

Extensibility of Adaptation Capabilities in the CAIN Content Adaptation Engine

Javier Molina, José M. Martínez, Víctor Valdés, Fernando López

Abstract—This paper describes the extensibility mechanism that has been incorporated to the CAIN Adaptation Engine, that provides audiovisual content adaptation based on user preferences, network capabilities and terminal limitations. The integration of new adaptation modules needs no code modifications in the core system, so it does not have to be recompiled for adding or modifying adaptation modules.

Index Terms—Adaptive Signal Processing, Multimedia Systems.

I. INTRODUCTION

The development of both, new access networks providing multimedia capabilities, and a wide and growing range of terminals, makes the adaptation of content an important issue in future multimedia services. Content adaptation is the main objective of a set of technologies that can be grouped under the umbrella of UMA (Universal Multimedia Access) [1]. In order to perform content adaptation it is necessary to have the description of the content and the description of the usage environment. To enhance the user's experience [2], not only terminal and network characteristics and conditions should be taken into account when adapting, but also user preferences and handicaps, as well as environmental conditions. All this information imposes some constraints to the content to be obtained after adaptation. In this paper section II presents the current architecture of CAIN (Content Adaptation Integrator), whilst section III details the differences with the previous version and therefore explains the extensibility mechanism. Section IV presents some conclusions and future research work.

Manuscript received June 21, 2006.

Javier Molina, Grupo de Tratamiento de Imágenes, EPS, Universidad Autónoma de Madrid, Madrid, Spain. phone: 0034-91-497-22-60; e-mail: javier.molina@uam.es).

J. M. Martínez, Grupo de Tratamiento de Imágenes, EPS, Universidad Autónoma de Madrid, Madrid, Spain. e-mail: jm.martinez@uam.es).

V. Valdés, Grupo de Tratamiento de Imágenes, EPS, Universidad Autónoma de Madrid, Madrid, Spain. phone: 0034-91-497-22-60; e-mail: victor.valdes@uam.es).

F. López, Grupo de Tratamiento de Imágenes, EPS, Universidad Autónoma de Madrid, Madrid, Spain. phone: 0034-91-497-22-60; e-mail: f.lopez@uam.es).

II. CAIN SYSTEM OVERVIEW

A. CAIN Architecture

CAIN is a complete multimedia Adaptation Engine integrating different content adaptation approaches[3]. It is constituted by three main modules: the DM (Decision Module), the EM (Execution Module), and the “battery” of CATs (Content Adaptation Tools), that is, the set of CATs CAIN can use for content adaptation and made available to CAIN thanks to the extensibility mechanism explained in III. Currently we have integrated CATs belonging to the four categories considered in the design of CAIN: ‘Transcoder CATs’, ‘Scalable Content CATs’, ‘Real-time Content-driven CATs’ and ‘Transmoding CATs’[4].

To make possible the work of the main modules, a set of support modules are also necessary, which are basically MPEG-7/21 XML parsers to make data (that is, descriptions) accessible for the EM and the DM.

B. CAIN Adaptation Process

First of all, we need a collection of CATs together with the descriptions of their adaptation capabilities [5]. These descriptions will be used by the DM, together with the content description and the usage environment description, in order to select the CAT that will perform the best adaptation.

For performing an adaptation we need to receive, at least, the input media content and a usage environment profile. It is desirable, but unusual, to receive the associated content description file. Therefore, the required media information description is usually extracted directly from the content, using *ffmpeg* library[6].

The EM collects all the information extracted from the required description files (media, usage environment, CAT capabilities) and launches the DM. The DM selects both the best CAT and its configuration parameters using a constraints satisfaction and optimization algorithm [7]. Afterwards the EM is able to execute the selected CAT, calling the adaptation method (see section III.B) of the chosen CAT.

III. CAIN EXTENSIBILITY

CAIN considers the integration of new CATs, as well as the update of existing ones. For this purpose, it provides a flexible extensibility mechanism in order to avoid recoding of the core of CAIN as well as recompiling. One of the integration requirements is to make the CAT's capabilities accessible from the DM. For this, we need to have a complete and lack of ambiguity file, describing what transformations a concrete

CAT is able to do and with which input and output formats. Another important requirement for CAT's integration is the compliance with interface defined for making the CAT accessible by the CAIN core.

A. Content Adaptation Tool Plug-in

The main differences with the previous CAIN version[3][4][5][7] are the following:

- The integration of new CATs does not need any modification in the system's core. This allows to change the set of available CATs before each execution of CAIN.
- The conditions to be fulfilled when incorporating a new CAT are only to be compliant with a defined interface, to follow a certain naming policy (see section III.B).
- The functionality of a working CAT is modifiable even without recoding its own code. This is possible by the edition of the correspondent CAT capabilities description file. This edition may allow the restriction of certain formats in the input or output. The adaptation schemes can also be modified in order to, for example, avoid certain non-desired transcoding operations.

B. Content Adaptation Tool's Interface

All the files implementing the CAT, or used by it, have to be packaged in a *.jar file. This file has to share name with the CAT it represents.

For integrating a new CAT we need, at least, two files:

- The CAT code in a *.java file, with the same name as the CAT. This code file should implement the `adapt(...)` method, declared as abstract in its parent class, `CATImplementation.java`.
- The CAT Capabilities description[5] in a *.xml file with the same name as the CAT. This description file will contain the adaptation capabilities of the module being integrated. This information is necessary for the DM, it will use it for deciding in which situations the new CAT should be launched by CAIN.

Apart from these mandatory files, it is also necessary the inclusion of any library used by the CAT. Any library should be packaged together with the mandatory CAT code file.

The `Adapt(...)` method constitutes the common core to every CAT. It is the method which, by a generic interface, performs the different adaptations. It will be called by the EM and it is expected to adapt in a proper fashion, because the DM has chosen it after checking out, in the CAT's capabilities description file, that it is able to do so. It returns a list containing the paths and formats of the adapted contents. It can return more than a unique output. A CAT, for example, could convert a video into a sequence of images (e.g., keyframes). In this case, the returned structure would contain the paths and characteristics of each of the output images.

C. CAIN Content Adaptation Tool's Management

The files of each integrated CAT should be packaged in a *.jar file. This file is stored in a CATs' folder together with the other available CATs. When CAIN execution starts, the CATs' folder is checked uncompressing each *.jar file in an execution space accessible by the EM and the DM.

IV. CONCLUSIONS AND FUTURE WORK

The proposed extensibility mechanism allows implementation-agnostic integration of new CATs in order to incorporate new functionalities without prior knowledge of each tool capabilities and limitations. This allows to use CAIN for service prototyping (as in currently being done in the EU IST aceMedia Integrated Project), service deployment (reducing the flexibility and numbers of CATs), and as a benchmarking framework. For the later, it will be necessary to incorporate the modules required for providing as result of the adaptation a report of resource consumption (CAT processing time, CPU and memory use, power consumed, temporary storage space, ...) as well as modules for comparing the quality of different versions of adapted content, both being the same or different adaptations, taking into account not only objective, but also subjective quality measures focus in the user's perceived quality even in the case of transmoding.

ACKNOWLEDGMENTS

This work is partially supported by the European Commission 6th Framework Program under project FP6-001765 (aceMedia). This work is also supported by the Ministerio de Ciencia y Tecnología of the Spanish Government under project TIN2004-07860 (MEDUSA) and by the Comunidad de Madrid under project P-TIC-0223-0505 (PROMULTIDIS). The authors want to thank Víctor Fernández-Carbajales for successful testing of the extensibility mechanism.

REFERENCES

- [1] A. Vetro, C. Christopoulos, T. Ebrahimi (eds.), "Universal Multimedia Access (special issue), IEEE Signal Processing Magazine, 20(2), March 2003.
- [2] F. Pereira, I. Burnett, "Universal Multimedia Experiences for Tomorrow", IEEE Signal Processing Magazine, 20(2):63-73, March 2003.
- [3] J.M. Martínez, V. Valdés, J. Bescós, L. Herranz, "Introducing CAIN: A metadata-driven content adaptation manager integrating heterogeneous content adaptation tools", Proceedings of the WIAMIS'2005 (CD-ROM), Montreux, April 2005.
- [4] V. Valdés, J.M. Martínez, "Content Adaptation Tools in the CAIN framework", in Visual Content Processing and Representation, L. Atzori, D.D. Giusto, R. Leonardi, F. Pereira (eds.), Lecture Notes in Computer Science, Vol. 3893, Springer Verlag, 2006, pp. 9-15
- [5] V. Valdés, J.M. Martínez, "Content Adaptation Capabilities Description Tool for Supporting Extensibility in the CAIN Framework", Multimedia Content Representation, Classification and Security-MCRS2006, B.Günzel, A.K.Jain, A.M. Tekalp, B. Sankur (eds.), Lecture Notes in Computer Science, Vol. 4105, Springer Verlag, 2006, pp. 395-402
- [6] <http://ffmpeg.sourceforge.net/>
- [7] F. López, J.M. Martínez, V. Valdés, "Multimedia Content Adaptation within the CAIN framework via Constraints Satisfaction and Optimization", Proceedings of the Fourth International Workshop on Adaptive Multimedia Retrieval-AMR06 (CD-ROM), 17 pp., Geneve, Suiza, July 2006, in press.