, the ambient condition, the temporal details and the actions that characterise the interaction of the users with the surrounding space. Light level, temperature, humidity, noise and motion are parts of ambient condition.
Finally, the ‘communication path’ describes the elements that characterise the interaction of the users with the communication platform used to access services. The considered model characterises the communication path through multimedia device features, network, network interface and application protocol characteristics. For each subclass, there are ontologies that specify the admissible terms and concepts.

Figure 1  The context ontology model (see online version for colours)

Figure 2 illustrates an example of ontology for the ‘action’ specification. As it is shown in Figure 2, an ‘action’ characterises the interaction of the users with the surrounding space and it has a name and a profile. An action is executed in a certain location and exacting software agents and current persons are involved in it. Moreover, a multimedia network that consists of resources, protocols and network topologies supports an action.

A large variety of data is stored and used to provide a significant context. These data have to be associated with metadata, both for distinguish them in their heterogeneity and for the need to formalise their usage and management policies.
4 The context management framework

Figure 3 The CMF (see online version for colours)
The proposed CMF draws attention to the necessary functionalities needed by a generic context-aware multimedia system that aims to manage its context using a semantics-based approach. However, the CMF does not provide a software system that can be used by context-aware information systems for managing their context. The CMF is a MAS architecture, in which agents are lightweight and expected to operate in pervasive environments. The term ‘agent’ identifies a person or a software component that interacts with the context-aware information system (Hendler, 2001). Agents of the proposed architecture are implemented in host multimedia computers and in user nodes. User nodes can support any available lightweight-agent platform, such as the light-weight extensible agent platform (LEAP), which is based on the Java agent development framework (JADE, http://jade.tilab.com). The inference agent and knowledge base repositories span several nodes to decrease bottlenecks and increase scalability and robustness against failure. Figure 3 depicts the proposed framework.

The CMF comprises of various agents and two repositories:

- **User agents** represent users in the multimedia environment and they are negotiating on users behalf to set both the required context specifications and the methods of context representation and delivery.

- **Service agents** represent multimedia services in the environment. They keep service profiles that define service capabilities and the knowledge required to negotiate each service’s best performance according to context.

- **Sensor agents** represent sensors in a multimedia environment, in which sensors record various context parameters such as light level, temperature, humidity, noise etc.

- **Event monitoring agents** represent device sensors and detect the location or the ambient conditions that may occur in a multimedia environment. For example, the rule: \( \text{If } (\text{BodyTemp} > 37.4) \text{ and BodyTemp} \leq 38 \text{ then EnableFeverEvent} \). This logic rule can be implemented using an artefact that monitors the health status of a person based on measurements of sensors attached to person’s body. An artefact has two different levels of context:
  1. the low level in which information is acquired from its own sensors
  2. the high level that is an interpretation of its low-level context information

In a multimedia environment, there are many artefacts used for the generation of audio messages (e.g., an eMP3Player) for sending SMS (e.g., an eMobilePhone) etc. Actually, in an ambient intelligence (AmI) environment, various artefacts can be treated as components of UbiComp applications and users can compose UbiComp applications by creating association between the artefacts (IST Advisory Group, 2001). The plug/synapse model, invented in the extrovert-gadgets project (http://www.extrovertgadgets.net), serves as a common interfacing mechanism between artefacts providing the means to create large scale systems based on simple building blocks. Plugs make visible the artefact’s properties, capabilities and services to people and to other artefacts, while synapses are associations between two compatible plugs.

- **The ontology agent** provides the semantic functionalities that other agents can use to represent and share context in the system.
The context management agent (CMA) is the core of the framework and the system administrator. The CMA manages and reasons on the context and is sharing context information with the other agents. The CMA negotiates with user agents to achieve context-level agreement (CLA). The CMA stores relevant information in a knowledge base repository for inference, consistency and knowledge sharing. In our framework, we adopt the context-level negotiation protocol introduced in (Khedr and Karmouch, 2004). Context-aware environments must allow adaptive and autonomous access to context information. A multi-agent middleware can use a negotiation protocol and ontology model to make the environment more easily personalised at run time and adapted and managed at provision time (Khedr and Karmouch, 2004). The CLA is a declarative method with which users state their context requirements and context providers offer dynamic customised context-based service information. According to Khedr and Karmouch (2004), we have three types of CLA:

- Passive context. Each user’s context specifications are simple, requiring little or no complex operation from the context provider.
- Active context. In multimedia environments, we have obviously an active context. Indeed, users’ context specifications result from complex content provider operations such as filtering, merging, composition and monitoring.
- Spontaneous context, in which context management and inference cope with the uncertainty at higher context levels.

The inference agent performs reasoning over the runtime context and is used for analysing the context during the runtime phase. In particular, it uses logic-reasoning mechanisms to ensure that context instances are consistent with each other and with arguments defined in the context ontology. In our framework, the inference agent is based on the Java expert system shell (JESS) rule engine, i.e., the rule engine for the Java platform (http://herzberg.ca.sandia.gov/jess).

Context provider agents are responsible for capturing the raw data from the source and interpreting it so that it’s understandable by other agents.

The repository (on the left side of the Figure 3) contains the context ontology, the metadata definitions and the rules for the context management. This component comprises the ontologies repository and the metadata and rules repository, which are the repositories employed for storing all the information needed by the framework. The first repository contains the context ontology and the domain specific ontologies, while the second stores the metadata and the respective rules. The context ontology, the metadata and the rules are defined during the design phase and are stored into the registries by the system designer. Domain specific ontologies can also be added and stored at runtime.

Once a context instance is created, it is saved into the runtime context repository, which is used for storing, coherently with the annotation (labelling), every instance of an agent’s context and it is modified every time that an agent changes its context. The initial instance of the context is also stored into the old context repository, which contains the information employed for tracing the evolution of the context over time. At runtime, the CMA interacts with the agents for gathering information about the context and keeping the registries up to date. This interaction is performed following the guidelines that are obtained combining the set of rules with the results that can be acquired using both the
runtime and the old context. The inference agent performs reasoning over the old context. For instance, the inference agent can be used for analysing the variations of the context over the time and retrieving information about the behaviour of agents (Middleton et al., 2004).

5 Conclusions

In ubiquitous, mobile and pervasive computing, the management of user context and profile is a primary concern. The characteristics of user context are heterogeneous and often span from domain dependent preferences to metadata expressed on context features. For this reason, many researchers have proposed special frameworks and context-aware applications for specific domains. In this paper, we proposed a flexible and extensible ontology-based model for defining and managing user contexts for adaptive multimedia systems. The ontology-based context model supports context reasoning and context management for adaptive multimedia systems. It is flexible and extensible as it provides mechanisms to add new multimedia domain specifications. The model is based on ontologies, which allow the definition of concepts and relations of different multimedia domains and permit the inclusion of metadata labelling definition for context management. In the near future, we aim to demonstrate the suitability and feasibility of the proposed framework considering services related to various multimedia sub-domains. Furthermore, we will focus on various context-reasoning mechanisms for reasoning about a range of adaptive multimedia environments that may adopt first-order logic (FOL), probabilistic logic and high order logic. Moreover, context specification will be used in the design of a prototype adaptive multimedia system.

References


