Context-Aware Multimedia Content Adaptation Framework

Georgia Troullinou

Thesis submitted in partial fulfillment of the requirements for the

Masters’ of Science degree in Computer Science

University of Crete
School of Sciences and Engineering
Computer Science Department
Knossou Av., P.O. Box 2208, Heraklion, GR-71409, Greece

Thesis Advisor: Prof. Dimitris Plexousakis

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH).
Context-Aware Multimedia Content Adaptation Framework

Thesis submitted by
Georgia Troullinou
in partial fulfillment of the requirements for the
Masters’ of Science degree in Computer Science

THESIS APPROVAL

Author: Georgia Troullinou

Committee approvals:
Dimitris Plexousakis
Professor, Thesis Supervisor

Constantine Stephanidis
Professor, Committee Member

Dimitris Grammenos
Principal Researcher, Committee Member

Departmental approval:
Angelos Bilas
Professor, Director of Graduate Studies

Heraklion, November 2012
Context-Aware Multimedia Content Adaptation Framework

Abstract

Nowadays, people become more and more absorbed with media content. A large number of digital media files such as video, image and audio are generated, transmitted and stored every single moment. Additionally, the range of devices which are available to access media content becomes both increasingly heterogeneous and ubiquitous too. Users can receive content through a number of devices, which provide network connectivity but they have different properties. As a result, they expect to access the information according to their preferences and abilities without paying attention to restrictions regarding the time or the location. Consequently, as the availability and consumption of multimedia content increases and in conjunction with the growing heterogeneity of devices, networks, users and delivery environment factors, the challenge of real time-content adaptation arises.

This thesis proposes a context-aware multimedia content adaptation framework (ADF) that achieves the targets of ubiquitous access of multimedia information and personalization to the user’s situation and preferences, overcoming any compatibility issues. Specifically, a set of upper-level ontologies provide a descriptive modelling of the context, exploiting the expressiveness of semantic technologies. Furthermore, a fundamental part of this modelling is the description of complex composition of multimedia content based on the MPEG-7 standard. The reasoning about the adaptation process is accomplished by multi-step decision rules, which are able to manage the captured knowledge and construct an adaptation plan. Finally, the proposed approach can be extended to different application domains by adding specific concepts and introducing new technologies.
Προσαρμογή Πολυμεσικής Πληροφορίας με Επίγνωση Συμφραζομένων

Περίληψη

Στη σημερινή εποχή δεχόμαστε καταγγελία πληροφοριών από διάφορα μέσα με προσδόξως ρυθμούς. Κάθε στιγμή, ένας μεγάλος αριθμός από ψηφιακά πολυμεσικά αρχεία όπως το βίντεο, η εικόνα και ο χώρος δημιουργούνται, μεταδίδονται και αποθηκεύονται. Επιπλέον, οι διαδέσμες συσκευές, οι οποίες έχουν πρόσβαση στην πληροφορία ενών χωρίς μόνο πολυάριθμες αλλά και ετερογενείς. Οι χρήστες έχουν στη διάθεσή τους ένα μεγάλο αριθμό από διαφορετικές συσκευές, που τους παρέχουν δικτυακή υποστήριξη, ώστε να λαμβάνουν πολυμεσική πληροφορία. Συνεπώς, αναμένουμε να έχουν πρόσβαση στην πληροφορία σύμφωνα με τις προτιμήσεις τους και τις δυνατότητές τους, χωρίς χρονικούς περιορισμούς και από οποιοδήποτε τοποθεσία. Έτσι, χαθεί τη διαθεσιμότητα και η κατανόηση της πολυμεσικής πληροφορίας διαγωνίζονται αλλά και σε συνδυασμό με την αύξηση της διαφορετικότητας των συσκευών, της δικτυακής υποδομής και του πλήθους παραγόντων που ισχύουν τους χρήστες και το περιβάλλον τους, η πρόκληση για την δυναμική προσαρμογή του περιεχομένου αυξάνεται.

Η μετατυπωγική αυτή εργασία αναπτύσσει την ιδέα ενός συστήματος που λαμβάνει υπόψη τις παραμέτρους του περιβάλλοντος και προσαρμόζει αναλόγως την πολυμεσική πληροφορία πάνω σε αυτές. Έτσι, ζητερώντας τυχών προβλήματα συμβατότητας, επιτυγχάνεται η διάχυτη πρόσβαση στην πληροφορία, εξασφαλίζοντας την πάνω σε κάθε χρήστη ξεχωριστά με βάση την κατάσταση και τις προτιμήσεις του. Συγκεκριμένα, μέσω ενός ανώτερου εισπράξιμου συνόλου οντολογιών προσφέρει μια περιγραφική μοντελοποίηση του περιβάλλοντος, εκμεταλλεύοντας την εκφραστικότητα της σημασιολογίας τεχνολογίας. Βασικά χωρίζει σύντομος της μοντελοποίησης αποτελεί η περιγραφή της συνθέτης πολυμεσικής πληροφορίας με βάση το MPEG-7 πρότυπο. Η συλλογιστική για την διάδοξα της προσαρμογής της πολυμεσικής πληροφορίας επιτεύχθηκε από ένα πολυπλεξίδε δίονο χαρτών απόφασης, που είναι σε θέση να διαχειριστεί την καταγραφική γραφή και να κατασκευάσει ένα σχεδίο προσαρμογής. Τέλος, η προταθεμένη προσέγγιση μπορεί να επικεντρώσει σε διαφορετικά πεδία εφαρμογής, προσλήφθει γενικά συγκεκριμένων έννοιας και εισαγόντας χαρακτηριστικές τεχνολογίες.
Ευχαριστίες

Θα ήθελα να ευχαριστήσω όλους εκείνους που ο καθένας με το δικό του τρόπο με βοήθησαν να ολοκληρώσω την μετατυπώση της εργασίας.

Επικίνδυνα, θα ήθελα να ευχαριστήσω θεραπεύοντα τον επίπτωση καθήγητή μου, κ. Δημήτρη Πελέκουσα για την επιστημονική καθοδήγησή της, συμβουλές, την υποστήριξη και την εμπιστοσύνη που μου έδωσε καθ’ όλη την διάρκεια της εκπόνησης της μετατυπώσης μου εργασίας.

Θα ήθελα επίσης να ευχαριστήσω τα μέλη της εξεταστικής μου επιτροπής κ. Κωνσταντίνος Απερανίτη και κ. Δημήτρη Γραμμένο για τις εποικοδομητικές παρατηρήσεις τους.

Κατά τη διάρκεια αυτής της μελέτης, είχα την τύχη να αποτελώ μέλος του Εργαστήριου Πληροφορικών Συστημάτων του Ινστιτούτου Πληροφορικής του Ιωάννικα Τεχνολογίας και Έρευνας. Ορέδεια ένα μεγάλο ευχαριστώ στα άτομα του εργαστηρίου και ιδιαίτερα στους «γείτονές μου», οι οποίοι με ενθάρρυναν και με ανέμειναν όλο αυτό το διάστημα. Ιδιαίτερη αναφορά αξιώνει ο Ιωάννης Χρυσάκης, που μου πρόσφερε ανεκτιμήτη βοήθεια για θεωρητικά και πρακτικά ζητήματα.

Ορέδεια να ευχαριστώ τις θερμότερες ευχαριστίες, στους άνδρες μου Μάρη, Διονυσία, Μάτθαιο και Μάνο που μου πρόσφεραν ψυχολογική υποστήριξη την κατάλληλη στιγμή, με βοηθήσαν να ξεπεράσω φάσματα και αναφαίλειες, και έκαναν πιο ευχάριστη αυτή τη διάρκεια. Ακόμα, θέλω να ευχαριστήσω τους ανθρώπους που γνώρισα και συναντήθηκα μαζί τους αυτή τη περίοδο, οι οποίοι μου πρόσφεραν όμορφες στιγμές. Στο σημείο αυτό, θα ήθελα να ευχαριστήσω και την εξαδέλφη μου Γεωργία, η οποία αν και μακριά, ήταν παρόν σε όλες τις ανησυχίες αυτών του διαστήματος.

Ολοκληρώνοντας, θέλω να προσφέρω την ευγνωμοσύνη μου στην οικογένεια μου, η οποία μου πρόσφερε όλα τα ερέθισα για να ανταποκρίνομαι και σε αυτό το βήμα. Ο πατέρας μου, Σπύρος, με ενθάρρυνε, δύναταις μου αξιολόγηση δύναμη σε όλες τις δυσκολίες αυτής της περιόδου. Συγκεκριμένα θα ήθελα να χάνω στην μητέρα μου Μαρία, η οποία ήταν πάντα υποστηρικτική και συμπαραστάθηκε μου σε όλες τις στιγμές της ζωής μου. Η αλοκλήρωση αυτής της εργασίας θα αποτελούσε μεγάλη χαρά για αυτήν και παρόλο που δεν πρόκειται να δεί ολοκληρωμένο το εγχείρημα αυτό, είμαι σίγουρη ότι η σκέψη της είναι που με καθιστήσασε να στέσω στο τέλος αυτής της διαδρομής. Για αυτούς τους λόγους η εργασία είναι αφιερωμένη σε αυτήν.
Στη μηνέρα μου, Μαρία
## Contents

1 Introduction .......................... 1
   1.1 Motivation .......................... 1
   1.2 Contribution of this thesis ............ 2
   1.3 Organization of the remaining chapters  3

2 Background Theory ...................... 5
   2.1 Ambient Intelligence (AMI) .......... 5
   2.2 Ambient Intelligent Context .......... 5
   2.3 Context-aware applications .......... 6
   2.4 Multimedia Content ................. 6
   2.5 Supporting standard technologies and prototypes ...
      2.5.1 MPEG-7 Standard ................. 7
      2.5.2 Semantic Web ..................... 9
      2.5.3 Ontologies in Semantic Web ...... 10
      2.5.4 Web Ontology Language (OWL) ... 11
      2.5.5 Reasoning ....................... 12

3 Literature Review ....................... 13
   3.1 Ontologies for Context Modelling .... 13
      3.1.1 General context ontology ......... 14
      3.1.2 Device ontology .................. 17
      3.1.3 Multimedia Content Ontology .... 19
   3.2 Context-Aware Content Adaptation Systems 21

4 Context-Aware Multimedia Content Adaptation Framework .... 25
   4.1 Pervasive Computing Environment ...... 25
      4.1.1 Middleware ....................... 26
   4.2 Adaptation Decision Framework (ADF) ... 26
   4.3 ADF Architecture ..................... 27
      4.3.1 Context-Content Provider ......... 27
      4.3.2 Knowledge Base (KB) .............. 29
      4.3.3 Adaptation Decision Engine (ADE) 29
      4.3.4 Content Adaptor (CA) ............ 30
### Contents

4.4 Ontology-Based Context Modelling

4.4.1 Core Ontology .................................................. 31
4.4.2 User Profile Ontology .......................................... 33
4.4.3 Device Ontology .................................................. 34
4.4.4 Multimedia Content Ontology ................................. 36

5 Methodology of Adaptation Decision Procedure .......................... 39

5.1 A use case scenario .................................................. 39
5.2 Briefly Syntax Description of Jess Engine ............................. 40
5.3 Decision Procedure .................................................. 40
    5.3.1 User in adaptation decision .................................. 41
    5.3.2 Device in adaptation decision ................................. 43
    5.3.3 Display adapted information .................................. 47

6 Concluding Remarks and Future Work ...................................... 49

6.1 Conclusion .......................................................... 49
6.2 Future Work ......................................................... 49

Bibliography ........................................................................ 51

A Implementation Tools ........................................................ 55

A.1 OWL Language & Protégé - Knowledge Base Representation .... 55
A.2 Jess, a rule-based system .............................................. 55

B A sample of rules for image adaptation .................................... 57
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Overview of MPEG-7 Multimedia DSs</td>
<td>8</td>
</tr>
<tr>
<td>3.1</td>
<td>Particular definition of CONON upper ontology</td>
<td>15</td>
</tr>
<tr>
<td>3.2</td>
<td>SOUPA consists of two sets of ontology documents: SOUPA Core and SOUPA Extension</td>
<td>16</td>
</tr>
<tr>
<td>3.3</td>
<td>MULTICAO ontological model</td>
<td>17</td>
</tr>
<tr>
<td>3.4</td>
<td>Device Ontology</td>
<td>18</td>
</tr>
<tr>
<td>3.5</td>
<td>Class hierarchy of the multimedia content and segment classes in the Harmony MPEG-7 based ontology</td>
<td>20</td>
</tr>
<tr>
<td>3.6</td>
<td>Class hierarchy of the multimedia content and segment classes in the SmartWeb MPEG-7 based ontology</td>
<td>22</td>
</tr>
<tr>
<td>4.1</td>
<td>Overview of Pervasive Computing Environment</td>
<td>26</td>
</tr>
<tr>
<td>4.2</td>
<td>Adaptation Decision Framework Architecture</td>
<td>27</td>
</tr>
<tr>
<td>4.3</td>
<td>XSLT Transformation</td>
<td>28</td>
</tr>
<tr>
<td>4.4</td>
<td>Knowledge of System</td>
<td>29</td>
</tr>
<tr>
<td>4.5</td>
<td>Core Ontology</td>
<td>31</td>
</tr>
<tr>
<td>4.6</td>
<td>User Profile Ontology</td>
<td>33</td>
</tr>
<tr>
<td>4.7</td>
<td>Device Ontology</td>
<td>35</td>
</tr>
<tr>
<td>4.8</td>
<td>Multimedia Content Ontology</td>
<td>37</td>
</tr>
<tr>
<td>5.1</td>
<td>The flow sequence of Adaptation Decision</td>
<td>41</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

In today’s world, people are more absorbed with media content than ever before. This holds particularly true for the Internet’s realm, where media files can be found everywhere, from some simple web pages to even large corporate portals. Multimedia on websites is what makes them all so attractive, entertaining and worth being visited repeatedly. The term multimedia refers to all types of media, such as text, graphics, animation, audio, video, as well as the combination of them. However, the composition and structure of multimedia content arises the complexity of consumption.

The technological evolution, which exists in the context of computational entities, lies in the fact that the range of devices available to access media content becomes increasingly heterogeneous and at the same time ubiquitous. The proliferation of this variety of devices such as PCs, smart phones, laptops, audio systems, PDAs makes the representation of media more challenging. The different device capabilities in terms of their computational power, memory size, display and network features result in the display requirements.

In a communication environment, this technological advancement has introduced the option of nearly ubiquitous access of multimedia information since users have more gadgets and equipment to interact in their environment. According to the pervasive computing vision, devices will seamlessly integrate into the life of users, providing them with services and information in a frame of a more sensitive approach. In this sense, the requested information should be personalized according to users’ preferences, interests, abilities and surrounding environment.

1.1 Motivation

According to the above requirements, multimedia content conversion are becoming an essential function of any service, application or system in ubiquitous environments. Thus, multimedia content adaptation has already gained a considerable importance in communications. More and more application scenarios in everyday activities depend on the processing of information in order to facilitate an efficient
CHAPTER 1. INTRODUCTION

representation of content. For instance, a man is going to his office and wants to watch a football match of the previous night from his mobile phone but unfortunately his battery is running out. Can the device receive this information? How it can display the game? Should the situation of users and their surrounding conditions should be taken into account? The challenge comes to the surface, since each combination of device, user, and delivery environment requires a different format for the content and thus a real-time adaptation is required.

The research’s questions which motivate this thesis can be formulated as follows:

- How can multimedia information be adapted in order to be accessible to users by using any device in a context of appropriate manner both anywhere and anytime?

- How efficient can the context be described and the multimedia content in order to achieve the above goal?

Two parties may be affected by changes in the existing technologies, and need to be considered as criteria to assess any context-aware content adaptation system.

Several approaches have been proposed to solve these challenges, ranging from production of multi-forms of multimedia content for specific devices and capabilities, to automatic adaptation approaches. The static adaptation perspectives are impractical in terms of development and maintenance costs and its inability to cover all terminals and user preferences. As a consequence, the inadequacy of static adaptation promoted research towards dynamic adaptation. In this field of real time conversion, various frameworks are proposed using different description mechanisms. A part of them was developed including only device capabilities and network characteristics in the modelling of the context. Another category of adaptation systems describes the general condition of delivery environment but lacks of expressiveness and adequacy.

1.2 Contribution of this thesis

The challenging of real time multimedia content adaptation, and the weakness of so far proposed perspectives, increase the motivation of developing a server-side context-aware multimedia content adaptation framework. In the context of pervasive computing, our approach achieves the target of ubiquitous access of multimedia information, personalization to the user's situation and preferences, overcoming compatibility issues. This means that in order to be able to implement the required processing, it is necessary to discover, describe and represent the capabilities of the involved devices, characteristics and preferences of the user and the conditions of the terminal environment. Moreover, fundamental part of this procedure is the mapping of multimedia content based on MPEG-7 standard (2.5.1). The mechanism of Semantic Web, specifically OWL DL, provides a vocabulary for representing knowledge and can be used to facilitate efficient context awareness. Using this
technology, four ontologies were created, in order to compose a descriptive modelling of all the parameters that take place in an adaptation procedure (namely Core Ontology, User Profile, Device Ontology, and Multimedia Content Ontology, respectively). Considering this knowledge, the adaptation decision mechanism, using multi-step decision rules, processes the multimedia information and proposes the optimal content version and the strategy for delivering it.

Our server-side architecture can achieve efficient content conversion, since it prevents the transmission of unnecessary information. The innovative characteristic of this framework is brought out by the modelling of knowledge. This mapping achieves the description of context components, not only as isolated entities but also as part of a general display situation. However, the adaptation decisions include the adjustment of terminal environment and especially the device in order to refine the representation of information. This processing power is accomplished by a multi-step decision which is able to process the captured knowledge (context and multimedia content) and construct adaptation plans. No changes are required in the general concept when new technologies are included in the communication environment and new tools become available as well. Moreover the objectives of our contextual model include mapping of a set of upper-level ontologies which provide flexible extensibility adding specific concepts in different application domains. The implementation of this framework is feasible with the current technology comparing to corresponding researches which is only in theoretical level until now.

1.3 Organization of the remaining chapters

This thesis is organized as follows: the background theory which is needed to understand the main aspects of this thesis, is discussed in Chapter 2. Related work is reviewed in the field of modelling context and content with ontologies and content adaptation systems in Chapter 3. Chapter 4 introduces the proposed Context-Aware Multimedia Adaptation Framework and describes Adaptation Decision Framework (ADF) architecture. In Chapter 5, a use case scenario is referred and the methodology of adaptation decision procedure is analysed. Finally, the future improvements and extensions of this framework are discussed in Chapter 6.
Chapter 2

Background Theory

In this chapter, we introduce the basic concepts, terms, technologies and prototypes in order to provide a better understanding of the thesis of this specific master.

2.1 Ambient Intelligence (AMI)

The vision of Ambient Intelligence was first proposed by Philips Research in 1999. The Information Society and Technology Advisory Group (ISTAG) was influenced by this initiative and it adopted this concept as one of its research focus. According to the ISTAG [19, 20], Ambient Intelligence (AmI) provides a vision of the Information Society where the emphasis is on greater user-friendliness, more efficient services support, user-empowerment, and support for human interactions. Undoubtedly, people are surrounded by both intelligent intuitive interfaces that are embedded in all kinds of objects and also an environment that is capable of recognizing and responding to the presence of different individuals in a seamless, unobtrusive and often invisible way.

2.2 Ambient Intelligent Context

A fundamental concept in the research area of Ambient Intelligence is "context". It is true that there are various definitions of the context in the bibliography. Clearly, a widely accepted meaning of this term, based on [16], includes any information that can be used to characterize the situation of an entity. Being more specific, an entity is a person, a place, or an object that is considered relevant to the interaction between a user and an application, including the user and the application themselves. The term "places" applies to geographical spaces like rooms, offices, buildings and so on. People need to be differentiated by groups or individuals. Things refer to physical objects or software components. To describe these entities, four categories are introduced:

- Identity: characterizes the entity with an explicit identifier, which has to be unique in the name domain of the application.
• Location: includes positioning data and orientation as well as information about regional relations to other entities (e.g. neighbouring entities). This comprises geographical data as well as spatial relations.

• Status: contains properties, which can be perceived by a user. For a place, this can be, for example, the current temperature, the ambient illumination or the noise level. For persons this refers to physical factors like vital signs, tiredness or the current occupation.

• Time: is both date and time

The context information, which exhibits a number of characteristics in intelligent environment, can be described and formed accordingly to a specific domain and its needs. Also, it can be any kind of information that is gathered and analysed to characterize or provide additional information regarding any feature or condition of specific interactive smart space. In the Ambience Intelligent environments, context provides information about the status of the people, activities, location, physical environment and computing entities.

2.3 Context-aware applications

Smart spaces use contextual information to become context-aware of the situations which are relevant to the intelligent interactions with users [17]. Dynamic context imposes context change awareness. More specifically, context-aware applications are those having the ability to detect, interpret and react to aspects of user’s preferences and environment characteristics, device capabilities or network conditions by dynamically changing or adapting their behaviour based on those aspects that describe the context of the application and the user.

2.4 Multimedia Content

Multimedia represents various types of media content, used together. More specifically multimedia content includes a combination of text, audio, still images, animation, video, or interactivity content forms. An incommensurable amount of audiovisual information is becoming available in digital form, in digital archives, on the World Wide Web, in broadcast data streams and in personal and professional databases, and this amount is still growing. Multimedia is usually recorded and played, displayed or accessed by information content processing devices, such as computerized and electronic devices, but can also be part of a live performance.

2.5 Supporting standard technologies and prototypes

In the following, we describe technologies and prototypes which are necessary for achieving the outlined functionalities in this thesis.
2.5. SUPPORTING STANDARD TECHNOLOGIES AND PROTOTYPES

2.5.1 MPEG-7 Standard

MPEG-7 [29], formally known as the Multimedia Content Description Interface, provides a standardized scheme for content-based metadata, termed descriptions by the standard. A broad spectrum of multimedia applications and requirements are addressed and as a consequence the standard describes both low- and high-level features for all types of multimedia content. The tree core elements of the standard are:

- Description tools consist of:
  - Descriptors (Ds) represent features, attributes, or groups of attributes of multimedia content. This means that they describe low-level audio or visual features such as color, texture, motion, audio energy, and so forth, as well as attributes of audio-visual contents such as location, time, quality, and so forth. Thus, Ds define the syntax and semantics of each feature and can be extracted automatically in applications.
  - Description Schemes (DSs) specify the structure and semantics of the relationships between their components, that may be both Descriptors and Description Schemes. They describe higher-level audio-visual features such as regions, segments, objects, etc. In some cases, automatic tools can be used for instantiating the DSs, but in many cases, instantiating DSs requires human-assisted extraction or authoring tools.

- Description Definition Language (DDL), based on XML Scheme Language [21], defines in XML the syntax of the description tools and enables the extension and modification of existing DSs and also the creation of new DSs and Ds.

- System tools which support two possible forms: (1) textual XML form which is suitable for editing, searching, filtering, and browsing and (2) a binary form that is suitable for storage, transmission, and streaming.

The specification of the MPEG-7 standard is divided into several parts, which correspond to the different groups by working on it. The basic part of the standard, since it specifies the bulk of the description tools, is the Multimedia DSs (MDS). They can be considered as a library of description tools and, in practice, an application should select an appropriate subset of DSs. Figure 2.1 represents the organization of MDS into different functional areas.

- The Basic Elements are the generic entities that are used as building blocks by different Description Tools. They include basic datatypes (e.g., numbers, matrices, vectors, country), links and locators (time, media locators and referencing tools), and other basic Description Tools for places, persons, textual annotations, controlled vocabularies, etc.
The **Schema Tools** are comprised by the tools (root and top-level elements) for wrapping Description Tools which are used by applications and the package tools for grouping related Description Tools into folders in order to personalize applications.

The **Content Description** Tools are the tools for the representation of perceivable information, comprising structural aspects (structure Description Tools) and conceptual aspects (semantic Description Tools). The Structure Description Tools allow the description of the content in terms of spatio-temporal segments which are organized in a hierarchical structure (allowing the definition of a table of contents or an index). Audio, visual, annotation and content management Description Tools can be attached to the segments to describe them in detail. The Semantic Description Tools allow the description of the content from the viewpoint of real-world semantics and conceptual notions: objects, events, abstract concepts and relationships. The semantic and structure Description Tools can be further related by a set of links.

The **Content Management** Tools are the tools that are related to classical archival of content, providing information about media features, creation and usage of the multimedia content. The Media Description Tools allow
the description of the storage media, the coding format, the quality and the transcoding hints for adapting the content to different networks and terminals. The Creation Description Tools allow the description of the creation process (title, agents, materials, places, dates, ...), classification (genre, subject, parental rating, languages, ...) and related materials. The Usage Description Tools allow the description of the conditions for use (rights, availability, ...) and the history of use (financial results, audience, ...).

• The **Content Organization Description** Tools allow the creation and modelling of collections of multimedia content and descriptions. Each collection can be described as a whole by their attribute values characterized by models and statistics.

• The **Navigation and Access Description** Tools allow the specification of summaries, partitions, decompositions, and variations of the multimedia content for facilitating browsing and retrieval. Summaries Description Tools provide both hierarchical and sequential navigation modes in order to provide efficient preview access to the multimedia material. Partitions and Decompositions Description Tools provide the tools for multi-resolution and progressive access in time, space and frequency. Variations Description Tools provide the tools for describing pre-existing views of the multimedia content: summaries, media modalities (e.g., image to text), scaled versions and so forth.

• The **User Interaction Description** Tools allow the description of user preferences (for personalized filtering) and usage history pertaining to the consumption of the multimedia content. These Description Tools have been also adopted within MPEG-21 DIA.

### 2.5.2 Semantic Web

The World Wide Web, commonly known as the Web, is a system of interlinked hypertext documents accessed via the Internet. With a web browser, one can view web pages that may contain text, images, videos, and other multimedia, and generally navigate between them via hyper-links. The Web has changed the way people communicate with each other, information is disseminated and retrieved, and business is conducted.

An alternative approach is to represent Web content in a form that is more easily machined processable and to use intelligent techniques to take advantage of these representations. We refer to this plan of revolutionizing the Web as the Semantic Web initiative [8] which provides richer and explicit descriptions of Web resources. The Semantic Web(SW) is propagated by the World Wide Web Consortium (W3C), an international standardization body for the Web and is actually the extension of the World Wide Web. The aim of the Semantic Web according to [2] is to allow much more advanced knowledge management systems:
• Knowledge will be organised in conceptual spaces according to its meaning.

• Automated tools will support maintenance by checking for inconsistencies and extracting new knowledge.

• Keyword-based search will be replaced by query answering: requested knowledge will be retrieved, extracted, and presented in a human friendly way.

• Query answering over several documents will be supported.

• Definition of views on certain parts of information (even parts of documents) will be possible.

2.5.3 Ontologies in Semantic Web

On the Semantic Web, ontology defines the concepts and relationships which are used to describe and represent an area of concern. A frequently cited, and perhaps the most prevalent definition of ontology is attributed by Tom Gruber [26], as "an explicit specification of conceptualization". Ontologies are used by people, databases, and applications that need to share domain information. A domain is just a specific subject area or area of knowledge, like medicine, tool manufacturing, real estate, automobile repair, financial management, etc. Typically, ontology consists of a finite list of terms, and relationships between these terms. The terms denote important concepts (classes of objects) of the domain. The Semantic Web relies heavily on formal ontologies to structure data for comprehensive and transportable machine understanding. In general, ontologies encode knowledge in a domain and also knowledge that spans domains. In this way, they make that knowledge reusable. At present, the most important ontology languages for the Web are as follows:

• The Extension Markup Language (XML) [11] provides a surface syntax for structured documents, but imposes no semantic constraints on the meaning of these documents. It is particularly suitable for sending documents across the Web. XML Schema is a language for restricting the structure of XML documents.

• The Resource Description Framework (RDF) [14] is a basic data model, like the entity-relationship model, for writing simple statements about Web objects (resources). These data models can be represented in an XML syntax. RDF Schema provides modelling primitives for organizing Web objects into hierarchies. Key primitives are classes and properties, with subclass and subproperty relationships, domain and range restrictions. RDF Schema is based on RDF.

• The Web Ontology Language (OWL) [35] is a richer vocabulary description language for describing properties and classes among others, relations between classes (e.g. disjointness), cardinality (e.g. exactly one), equality,
2.5. SUPPORTING STANDARD TECHNOLOGIES AND PROTOTYPES

richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes. OWL builds on RDF and RDF Schema, and uses RDF’s XML syntax. The above benefits lead us to the selection of this language for the description of context models.

2.5.4 Web Ontology Language (OWL)

The Web Ontology Language is designed to be used by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema because it provides additional vocabulary along with a formal semantics. OWL has three increasingly-expressive sub-languages: OWL Lite, OWL DL, and OWL Full.

- OWL Lite supports those users who primarily need a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than its more expressive relatives, and OWL Lite provides a quick migration path for thesauri and other taxonomies. Owl Lite also has a lower formal complexity than OWL DL.

- OWL DL supports those users who want the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time). OWL DL includes all OWL language constructs, but they can only be used under certain restrictions. For example, while a class may be a subclass of many classes, a class cannot be an instance of another class. OWL DL is so named due to its correspondence with description logics, a field of research that has studied the logics that form the formal foundation of OWL.

- OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.

Each of these sublanguages is an extension of its simpler predecessor, both in what can be legally expressed and in what can be validly concluded. Although, the following set of relations holds, their inverses do not.

- Every legal OWL Lite ontology is a legal OWL DL ontology.
- Every legal OWL DL ontology is a legal OWL Full ontology.
- Every valid OWL Lite conclusion is a valid OWL DL conclusion.
• Every valid OWL DL conclusion is a valid OWL Full conclusion

OWL distinguishes three main categories of entities:

• Individuals represent entities or phenomena in a domain of interest (e.g. Room-160).

• Classes are groups of individuals with similar characteristics (e.g. Cars, Humans, Animals). Classes can be also organized into a superclass/subclass hierarchy, known as taxonomy. If an individual is a member of a class, it is member of all its superclasses.

• Properties represent binary relationships between two individuals or between an individual and a data value. Properties can be defined independently from any individual. Owl distinguishes two categories of properties:
  
  – Object Properties link individuals to individuals. We can state that this type of property can only relate members of class D (domain) to members of class R (range). Also, it is possible to further constrain the range of a property in specific contexts in a variety of property restrictions (allValuesFrom, someValuesFrom, Cardinality, etc.).

  – Datatype Properties, which link individuals to data values.

OWL is a component of the Semantic Web activity. This effort aims to make Web resources more readily accessible to automated processes by adding information about the resources that describe or even provide Web content. As the Semantic Web is inherently distributed, OWL must allow information to be gathered from distributed sources. This is partly done by allowing ontologies to be related, including explicitly importing information from other ontologies.

2.5.5 Reasoning

Semantics is a prerequisite for reasoning support. Reasoning is important because it allows one to check the consistency of the ontology and generate additional knowledge. Also it inspects for unintended relationships between classes and it automatically classifies instances in them. The implementation of reasoning can be achieved with the help of a rule-based system. A rule-based system is a computer declarative program that uses rules to reach conclusions from a set of premises.

There are two main classes of rules: forward-chaining and backward-chaining rules. Forward-chaining rules are somewhat like if...then statements in a procedural language, and they are the most common and important kind of rule. Backward-chaining rules, on the other hand, do not have a clear analogy in procedural programming. They are also similar to if...then statements, but a backward-chaining rule actively tries to satisfy the conditions of its if-part.
Chapter 3

Literature Review

3.1 Ontologies for Context Modelling

Ontology is a vehicle to enable interoperability from the semantic point of view. In the following paragraphs on this section, we refer in ontologies that gather, model, store, distribute and monitor context. The definition of context includes any information that describes physical objects, applications and users in any domain. Specifically we focus on general context knowledge about the physical world, users’ characteristics, devices and the multimedia information consumed and exchanged in pervasive computing environments.

Ontologies in Context-Aware System

An ontology-based approach for developing a context-aware system is an efficient and suitable choice based on \[41\] for the following reasons:

- Ontologies with declarative semantics provide multiple policies to support context inference.
- A well-defined ontology enables the sharing and reuse of knowledge.
- Ontologies provide variable complex efficient inference mechanisms to deduce high-level contexts from low-level, raw context data, check inconsistent contextual information and also reason about knowledge.
- Represented ontologies enhance the development of context-aware systems with semantic web technologies.

The following elements are typically found in ontology-based systems \[40\]:

- Ontologies describe the concepts (classes), the entities (instances) and the relationships (properties) between these concepts within a domain of discourse.
Knowledge bases contain the actual information or knowledge. This information is structured according to the ontology language that is used and the ontologies that have been defined by this language. This knowledge can be seen as a collection of instances of the classes defined by the ontologies.

Reasoners, which are used to reason on the information contained in the knowledge base, use the definitions provided in the ontology. These reasoning capabilities are normally based on assertions. Two examples of these capabilities are the classification of information and the inference of additional relationships between concepts.

Applications, which query the knowledge base for information (possibly through reasoners), use the reasoning capabilities of the reasoners to implement their own service logic.

3.1.1 General context ontology

CONtext ONtology (CONON)

CONON [47] defines general concepts such as location, activity, person and computational entity, whose terms are thought to be extensible in a hierarchical way by adding domain specific concepts. The authors divide their context model into an upper ontology and a specific ontology (see 3.1). On the one hand, the upper ontology is a high-level ontology that captures general features of basic contextual entities. On the other hand, the specific ontology defines the details of the general concepts and their features in each subdomain covered.

This modelling of context achieves to describe generally a pervasive computing environment. Moreover, the flexibility of the upper ontology provides itself the capability to be extended and to add specific concepts in different application domains. It also makes this description comprehensive and powerful in future expanding. Therefore, CONON’s model is unable to satisfy the description of physical environment and general conditions. This is a major drawback of their description. On the contrary, our approach facilitates the complete description of the environment, conditions and the changes that happened. Our modelling and description of the context is influenced by these basic contextual entities. We adopt these basic concepts and analyse further the notion of activity to a simple and complex event.

Standard Ontology for Ubiquitous and Pervasive Applications) (SOUPA)

SOUPA [12] is a sharing ontology that supports and models knowledge in pervasive computing applications. The ontology is expressed by using the OWL and it is divided into two main blocks called SOUPA-Core and SOUPA-Extensions (see Figure 3.2). SOUPA-Core defines concepts that should appear in a lot of scenarios (e.g., person, agent, policy, time, space), while SOUPA-Extensions supports particular concepts in narrower domains (e.g., home, office, entertainment). The goal
of this design is to reduce the efforts of other systems in creating ontologies and to be more focus on the actual system implementation. Particularly, SOUPA can be extended and used to support the applications of CoBrA. The general idea behind this modelling is to focus only on entities and how these entities can be analysed. In contrast to our approach, the general representation of the smart space and the events taken place are out of the scope.

**Context Broker Architecture (CoBrA)**

CoBrA [30] is a broker-centric agent architecture, that supports context-aware systems in smart spaces. More detailed, the role of this agent is to maintain a model of the present context and to share this model of context knowledge with other agents, services and devices. The key components of a Context Broker is CoBrA Ontology which is a set of ontologies for agents to describe contextual information and share context knowledge. These ontologies are categorized to action, agent, device, time, space, etc. CoBrA has been used to prototype a smart meeting system. This architecture does not have any general or upper level ontology model. In contrast to our approach, this description lacks extensibility. Moreover, the description of context is limited to the categories referred above and also there isn’t connection between the informations which is collected by every ontology separately. Less emphasis is put on the notion of user interfaces and mobile devices. To summarize, CoBrA models ubiquitous domain knowledge rather than context information. However, our approach is feasible for both.
Figure 3.2: SOUPA consists of two sets of ontology documents: SOUPA Core and SOUPA Extension.

**Ontology-Based Context-Aware Middleware for Smart Spaces**

This middleware [40] forms context knowledge with the combination of first-order probabilistic logic and explicitly represented ontologies. The contextual ontology is divided into a core context ontology for general conception entities in the smart space and an extended ontology for the domain specific environments. According to the description of core ontology, the instances of the smart space are consisted of User, Location, Activity, Service, Environment, Platform and Time. These entities and the correlations between them are enabled to describe a high-level abstraction of contextual information and give an expressive representation of the condition of smart space. The extended context ontology extends the core ontology and defines additional information of various domains. This ontology-based context model has common entities with CONtext ONtology (CONON) and includes additional concepts like Environment and Time. In contrast to CONON, this modelling of context focuses on user and his activities. Our approach tries to correlate the benefits of context ontologies and monitor user, his intentions and the the conditions of the surrounding environment.
3.1. ONTOLOGIES FOR CONTEXT MODELLING

Context-Aware Ontology for Multimedia applications (MULTICAO)

The MULTICAO was designed to provide the support for context-aware and Digital RIGHTS Management (DRM) enabled multimedia content adaptation decision making operation. The part of modelling contextual information [6] is divided into a core context ontology layer, which describes the common knowledge denominator and an extended context ontology layer which is domain-specific and only relevant for the application under consideration (see Figure 3.3). The core ontology defines the following basic concepts: user, terminal, network and natural environment. This layer is based on MPEG-21 [28] Digital Item Adaptation (DIA), in particular the Usage Environment Description tool. The basic drawback of MULTICAO ontological model is the unrefined forming of context information and as a result the narrow description of real-world situations.

![MULTICAO ontological model](image)

Figure 3.3: MULTICAO ontological model.

It should be mentioned that the approach in [3] follows the same modelling on core ontology based on MPEG21 standard.

3.1.2 Device ontology

In this subsection we mention specific ontologies which focus on diverse terminals (devices) as a part of a pervasive computing environment and we describe their characteristics and capabilities.

An Ontology Framework for Semantic Description of Devices

In this work [4], an ontology has been proposed with the aim of providing a formal framework to describe devices and their services to support effective service discovery. The characteristics and capabilities of a device play an important role of the description and behaviour of the service hosted in. The information related to a device is divided into six entities depending of the type of information provided, namely: Device, Device Description, Hardware Description, Software Description, Device Status and Services. This Device ontology (see Figure 3.4) is intended to
provide a general framework to describe any type of device and can be extended with the construction of an hierarchy of sub-classes. This modelling inspired us and we adopt the concept of separating the characteristics of a device to a number of classes which describe each technological part of a device. Moreover, we extend the mapping of capabilities with additional classes and include more details.

**FIPA Device Ontology Specification**

The Foundation for Intelligent Physical Agents (FIPA)\(^7\) is aimed at producing standards for the interoperation of heterogeneous software agents. Its Technical Committee, which was responsible for the agent communications, began the work of ontology service for dealing with the problem of how agents can share a common ontology in order to make a meaningful conversation. The Device Ontology \(^39\) specifies a frame-based structure to describe devices and it is intended to facilitate agent communication for purposes such as content adaptation. The device ontology analyses the basic input and output capabilities of a device and does not refer to other features. Thus, this modelling neither facilitates an effective description of
3.1. ONTOLOGIES FOR CONTEXT MODELLING

Device and Service Description for Ontology-based Ubiquitous Multimedia Services

In the context of the Ubiquitous Service Core Research Programme of Mobile VCE, that provides intelligent multimedia processing, has developed an approach for modelling device and service descriptions, constructed in Web Ontology Language. This ontology model is based on the resource discovery protocol UPnP. The description of terminal domain is divided into Terminal, Terminal Parameters, Terminal Capabilities, which are analysed further to other corresponding sub-classes. The first class, Terminal, refers to each device that can receive and process multimedia content. The entity Terminal Capabilities is analysed to hardware components and software aspects (services hosted). The latter represents characteristics of a device that can be directly compared with the properties of multimedia content for a much better display. Generally, this ontology does not facilitate an effective modelling of terminals since the description of device characteristics is not organized well and lacks important information for the purpose of this ontology.

3.1.3 Multimedia Content Ontology

The amount of digital multimedia information, which is accessible to the end-users, is growing every day. Semantic descriptions of available media can be used to facilitate retrieval, presentation and exploitation of media and documents that contain them. In order to accomplish the above goal, ontologies are introduced in the field of multimedia. As a result, machine-understandable descriptions of the multimedia content based on MPEG-7 standard are constructed. In this subsection we present several multimedia ontologies created according to MPEG-7 prototype.

In contrast to our approach, the following ontologies are developed for facilitating the effective retrieval and recognition of multimedia information. We focus on the improvement of presentation of multimedia content based on the context-aware. In order to accomplish the above goal, we analyse the multimedia content and their features for exploiting and adapting the content. The influence of these ontologies and the adopted elements actually implemented for our purpose. There are some common ideas of the following modelings but our approach includes the path of transformation of information, in an attempt to capture this procedure.

Hunter's MPEG-7 ontology

The MPEG-7 ontology proposed by Hunter within the Harmony project constitutes chronologically the first attempt to model parts of MPEG-7 in RDFS. This multimedia ontology was translated into OWL, extended and harmonized using the ABC
upper ontology. The current version is an OWL Full ontology containing classes which define the media types (Image, Multimedia, Audio, AudioVisual, Video) and the decompositions (temporal, spatial) from the MPEG-7 Multimedia Description Scheme (MDS) (see Figure 3.5). Furthermore a set of descriptors representing information about the production, creation, usage and the media features are also included. This ontology has usually been applied to describe the decomposition of images and their visual descriptors. The basic drawback of this modelling is that the defined semantic entities which are allowed have more than one semantic interpretation, while different entities share the same meaning. According to our perspective, each entity is defined without ambiguities and spatial or temporal decomposition properties promote clarity.

Figure 3.5: Class hierarchy of the multimedia content and segment classes in the Harmony MPEG-7 based ontology

The aceMedia MPEG-7 based ontology

Two RDFS ontologies [38], namely the Multimedia Structure Ontology (MSO) and the Visual Description Ontology (VDO) have been developed within the aceMedia project. MSO covers the complete set of structural description tools from MDS, while VDO addresses the Visual part and provides the formalisation of low-level descriptions. Although this modelling follows the same engineering principles with
Hunter's ontology resulting in an analogous class hierarchies, new classes are introduced. Moreover this approach addresses the structural and low level features definitions in two separate ontologies. Although these particular groups of ontologies enhance Hunter's modelling, the problems about the conceptual clarity continue to exist and there is focus only on the visual part.

**Boemie Ontology**

In an attempt to capture the MPEG-7 structural descriptions, as well as, the Visual and Audio parts in a more declarative way, two OWL DL ontologies [15] have been developed within the context of the BOEMIE project, namely the Multimedia Content Ontology (MCO) and the Multimedia Descriptors Ontology (MDO). As the respective names denote, the first ontology examines the structure of multimedia documents, including the representation of the types of multimedia content and their decomposition schemes, whereas the second ontology addresses the descriptors that are used to describe the various low-level features. Not only distinct classes have been introduced to model the different content and segment entities, but disjoint axioms explicitly model that they form non intersecting sets. This perspective influences us and we adapt the way that classes represent the multimedia content and its spatial/time segmentation. In contrast to our modelling, linking with domain specific ontologies is implemented through a pair of generic properties that captures the relation between a content/segment instance and the depicted semantics, and additionally the relation between a content/segment instance and its extracted low-level features.

**SmartWeb Ontology**

In the context of the SmartWeb project [46], a set of ontologies, relevant for query-answering and information services on the Web, has been developed. Among them, an MPEG-7 based ontology [42] supports the annotation of multimedia content. The developed description includes the structural, localisation, media and low-level description tools of MPEG-7. Two classes hierarchies are modelling, namely the Segment and the Multimedia Content. This ontology (see Figure 3.6) does not treat Segment classes as specialisations of the Multimedia Content class. Furthermore the decomposition schemes are also modelled also as classes, each denoting a valid decomposition pattern per content/segment type and a spatial/temporal dimension. This approach may be closer to the original MPEG-7 Schemas but introduces peculiarities and semantic ambiguities, especially in the case of recursive content decomposition.

### 3.2 Context-Aware Content Adaptation Systems

Content adaptation has already gained a considerable importance in multimedia communications and become an essential functionality of any service, application or
Figure 3.6: Class hierarchy of the multimedia content and segment classes in the SmartWeb MPEG-7 based ontology

system. From a systems’ perspective, delivering content to heterogeneous clients/terminals through diverse networks and delivery environments, requires to process context and content metadata and subsequently perform content adaptation operations. The purpose of this procedure is to convert the information to a suitable version based on several parameters and requirements. The research spectrum in content adaptation is remarkable. The categorization of each survey and the corresponding criteria may be controversial, since each framework has been designed with different goals and approaches. In this section, we will describe researches regarding techniques, procedures and approaches that focus on context-aware multimedia content adaptation and we will perform a briefly comparison with our perspective. It should be mentioned that except the first two systems, the approaches, which are refereed in the rest of section, use ontology technology.

InfoPyramid

[37] is one of the early adaptation systems that used a representation scheme called the InfoPyramid to represent hierarchy of multimedia data. The system stores multi version content and will be selected to suit best the targeted device. In contrast to our approach, this static adaptive adaptation system requires pre-processing task and bigger storing allocation.
Digital Item Adaptation

The Digital Item Adaptation (DIA) standard [45] was introduced by the ISO MPEG group as part of the MPEG-21 framework [28]. DIA aims to assist multimedia content adaptation to different storage, transition and consumption environments. In order to fulfill this goal, DIA provides standard metadata enabling in that way the description of terminals, network, users and consumption environments characteristics. It is important to emphasize that this standard only provides description tools, using the tools for adaptation is left to implementers. Examples of researches that are based on DIA and extend the standard are presented in the following paragraphs. MPEG-21 is based on XML which is efficient for syntax definition but limited in semantic definition.

MULTICAO: A Semantic Approach to Context-Aware Adaptation Decision taking

In the context of the European project CISNET-II, an Adaptation Decision Engine (ADE) [5] is developed as part of a context-aware content adaptation platform for Virtual Collaboration. The ADE is responsible for deciding how to react and allow the consumption of multimedia content under changes in the usage environment, in a way that best meets users’ expectations. This architecture is designed based on a multi-module service oriented approach. A Context Service Manager gathers low-level context information processed and saved in the Context Knowledge Base (KB) for knowledge sharing and reuse. In the next level, the Ontology Service Manager integrates the contextual knowledge with the concepts and infers a high-level context using rules appropriate to specific application scenarios. Specifically, part of this operation is two-layer context-aware ontology MULTICAO, described in previous section and enables the formal characteristics of generic context as well as the extension to specific applications scenarios. The modelling of this ontology is based on the form of MPEG-21 DIA, since the received contextual information is formatted in this standard (XML format). Finally the Adaptation Decision Engine Manger is responsible for the selection the most appropriate adaptation and corresponding service parameters. Whenever new or updated contextual information is available, the Rules Interface Engine is invoked and interacts with the Decision Taking module. Not only this ontology model of contextual knowledge lacks expressiveness but also this architecture affects the decisions taken since the corresponding parameters are limited. In our approach, the modelling of context is more descriptive and includes more details for better perspective of the adaptation.

Context-Aware Service Adaptation Management

An Adaptation Management Framework (AMF) defined as part of Mobile VCE Core 4 Removing the Barriers to Ubiquitous Services Project. The AMF [3] has two main components, the Adaptation Manager (AM) that manages adaptation and invokes other framework components and the Content Adaptor (CA) that discovers
adaptation services and collects their profiles. The CA activates the AM. This architecture analyse the delivery context which includes device characteristics, user preferences, natural environment characteristic and content description, in order to enable efficient context-aware service adaptation. A service is multiple content items (e.g. hotel booking, tourist guide) that in the most cases would be the content it delivered. The AM determines processing stages that define the procedure and the parameters checked until taking an adaptation decision. In contrary to our architecture, the adaptation decision is included only in the transforming of the content and does not affect the context. It is worth to be mentioned that the collection of contextual information and the mapping to OWL ontology can be done with the automatic conversion of MPEG-21 DIA using XSLT transformation \[13\]. This means that the contextual information is gathered only from this standard and its corresponding description. Our approach uses sensors and other techniques which provide a fully description of low-level of context.

**Context-aware Adaptation of Mobile Multimedia Presentation**

This middleware architecture \[18\] performs an automatic and distributed service adaptation process. Specifically, the adaptation of the presentation of multimedia content distributed among the server and the client side part of the middleware located between the applications and services. The presentation content is adapted according to device characteristics, user preferences and to the current context that refers to dynamic parameters, especially the device status. The description of the environment conditions and events happened are out of the scope. While the server provides a preprocessing of the content based on information sent by client, the client completes the overall adaptation. In contrast to our architecture, the collection and the processing of context information about device and users are performed in the client side. At that point a reasoner creates guidelines about the adaptation process and these decisions are sent to server that transforms the content. An XML-based Adaptation Decision Language (ARL), developed in this work, specifies the situations and rules that support the adaptation decision. This language lacks expressiveness, thus it does not achieve to efficient adaptation.
Chapter 4

Context-Aware Multimedia Content Adaptation Framework

Considering the dramatic growth of digital media reproduction, users continuously consume and exchange multimedia information. End user devices are becoming more and more diverse, and users with different preferences access the multimedia contents through heterogeneous networks. Moreover, all these different usage environments, where conditions are continuously changing due to occurring events, arise the need for multiple versions of the same content, providing a real time content adaptation.

This thesis proposes a context-aware multimedia content adaptation framework that is integrated and collaborates with a pervasive computing environment, in order to implement the required adaptation in the multimedia information consumed and also in the delivery environment. The Figure 4.1 illustrates a generalized view of what will be discussed further in the following sections. We will analyse the specification of the ambient intelligent environment of which our system is part and we will focus on the architecture and the components of our Adaptation Framework.

4.1 Pervasive Computing Environment

Ubiquitous computing requires the deployment of a wide variety of smart devices throughout our working and living spaces. Computational power is available everywhere. Computing systems are intended to react to their environment and coordinate with each other and with network services. The overall goal is to provide users with universal and immediate access to the required format of digital information and to support them in their tasks transparently, satisfying their preferences and abilities.

From a systems’ viewpoint, the pervasive computing space presents the unique challenge of a large and highly dynamic distributed computing environment in which they can be integrated in order to expand the existing functionality. Ambient Intelligent systems need to represent information about the environment and
recognize relevant situations to perform appropriate actions according to the purpose of their development.

4.1.1 Middleware

In such environments, the different interconnected computing systems aim to improve human quality of life without explicit awareness of the underlying communications and computing services. The heterogeneous systems, that are built using diverse technologies from diverse research fields, should be organized and coordinated from a central infrastructure. According to [24], "a Middleware is a set of programming libraries and programs (services) that constitute an indivisible platform which offers a comprehensible abstraction over the complexities and potential heterogeneity of the target problem domain."

4.2 Adaptation Decision Framework (ADF)

Delivering multimedia content to heterogeneous clients through diverse networks and environments required the provision of structural support to process context and content related metadata and subsequently perform content adaptation procedure. In the context of the vision of ambient intelligent, our approach achieves the target of ubiquitous access of multimedia content, personalization to the user situation and preferences, overcoming compatibility issues. Not only the content is transformed according to the above requirements but also environment and especially devices are adjusted in order to refine the representation of information.

Adaptation Decision Framework takes over all the procedure of collecting and modelling contextual and content information, deciding about the conversion of
original multimedia content and finally sending the adapted information in the appropriate terminal. In contrast to other approaches that adaptation process is divided in server-client part, our server-side perspective has more processing power and can achieve content adaptation more efficiently, since processing prevents the transmission of unnecessary information. It should be mentioned that part of the adaptation concept is completed in the consumption terminal(s), since in some cases there is the necessity for changing the settings of device in order to ameliorate the conditions for a more efficient display. However, ADF takes the required decisions. It is the central coordinator of procedure and is located to a processing server in the network (Proxy Server). The key position of framework enables transparent access to both the multimedia information requested and the terminal environments.

4.3 ADF Architecture

The architecture of our framework is capable of analysing the delivery context and multimedia content requested, determining multi-step processing of these inputs, and computing and executing the adaptation decision if it is necessary. The organization of the overall system architecture is illustrated in Figure 4.2.

4.3.1 Context-Content Provider

The Context Provider (CP) is the gate of this framework, providing the required information for the adaptation decision procedure. The role of the CP is to gather
contextual and multimedia content information and form it to a suitable format that is the mapping to OWL ontologies (KB). Specifically, the delivery context includes device characteristics and status, available user peripherals, network properties, user situation and preferences, and delivery environment. The aim is the efficient context-content representation and the inference for accurate adaptation decisions. The processing of inputs data can be categorized in two parts depending on the kind of data.

The first part refers to the contextual information that is collected and refreshed continuously in an ambient intelligent environment through middleware. This platform, as a collection of services, is able to monitor and control the overall conditions of such environment and also focuses on each part of it separately, in order to facilitate the derivation of the contextual description from distributed sources. It should be mentioned that if each device is considered as an isolated entity, there are various industry-standard discovery protocols such as Bluetooth, UPnP [36], UAProf (WAP) [22] that are description models for devices (hardware aspects) and hosted services (software aspects). However, they do not accomplish the most efficient description.

![Figure 4.3: XSLT Transformation](image)

Figure 4.3: XSLT Transformation

The other kind of the information captured by CP describes the multimedia delivery content and the way that its metadata are extracted. Specifically, the modeling of the multimedia content using OWL ontology is based on the Multimedia Description Standard MPEG-7. This audiovisual description standard provides the mechanisms which describe information about the content and about the information contained within the content. MPEG-7 uses XML as the language of choice for the textual representation of content description. The format and vocabulary mapping to OWL ontology can be accomplished using XML Stylesheet Language Transformation (XSLT) [13] by applying corresponding rules into transformation file. The conversion has to examine the elements and the attributes of the XML schema, evaluate the semantics, and translate them into OWL constructs. The process is depicted in Figure 4.3. An interesting point that has to be mentioned is that the low level audio-visual features can be extracted automatically in applications.
4.3. ADF ARCHITECTURE

4.3.2 Knowledge Base (KB)

A fundamental part of this framework is the modelling of the input data which include device capabilities and status, available user peripherals, network properties and state, user situation and preferences, natural environment characteristics and content description. The mechanism of Semantic Web, specifically OWL DL, provides a vocabulary for representing knowledge and can be used to facilitate efficient context awareness [9]. In other words, the representation of context information with declarative semantics aims to deduce high-level contexts from low-level, raw context data.

In order to compose a descriptive modelling of all the parameters that take place in a efficient adaptation procedure, four ontologies are created, namely Core Ontology, User Profile, Device Ontology, and Multimedia Content Ontology. In the following sections, we describe further these ontologies. We analyse their features and report the concept of their design. We model each different category of contextual information to separate ontology; in order to organize and define more intelligent the captured knowledge. However, these ontologies are coordinated and connected with each other since they are imported to a unique general ontology. The KB includes all the required knowledge that is illustrated with a descriptive sketch in Figure 4.4 providing context awareness to framework. The objectives of our context model are comprised of modelling a set of upper-level ontologies and as a consequence this modelling provides flexible extensibility by adding specific concepts in different application domains.

![Figure 4.4: Knowledge of System](image)

4.3.3 Adaptation Decision Engine (ADE)

This component of ADF is responsible for the processing of the requested multimedia content in order to facilitate efficient presentation in the delivery environment. The Adaptation Decision Engine decides the optimal content version and the best strategy for delivering that version. In other words, it defines the type of adaptation
to be performed on the multimedia content and selects the appropriate parameters, taking into account the entire context knowledge that is captured in Knowledge Base. The mechanisms that are hidden behind this procedure are a set of rules and a reasoning engine [10] that evaluate these rules. Specifically, Java Expert System Shell (Jess) is used for reasoning. Jess is a Java framework for editing and applying rules and is considered the most suitable choice between reasoners.

In the context of adaptation decision, the rules, which were created, represent the logic of the adaptation engine. These rules manage the adaptation decision steps and are capable of extending the expressiveness through the definition of more complex relationships between individuals. Notably, ADE controls the adaptation and also the delivery procedure, deciding how the system should react when changes occur in the context of usage. Whenever new or updated contextual information is available, the ADE is invoked in order to evaluate the conditions interacting with the KB to make the necessary interference. We analyse further the reasoning of adaptation decision in Chapter 5.

4.3.4 Content Adaptor (CA)

The CA executes the adaptation decisions defined by ADE. In particular, this part of framework implements the process of converting the multimedia content from the original format into a format which can be consumed under all the premises analysed above. As no single software piece exists that can handle the huge variety of multimedia adaptation method, an intelligent adaptation mechanism has to be extensible in order to integrate the already existent or the new adaptation tools such as FFmpeg. It should be mentioned that our research does not focus on investigation transforming techniques or the actual adaptation operation. Our perspective aims to define the appropriate transformation of multimedia information according to the captured knowledge for terminal environment. However, it can be robust and extensible with respect to additions of new technologies.

4.4 Ontology-Based Context Modelling

Context-awareness is one of the drivers of ubiquitous computing environment and our framework aims to provide a fully context-aware dynamic adaptation and personalization according to the delivery context. A well designed model is a mechanism to understand better the user's situation and environment condition, and hence optimizes the quality of the adaptation decisions. Ontology has been introduced as a means to provide declarative formal representation of the domain knowledge. This section focuses on modelling and representation of delivery context and multimedia content using Web Ontology Language (OWL). Particularly, this set of ontologies composes the Knowledge Base (KB) of Adaptation Decision Framework.

\[1\text{http://www.ffmpeg.org/}\]
4.4. ONTOLOGY-BASED CONTEXT MODELLING

4.4.1 Core Ontology

There are number of factors coming together in a simple content delivery scenario. Such delivery management requires not only constant awareness of the user's surrounding context, in particular the awareness about the locally available devices and their capabilities, but it also needs to reflect any context change in the way the multimedia content is delivered. The Core Ontology accomplishes these goals and describes not only the specific components of a delivery environment but also the general condition and the various events that may occur, in order to understand the user's situation better. The schema of this general context mapping is depicted in the following Figure 4.5. It should be mentioned that we are influenced by CONtext ONtology (CONON) and its basic contextual entities. We adopt some basic concepts and analyse further the notion of activity to simple and complex event.

![Figure 4.5: Core Ontology](image)

**Classes**

- **Person:** this class represents a vital part of general context description, since a human plays a central role in ambient intelligent environment. The profile and contact information, user's preferences and so forth are further analysed in the ontology of User Profile.

- **Location:** this class provides the location of an entity in the space. It is divided into two subclasses.
  - **Room Location:** this class represents a room in space, in order to define the location of an entity. An example of instances of this class can be
bedroom, living-room, meeting-room.
- Absolute Location: this subclass describes the coordinates of an entity in a place according to the specifications of planning.

- Environment: the environment class holds information on physical environment conditions, such as humidity, lighting, noise, pressure, temperature.

- Computational Entity: This class defines all the objects that exist in a space. It includes not only the computational devices but also the resources and is divided in the corresponding subclasses.
  - Device: This subclass includes all the computational devices that are able to interact with a user. Personal computers, smart phones, television, PDA, so forth can be instances of this class. The description of device is extended to the Device ontology (§4.3).
  - Resources: Furniture, windows, doors, lights are a small category of instances of this subclass.

- Complex Event: this class describes all activities carried out in a pervasive environment (i.e. working meetings, lectures, request a video). Each Complex Event consists of a series of Simple Events that are not necessarily indivisible. The duration of a Complex event and the start time are data properties that characterize this entity. Moreover a set of object properties are defined, in order to exist correlation between activities (i.e. meets, overlap, start, finishes, is equal). The definition of the relation primitives is based on a general theory of action and time (Allen Algebra) [1].

- Simple Event: this class describes an event that happens in the space and its time duration is negligible. For example, instances of this class can be the act of a person who sits in a chair, or that the television is turned on. Along with the Complex Event, relations are derived between Simple Events according to the Allen Algebra (follows, proceeds).

The Core Ontology is able to describe any domain. This general mapping of user environment is able to give us information about the location of a user and a computational entity. At this point, we should refer that the object property has location is analysed into two sub-properties, namely has relative location and has absolute location. Using the power and expressiveness of OWL DL, we define that a person or a computational entity can be located to a specific place but in the same time they can be correlated to each other, using the property which has relative location (above, in front of, behind under, facing). Moreover, every location is characterized by the Environment conditions. In the context of continuously monitoring, Simple and Complex Events [44] capture the changes in the environment and also the entities that take part in them.
4.4.2 User Profile Ontology

Existing literature, applications and ontologies related to the domain of user context and profiling have been taken into account in order to create a general, comprehensive and extensible User Profile Ontology. This ontology models all the information that describe a user and is illustrated in Figure 4.6.

Figure 4.6: User Profile Ontology

Classes

- **Person**: this class is the central one in the ontology, since it contains all the user profile characteristics. Especially, it can be described by first name, last name, date of birth, gender, ID and the situation of user. The information about the current condition of a person is always updated and refers to the specific activities that a person takes part and could define his status. Possible values of this data property can be meeting, sleeping and so forth.

- **Contact Profile**: this entity gives additional information about the ways to keep in contact with a person, such as e-mail, phone, address.

- **Ability**: this superclass refers to user abilities and disabilities, both mental and physical. Two corresponding subclasses are created, namely Cognitive Ability and Physical Ability. These classes are characterized by the type and the level of the ability.

- **Preference**: This class describes the user’s preferences. It is defined by the type and the level of preference, as well as the environment condition and the
user situation that are the ideal for satisfying the user's desire. In the context of our perspective, this class focuses on preferences that refer to information and how it can be displayed in a device. In order to achieve this goal, the object property relate to links the Preference with the Device class, providing mainly personalization. We describe the utility of Preference in the next chapter. This class is analysed in a subclass, in order to understand better the preferences of user, regarding the consumption of multimedia information. The subclass describes how the user prefers the contented to be adopted and inherit the properties of the superclass.

- Specific Preference: this subclass describes specific preferences that are related to the multimedia content. The data property category characterizes the type of multimedia content that is referred to each instance of this class. The range value of this property can be the strings: Image, Video, Audio, Audiovisual and Text. For example, a user prefers the selected images to be analysed as black and white images in a device.

This User Profile Ontology expresses the characteristics, the preferences and the abilities of a person. The purpose was to create a general yet extensible ontology, offering a complete profile of a user with static and dynamic information.

4.4.3 Device Ontology

Heterogeneity and mobility of devices pose challenges for information delivery mechanisms. In order to implement a required adaptation, it is necessary to describe and represent the capabilities of all devices \[31\] that are in reach and can be used by a user to receive their requested media content. The Device Ontology achieves these goals and analyses the characteristics and the status of each device, according to the supported technical and software infrastructure. The Figure 4.7 illustrates the concept of this ontology. We have been influenced by the approach \[3.1.2\] that categorizes the characteristics of a device according to their type (software, hardware) and create a number of corresponding classes. Our perspective extends the mapping of capabilities with additional classes and includes more details. Although we modify some parts of ontology in order to ameliorate it and facilitate the adaptation procedure.

Classes

- Device: this class is the core entity of this modelling. The terminals such PCs, mobile phones, projectors, PDAs and whatever is able to present multimedia information are included in this class. Each device is characterised by model, vendor, type and a short description of device.

- Device Status: this class represents the current condition of a device. Specifically, it contains the details of its CPU usage, power resource, remaining
4.4. ONTOLOGY-BASED CONTEXT MODELLING

Figure 4.7: Device Ontology

power, wireless connection, connection status, volume level, brightness and if it is on/off. Point of reference is the data property profile, which declares the level of requirements (high, normal, low) that satisfy each user of this device.

- **Hardware Description**: this class contains the details about the hardware resources of a device, such as CPU (version, type, name),graphics card, total memory,sound and screen capability.

- **Screen Characteristics**: this class provides the display characteristics of the device, such as resolution, aspect ration, color format, DPI, bits per pixel, contrast and brightness.

- **Sound Characteristics**: this class includes the features which describe the RMS power, peak power, number of channel.

- **Software Description**: this class defines details about operation system (version, vendor, name), SDK capabilities.

- **Browser Description**: this entity provides information about the name, version and applications supported of a browser installed on device.

- **Application Software**: this class describes the application or program that is installed on a device. Application name, memory requirements and supported media types are the data properties that characterize the Application Software.
Network Characteristics: this class refers to the networking capabilities of a device.

This Device ontology is intended to provide a general framework to describe any type of device and can be extended with the construction of a hierarchy of sub-classes. Each type of device (e.g., PCs, PDAs, smart phones) corresponds to a sub-class. Using the expansiveness of OWL, some devices are restricted to inherit from the Device superclass specific object properties and subsequently connect with corresponding classes, since these devices may not afford the entire description of this ontology. For example, an audio player could not support screen characteristics. The concept of modelling separates each category of device characteristics to different classes, facilitating in that way the adaptation decision procedure.

4.4.4 Multimedia Content Ontology

A prerequisite for efficient adaptation of multimedia information is a careful analysis of the characteristics of different media types[43]. The term multimedia refers to all types of media files, such as text, graphics, animation, audio, video, etc., that require different adaptation procedures. The complex nature of multimedia makes the adaptation difficult to design and implement. The Multimedia Content Ontology, whose modelling is based on MPEG-7 standard [ref: MPEG7], represents the various types of media, as well as the combination of them. Moreover, a part of this ontology describes the segmentation of each multimedia content in time and space according to the standard. The context under which this ontology (see Figure 4.8) has been utilized is not only the comprehensive representation of multimedia information but also the way that each multimedia content can be converted. In particular, this modelling illustrates a part of adaptation process.

Classes

• Multimedia Content: this class is the central one in the ontology because it describes the original multimedia information. The content is described by the following characteristics: title, creator, date of creation, file format, URL that defines the file path and file size. According to MPEG-7 standard, multimedia content is classified into five types: Image, Video, Audio, Audiovisual, Multimedia. Each of these types consists of distinct and disjoint subclass that inherits the properties of the Multimedia Content superclass.

  – Image: this subclass includes the characteristics of an image, such as aspect ratio, resolution, DPI, color analysis.
  – Video: this media can be interpreted as a sequence of still images that represents scenes in motion. This class is described by the following features: duration, frame rate, resolution, color space, bit rate,
  – Audio: the representation of sound is characterized by audio frequency and duration.
Text: this subclass represents the written language that is the cardinal component of the multimedia. Font and character set describe the Text.

Audiovisual: it refers to content with both sound and visual component. The data properties that give additional description to this class is the duration, the frame rate, the resolution, the color space, the bit rate, the audio availability and audio encoding format.

Multimedia: this class represents the composition of various types of media content. For example if we have a text with images, we have Multimedia.

Multimedia Segment: this class describes a spatial and/or temporal fragment of multimedia content. The instances of Multimedia Segment are derived from the adaptation procedure and are a transformation of the original content. Title, file format, URL, file size give additional details about this class. Each Multimedia Content can be converted to various Multimedia Segments depending on the kind of modification that generates the following subclasses.

Space Segment: this subclass derives from spatial segmentation of multimedia information and the data property region bounds the new region according to the original.

Still Region: this subclass of Space Segment follows the same concept of its superclass. Particularly, it can derive only from Image and is described by the corresponding data properties. For example, the resulting resize image can be an instance of Still Region.
* Text Segment: this subclass derives from text conversion.

- Time Segment: this subclass results from temporal decomposition of multimedia information. Specifically, these instances are created when the conversion refers only to duration issues.

* Audio Segment: the instances of this class characterize a particular time period and are linked only to Audio class. Audio Segment is described by corresponding features.

* Video Segment: this entity is a set of frames in a video sequence. An instance of this class is described by corresponding features of a video.

* Audiovisual Segment: this subclass combines a set both frames and audio samples. It is related only to Audiovisual Content.

* Video Frame: this entity defines one of the many images which compose a video. It can derive from Video and Audiovisual classes.

- Space-Time Segment: this subclass results from spatial-temporal segmentation of multimedia information.

* Moving Region: the instance of this class is the result of a Video instance and is linked to the corresponding data properties.

* Audiovisual Region: this subclass derives only from Audiovisual class and is described by the same features.

The Multimedia Content Ontology provides the formalism of low-level feature descriptions derived from MPEG-7 standard and re-engineers the structural description of Multimedia Description Schema (MDS). The hierarchy of classes is organized in order to describe the structure of multimedia content in time and space. Using the expressiveness of OWL, each Multimedia Segment is restricted to derive from specific Multimedia Content as we have already analysed above. This modelling facilitates the adaptation procedure, since each transformation of original information can be described step by step.
Chapter 5

Methodology of Adaptation
Decision Procedure

As we mentioned in the previous chapter, the Adaptation Decision Engine which is part of ADF decides the optimal multimedia content version and the best strategy for delivering that version. In this chapter, we analyse the adequate sequences of adaptation decisions, in order to accomplish the adaptation of requested information and also the delivery procedure. In other words, a set of rules determines the steps and reveals the path for the final result. Jess rule engine is the operator that manages the activation of appropriate rules according to the contents of Knowledge Base. Below, we present a scenario of the ubiquitous environment that leads in understanding the evaluation and implementation of adaptation decisions.

5.1 A use case scenario

Bob, a game fan, wakes up early in the morning. While he is going to his office, he wants to watch from his mobile phone a football match from the previous night but unfortunately his battery is running out. Therefore he can listen to the game since the video is adapted to audio. His schedule for the day involves a meeting to the office and visiting his friend to watch live formula one. Bob’s meeting in the office is running longer than he has expected and misses the game. He decides to watch the formula crashes during the meeting on his tablet, since the conversation isn’t so interesting. Thus, he accesses the multimedia news formula portal and chooses an audio-visual report. Since the meeting is still going on, he watches the converted report with the absence of sound, so other co-workers are not disturbed. Later at home, while he is sitting in the living-room he is informed about the news on his laptop. He checks financial charts for the stock market. These image charts are converted to the resolution of tablet. He clicks to a audio-visual report. Although, the video is displayed on the TV which is front of him, he prefers high quality presentation on his laptop at home since the battery is low.
5.2 Briefly Syntax Description of Jess Engine

In order to be more perceived in the rest of the chapter, we mention to some parts of the functionality of Jess rule engine. The term fact is the basic component of this rule engine, since the set of facts capture the content of the knowledge base (working memory). Particularly, each fact in working memory is related to an instance of a class and includes the corresponding features (slots) that characterize it (data/object property). A Jess rule is something like an if... then statement. Jess rules are activated whenever their if parts (their left-hand-sides or LHSs) are satisfied. Each rule has two parts, separated by the ";=>" symbol (read as "then"). The first part consists of the LHS pattern which consists of patterns that match facts, and the second part consists of the RHS action. Below we give the general syntax of a rule.

- (defrule Rule_Name (fact-1 (slotName1 ?x) (slotName2 ?y)...) ... (fact-i (slotName1 ?z) (slotName2 ?t) ) ... => ...action)

Each slot consists of the name of the slot (slotName) and variable (?x) that is the corresponding value of the slot. When the slot corresponds to a object property the variable is a list of corresponding instances. For example, the fact that describes the instance Bob of class Person is:

- (Person (instance Bob) (last name Marley ) ... (has Preferences ?x) ...)

5.3 Decision Procedure

The decision of the optimal form of requested multimedia information comprises of various steps. When a user chooses a multimedia content to be displayed in the device, a mechanism is activated and a series of actions leads to the final decision in order to have efficient consumption of content. The flow sequence of adaptation decision that is illustrated in Figure 5.1 describes all the processing stages that should be checked and how the system reacts when an event occurs during the presentation of content. Each cycle corresponds to a set of rules that manages a particular part of process and decides about the flow. To be more precise, each step depends on the previous steps, implementing the concept of backward-chaining. In the context of this concatenation of rules, the modules, which are a functionality of Jess, divide rules and facts into distinct groups. Particularly, modules organize the large number of rules into logical groups. They provide a control mechanism that the rules in a module fire only when that module has the focus and only one module can be in focus at a time. Moreover the action of a rule can change the current focus and respectively the flow. In the following description of each step, we provide a representative sample of rules and describe their implementation, using the above scenario. For the sake of clarity, the precise syntax of rules has been modified. The interested reader can refer to Appendix B where a sample of rules is presented.
5.3. DECISION PROCEDURE

Figure 5.1: The flow sequence of Adaptation Decision.

5.3.1 User in adaptation decision

Request Information

The process begins when a user submits a request for displaying multimedia information through a device.

- defrule Request Information
  ( Complex Event (instance ?instance-Event = Request Information) )
  ( Person (instance ?person) (partOf ?instance-Event) )
  ( Device (instance ?device) (partOf ?instance-Event) )
  ( Multimedia Content (instance ?Multimedia) (partOf ?instance-Event) )
Move on to the next stage. Starting the process of the above instances (?person, ?device, ?Multimedia)

The request for some information is considered as an instance of Complex Event class. When such an instance is created, a person has requested for a multimedia content in a particular device. One of the requests in the above scenario is the choice of Bob to watch the football match in mobile phone. Based on this information, the procedure is navigated to the next step, in order to personalize the adaptation according to the user’s preferences. Moreover, this modelling can support the simultaneous activation of adaptation procedure by different users.

User Preferences

User’s preferences play a significant role, since they influence the behaviour of decision engine. The user profile offers all the information about the characteristics and preferences of a user. The Specific Preference class describes how the user prefers the content to be adopted according to the condition of environment and his/her situation. When Bob watches a video during the meeting, prefers to lack sound. The data property category is the type of multimedia content that is referred in this preference. Consequently, Bob has declared to the value of category Audio-visual, the type audio and the level mute (0) when his situation is in a meeting. If there are instances in this class that refer to the type of the current requested information (e.g. audiovisual content), the following rule is activated.

- defrule User Specific Preferences
  
  (Multimedia Content (instance ?MultimediaContent))
  
  (Person (instance ?person) (hasPreferences ?specificPreference)) (situation ?UserSituation))
  
  
  (Environment (instance ?EnvironmentInstance))
  
  =>
  
  - Create Multimedia Content Preference facts which refer to Multimedia Content.
  
  - Move on to the next stage.

This rule creates the Multimedia Content Preference facts in order to specialize the preferences on current Multimedia Content.
Creation of Multimedia Segment (First-Level Adaptation)

The next stage processes the Multimedia Content Preference facts and creates an instance of the Multimedia Segment class. According to the kind of conversion that the user prefers and the type of the multimedia content, an instance is generated in the corresponding class. An example is the audiovisual content (formula crashes) that is converted to audiovisual region, since audio is removed. Specifically, the new instance represents the result of the first-level adaptation of the requested information. It should be noticed that an automatic mechanism selects the type of new instance according to the above requirements. In the context of this functionality, a kind of language was created that relates the value of data properties of Preference class with the corresponding properties of Multimedia Content. The following rule is a sample of the rules that implement this step and refers to the creation of an Audiovisual Region instance. According to the requirements mentioned above, rules generate the corresponding instances.

- **defrule** Multimedia Content Preferences create AudioVisual Instance

  \[
  \text{Multimedia Content Preferences} \quad \text{(instance ?MultimediaPreference) (type ?type) (level ?level) (referTo ?audiovisual) (user situation ?UserSituation) (environmental condition ?EnvironmentalCondition)}
  \]

  \[
  \text{(Person} \quad \text{(instance ?person) (hasPreferences ?MultimediaPreference) (situation ?UserSituation)}
  \]

  \[
  \]

  \[
  \Rightarrow
  \]

  \[
  \text{if} \quad \text{(?type = Duration)} \quad \text{then}
  \]

  \[
  \text{(Create (Audiovisual Segment instance)} \quad \text{)}
  \]

  \[
  \text{else} \quad \text{(?type = sound availability)} \quad \text{then}
  \]

  \[
  \text{(Create (Audiovisual Region instance (Sound availability = level)} \quad \text{)} \quad \text{)}
  \]

  - Move on to the next step.

5.3.2 Device in adaptation decision

The basic part of multimedia content display is the terminal. The currently adaptation decision is considered as the optimal choice but the device sets the limits on how the information is presented.
CHAPTER 5. METHODOLOGY OF ADAPTATION DECISION PROCEDURE

General features of device

Since the device has determined, general characteristics of the terminal and network should be examined in order to refine the consumption of information. This step consists of three simultaneous action categories.

The first set of rules examines the current status of the device and the basic capabilities like, the memory availability, the supported file formats, CPU usage, the output capabilities, the remaining battery life, and the network connection status. In this category there are two kinds of rules. The first kind of rules is activated if the procedure can be continued without proposed changes in multimedia content, having checked the above features. The other kind is activated when transformations should be applied to the content or the device is not able to represent the information. According to the feature examined in each rule, the actions are different. For example, the rule about the modification of multimedia format can be activated, if the device doesn’t support the appropriate program (Application Software) for the multimedia format. It decides the new format and transfers the modification instruction to the next level of the procedure, that will invoke the corresponding rules.

- **defrule** Check/Change FileFormat
  
  (Device (instance ?device) (has Software Description ?SoftwareDescription) )


  (Software Description (instance ?SoftwareDescription) (support ?ApplicationsSoftware) )


  =>

  - Create modification instructions
  - Move on to the next level

The next part of this phase refers to the requirements of a user for the target device. The selected device should be adapted to the Preference of the user that captured on User Profile ontology (Preferences relate to Device). The rules of this part manage the device characteristics and the current status such as brightness, volume level and so on.

- **defrule** User Preferences For Devices

5.3. DECISION PROCEDURE

(Person (instance ?person) (hasPreferences ?Preference) (situation User-Situation))

(Environment (instance ?EnvironmentalCondition))

(Device (instance ?device) (has current Device Status ?deviceStatus))


=>

- The data property ?type of instance ?device take the value ?level
- Move on to the next level

The latter set of rules refers to the attribute profile of Device Status class that describe the requirements of a user for the quality of the adapted information. The values (low, normal, high) of the property are updated for each user of the device. It is a kind of preference that is adjusted by the Preference class according to the corresponding situations. This attribute directs the selection of what transformation the user prefers. Particularly, in the use case scenario, Bod prefers high quality presentation on his laptop and the video is displayed on the TV in front of him, instead of being adapted to audio as the football game.

Change Device

Anytime when the set of rules in the above step decides that the display device should be changed, the control comes to this phase. This stage is an extended part of the flow and checks the location of both the user and the display device. The location of the device is a primary issue in this procedure. Moreover, the rules, that accomplish this stage, can be activated anytime the user changes position relative to the device. Subsequently, the flow continues from this step and the procedure runs back to the Device examination as we see in Figure 5.1. Particularly, the first rule is invoked when the user is not in front of the terminal device and selects the appropriate device for display, according to the location of the user. Afterwards it creates the appropriate fact to activate the second rule. The latter applies the decision and defines the new target device. It is important to mention that due to the lack of presenting the information in the case of the device change, only the second rule is activated, since the the decision for the terminal device has been taken in the previous step.

- defrule Check LocationUser
  
  (Check Location (person ?person) (device ?current device) (Device (instance ?device)) )
  
  (Person (instance ?person) (facing ?facing device != ?current device))
CHAPTER 5. METHODOLOGY OF ADAPTATION DECISION PROCEDURE

(MultimediaContent (instance ?MultimediaContent) )

=>

- Add Fact ( Changing Device (person ?person) (device ?facing-device) (multimedia content ?MultimediaContent) )
- Activate Changing Display Device rule

- defrule Changing Display Device

  ( Changing Device (person ?person) (device ?facing_device) (multimedia content ?MultimediaContent) )

  =>

  - Change the display device
  - Change the flow

Hardware Capabilities

Finally, all capabilities of the device are checked in detailed. According to the type of multimedia content, the corresponding rule of this step is activated. All the previous phases guarantee that the device is able to present the content even if transformations are required. Providing the information that result until now, this stage converts, if it is necessary, the current decided content according to the hardware restrictions. The screen and sound characteristics are examined and the final form of the delivered content is decided. Specifically, the characteristics of multimedia segment instance, which was created in previous step, are compared with the corresponding hardware features. For example, in the above scenario the image charts’ resolution is converted to the resolution of tablet because the image is adapted in this stage according to the capability of the device. The following rule is activated for this transformation decision. Priority of the procedure is the maintenance of the current multimedia segment, since being proposed as optimal adaptation decision. But as we have mentioned, the device set the limits on how the information is presented. According to this concept, a new multimedia segment instance is generated and defines the proposal of the adaptation process.

- defrule Check Hardware Characteristics according to Image Requirements

  (Device (instance ?device) (has Hardware Description ?HardwareDescription) )

  (HardwareDescription (instance ?HardwareDescription) (image capability ?ImageCapable) (has Screen Description ?ScreenDescription) )

5.3. DECISION PROCEDURE

\[
(\text{Image} \ (\text{class} \ ?\text{MultimediaContent}) \ (\text{convertTo} \ ?\text{StillRegion})) \ \\
(\text{StillRegion} \ (\text{class} \ ?\text{StillRegion}) \ (\text{fileFormat} \ ?\text{FileFormat}) \ (\text{URL} \ ?\text{URL}) \ \\
(\text{fileSize} \ ?\text{FileSize}) \ (\text{aspectRation} \ ?\text{Image_AspectRation}) \ (\text{resolution} \ ?\text{Image_Resolution}) \ (\text{DPI} \ ?\text{Image_DPI}) \ (\text{color analysis} \ ?\text{Image_ColorAnalysis}) \ \\
(\text{title} \ ?\text{Title})) \Rightarrow
\]

- Create Final StillRegion instance
- Create Fact: Check Device (person ?person) (device ?device) (multimedia content ?MultimediaContent) (multimedia segment ?StillRegion)
- Move on to the next step.

5.3.3 Display adapted information

We built a framework which is capable of computing multi-step adaptation decision sequences. After the completion of the above actions, the Adaptation Decision Framework defines the final appropriate form of the requested multimedia content. This adopted content will be consumed by the selected device according to the rule below. It should be indicated that the final processed information may be the same with the original due to the fact that changes are not required.

- defrule Display Multimedia Content
  
  \[
  (\text{Check Device} \ (\text{person} \ ?\text{person}) \ (\text{device} \ ?\text{device}) \ (\text{multimedia content} \ ?\text{MultimediaContent}) \ (\text{multimedia segment} \ ?\text{StillRegion})) \Rightarrow
  \]

  - ?StillRegion is going to be displayed in ?device
Chapter 6

Concluding Remarks and Future Work

6.1 Conclusion

Adaptation procedure can be considered as an effective approach to allow the consumption of multimedia content under usage conditions that prevented the display in the original form. In this thesis, we have presented the concept and architecture of a scalable framework for context-aware multimedia content adaptation, especially suited for ambient intelligent environments. One of the most important challenges of this thesis is to propose the optimal content version, considering the heterogeneous of user preferences, the diverse of device and network capabilities, and the condition of terminal environments. In this concept, a set of upper-level ontologies, adopted by Semantic expressiveness, was created and mapped the above required aspects. The reasoning of adaptation process was accomplished by a multi-step decision rules, using Jess rule engine.

Summarizing, an ontology-based approach to map context, provides the means to enable formal description of high-level notions of context. This descriptive modelling of context and multimedia content enhanced the adaptation decision, providing the potential to overcome the limitations and take decisions that meet better the user's expectations. Moreover this approach is flexible for extending to specific concepts in different application domains such as classrooms, e-health, assisting living etc. The structure of this framework is able to embed different technologies in order to collect the knowledge and perform the adaptation decision.

6.2 Future Work

The flexible structure of the suggested adaptation content framework (ADF) facilitates its extension in order to improve knowledge modelling. Specifically, MPEG-7 standard aims at providing structural and semantic description mechanism for multimedia contents. The exploiting of the semantic part of this standard can refine
the current knowledge and consequently the transformation decision. The content
of information such as, what is illustrated in the picture or what are the highlights
of a game is still an open issue. The semantic description of multimedia inform-
ation can offer new perspectives in the adaptation procedure, providing more
expressiveness in the modelling of the content.

In the field of knowledge representation can include the extension in specific
domains, such as e-learning, classrooms, smart meeting rooms, assisted living. Spe-
cific information should be modelled according to the requirements of each domain
and additional rules will extend the reasoning process.

Finally, the World Wide Web is famous for its extremely rapid evolution and in
a short time, different websites started including pictures, audio, video files, making
the web more multimedia rich. The current condition can motivate the future steps
in order to adapt not only the included multimedia information but also the whole
presentation of a site. Multimedia on websites is what makes them all so attractive,
entertaining and is worth being visited repeatedly. Undoubtedly, the adaptation
of multimedia content as component of a web page arises the challenge!
Bibliography


[38] Motorola. The acemedia project. 2008.


Appendix A

Implementation Tools

In this section we briefly describe the main tools used for the implementation of the context-aware multimedia content adaptation framework.

A.1 OWL Language & Protégé - Knowledge Base Representation

We have chosen OWL to realize our context model and define our context ontologies for various reasons. First it is much expressive compared to other ontology languages such as XML and RDFS. Second, it has the capability of supporting semantic interoperability to exchange and share context knowledge between different systems. To be more precise, contexts can be exchanged and understood between different systems in various domains and enabling automated reasoning to be used by automated process.

For ontology modelling we use Protégé OWL [32], an open-source ontology editor. Protégé, besides the editor, provides a suite of tools that support the creation, visualization, and manipulation of ontologies in various representation formats. Furthermore, Protege can be extended in a way of a plug-in architecture and a Java-based Application Programming Interface (API) for building knowledge based tools and applications. In order to establish communication between the designed ontology and the application’s API, we use the Protégé-OWL API, an open-source java library for OWL and RDF(S). The API provides classes and methods to load and save OWL files, to query and manipulate OWL data models.

A.2 Jess, a rule-based system

A rule-based system is a computer declarative program that uses rules to reach conclusions from a set of premises. Jess [23] is a rule engine and scripting environment written entirely in Sun’s Java language by Ernest Friedman-Hill at Sandia National Laboratories in Livermore, CA. Using Jess, you can build Java software
that has the capacity to "reason" using knowledge you supply in the form of declarative rules. Jess is small, light, and one of the fastest rule engines available. Its powerful scripting language gives you access to all of Java’s APIs. A typical rule engine includes three entities:

- **A working memory** where engine holds a collection of knowledge nuggets that are called facts. In a typical rule engine, the working memory, which is sometimes called the fact base, contains all the pieces of information that the rule-based system is working with. This part of the engine is important because rules can only react to additions, deletions, and changes to working memory.

- **A rule base** which contains all the rules the system knows. A Jess rule is something like an if... then statement. Jess rules are executed whenever their if parts (their left-hand-sides or LHSs) are satisfied. Given only that the rule engine is running. Each rule has two parts, separated by the "=>" symbol (read as "then"). The first part consists of the LHS pattern which comprises patterns that match facts, and the second part consists of the RHS action.

- **An inference engine** controls the whole process of applying the rules to the working memory to obtain the outputs of the system. All the rules are compared to working memory to decide which ones should be activated. The rule engine will decide the order with which the rules will be fired since many rules can be activated simultaneously.
Appendix B

A sample of rules for image adaptation

This chapter presents a sample of decision rules that accomplishes the context-aware adaptation of an image, according to the concept which has analysed in the previous chapters.

- (defrule ComplexEvent_Request
  ?ComplexEventFact<-(ComplexEvent (instance ?instanceEvent:& (call ?*functions* Equal ?instanceEvent requestInformation) ))
  (Person (instance ?person)(partOf?complexEventPerson&:(call ?*functions* contain ?complexEventPerson ?instanceEvent)) (hasPreferences ?Preferences))
  )
  =>
  (bind ?t (call ?*ontologyHandler* setObjectPropertyValue ?MultimediaContent ?device isDisplayed ?*AdaptationSystemOntology*)))
  (bind ?*user* ?person )
  (bind ?*multimediaInformation* ?MultimediaContent )
  (bind ?*Device* ?device )
  (retract ?ComplexEventFact)
  ;(assert (Start_Preferences (person ?person) (multimediaContent ?MultimediaContent) (device ?device)))

57
APPENDIX B. A SAMPLE OF RULES FOR IMAGE ADAPTATION

(bind ?rangeInstancesList (call ?*jessHandler* getRangeInstances Person ?person gethasPreferences SpecificPreference) )

(if (eq ?rangeInstancesList nil) then
  (assert (Create_firstMultimediaSegment (person ?person) (multimediaContent ?MultimediaContent)(device ?device) )))
else
  (bind ?*numberOf_SpecificPreferenceInstances* 0)
  (bind ?*countOf_SpecificPreferenceInstances* 0)
  (bind ?i 0)
  (while (< ?i ?*numberOf_SpecificPreferenceInstances*)
    (bind ?item (call ?rangeInstancesList get ?i ) )
    (bind ?object (call ?*jessHandler* getSpecificShadowObject SpecificPreference ?item) )
    (if (eq (call ?object getcategory ) (call ?*functions* getClassName ?multimediaFact ) )then (assert (Start_Preferences (person ?person) (multimediaContent ?MultimediaContent)(device ?device) (preference ?item))))
    (bind ?i (+ ?i 1))
    (bind ?*numberOf_SpecificPreferenceInstances*(+ ?*numberOf_SpecificPreferenceInstances* 1))
  )
  (assert (Create_firstMultimediaSegment (person ?person) (multimediaContent ?MultimediaContent)(device ?device) )))
)
)
(if (eq ?*numberOf_SpecificPreferenceInstances* 0 )then
  (assert (Create_firstMultimediaSegment (person ?person) (multimediaContent ?MultimediaContent)(device ?device) )))
)
)

• (defrule User_Specific_Preferences
  (declare (no-loop TRUE))
  ?multimediaFact <- (MultimediaContent (instance ?MultimediaContent ) )
)
(Person (instance ?person) (hasPreferences ?Preferences & : (call **functions** contain ?Preferences ?specificPreference) ) (situation ?Situation & : (or (call **functions** Equal ?Situation ?UserSituation) (call **functions** Equal ?UserSituation all)) ) ) 

(Environmen t (instance ?Enviromen tInstance & : (or (call **functions** Equal ?Enviromen tInstance ?EnviromentalCondition) (call **functions** Equal ?EnvironmentalCondition all)) ) )

=>

(bind ?in 0)

;..........Image.......

(if (and (eq (call **functions** getClassName ?multimediaFact ) "Image") (eq ?category "Image")) then

(bind ?name (call **functions** concat "multimediaPref_" ?MultimediaContent ?type ))

(bind ?t (call **ontologyHandler** createInstance MultimediaContentPreference ?name ?*PersonProfileOntology*))

(bind ?t (call **ontologyHandler** addObjectPropertyValue ?name ?MultimediaContent referTo ?*AdaptationSystemOntology*))


(bind ?t (call **ontologyHandler** addObjectPropertyValue ?person ?name hasPreferences ?*PersonProfileOntology*))

(bind ?in 1 )

;..........Video.......

(if (and (call **functions** Equal (call **functions** getClassName ?multimediaFact ) "Video") (eq ?category "Video")) then

(bind ?name (call **functions** concat "MultimediaPreferences_" ?MultimediaContent ?type ))

(bind ?t (call **ontologyHandler** createInstance MultimediaContentPreference ?name ?*PersonProfileOntology*))


(bind ?t (call **ontologyHandler** addObjectPropertyValue ?person ?name hasPreferences ?*PersonProfileOntology*))

(bind ?t (call **ontologyHandler** addObjectPropertyValue ?name ?MultimediaContent referTo ?*AdaptationSystemOntology*))

)
(if (and (eq (call ?*functions* getClassName ?multimediaFact) "Audio") (eq ?category "Audio")) then (bind ?name (call ?*functions* concat "multimediaPref_" ?MultimediaContent ?type)))

(bind ?t (call ?*ontologyHandler* createInstance MultimediaContentPreference ?name ?*PersonProfileOntology*))

(bind ?t (call ?*ontologyHandler* addObjectPropertyValue ?name ?MultimediaContent referTo ?*AdaptationSystemOntology*)))


(bind ?t (call ?*ontologyHandler* addObjectPropertyValue ?person ?name hasPreferences ?*PersonProfileOntology*))

) ;........Text.......(if (and (eq (call ?*functions* getClassName ?multimediaFact) "Text") (eq ?category "Text")) then

(bind ?name (call ?*functions* concat "multimediaPref_" ?MultimediaContent ?type)))

(bind ?t (call ?*ontologyHandler* createInstance MultimediaContentPreference ?name ?*PersonProfileOntology*)))

(bind ?t (call ?*ontologyHandler* addObjectPropertyValue ?name ?MultimediaContent referTo ?*AdaptationSystemOntology*)))


(bind ?t (call ?*ontologyHandler* addObjectPropertyValue ?person ?name hasPreferences ?*PersonProfileOntology*)))

(retract ?Start_PreferencesFact)

(assert (Start_MultimediaPreferences (person ?person) (multimediaContent ?MultimediaContent)(device ?device) (multimediaContentPreference ?name)))

(focus MULTIMEDIAPREFERENCES )

- (defmodule MULTIMEDIAPREFERENCES)
- (defrule MultimediaContentPref_Image
  (declare (no-loop TRUE))


=>

(bind ?name (call ?*functions* concat "StillRegion_" ?image ))

(if (call ?*functions* Equal ?type AspectRation) then

(bind ?t (call ?*ontologyHandler* setDataType PropertyValue ?name AspectRation ?level ?*MultimediaContentOntology*))

(if (call ?*functions* Equal ?type Resolution) then

(bind ?t (call ?*ontologyHandler* setDataType PropertyValue ?name Resolution ?level ?*MultimediaContentOntology*))

(if (call ?*functions* Equal ?type DPI) then

(bind ?t (call ?*ontologyHandler* setDataType PropertyValue ?name DPI ?level ?*MultimediaContentOntology*)))

(if (call ?*functions* Equal ?type ColorAnalysis) then

(bind ?t (call ?*ontologyHandler* setDataType PropertyValue ?name colorAnalysis ?level ?*MultimediaContentOntology*)))

(retract ?Start_MultimediaPreferencesFact)

(bind ?*countOf_SpecificPreferenceInstances* (+ ?*countOf_SpecificPreferenceInstances* 1 ))

(if (= ?*countOf_SpecificPreferenceInstances* ?*numberOf_SpecificPreferenceInstances* ) then

(assert (Check_Preferences (person ?person) (device ?device) (multimediaContent ?image))))

(assert (Check_Location (person ?person) (device ?device)))

(assert (Check_DevicePreferences (person ?person) (device ?device))))
APPENDIX B. A SAMPLE OF RULES FOR IMAGE ADAPTATION

• (deffrule Create_First_S StillRegion
  (declare (no-loop TRUE))
  =>
  (bind ?name (call ?*functions* concat "StillRegion_" ?image))
  (bind ?exist (call ?*ontologyHandler* getIndividual ?name))
  (bind ?t (call ?*ontologyHandler* createInstance StillRegion ?name ?*MultimediaContentOntology*))
  (bind ?t (call ?*ontologyHandler* addObjectPropertyValue ?image ?name convertTo ?*MultimediaContentOntology*))
  (bind ?t (call ?*ontologyHandler* setObjectPropertyValue ?name ?image deriveFrom ?*MultimediaContentOntology*))
  (retract ?Create_firstMultimediaSegmentFact)
  (assert (Check_Preferences (person ?person) (device ?device) (multimediaContent ?image)))
  (assert (Check_Location (person ?person) (device ?device)))
  (assert (Check_DevicePreferences (person ?person) (device ?device)))

• (deffrule Check_General_Characteristics
  (declare (no-loop TRUE))
  ?CheckPreferencesFact <- (Check_Preferences (person ?person) (device ?device) (multimediaContent ?image))
  (Device (instance ?device) (has_CurrentDeviceStatus ?deviceStatus) (has_SoftwareDescription ?SoftwareDescription))
  (SoftwareDescription (instance ?SoftwareDescription) (support ?ApplicationSoftware))

)
(ApplicationSoftware (instance ?applicationSoftware & : (call ?*functions* contain ?ApplicationsSoftware ?applicationSoftware) ) (memoryRequirements ?MemoryRequirements) (multimediaType ?MultimediaType & : (call ?*functions* contain ?MultimediaType (call ?*functions* getClassName ?multimediaFact ))))


=>

(retract ?CheckPreferencesFact)

(assert (Check_GeneralCharacteristics(person ?person) (device ?device) (multimediaContent ?MultimediaContent )))
)

• (defrule Check/Change_FileFormat


( Device (instance ?device ) (has_SoftwareDescription ?SoftwareDescription) )


(SoftwareDescription (instance ?SoftwareDescription) (support ?ApplicationsSoftware))


=>

(retract ?generalCharacteristicsFact )

(bind ?newFormat (call ?*functions* getFirstPartOfString ?fileFormat ,))

(assert(Change_MultimediaContent_Characteristic(multimediaContent ?MultimediaContent) (characteristic fileFormat) (value ?newFormat)))

(assert( Check_FileFormatCharacteristics(person ?person) (device ?device) (multimediaContent ?MultimediaContent )))
)
• (defrule Unable_Using_Device
  (declare (no-loop TRUE))
  (?CheckPreferencesFact <- (Check_Preferences (person ?person) (device ?device) (multimediaContent ?MultimediaContent)))
  (Device (instance ?device) (has_CurrentDeviceStatus ?deviceStatus) (has_SoftwareDescription ?SoftwareDescription))
  (SoftwareDescription (instance ?SoftwareDescription) (support ?ApplicationSoftware))
  (ApplicationSoftware (instance ?ApplicationSoftware & : (call ?*functions* contain ?ApplicationsSoftware ?applicationSoftware)) (memoryRequirements ?MemoryRequirements) (multimediaType ?MultimediaType))
  (Person (instance ?person) (inFrontOf ?inFrontOf_device))
  =>
  (if (or (eq ?inFrontOf_device nil) (eq ?inFrontOf_device ?device))
      (assert (Unable_Display (person ?person) (multimediaContent ?MultimediaContent)))
  else
      (bind ?t (call ?*ontologyHandler* setObjectPropertyValue ?person ?inFrontOf_device facing ?*ContextOntology*))
  (retract ?CheckPreferencesFact))

• (defrule User_PreferencesForDevices
  (declare (no-loop TRUE))
  (?CheckDevicePreferencesFact <- (Check_DevicePreferences (person ?person) (device ?device))
  (Person (instance ?person) (hasPreferences ?Preferences & : (call ?*functions* contain ?Preferences ?Preference)) (situation ?Situation & : (or (call...


(Device (instance ?device) (has_CurrentDeviceStatus ?deviceStatus))


=>

(retract ?CheckDevicePreferencesFact)


• (defrule Check_LocationUser

  ?Check_LocationFact <- (Check_Location (person ?person) (device ?device))

  (Device (instance ?device))


  (MultimediaContent (instance ?MultimediaContent))

  =>

  (assert (ChangingDevice (person ?person) (device ?facing_device) (multimediaContent ?MultimediaContent)))

  )

• (defrule Changing_DisplayDevice

  (declare (no-loop TRUE))

  ?CheckLocationFact <- (Check_Location (person ?person) (device ?device))


  =>

  (modify ?CheckLocationFact(device ?facing_device))

  (retract ?ChangingDeviceFact)

  (assert (Check_Preferences (person ?person)(device ?facing_device) (multimediaContent ?MultimediaContent)))

  (assert (Check_DevicePreferences (person ?person)(device ?device))))
• (defrule Check_Device_ImageRequirements_Create_FinalSegment
  (declare (no-loop TRUE))
  ?CheckDeviceFact<-\((Check_Device( person \ ?person \ ) (device \ ?device) (multimediaContent \ ?MultimediaContent ))\)
  ( Device (instance \ ?device \ ) (has_HardwareDescription \ ?HardwareDescription) )
  (HardwareDescription (instance \ ?HardwareDescription \ ) (imageCapable \ ?ImageCapable) (has_ScreenDescription \ ?ScreenDescription))
  (ScreenCharacteristics (instance \ ?ScreenDescription \ ) (aspectRation \ ?Device_AspectRation) (resolution \ ?Device_Resolution) (DPI ?DPI) (colorFormat \ ?Device_ColorFormat))
  (Image (instance \ ?MultimediaContent \ ) (convertTo \ ?MultimediaSegment))

  =>
  (bind \ ?name (call \ ?*functions* concat "FinalStillRegion_" \ ?MultimediaContent))
  (bind \ ?t (call \ ?*ontologyHandler* createInstance StillRegion \ name \ ?*MultimediaContentOntology*))
  (bind \ ?t (call \ ?*ontologyHandler* addObjectPropertyValue \ MultimediaContent \ name \ convertTo \ ?*MultimediaContentOntology*))
  (bind \ ?t (call \ ?*ontologyHandler* setDataPropertiesValue \ name \ Title \ ?Title \ FileFormat \ ?URL \ ?FileSize \ ?*MultimediaContentOntology*))
  (bind \ ?t (call \ ?*ontologyHandler* addObjectPropertyValue \ name \ MultimediaContent deriveFrom \ ?*MultimediaContentOntology*))

  ; Compare Resolution
  (bind \ ?result (call \ ?*functions* compareResolution \ Device_Resolution \ Image_Resolution))
  (bind \ ?t (call \ ?*ontologyHandler* setDataPropertyValue \ name \ Resolution \ ?result \ ?*MultimediaContentOntology*)))

  ; Compare Aspect Ration
  (bind \ ?result (call \ ?*functions* compareAspectRation \ Device_AspectRation \ Image_AspectRation))
  (bind \ ?t (call \ ?*ontologyHandler* setDataPropertyValue \ name \ AspectRation \ ?result \ ?*MultimediaContentOntology*)))
(if (< ?Device_ColorFormat ?Image_ColorAnalysis ) then
  (bind ?t (call *ontologyHandler* setDataTypeValue ?name ColorAnalysis ?Device_ColorFormat *(MultimediaContentOntology*)))
else
  (bind ?t (call *ontologyHandler* setDataTypeValue ?name ColorAnalysis ?Image_ColorAnalysis *(MultimediaContentOntology*)))
)

(assert (Check_Device_MultimediaContent ( person ?person ) (device ?device) (multimediaContent ?MultimediaContent) (multimediaSegment ?name)))

(retract ?CheckDeviceFact)

(focus CHANGES)
)

(defmodule CHANGES)

(defrule Change_Characteristics ( Check_Device_MultimediaContent(person ?person ) (device ?device) (multimediaContent ?MultimediaContent ) (multimediaSegment ?multimediaSegment ) )

=>

(if (call *functions* Equal ?characteristic fileFormat) then
  (bind ?t (call *ontologyHandler* setDataTypeValue ?multimediaSegment FileFormat ?value *(MultimediaContentOntology*)))

(if (call *functions* Equal ?characteristic Resolution) then
  (bind ?t (call *ontologyHandler* setDataTypeValue ?multimediaSegment Resolution (/ (call *ontologyHandler* getDataPropertyValue ?multimediaSegment Resolution ?MultimediaContentOntology*) 2) *(MultimediaContentOntology*)))

;value = NO /only

(if (call *functions* Equal ?characteristic SoundAvailableOnly) then
  (bind ?t (call *ontologyHandler* setDataTypeValue ?multimediaSegment SoundAvailable ?value *(MultimediaContentOntology*)))

(retract ?Change_MultimediaContentCharacteristicFact );If the function return is called from a rule's right-hand-side, it immediately terminates the execution of that rule's RHS. Furthermore, the current focus module is popped from the focus stack. (return)
APPENDIX B. A SAMPLE OF RULES FOR IMAGE ADAPTATION

- (defrule Display_MultimediaContent

  ?Check_Device_MultimediaContentFact <- (Check_Device_MultimediaContent
  (person ?person) (device ?device) (multimediaContent ?MultimediaContent) (multimediaSegment ?multimediaSegment)
  )

  =>

  (bind ?t (call ?*ontologyHandler* addObjectPropertyValue ?multimediaSegment ?device isDisplayed ?*AdaptationSystemOntology*))

  (retract ?Check_Device_MultimediaContentFact))